Adaptive Knowledge-Enhanced Bayesian Meta-Learning for Few-shot Event Detection

Shirong Shen[1] and **Tongtong Wu**[1] and **Guilin Qi**[1*] and **Yuan-Fang Li**[2] and
Gholamreza Haffari[2] and **Sheng Bi**[1]

[1]School of Computer Science and Engineering, Southeast University, China
[2]Faculty of Information Technology, Monash University, Melbourne, Australia
{ssr, wutong8023, gqi}@seu.edu.cn, yuanfang.li@monash.edu,
Gholamreza.Haffari@monash.edu, bisheng@seu.edu.cn

Abstract

Event detection (ED) aims at detecting event trigger words in sentences and classifying them into specific event types. In real-world applications, ED typically does not have sufficient labelled data, thus can be formulated as a few-shot learning problem. To tackle the issue of low sample diversity in few-shot ED, we propose a novel knowledge-based few-shot event detection method which uses a definition-based encoder to introduce external event knowledge as the knowledge prior of event types. Furthermore, as external knowledge typically provides limited and imperfect coverage of event types, we introduce an adaptive knowledge-enhanced Bayesian meta-learning method to dynamically adjust the knowledge prior of event types. Experiments show our method consistently and substantially outperforms a number of baselines by at least 15 absolute F_1 points under the same few-shot settings.

1 Introduction

Event detection is an important task in information extraction, aiming at detecting event triggers from text and then classifying them into event types (Chen et al., 2015). For example, in "*The police **arrested** Harry on charges of manslaughter*", the trigger word is *arrested*, indicating an **"Arrest"** event. Event detection has been widely applied in Twitter analysis (Zhou et al., 2017), legal case extraction (de Araujo et al., 2017), and financial event extraction (Zheng et al., 2019), to name a few.

Typical approaches to event detection (Chen et al., 2015; McClosky et al., 2011; Liu et al., 2019) generally rely on large-scale annotated datasets for training. Yet in real-world applications, adequate labeled data is usually unavailable. Hence, methods that generalize effectively with small quantities
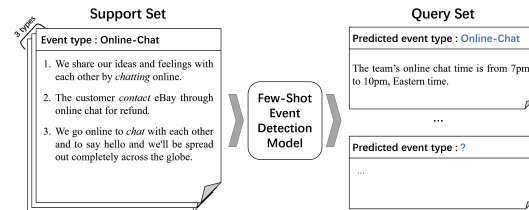
---

* Corresponding author.

Figure 1: A 3-way 3-shot event detection example, in which the model uses the support set to predict the event types of query samples.

of labeled samples and adapt quickly to new event types are highly desirable for event detection.

Various approaches have been proposed to enable learning from only a few samples, i.e., few-shot learning (Finn et al., 2017; Snell et al., 2017; Zhang et al., 2018a). Yet few-shot event detection (FSED) has been less studied until recently (Lai et al., 2020a; Deng et al., 2020). Although these methods achieve encouraging progress on typical N-way M-shot setting (Figure 1), the performance remains unsatisfactory as the *diversity* of examples in the support set is usually limited.

Intuitively, introducing high-quality semantic knowledge, such as FrameNet (Baker et al., 1998), is a potential solution to the insufficient diversity issue (Qu et al., 2020; Tong et al., 2020; Liu et al., 2016, 2020). However, as shown in Figure 2, such knowledge-enhanced methods also suffer from two major issues: (1) the incomplete coverage by the knowledge base and (2) the uncertainty caused by the inexact alignment between predefined knowledge and diverse applications.

To tackle the above issues, in this paper, we propose an Adaptive Knowledge-Enhanced Bayesian Meta-Learning (AKE-BML) framework. More specifically, (1) we align the event types between the support set and FrameNet via heuristic rules.[1] (2) We propose encoders for encoding the sam-

---

[1]For event types that cannot be accurately aligned to FrameNet, we match the nearest super-ordinate frame for them.

Chatting

Definition: A group of people (the Interlocutors or Interlocutor_1 and Interlocutor_2 together) have a conversation. No person is construed as only a speaker or only an addressee.

Arguments: Interlocutors, Interlocutor_1, Interlocutor_2.

LUs: badinage.n, banter.n, chat.n, chat.v, chit-chat.n, colloquy.n,…

Discussion

Definition: Two (or more) people (the Interlocutors, also encodable as Interlocutor_1 and Interlocutor_2) talk to one another. No person is construed as only a speaker or only an addressee. This frame differs from the Chatting frame in that the Interlocutors have a particular question that they are attempting to decide or understand.

Arguments: Interlocutors, Interlocutor_1, Interlocutor_2.

LUs: communicate.v, conference.n, consultation.n, debate.n, talk.n,…

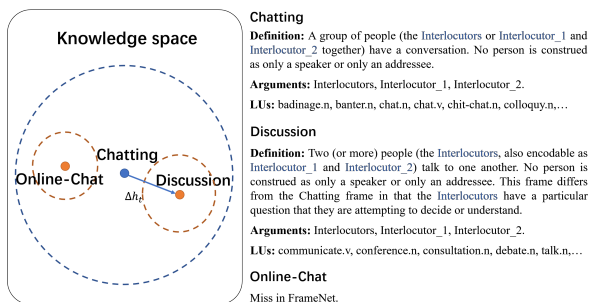**Online-Chat**

Miss in FrameNet.

Figure 2: An example of FrameNet. Left side: the relation between frame 'Chatting' and its sub-frame. Right side: the definitions and LUs (Lexical Units) of frame Chatting and Discussion. The blue words represent the mentions of arguments in definition. It can be seen that, in FrameNet, the definition of the sub-frame is similar to the definition of its super-frame. External knowledge base can provide rich semantic information, yet the knowledge base is typically incomplete, such as the missing of a desired frame "online-chat".

ples and knowledge-base in the same semantic space. (3) We propose a learnable offset for revising the aligned knowledge representations to build the knowledge prior distribution for event types and generate the posterior distribution for event type prototype representations. (4) In the prediction phrase, we adopt the learned posterior distribution for prototype representations to classify query instances into event types.

We conduct comprehensive experiments on the aggregated benchmark dataset of few-shot event detection (Deng et al., 2020). The experimental results show that our method consistently and substantially outperforms state-of-the-art methods. In all six N-way-M-shot settings, our model achieves a large F_1 superiority of at least 15 absolute points.

2   Related Work

Event Detection.   Recent event detection methods based on neural networks have achieved good performance (Chen et al., 2015; Sha et al., 2016; Nguyen et al., 2016; Lou et al., 2021). These methods use neural networks to construct the context features of candidate trigger words to classify events. Pre-trained language models such as BERT (Devlin et al., 2019) have also become an indispensable component of event detection models (Yang et al., 2019; Wadden et al., 2019; Shen et al., 2020). However, neural models rely on large-scale labeled event datasets and fail to predict the labels of new event types. A recent study utilized the basic metric-based few-shot learning method for event detection (Lai et al., 2020b). Deng et

al. (2020) tackles few-shot learning for event classification with a dynamic memory network. To enhance background knowledge, ontology embedding is used in ED (Deng et al., 2021). These methods have achieved encouraging results in the few-shot learning setting. However, they do not address the problem of insufficient sample diversity in the support set. Our method leverages the knowledge in *FrameNet* to augment the support set for event detection.

