Towards Cross-lingual Social Event Detection with Hybrid Knowledge Distillation

JIAQIAN REN, China Mobile (Hangzhou) Information Technology Co., Ltd., China
HAO PENG*, Beihang University, China
LEI JIANG, Institute of Information Engineering, Chinese Academy of Sciences, China
ZHIFENG HAO, University of Shantou, China
JIA WU, Macquarie University, Australia
SHENGXIANG GAO, Kunming University of Science and Technology, China
ZHENGTAO YU, Kunming University of Science and Technology, China
QIANG YANG, Hong Kong University of Science and Technology, China and WeBank Co., Ltd.,, China

Recently published graph neural networks (GNNs) show promising performance at social event detection tasks. However, most studies are oriented toward monolingual data in languages with abundant training samples. This has left the common lesser-spoken languages relatively unexplored. Thus, in this work, we present a GNN-based framework that integrates cross-lingual word embeddings into the process of graph knowledge distillation for detecting events in low-resource language data streams. To achieve this, a novel cross-lingual knowledge distillation framework, called CLKD, exploits prior knowledge learned from similar threads in English to make up for the paucity of annotated data. Specifically, to extract sufficient useful knowledge, we propose a hybrid distillation method that consists of both feature-wise and relation-wise information. To transfer both kinds of knowledge in an effective way, we add a cross-lingual module in the feature-wise distillation to eliminate the language gap and selectively choose beneficial relations in the relation-wise distillation to avoid distraction caused by teachers' misjudgments. Our proposed CLKD framework also adopts different configurations to suit both offline and online situations. Experiments on real-world datasets show that the framework is highly effective at detection in languages where training samples are scarce.

CCS Concepts: • **Information systems** → **Social networks**; **Data mining**; **Web mining**; *Clustering and classification*; • **Computing methodologies** → **Artificial intelligence**; **Artificial intelligence**; • **Applied computing** → **Document management and text processing**;

*Corresponding author

Authors' addresses: J. Ren, China Mobile (Hangzhou) Information Technology Co., Ltd., No. 1600 Yuhangtang Road, Hangzhou, 311121, China; E-mail: renjiaqian@iie.ac.cn; H. Peng, School of Cyber Science and Technology, Beihang University, No. 37 Xue Yuan Road, Haidian District, Beijing, 100191, China; E-mail: penghao@buaa.edu.cn; L. Jiang, Institute of Information Engineering, Chinese Academy of Sciences, No. 19 Shu Cun Road, Haidian District, Beijing, China; E-mail: jianglei@iie.ac.cn; Z. Hao, College of Science, University of Shantou, No. 243, University Road, Shantou, 515063, China; E-mail: zfhao@stu.edu.cn; J. Wu, Department of Computing, Macquarie University, Balaclava Road, North Ryde, NSW, 2109, Australia; E-mail: jia.wu@mq.edu.au; S. Gao, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China. E-mail:gaoshengxiang.yn@foxmail.com; Z. Yu, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China. E-mail:ztyu@hotmail.com; Q. Yang, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, China, and AI Group, WeBank Co., Ltd., China; E-mail: qyang@cse.ust.hk.

1556-4681/2024/8-ART37 $15.00
https://doi.org/XXXXXXX.XXXXXXX

Additional Key Words and Phrases: Social event detection, low-resource languages, cross-lingual knowledge distillation, graph neural network

ACM Reference Format:
Jiaqian Ren, Hao Peng, Lei Jiang, Zhifeng Hao, Jia Wu, Shengxiang Gao, Zhengtao Yu, and Qiang Yang. 2024. Towards Cross-lingual Social Event Detection with Hybrid Knowledge Distillation. *ACM Trans. Knowl. Discov. Data.* 1, 1, Article 37 (August 2024), 37 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The task of social event detection aims to extract information about important and often newsworthy, real-world occurrences from social media data streams [1]. It benefits greatly in fields like marketing, disaster risk management, public opinion analysis, and decision-making. Due to its wide applications, social event detection has been the research hot spot for the last decade. When this task first comes to the fore, most studies on the subject treat event detection as either an incremental clustering problem [1, 38, 54, 85], a community detection problem [25, 44–46, 84], or a topic modeling problem [7, 16, 20, 89]. However, in using these approaches, scholars are ignoring much of the rich semantics and structural information social streams contain. Hence, more recently, researchers have turned to the expressive power of graph neural networks (GNNs) [18, 69] – both homogeneous and heterogeneous – to capture this information [11, 21, 58, 60, 61, 63, 64]. Among these new approaches, one called KPGNN [11] has successfully addressed the incremental detection of events in Twitter data from a knowledge-preserving perspective.