Few-shot Learning and Meta-learning.   Few-shot learning trains a model with only a few labeled samples in a support set and predicts the labels of unlabeled samples in the query set. Various approaches have been proposed to solve the few-shot learning problem, which mainly fall into three categories: (1) metric-based methods (Vinyals et al., 2016; Snell et al., 2017; Garcia and Bruna, 2012; Sung et al., 2018), (2) optimization-based methods (Finn et al., 2017; Nichol et al., 2018; Ravi and Larochelle, 2016), and (3) model-based methods (Yan et al., 2015; Zhang et al., 2018b; Sukhbaatar et al., 2015; Zhang et al., 2018a). However, these methods rely heavily on the support set and suffer from poor robustness caused by insufficient sample diversity of the support set.

Bayesian meta-learning (Ravi and Larochelle, 2016; Yoon et al., 2018) can construct the posterior distribution of the prototype vector through external information outside the support set. The effectiveness of this method has been shown in the few-shot relation extraction task (Qu et al., 2020). It inspires us to solve the problem of insufficient sample diversity in the task of few-shot event detection by introducing external knowledge. However, this method ignores the semantic deviation between knowledge and target types. Specifically, a knowledge base may provide incomplete coverage of target types in a given support set, which leads to inaccurate matching between a target type and knowledge.

3   Problem Definition

In this paper, the Few-Shot Event Detection (FSED) problem is defined as a typical *N-way-M-shot* problem. Specifically, a tiny labeled *support set S* is provided for model training. S contains N distinct event types and each event type has only M labeled samples, where M is typically small (e.g. $M = 5, 10, 15$). More precisely, in each FSED task we are given a small support set $S = \{(x_s, y_s)\}$.

Let $X_S = \{x_s\}_{s \in S}$ represent the samples in the support set S, i.e. $x_s = (I_s, tt_s)$, where I_s is the sentence of the sample x_s and tt_s is the candidate trigger word of x_s. We denote by Y_S an ordered list of event types, i.e. $Y_S = \{y_s\}_{s \in S}$, where each y_s is the ground-truth event type of sample x_s. For each support set S, we only consider a subset of event types T_s from the entire set of event types T. Hence, in the N-way-M-shot setting, $|T_S| = N$ and $|X_S| = |Y_S| = N * M$.

Moreover, we assume an external knowledge base $\mathscr{F}$ that contains a number of *frames*. Each frame $F_t \in \mathscr{F}$ consists of three parts: $F_t = (D_t, A_t, L_t)$, where D_t, A_t and L_t are the definition, arguments, and linguistic units (LUs) of the frame respectively. Please see Appendix A for details of FrameNet.

For each support set S, we are also given a query set Q composed of some unlabeled samples $X_Q = \{x_q\}_{q \in Q}$, where $x_q = (I_q, tt_q)$, I_q is the sentence of sample x_q, and tt_q is the candidate trigger word of x_q. Our goal is to learn a neural classifier for these event types by using the external knowledge and the support set. We will apply the classifier to predict the labels of the query samples in Q, i.e., $Y_Q = \{y_q\}_{q \in Q}$ with each $y_q \in T_S$. We do this by learning $p(Y_Q|X_Q, X_S, Y_S, \mathscr{F})$.

4 Adaptive Knowledge-Enhanced Bayesian Meta-Learning

We now present our adaptive knowledge-enhanced few-shot event detection approach. The overall structure of our method is shown in Figure 3. Our method represents each event type t with a prototype vector $\mathbf{v}_t$, which is then used to classify the query sentences. We use $\mathbf{V}_{T_S} = \{\mathbf{v}_t\}_{t \in T_S}$ to represent the collection of prototype vectors for all event types in T_S. Then the conditional distribution $p(Y_Q|X_Q, X_S, Y_S, \mathscr{F})$ can be represented as:

$$\int p(Y_Q|X_Q, V_{T_S})p(V_{T_S}|X_S, Y_S, \mathscr{F})\, d\mathbf{V}_{T_S}. \quad (1)$$

To calculate Eq. 1, we first introduce the sample encoder and knowledge encoder to give the vector representations of samples and the knowledge of event types. Then we use the sample representations and knowledge representation to construct the adaptive knowledge-enhanced posterior distribution $p(V_{T_S}|X_S, Y_S, \mathscr{F})$ of $\mathbf{V}_{T_S}$ and give the likelihood $p(Y_Q|X_Q, V_{T_S})$ by $\mathbf{V}_{T_S}$ and sample representations. Finally we leverage Monte Carlo sampling

to approximate the posterior distribution and draw each prototype sample by the stochastic gradient Langevin dynamics (Welling and Teh, 2011) to optimize model parameters in an end-to-end fashion. We now explain the framework in more details.

4.1 Sample and Knowledge Encoder

The purpose of encoding knowledge is to make up for the lack of diversity and coverage of the support set. Thus we align the knowledge and sample encoding and map them into the same semantic space. Intuitively, trigger and arguments are the main factors for entity detection. Hence, to align the trigger and arguments from samples and external knowledge, we design two encoders for the knowledge and samples, generating the final knowledge encoding $\mathbf{h}_t$ and the sample encoding $\mathcal{E}(x)$ with the same dimensions.

Knowledge Encoder. Given a knowledge frame $F_t = \{D_t, A_t, L_t\}$ for the event type t, we encode it into a real-valued vector to represent the semantics of t. As shown in Figure 2, for a frame F_t, the linguistic units L_t can represent the features of the trigger words, the arguments A_t can represent the context of the trigger words in samples, and D_t describes the semantic relationship between A_t and t.

For each event type t, the proposed knowledge encoder uses BERT to generate the text encoding $\mathbf{E}_{D_t}$ and $\mathbf{E}_{L_t}$ from the description D_t and the LUs L_t respectively. Moreover, the arguments encoding $\mathbf{E}_{A_t}$ is a sequence of $\mathbf{e}_{A_t}^{(i)}$, i.e., the average token encoding in the i-th argument mention in D_t, which ensures that the encoding of A_t fully contains the semantics of the event type t. Then, as shown in Figure 3, the trigger word prior encoding and argument prior encoding are generated by follows:

- **Trigger word prior encoding.** We use attention to get the weighted sum of words in L_t as the trigger word prior encoding $\mathbf{e}_{L_t}^*$. The *query* of the attention is $\mathbf{E}_{D_t}$, *key* and *value* are both $\mathbf{E}_{L_t}$.
- **Argument prior encoding.** An attention mechanism is used to aggregate the arguments information into $\mathbf{e}_{A_t}^*$, where the *query* of the attention is $\mathbf{e}_{L_t}^*$, *key* and *value* are both $\mathbf{E}_{A_t}$.

Finally, we concatenate the trigger word prior encoding $\mathbf{e}_{L_t}^*$ and the argument prior encoding $\mathbf{e}_{A_t}^*$, and use a feed forward network f^h to generate the knowledge encoding vector $\mathbf{h}_t$ of event type t,

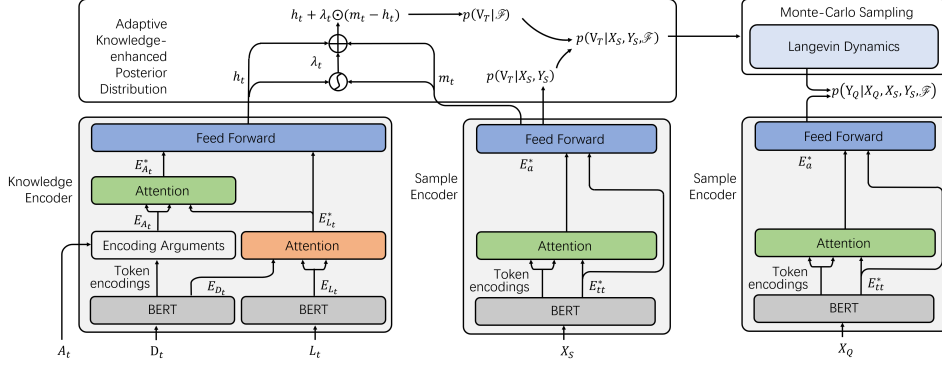$$\mathbf{h}_t = f^h\left([\mathbf{e}_{A_t}^*; \mathbf{e}_{L_t}^*]\right). \quad (2)$$

Figure 3: Framework overview. Our method combines both the external knowledge and the support set into a prior distribution of event prototype. We customize two encoders to generate sample representations and knowledge representations. Then we utilize the support set to generate a learnable offset for revising the aligned knowledge representations to generate the prior distribution for prototype representations. Finally, we use Monte-Carlo sampling and stochastic gradient Langevin dynamics to draw samples of prototypes for prediction.

Sample encoder. We follow the same strategy to build a sample encoder. Given each sample $x = (I, tt)$, i.e., a candidate trigger word tt and its context I, we first utilize BERT to encode x and select the encoding of tt as the trigger representation $\mathbf{e}_{tt}^*$. As arguments are not explicitly given in x, we use an attention mechanism to aggregate the implicit argument information for current trigger tt, in which the *query* is $\mathbf{e}_{tt}^*$, *key* and *value* are both token encoding generate from I. We denote the argument encoding as $\mathbf{e}_a^*$.

Finally, we concatenate the trigger word encoding $\mathbf{e}_{tt}^*$ and the argument encoding $\mathbf{e}_a^*$, and use a feed forward network $f^{\mathcal{E}}$ to generate the sample encoding vector $\mathcal{E}(x)$,

$$\mathcal{E}(x) = f^{\mathcal{E}}\left([\mathbf{e}_a^*; \mathbf{e}_{tt}^*]\right). \tag{3}$$

4.2 Adaptive Knowledge-Enhanced Posterior

The posterior distribution can be factorized into a prior distribution (given the event knowledge) and a likelihood on the support set (Qu et al., 2020) as,

$$p(\mathbf{V}_{T_S}|X_S, Y_S, \mathscr{F}) \propto p(Y_S|X_S, \mathbf{V}_{T_S})p(\mathbf{V}_{T_S}|\mathscr{F}), \tag{4}$$

where $p(Y_S|X_S, \mathbf{V}_{T_S})$ is the likelihood on the support set, and $p(\mathbf{V}_{T_S}|\mathscr{F})$ is the adaptive knowledge-based prior for the prototype vectors. We describe the details of these two components as follows:

Adaptive Knowledge-based Prior. As we discussed in Section 1, an event type t may not have an exact/perfect match in the knowledge base $\mathscr{F}$. In such situations, we resort to finding the super-ordinate frame of t, which is semantically closest to t. As shown in Figures 1 and 2, where

the event type t in the support set 'online-chat' is matched against the knowledge prior F_t 'Chatting' in FrameNet, a super-ordinate frame. In order to enable the knowledge encoding to accurately reflect the characteristics of the corresponding event type, we add a learnable *knowledge offset* to $\mathbf{h}_t$. We denote the knowledge offset between the event type t and its knowledge encoding $\mathbf{h}_t$ by $\Delta \mathbf{h}_t$. Recall that the knowledge in $\mathbf{h}_t$ is encoded from the exactly-matched frame or the super-ordinate frame. $\Delta \mathbf{h}_t$ is defined as follows:

$$\Delta \mathbf{h}_t = \boldsymbol{\lambda}_t \odot (\mathbf{m}_t - \mathbf{h}_t), \tag{5}$$

where $\odot$ is the element-wise product, and $\mathbf{m}_t$ is the mean of the encodings $\mathcal{E}(x)$ of all the samples x in the support set. $\boldsymbol{\lambda}_t \in [0, 1]^{|\mathbf{h}_t|}$ is the adaptive weight (gate), which is obtained from the sample encoding $\mathbf{m}_t$ and the knowledge encoding $\mathbf{h}_t$:

$$\boldsymbol{\lambda}_t = \sigma(\mathbf{W}_\lambda [\mathbf{m}_t; \mathbf{m}_t - \mathbf{h}_t; \mathbf{h}_t] + \mathbf{b}_\lambda), \tag{6}$$

where σ is the nonlinear sigmoid function, and $\mathbf{W}_\lambda$ and $\mathbf{b}_\lambda$ are trainable parameters.

Putting it altogether, the knowledge prior distribution has the following form,

$$p(\mathbf{V}_{T_S}|\mathscr{F}) = \prod_{t \in T_S} p(\mathbf{v}_t|\mathbf{h}_t, \Delta \mathbf{h}_t)$$
$$= \prod_{t \in T_S} \mathcal{N}(\mathbf{v}_t|\mathbf{h}_t + \Delta \mathbf{h}_t, \boldsymbol{I}), \tag{7}$$

where $\mathcal{N}(\mathbf{v}_t|\mathbf{h}_t + \Delta \mathbf{h}_t, \boldsymbol{I})$ is multivariate Gaussian with the mean $\mathbf{h}_t + \Delta \mathbf{h}_t$ and covariance $\boldsymbol{I}$ (the identity matrix). So, each prototype vector has a prior distribution containing knowledge from *FrameNet* adaptively adjusted according to the support set.

Likelihood. With the given prototype vectors $\mathbf{V}_{T_s}$ distributed according to $p(\mathbf{V}_{T_S}|X_S, Y_S, \mathscr{F})$, the likelihood for support samples is defined as,

$$p(Y_S|X_S, \mathbf{V}_{T_S}) \quad = \quad \prod_{s \in S} p(y_s|x_s, \mathbf{V}_{T_S}) \qquad (8)$$

$$p(y_s = t|x_s, \mathbf{V}_{T_S}) \quad := \quad \frac{\exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \mathbf{v}_t)}{\sum_{t' \in T_S} \exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \mathbf{v}_{t'})}.$$

The dot product of the sample encoding $\boldsymbol{\mathcal{E}}(x_q)$ and the event type prototype vector $\mathbf{v}_t$ estimates their similarity. We use $softmax$ to normalize the result to the probability of x_s belonging to event type t.

4.3 Optimization and Prediction

For prediction, the model computes and maximizes the log-probability $\log p(Y_Q|X_Q, X_S, Y_S, \mathscr{F})$. However, according to Eqn (1), the log-probability relies on the integration over prototype vectors, which is difficult to compute. Hence, we estimate it with Monte Carlo sampling (Qu et al., 2020),

$$
\begin{aligned}
&p(Y_Q|X_Q, X_S, Y_S, \mathscr{F}) \\
&= \mathbb{E}_{p(\mathbf{V}_{T_S}|X_S, Y_S, \mathscr{F})} \left[p(Y_Q|X_Q, \mathbf{V}_{T_S}) \right] \\
&\approx \frac{1}{N_s} \sum_{i=1}^{N_s} p(Y_Q|X_Q, \mathbf{V}_{T_S}^{(s)}) \qquad (9)
\end{aligned}
$$

where N_s is the number of samples, and $\mathbf{V}_{T_S}^{(s)}$ is a sample drawn from the posterior distribution, i.e. $\mathbf{V}_{T_S}^{(s)} \sim p(\mathbf{V}_{T_S}|X_S, Y_S, \mathscr{F})$. $p(Y_Q|X_Q, \mathbf{V}_{T_S}^{(s)})$ is the likelihood for query samples which has the same form as Eqn 8. To sample from the posterior, we use the stochastic gradient Langevin dynamics (Welling and Teh, 2011) with multiple stochastic updates. Formally, we initialize the sample $\hat{\mathbf{V}}_{T_S}$ and iteratively update the sample as,

$$
\begin{aligned}
\hat{\mathbf{V}}_{T_S} \leftarrow &\hat{\mathbf{V}}_{T_S} + \sqrt{\epsilon}\mathbf{z} \qquad (10)\\
&+ \frac{\epsilon}{2} \nabla_{\hat{\mathbf{V}}_{T_S}} \log \ p(Y_S|X_S, \hat{\mathbf{V}}_{T_S}) p(\hat{\mathbf{V}}_{T_S}|\mathscr{F}),
\end{aligned}
$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and ϵ is a small real number representing the update step size. The gradient $\nabla_{\hat{\mathbf{V}}_{T_S}} \log \ p(Y_S|X_S, \hat{\mathbf{V}}_{T_S}) p(\hat{\mathbf{V}}_{T_S}|\mathscr{F})$ in Eqn 10 balances the effect of the knowledge and the support set on the prototype vector. Please see Appendix B for derivation details and intuitive explanations of its influence.