This is tremendous progress, but there are still questions to be answered. For instance, in general, GNN-based event detection [11, 58, 60] is oriented toward monolingual data where there are ample instances for training (i.e., high-resource languages), particularly English. What can we do for languages where training samples are scarce? As events and news are not limited to widely spoken languages. People speaking low-resource languages also deserve access to relevant and timely information. Developing methods that can handle low-resource languages ensures a more inclusive and global perspective in event detection and media monitoring. Thus, there is an urge demand to strengthen detection capabilities in low-resource languages.

Turning to the low-resource language problems, though multilingual models such as M-BERT from BERT [23] and XLM-R [19] and mT5 [79] have certain abilities to handle them, their results are still far from satisfactory. One of the main limitations of multilingual models in handling less-spoken languages lies in the underrepresentation of low-resource languages in the subword vocabulary and the common vector space [17]. For this, we opt to leverage cross-lingual word embeddings (CLWE) to solve low-resource language detection problems. As shown in Fig. 1, we aim to leverage knowledge learned from high-resource languages to strengthen detection capabilities in low-resource languages. Our method can be referred to as cross-lingual knowledge distillation. This concept draws inspiration from cross-modal knowledge distillation, as proposed in work [31]. In cross-modal knowledge distillation, priors from a model trained with a more robust modality (referred to as the "teacher") are used to guide the segmentation of another model trained with a less reliable modality (referred to as the "student") by aligning their predicted distributions. Usually, the superior modality should have an adequate amount of high-quality data while the student modalities only have access to small or poor-quality datasets. Despite their tremendous variations and diversity, languages do share some things in common. For us, the superior "teacher" modality is English, while the weak "student" modality is the low-resource language – more specifically, in our experiments, these are Arabic and French. Hence, we propose a cross-lingual knowledge distillation (CLKD) framework that leverages the prior knowledge learned from a high-resource language (i.e. English) to make up for a lack of annotated samples in the low-resource languages (i.e. Arabic and French). Meanwhile, to guarantee effective and beneficial guidance during the training process of student models, in CLKD, we carefully consider the two key factors of distillation: what knowledge is helpful for the final detection and how to effectively transfer