The Langevin dynamics requires a burn-in period. To speed up the convergence, we follow the previous method (Qu et al., 2020) and initialize the sample as follows,

$$
\begin{aligned}
\hat{\mathbf{V}}_{T_S} &\leftarrow \{\hat{\mathbf{v}}_t\}_{t \in T_S} \\
\hat{\mathbf{v}}_t &\leftarrow \mathbf{m}_t + \mathbf{h}_t + \Delta\mathbf{h}_t - \mathbf{m},
\end{aligned} \qquad (11)
$$

where $\mathbf{m}$ is the mean encoding of all the samples in the support set.

Algorithm 1: Training Process

Input: Event type set T
Input: Event knowledge from *FrameNet*
1 **while** *not convergence* **do**
2   Sample a subset T_S from T to build a FSED task
3   Sample disjoint support and query sets for T_S
4   Compute the sample encodings (Eq. 3)
5   Compute the $\{\mathbf{m}_t\}_{t \in T_S}$ for each $t \in T_S$
6   Compute knowledge encodings $\{\mathbf{h}_t\}_{t \in T_S}$ (Eq. 2)
7   Compute knowledge offset $\{\Delta\mathbf{h}_t\}_{t \in T_S}$ (Eq. 5)
8   Initialize prototype vectors $\{\mathbf{V}_{T_S}^s\}_{s=1}^{N_s}$ (Eq. 11)
9   Update prototype vectors iteratively (Eq. 12)
10   Compute and maximize log-likelihood (Eq. 9)

After obtaining prototype samples from the posterior, $\log p(Y_Q|X_Q, X_S, Y_S, \mathscr{F})$ is end-to-end approximated according to Eqn (9). During the training stage, we optimize the log-likelihood of the query set and update the model parameters by gradient descent. In the prediction stage, the log-likelihood will determine the probability that a query sample belongs to each event type. The training process is shown in Algorithm 1.

5 Experiments

We conduct evaluation with the following goals: (1) to compare our adaptive knowledge-enhanced Bayesian meta-learning method with existing few-shot event detection methods and few-shot learning baseline methods; (2) to assess the effectiveness of introducing external knowledge in different N-way-M-shot settings; and (3) to provide empirical evidence that our adaptive knowledge offset can flexibly adjust the impact of the support set and prior knowledge on event prototypes, making the model more accurate and generalizable.

5.1 Experimental Settings

We evaluate our method on an aggregated few-shot event detection dataset *FewEvent*[2] (Deng et al., 2020). *FewEvent* combines two currently widely-used event detection datasets, the ACE-2005 corpus[3] and the TAC-KBP-2017 Event Track Data[4], and adds external event types in specific domains including music, film, sports and education (Deng et al., 2020). As a result, *FewEvent* contains 70,852 samples for 19 event types that are further divided into 100 event subtypes.

In order to match the few-shot settings , we use

---

[2] https://github.com/231sm/Low_Resource_KBP
[3] http://projects.ldc.upenn.edu/ace/
[4] https://tac.nist.gov/2017/KBP/Event/index.html

2421

88 event types covering a total of 15,681 samples to construct experimental data. 68 event types are selected for training, 10 for validation, and the rest 10 for testing. Note that there are no overlapping types between the training, validation and testing sets. In order to obtain a convincing result, we conducted 5 random divisions of training and testing for all event types, and the experimental results are averaged as the final result.

The comparisons with our AKE-BML are performed in two aspects, the *sample encoder* and the *few-shot learner*. We combine different encoders and few-shot learners to obtain different baseline models. We consider four sample encoders including CNN (Kim, 2014), Bi-LSTM (Huang et al., 2015), DMN (Kumar et al., 2016) and our trigger-attention-based sample encoder TA. For few-shot learners, we consider Matching Networks (MN) (Vinyals et al., 2016) and Prototypical Networks (PN) (Snell et al., 2017). We also compare to the SOTA few-shot event detection method DMN-MPN (Deng et al., 2020), which uses a dynamic memory network (DMN) as the sample encoder and a memory-based prototypical network as the few-shot learner. In addition, in order to verify the effectiveness of our proposed method, we perform an ablation study on our model, which evaluate the model without external knowledge and without dynamic knowledge adaptation.

As a result, the following methods are compared in our experiments:

- **AKE-BML**, our adaptive knowledge-enhanced Bayesian meta-learning method which uses TA encoder as the sample encoder.
- **KB-BML**, a variant of AKE-BML without dynamic knowledge adaption.
- **TA-BML**, a variant of AKE-BML using our TA encoder but without using external knowledge.
- **DMN-MPN**, dynamic-memory-based prototypical network (Deng et al., 2020).
- **Encoder+Learner**, combinations of various sample encoders and event type learners (e.g. CNN+MN and TA+PN).

We use stochastic gradient descent (Bottou, 2012) as the optimizer in training with the learning rate 1×10^{-5}. The sampling times N_s of Monte Carlo sampling and update step size ϵ are set to 10 and 0.01 respectively. The update times of stochastic gradient Langevin dynamics M is set to 5. We use dropout after the sample encoder and the knowledge encoder to avoid over-fitting; the dropout rate

is set to 0.5. We evaluate the performance of event detection with F_1 and $Accuracy$ scores.

5.2 Main Results

As shown in Table 1, we compare methods on F_1 and $Accuracy$ scores. We observe the followings:

- Our full model AKE-BML outperforms all other methods on both $Accuracy$ and F_1 scores across all settings. Compared with the SOTA method DMN-MPN, AKE-BML achieves a substantial improvement of 15–23 absolute F_1 points in all N-way-M-shot settings. It shows our adaptive knowledge-enhanced Bayesian meta-learning method can effectively utilize external knowledge and adjust it according to the support set, thus build better prototypes of event types. Please see Appendix C and D for a detailed performance analysis over various N-way and M-shot settings.

- With the sample encoders (Bi-LSTM, CNN, DMN and TA) fixed, it can be observed that prototypical networks (PN) consistently outperforms matching networks (MN). DMN-MPN performs better than PN-based methods, because the dynamic memory network can extract key information from the support set through multiple iterations. However, DMN-MPN only considers the information of a few samples in each support set, hence suffering from insufficient sample diversity similar to PN- and MN-based methods.

- TA-BML performs similarly with DMN-MPN under the settings of N-way-5-shot and N-way-10-shot, but slightly worse under the N-way-15-shot setting. One possible explanation is that when the number of samples in the support set is larger, MPN can generate higher-quality prototypes. In addition, the performance of TA-BML is not as good as KB-BML, which shows the importance of introducing external knowledge.

- Compared with KB-BML, our full model AKE-BML can effectively solve the problem of deviation between knowledge and event types, and generate event prototypes with better generalization through knowledge. Compared with TA-BML, which does not incorporate external knowledge, AKE-BML achieves an even larger performance advantage, which further demonstrates the effectiveness of external knowledge.

5.3 Case Study

We present a case study on the dynamic knowledge adaptation between the support set and the

Model	5-Way-5-Shot F_1/Accuracy	5-Way-10-Shot F_1/Accuracy	5-Way-15-Shot F_1/Accuracy	10-Way-5-Shot F_1/Accuracy	10-Way-10-Shot F_1/Accuracy	10-Way-15-Shot F_1/Accuracy
Bi-LSTM+MN[§]	58.19/58.48	61.26/61.45	65.55/66.04	46.43/47.62	51.97/52.60	56.27/56.47
CNN+MN[§]	59.30/60.04	64.81/65.15	68.35/68.58	44.85/45.80	50.14/50.67	54.13/54.49
DMN+MN[§]	66.09/67.18	68.92/69.33	70.88/71.17	52.81/54.12	58.04/58.38	61.63/62.01
TA+MN	66.83/67.55	69.12/69.64	71.13/71.59	53.49/55.47	59.58/60.01	62.41/63.11
Bi-LSTM+PN[§]	62.42/62.72	64.65/64.71	68.23/68.39	53.15/53.59	55.87/56.19	60.34/60.87
CNN+PN[§]	63.69/64.89	69.64/69.74	70.42/70.52	51.12/51.51	53.80/54.01	57.89/58.28
DMN+PN[§]	72.08/72.43	72.47/73.38	73.91/74.68	59.95/60.07	61.48/62.13	65.84/66.31
TA+PN	73.66/73.92	73.81/74.63	75.69/76.31	61.25/61.88	63.89/64.31	66.21/67.59
DMN-MPN[§]	73.59/73.86	73.99/74.82	76.03/76.57	60.98/62.44	63.69/64.43	67.84/68.35
TA-BML	73.37/73.59	74.02/74.63	75.52/75.83	61.43/62.59	63.28/63.96	66.27/67.49
KB-BML	74.63/75.07	75.06/75.63	80.69/81.12	65.99/66.82	67.47/68.08	73.89/74.06
AKE-BML	**88.99/89.36**	**90.10/91.48**	**91.40/92.34**	**84.55/84.94**	**86.03/87.73**	**87.13/87.45**

Table 1: Accuracy and F_1 scores of all compared methods. [§] denotes the results that are directly taken from the original paper (Deng et al., 2020), due to the unavailability of the source code.

corresponding event knowledge to demonstrate our model's ability to learn robust event prototypes.

5.3.1 Predictions for Specific Cases

We select three event types as target categories to illustrate the contributions of each main component of our model. The event types are *Music.Compose*, *Music.Sing* and *Film.Film_Productution*. The sample contexts of *Music.Compose* and *Music.Sing* are similar, while *Music.Compose* and *Film.Film_Productution* share the same frame, which is *Behind_the_scenes*.

As shown in Table 2, only AKE-BML correctly predicts on all samples. TA-BML, the model without introducing knowledge, wrongly predicts the second sample of *Music.Compose* to be *Music.Sing*, due to their similar contexts. By introducing knowledge, both KB-BML and AKE-BML avoid this error, indicating that external knowledge can enrich event information based on the support set. For KB-BML, as *Music.Compose* and *Film.Film_Productution* share the same superordinate frame, the prototype of *Music.Compose* cannot distinguish between *Music.Compose* and *Film.Film_Productution*, so it wrongly classifies the third sample as *Music.Compose*. With our adaptive knowledge offset, AKE-BML can deal with sample similarity and knowledge deviation issues at the same time, thus it correctly classifies all samples.

5.3.2 Visualization of Prototypes

We use Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to reduce the dimensionality of the prototypes, sample encodings and prior knowledge encodings. Figure 4 visualizes five event type prototypes (large solid shapes), their aligned frames in FrameNet (large solid shapes with circle outlines) and some corresponding samples (small
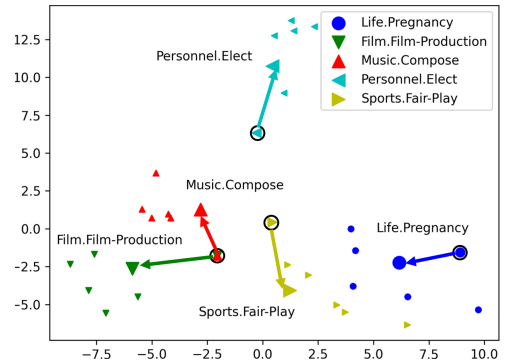

Figure 4: Visualization of event prototypes, prior knowledge, and event samples learned by AKE-BML in the 5-way-5-shot setting. The large, solid shapes denote event prototypes, the large shapes with circle outlines denote the prior knowledge, and the small shapes denote samples. Samples are marked by the color of their corresponding event types. The arrows indicate the adaption of prior knowledge to the prototype. Note that *Music.Compose* and *Film.Film-Production* share the same frame *Behind_the_scenes*.

solid shapes). Each event type and its samples are coded with the same color.

In general, the samples and prototypes belonging to one event type are close in the space and different event types are far away from each other. Prior knowledge is distributed in different places in the space, which roughly determines the distribution of event prototypes. For example, the samples of *Life.Pregnancy* and *Sports.Fair-Play* are close to their respective event prototypes. Meanwhile, the distances between their prior knowledge is large, making their prototypes easily distinguishable.

It can also be seen that the event prototypes are closer to their samples than to the prior knowledge, which reflects the benefits of our proposed learnable knowledge offset. The visualization demonstrates the effectiveness of introducing external

2423

No.	Queries	Event types	Corresponding frame	Prediction		
				TA-BML	KB-BML	AKE-BML
1	Since the mid-20th century, the composers directly **write** sonic material in any of the various electronic media.	Music.Compose		Music.Compose	Music.Compose	Music.Compose
2	Performers of Western popular music **created** earlier popular songs.	Music.Compose	Behind_the_scenes	Music.Sing	Music.Compose	Music.Compose
3	The film is **created** by the director, and may be storyboarded with the help of illustrators and concept artists.	Film.Film_Production		Film.Film_Production	Music.Compose	Film.Film_Production
4	They came with rainbow flags and **sang** the Russian national anthem.	Music.Sing	Performing_arts	Music.Sing	Music.Sing	Music.Sing

Table 2: A case study of three event types. Words in bold indicate candidate trigger words.

Event Type	Explanation of Event Type	Examples in Support Set	λ_t	Corresponding Frame	Frame Definition
Contact.Letter-Communication	Communicate by writing letters.	(1)…*write* this letter to you… (2)…*sends* you this letter… (3)…the letter *passed* from…	0.228	Communication	A Communicator conveys a Message to an Addressee; the Topic and Medium of the communication also may be expressed.
Conflict.Attack	An assailant attacks a victim.	(1)He *shoot* them all with… (2)… *firing* at close… (3)Soldier hacked to death…	0.132	Attack	An Assailant physically attacks a Victim causing or intending to cause the Victim physical damage…
Film.Film-Production	The production process of the film.	(1)… *making* a film… (2)…*creating* the film… (3)…the action is *shot* as…	0.386	Behind_the_scenes	Artists work at various tasks that further the completion of a film, play, television show, or other similar Production
Music.Compose	Music creation process.	(1)…*composes* musical … (2)…*co-written* by him and… (3)… *create* musical sounds…	0.421		

Table 3: λ_t values corresponding to different event types. The larger the λ_t, the greater the dependence of the prototype vector on the support set.

knowledge and our adaptive knowledge offset's ability to balance the impact of the support set and prior knowledge on the event prototypes.