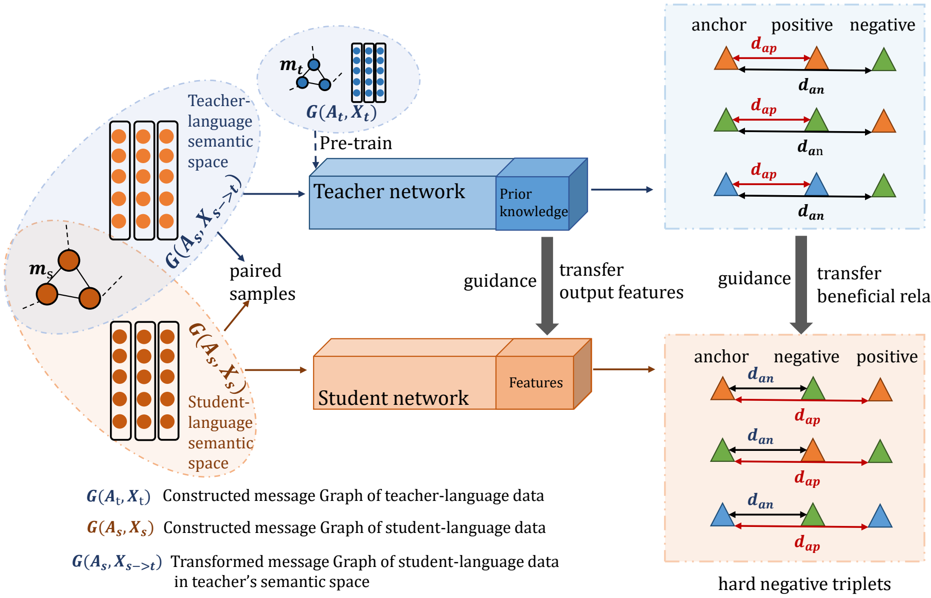

m_t
$G(A_t, X_t)$
Pre-train
Teacher-language semantic space
$G(A_s, X_{s->t})$
m_s
paired samples
$G(A_s, X_s)$
Student-language semantic space
Teacher network
Prior knowledge
guidance
transfer output features
Student network
Features
anchor
positive
negative
d_{ap}
d_{an}
guidance
transfer beneficial relations
anchor
negative
positive
d_{an}
d_{ap}
hard negative triplets
$G(A_t, X_t)$ Constructed message Graph of teacher-language data
$G(A_s, X_s)$ Constructed message Graph of student-language data
$G(A_s, X_{s->t})$ Transformed message Graph of student-language data in teacher's semantic space

Fig. 1. **An illustration of the cross-lingual knowledge distillation framework.** We utilize the priors learned from the teacher model pre-trained on teacher language data to segment the student model trained on student language data. To transfer sufficient information that helps detection, the distillation knowledge includes two parts: 1) the learned output features of the teacher model, and 2) the selected beneficial relations extracted by the teacher model.

knowledge. Most previous studies that apply KD in detection tasks [15, 22, 29, 77] directly treat the teacher's representations of a hidden or the last layer as extra supervision. Nevertheless, due to the existence of language discrepancy, the exact match of features may be detrimental. Thus, as demonstrated in Fig. 1, in our work, when transferring output, we also add the cross-lingual module to alleviate discrepancy. Besides, some recent works [28, 56, 71, 76] argue that representation itself is not informative enough. To get sufficient knowledge, they propose relational knowledge distillation. However, these methods often take a large time consumption and ignore the mispredictions made by the teacher, which may disturb the training of students. We solve this problem by establishing contrasting activation criteria for negative pairs and positive pairs to transfer only beneficial relations.

Our proposed CLKD framework also adopts different configurations to suit both offline and online situations. As shown in Fig. 4(a), this is an offline solution that combines CLWE with knowledge distillation and follows a Teacher-Student structure – offline because traditional Teacher-Student knowledge distillation requires a two-stage training strategy that does not suit online situations. As aforementioned, before inputting student language samples into the pre-trained teacher model, we leverage the cross-lingual module to alleviate the difficulties caused by language discrepancy. To remove distractions caused by misjudgments from the teacher model, only beneficial relations are transferred. The framework also incorporates a Mutual-Learning structure that follows a one-stage learning strategy, which allows it to be applied to detect events in an incremental setting. As shown in Fig. 4(b), the online mutual distillation thoroughly exploits language-shared knowledge through mutual instruction between two GNN peers to facilitate each network learning. The distillation works in both directions.

Experimental results demonstrate the superiority of our models in detecting events from low-resource language data. Specifically, to evaluate the CLKD framework for low-resource transfer scenarios, we use three publicly-available datasets of Twitter data – English [49], French [48], and Arabic [2]. The results show that our CLKD framework, which relies on prior knowledge from English, substantially improves detection in French and Arabic. Note that all code and data used for these tests are publicly available from Github[1]. The main contributions of this paper therefore include:

(1) A novel cross-lingual knowledge distillation (CLKD) framework for social event detection in low-resource languages. This framework effectively integrates cross-lingual word embeddings into the graph knowledge distillation process. It successfully detects events in a low-resource language by borrowing prior knowledge from a high-resource language (i.e., English).
(2) A hybrid graph knowledge distillation strategy. This strategy takes into account both feature-wise and relation-wise information and focuses on how to effectively transfer them. In feature-wise distillation, we add a cross-lingual module to tackle language discrepancy. In relation-wise distillation, we only transfer beneficial relations by establishing contrasting activation criteria for negative pairs and positive pairs in those carefully selected triplets.
(3) Experimental results on two real-world datasets – one in French and one in Arabic demonstrate that the performance of social event detection can be greatly improved in situations of data paucity by combining a GNN model with cross-lingual knowledge distillation into a CLKD framework.