5.3.3 λ_t of Different Event Types

As shown in Formula (5), we use the learnable parameter λ_t to generate knowledge offsets. λ_t accounts for the deviation of the prior knowledge (i.e. a frame) from the event type it represents, and adaptively corrects this deviation using information of the support set. When the frame corresponding to the event type accurately expresses its semantics, the λ_t value should be small. When the knowledge is the super-ordinate frame of the event type (i.e., the frame cannot accurately describe the event semantics), the λ_t value should be large, so that the support set can be used to modify the prior knowledge to ensure that the prototype precisely represents the current event type.

Table 3 shows four different event types, their corresponding frames and λ_t values. The λ_t of *Conflict.Attack* is a small value 0.132, as the event type *Conflict.Attack* closely matches the frame *Attack*. The event type *Contact.Letter-Communication* matches the frame *Communication*. *Communication* does not contain the semantics of "by writing letters", but the core semantics is the same as *Contact.Letter-Communication*. Therefore, λ_t is small, at 0.228, which is still larger than the λ_t of *Conflict.Attack*. The event types *Film.Film-Production* and *Music.Compose* share the same super-ordinate frame *Behind_the_scenes* as prior

knowledge, but the semantics of *Behind_the_scenes* is too abstract for *Film.Film-Production* and *Music.Compose*. Thus, it can be seen that the λ_t values corresponding to these two event types are relatively large: 0.386 for *Film.Film-Production* and 0.421 for *Music.Compose*.

The above cases demonstrate that our model is able to balance the influence of the support set and the knowledge on event prototypes through λ_t, and consequentially obtain highly accurate and generalizable prototypes.

6 Conclusion

In this paper, we proposed an Adaptive Knowledge-enhanced Bayesian Meta-Learning (AKE-BML) method for few-shot event detection. We alleviate the insufficient sample diversity problem in few-shot learning by leveraging the external knowledge base *FrameNet* to learn prototype representations for event types. We further tackle the uncertainty and incompleteness issues in knowledge coverage with a novel knowledge adaptation mechanism.

The comprehensive experimental results demonstrate that our proposed method substantially outperforms state-of-the-art methods, achieving a performance improvement of at least 15 absolute points of F_1. In the future, we plan to extend our proposed AKE-BML method to the few-shot event extraction task, which considers both event detection and argument extraction. We also plan to explore the zero-shot and incremental event extraction scenarios.

7   Acknowledgement

References

Denis A. de Araujo, Sandro José Rigo, and Jorge Luis Victória Barbosa. 2017. Ontology-based information extraction for juridical events with case studies in brazilian legal realm. *Artif. Intell. Law*, pages 379–396.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of ACL*, pages 86–90.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, (Jan):993–1022.

Léon Bottou. 2012. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of ACL&IJCNLP*, pages 167–176.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of WSDM*, pages 151–159.

Shumin Deng, Ningyu Zhang, Luoqiu Li, Hui Chen, Huaixiao Tou, Mosha Chen, Fei Huang, and Huajun Chen. 2021. Ontoed: Low-resource event detection with ontology embedding. In *ACL*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Charles J Fillmore, Srini Narayanan, and Collin F Baker. 2006. What can linguistics contribute to event extraction. In *Proceedings of AAAI*, pages 18–23.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*.

Victor Garcia and Joan Bruna. 2012. Few-shot learning with graph neural networks. *ICLR*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR.

Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2020a. Exploiting the matching information in the support set for few shot event classification. In *Proceedings of PAKDD*, pages 233–245.

Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2020b. Extensively matching for few-shot learning event detection. *arXiv preprint arXiv:2006.10093*.

Jian Liu, Yubo Chen, and Jun Zhao. 2020. Knowledge enhanced event causality identification with mention masking generalizations. In *Proceedings of IJCAI*, pages 3608–3614.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging framenet to improve automatic event detection. In *Proceedings of ACL*, pages 2134–2143.

Shulin Liu, Yang Li, Feng Zhang, Tao Yang, and Xinpeng Zhou. 2019. Event detection without triggers. In *Proceedings of NAACL-HLT*, pages 735–744.

Dongfang Lou, Zhilin Liao, Shumin Deng, Ningyu Zhang, and Huajun Chen. 2021. Mlbinet: A cross-sentence collective event detection network. In *ACL*. Association for Computational Linguistics.

David McClosky, Mihai Surdeanu, and Christopher D Manning. 2011. Event extraction as dependency parsing. In *Proceedings of NAACL-HLT*, pages 1626–1635.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of NAACL-HLT*, pages 300–309.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999.

Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via Bayesian meta-learning on relation graphs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7867–7876. PMLR.

Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. *ICLR*.

Lei Sha, Jing Liu, Chin-Yew Lin, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. Rbpb: Regularization-based pattern balancing method for event extraction. In *Proceedings of ACL*, pages 1224–1234.

Shirong Shen, Guilin Qi, Zhen Li, Sheng Bi, and Lusheng Wang. 2020. Hierarchical chinese legal event extraction via pedal attention mechanism. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 100–113.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of NeurIPS*, pages 4077–4087.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Proceedings of NeurIPS*, pages 2440–2448.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of CVPR*, pages 1199–1208.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of ACL*, pages 5887–5897.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Proceedings of NeurIPS*, pages 3630–3638.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations.

Max Welling and Yee W Teh. 2011. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of ICML*, pages 681–688.

Wang Yan, Jordan Yap, and Greg Mori. 2015. Multi-task transfer methods to improve one-shot learning for multimedia event detection. In *Proceedings of BMVC*, pages 37.1–37.13.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of ACL*, pages 5284–5294.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. 2018. Bayesian model-agnostic meta-learning. In *Proceedings of NeurIPS*, pages 7332–7342.

Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. 2018a. Metagan: An adversarial approach to few-shot learning. In *Proceedings of NeurIPS*, pages 2371–2380.

Yabin Zhang, Hui Tang, and Kui Jia. 2018b. Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. In *Proceedings of ECCV*, pages 241–256.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2edag: An end-to-end document-level framework for chinese financial event extraction. In *Proceedings of EMNLP*, pages 337–346.

Deyu Zhou, Xuan Zhang, and Yulan He. 2017. Event extraction from twitter using non-parametric bayesian mixture model with word embeddings. In *Proceedings of EACL*, pages 808–817.

2426

Appendix

A FrameNet

An important problem in the few-shot event detection task is the insufficient diversity of support set samples. There are only a few labeled samples in the support set, which results in the model unable to construct high-quality prototype features of event types. To address this problem, we introduce the *FrameNet* (Baker et al., 1998) as an external knowledge base of event types. *FrameNet* is a linguistic resource storing information about lexical and predicate-argument semantics. Each frame in *FrameNet* can be taken as a semantic frame of an event type (Liu et al., 2016), which can be used as background knowledge for event types to assist event detection (Liu et al., 2016; Fillmore et al., 2006). Figure 2 shows an example frame defining *Attack*. We can see the arguments involved in an *Attack* event and their roles. The linguistic units (LUs) of the frame *Attack* are the possible trigger words for the corresponding event. The frame is an important complementary source of knowledge to the support set. We match a frame in *FrameNet* to each event type, based on the event name, as its knowledge. In practice, *FrameNet* does not provide complete coverage of all event types, nor does every event type have an exact frame matched in *FrameNet*. For event types that cannot be exactly matched, we assign the frame corresponding to their super-ordinate event. For example, there is no corresponding frame for *Contact.Online-Chat*, so we assign it to the frame *Chatting*, which corresponds to the event type *Contact.Chat*.

B Gradient of posterior distribution

In order to show the change of the prototype vector after adding the knowledge shift, the gradient $\nabla_{\hat{\mathbf{V}}_{T_S}} \log\ p(Y_S|X_S, \hat{\mathbf{V}}_{T_S})p(\hat{\mathbf{V}}_{T_S}|\mathscr{F})$ in iteration

$$\begin{aligned}
\hat{\mathbf{V}}_{T_S} \quad &\leftarrow \hat{\mathbf{V}}_{T_S} + \sqrt{\epsilon}\mathbf{z} \\
&+ \frac{\epsilon}{2}\nabla_{\hat{\mathbf{V}}_{T_S}} \log\ p(Y_S|X_S, \hat{\mathbf{V}}_{T_S})p(\hat{\mathbf{V}}_{T_S}|\mathscr{F}),
\end{aligned} \tag{12}$$

is expanded. For ease of explanation, we only calculate the gradient of the prototype vector $\hat{\mathbf{v}}_t$. We denote the gradient of the original posterior distribution as $\mathbf{g}^o_{\hat{\mathbf{v}}_t}$, the gradient of the knowledge-shifted posterior distribution as $\mathbf{g}^s_{\hat{\mathbf{v}}_t}$. We first calculate $\mathbf{g}^o_{\hat{\mathbf{v}}_t}$:

$$\begin{aligned}
\mathbf{g}^o_{\hat{\mathbf{v}}_t} =& \nabla_{\hat{\mathbf{v}}_t} \log\ p(Y_S|X_S, \hat{\mathbf{V}}_{T_S})p^o(\hat{\mathbf{V}}_{T_S}|\mathscr{F}) \\
=& \nabla_{\hat{\mathbf{v}}_t} \log\ p(Y_S|X_S, \hat{\mathbf{V}}_{T_S}) + \log\ p^o(\hat{\mathbf{V}}_{T_S}|\mathscr{F}) \\
=& \nabla_{\hat{\mathbf{v}}_t} \log\ p(Y_S|X_S, \hat{\mathbf{V}}_{T_S}) + \\
& \nabla_{\hat{\mathbf{v}}_t} \log\ p^o(\hat{\mathbf{V}}_{T_S}|\mathscr{F}) \\
=& \nabla_{\hat{\mathbf{v}}_t} \log\ \prod_{s \in S, y_s = t} p(y_s|x_s, \hat{\mathbf{v}}_t) + \\
& \nabla_{\hat{\mathbf{v}}_t} \log\ p^o(\hat{\mathbf{v}}_t|\mathscr{F}) \\
=& \mathbf{g}^{o,l}_{\hat{\mathbf{v}}_t} + \mathbf{g}^{o,p}_{\hat{\mathbf{v}}_t},
\end{aligned} \tag{13}$$

where $\mathbf{g}^{o,l}_{\hat{\mathbf{v}}_t} = \nabla_{\hat{\mathbf{v}}_t} \log \prod_{s \in S, y_s = t} p(y_s|x_s, \hat{\mathbf{v}}_t)$ and $\mathbf{g}^{o,p}_{\hat{\mathbf{v}}_t} = \nabla_{\hat{\mathbf{v}}_t} \log\ p^o(\hat{\mathbf{v}}_t|\mathscr{F})$. The prior distribution is

$$\begin{aligned}
p^o(\hat{\mathbf{v}}_t|\mathscr{F}) &= \mathcal{N}(\mathbf{v}_t|\mathbf{h}_t, \boldsymbol{I}) \\
&= (2\pi)^{-\frac{d}{2}}\ e^{-\frac{1}{2}(\hat{\mathbf{v}}_t - \mathbf{h}_t)^2},
\end{aligned} \tag{14}$$

the gradient of the logarithm of prior distribution to $\hat{\mathbf{v}}_t$ is:

$$\begin{aligned}
\mathbf{g}^{o,p}_{\hat{\mathbf{v}}_t} &= \left(\log (2\pi)^{-\frac{d}{2}}\right)(\mathbf{h}_t - \hat{\mathbf{v}}_t) \\
&= C\ (\mathbf{h}_t - \hat{\mathbf{v}}_t),
\end{aligned} \tag{15}$$

where $C = \log (2\pi)^{-\frac{d}{2}}$ is a constant, d is the dimension of prototype. The gradient of the log-likelihood on support set is

$$\begin{aligned}
&\log \prod_{s \in S, y_s = t} p(y_s|x_s, \hat{\mathbf{v}}_t) \\
&= \log \prod_{s \in S, y_s = t} \frac{exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_t)}{\sum_{t' \in T_S} exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_{t'})} \\
&= \sum_{s \in S, y_s = t} \log \frac{exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_t)}{\sum_{t' \in T_S} exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_{t'})}.
\end{aligned} \tag{16}$$

The gradient of the log-likelihood to $\hat{\mathbf{v}}_t$ is

$$\begin{aligned}
\mathbf{g}^{o,l}_{\hat{\mathbf{v}}_t} =& \nabla_{\hat{\mathbf{v}}_t} \sum_{s \in S, y_s = t} \log \frac{exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_t)}{\sum_{t' \in T_S} exp((\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_{t'})} \\
=& \sum_{s \in S, y_s = t} \nabla_{\hat{\mathbf{v}}_t} \log \frac{exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_t)}{\sum_{t' \in T_S} exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_{t'})} \\
=& \sum_{s \in S, y_s = t} \nabla_{\hat{\mathbf{v}}_t}(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_t) \\
& - \nabla_{\hat{\mathbf{v}}_t} \log \sum_{t' \in T_S} exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_{t'}) \\
=& \sum_{s \in S, y_s = t} \boldsymbol{\mathcal{E}}(x_s) - \frac{\boldsymbol{\mathcal{E}}(x_s)exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_t)}{\sum_{t' \in T_S} exp(\boldsymbol{\mathcal{E}}(x_s) \cdot \hat{\mathbf{v}}_{t'})} \\
=& \sum_{s \in S, y_s = t} (1 - p^{(t)}_s)\boldsymbol{\mathcal{E}}(x_s),
\end{aligned} \tag{17}$$

where $p_s^{(t)} = p(y_s|x_s, \hat{\mathbf{v}}_t)$ is the probability of correct classification of sample s in support set. Then we get

$$\mathbf{g}_{\hat{\mathbf{v}}_t}^o = \sum_{s \in S, y_s = t} (1 - p_s^{(t)}) \mathcal{E}(x_s) + C(\mathbf{h}_t - \hat{\mathbf{v}}_t). \tag{18}$$

Then we calculate $\mathbf{g}_{\hat{\mathbf{v}}_t}^s$, The only difference between calculating $\mathbf{g}_{\hat{\mathbf{v}}_t}^s$ and $\mathbf{g}_{\hat{\mathbf{v}}_t}^o$ is $\mathbf{g}_{\hat{\mathbf{v}}_t}^s$ use the knowledge-shifted prior distribution

$$\begin{aligned} p(\hat{\mathbf{v}}_t|\mathscr{F}) &= \mathcal{N}(\mathbf{v}_t|\mathbf{h}_t + \Delta\mathbf{h}_t, \mathbf{I}) \\ &= (2\pi)^{-\frac{d}{2}} e^{-\frac{1}{2}(\hat{\mathbf{v}}_t - \mathbf{h}_t - \Delta\mathbf{h}_t)^2}. \end{aligned} \tag{19}$$

Same as the original posterior gradient, we have

$$\begin{aligned} \mathbf{g}_{\hat{\mathbf{v}}_t}^s &= \nabla_{\hat{\mathbf{v}}_t} \log \prod_{s \in S, y_s = t} p(y_s|x_s, \hat{\mathbf{v}}_t) \\ &\quad + \nabla_{\hat{\mathbf{v}}_t} \log p(\hat{\mathbf{v}}_t|\mathscr{F}) \\ &= \mathbf{g}_{\hat{\mathbf{v}}_t}^{s,l} + \mathbf{g}_{\hat{\mathbf{v}}_t}^{s,p}, \end{aligned} \tag{20}$$

where $\mathbf{g}_{\hat{\mathbf{v}}_t}^{s,l} = \mathbf{g}_{\hat{\mathbf{v}}_t}^{o,l} = \sum_{s \in S, y_s = t}(1 - p_s^{(t)})\mathcal{E}(x_s)$. The gradient of the logarithm of knowledge-shifted prior distribution to $\hat{\mathbf{v}}_t$ is:

$$\begin{aligned} \mathbf{g}_{\hat{\mathbf{v}}_t}^{s,p} &= \left(\log(2\pi)^{-\frac{d}{2}}\right)(\mathbf{h}_t + \Delta\mathbf{h}_t - \hat{\mathbf{v}}_t) \\ &= C(\mathbf{h}_t + \boldsymbol{\lambda}_t \odot (\mathbf{m}_t - \mathbf{h}_t) - \hat{\mathbf{v}}_t) \\ &= C((\mathbb{1} - \boldsymbol{\lambda}_t) \odot \mathbf{h}_t - \hat{\mathbf{v}}_t) + C\boldsymbol{\lambda}_t \odot \mathbf{m}_t, \end{aligned} \tag{21}$$

where $\mathbb{1}$ is a $|\mathbf{h}_t|$-dimensional vector, and each element of $\mathbb{1}$ is 1. then we get

$$\begin{aligned} \mathbf{g}_{\hat{\mathbf{v}}_t}^s &= \mathbf{g}_{\hat{\mathbf{v}}_t}^{s,l} + \mathbf{g}_{\hat{\mathbf{v}}_t}^{s,p} \\ &= \sum_{s \in S, y_s = t} (1 - p_s^{(t)})\mathcal{E}(x_s) \\ &\quad + C((\mathbb{1} - \boldsymbol{\lambda}_t) \odot \mathbf{h}_t - \hat{\mathbf{v}}_t) \\ &\quad + C\boldsymbol{\lambda}_t \odot \mathbf{m}_t, \end{aligned} \tag{22}$$

bring $\mathbf{m}_t = \frac{1}{M}\sum_{s \in S, y_s = t} \mathcal{E}(x_s)$ into the above formula, we get

$$\begin{aligned} \mathbf{g}_{\hat{\mathbf{v}}_t}^s &= \sum_{s \in S, y_s = t} \left[(1 - p_s^{(t)})\mathcal{E}(x_s) + \frac{C\boldsymbol{\lambda}_t}{M} \odot \mathcal{E}(x_s)\right] \\ &\quad + C((\mathbb{1} - \boldsymbol{\lambda}_t) \odot \mathbf{h}_t - \hat{\mathbf{v}}_t). \end{aligned} \tag{23}$$

Note that, when knowledge adaption is not used, the form of the prior knowledge distribution of the prototype is as follows,

$$p^o(\mathbf{V}_{T_S}|\mathscr{F}) = \prod_{t \in T_S} p^o(\mathbf{v}_t|\mathbf{h}_t) = \prod_{t \in T_S} \mathcal{N}(\mathbf{v}_t|\mathbf{h}_t, \mathbf{I}). \tag{24}$$

To intuitively show the influence of the knowledge-adapted posterior distribution on the prototype vector, we expand the gradient $\nabla_{\hat{\mathbf{V}}_{T_S}} \log p(Y_S|X_S, \hat{\mathbf{V}}_{T_S})p(\hat{\mathbf{V}}_{T_S}|\mathscr{F})$ in Eqn 12. For ease of explanation, we only calculate the gradient of the prototype vector $\hat{\mathbf{v}}_t$. Denote the gradient of the original posterior distribution without knowledge adaption as $\mathbf{g}_{\hat{\mathbf{v}}_t}^o$,

$$\mathbf{g}_{\hat{\mathbf{v}}_t}^o = \sum_{s \in S, y_s = t} (1 - p_s^{(t)})\mathcal{E}(x_s) + C(\mathbf{h}_t - \hat{\mathbf{v}}_t), \tag{25}$$

and the gradient of the knowledge-adapted posterior distribution as $\mathbf{g}_{\hat{\mathbf{v}}_t}^s$ from Eqn 23.

Comparing Eqn 23 and Eqn 25, it can be seen that the posterior distribution without knowledge adaption cannot dynamically balance the influence of the knowledge and the support set on the prototype vector, whereas the knowledge-adapted posterior distribution can adjust their contributions to the prototype vector through $\boldsymbol{\lambda}_t$. The parameters in Eqn 6 will be updated by the log-likelihood on the query set. This allows the model to reasonably choose the weight of the knowledge and the support set, and obtain prototype vectors with better generalization.

C M-shot Evaluation

In this section, we illustrate the effectiveness of adaptive knowledge-enhanced Bayesian meta-learning under different M-shot settings, such as N-way-5-shot, N-way-10-shot and N-way-15-shot. As shown in Table 1 in the main paper, as M increases, the performance of all models improves, which shows that increasing the number of samples in the support set can provide more pertinent event type-related features. At the same time, it can be seen that from 15-shot to 5-shot, the previous methods suffer a significantly larger performance degradation than AKE-BML. This observation shows our model's strong robustness against low sample diversity due to the incorporation of external knowledge.

The performance of KB-BML is close to that of DMN-MPN in the case of N-way-5-shot and N-way-10-shot, and the performance is better in the case of N-way-15-shot. This can be attributed to two factors: (1) the introduction of knowledge can improve the generalization of event prototypes; and (2) increasing the number of samples can reduce the impact of the deviation between knowledge and
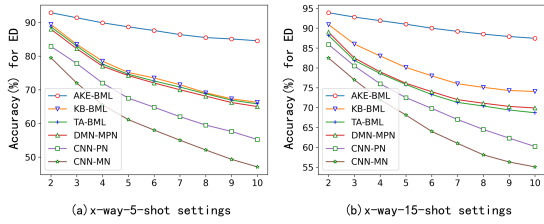
Figure 5: N-way ($N = 2, \ldots, 10$) evaluation and fixed shot numbers. (a) N-way-5-shot. (b) N-way-15-shot.

enhanced Bayesian meta-learning is more robust to the changes in the number of ways.

event types. When the support set is sufficiently large, the samples in the support set can compensate for the deviation between knowledge and event types, and the knowledge can also improve the generalization of the prototype vector. However, when M is small, the deviation between knowledge and event types will affect the quality of the prototype vectors.

AKE-BML can well balance the effects of samples and knowledge on the event type prototypes. It can be seen that when M is small, the performance of AKE-BML does not decline as quickly as other models, which also proves the effectiveness of knowledge in dealing with the problem of insufficient diversity of the support set. At the same time, compared with KB-BML, our adaptive knowledge offset can effectively use the information in the support set to correct the knowledge deviation.

D N-Way Evaluation

Figure 5 also illustrates model performance with respect to different way values (i.e. N), while fixing the shot values. It can be seen from the figure that when N increases, the performance of previous models decreases faster than AKE-BML, which shows that those models, only relying on the support set, cannot generate more recognizable event prototypes. The performance of KB-BML also declines significantly when N increases. This is because many event types can only be partially aligned in FrameNet, to its super-ordinate frame, which causes the event prototypes to be indistinguishable to similar event types.

On the contrary, the performance of AKE-BML does not decrease significantly when N increases, which shows that our adaptive knowledge-enhanced Bayesian meta-learning method can enhance the distinguishability of prototype vectors through the learnable knowledge offset. These results indicate that our adaptive knowledge-