The rest of the paper is organized as follows. We first give the preliminaries in Section 2. Second, we present the CLKD framework in Section 3. Third, we evaluate the proposed models in Section 4. Fourth, we describe the extensions and limitations of our work in Section 5. Then, we introduce the related work in Section 6. Finally, we conclude this study and discuss the future works in Section 7.

2 PRELIMINARY

This section sets out the main notations used in this paper, as shown in Table I. Similar to work [11], we first describe the definitions of Social Stream and Social Event, then give the definition of Social Event Detection and of significant importance, Cross-lingual Social Event Detection.

Definition 2.1. *A **social stream** $S = M_0, ..., M_{i-1}, M_i, ...$ is a continuous and temporal sequence of blocks of social messages. Each block M_i contains messages that arrive during a certain period, i.e., $M_i = \{m_j | t_{m_j} \in [t^i, t^{i+1})\}$, where t_{m_j} denotes the posted time of message m_j. t^i and t^{i+1} are the split beginnig time and ending time of block M_i.*

Definition 2.2. *A **low-resource social event** e **in a social stream** is a set of correlated messages that discuss the same real-world happening in a less-spoken language, i.e., $e = \{m_j | e_{m_j} = e\}$, where e_{m_j} denotes the event which message m_j belongs to. In this work, it is assumed that each message discusses at most one event.*

Definition 2.3. *A **social event detection** algorithm learns a model $f(M_i; \theta)$, where θ denotes the parameters in the network, and M_i denotes a message block containing a set of social events e.*

Definition 2.4. *For **cross-lingual social event detection**, priors learned from a high-resource language are used to segment the performance of a low-resource language. The message block in the high-resource language is denoted as M_i^h and as M_i^l for the low-resource language. Both M_i^h and M_i^l are monolingual. The model $f(M_i^h; \theta^h)$ is learned through a specific social event detection algorithm. The model for the low-resource language M_i^l, $f(M_i^l; \theta^l)$, is learned through the same event detection algorithm but under the supervision of the output of $f(M_i^{l\rightarrow h}; \theta^h)$ in both feature-wise and relation-wise ways, where θ^h is the already trained network parameters by high-resource language data, and $M_i^{l\rightarrow h}$ is the transformed low-resource language data in high-resource language semantic space.*

[1] https://github.com/RingBDStack/CLKD

Table I. NOTATIONS.

Symbol	Meaning
$S; M$	Social stream; Message block
$M^h; M^l$	Message block in high-resource language; Message block in low-resource language
$M^{l\to h}$	Message block in which the original low-resource language data is transformed into the high-resource language semantic space
m	Message or message as a node type
$t_{m_j}; [t^i, t^{i+1})$	The arriving time of message m_j; Time period of the i-th message block
$m_{j+}; m_{j-}$	A message in the same class as m_j; A message not in the same class as m_j
$e; e_{m_j}$	Event; The specific event message m_j belongs to
$\boldsymbol{X}$	Non-English (French or Arabic) word embeddings
$\boldsymbol{Y}$	English word embeddings
$\boldsymbol{W}$	The cross-lingual transformation matrix
a	The additional margin set in the triplet loss
w	The window size for maintaining the model
$b; m_b; B;$	Mini-batch number; A set of messages in the b-th mini-batch; Total number of mini-batches
$\boldsymbol{\theta}$	The parameters of GNN model
$\boldsymbol{\theta^s}; \boldsymbol{\theta^t}$	The parameters of student model; The parameters of teacher model
$\boldsymbol{\theta^{p1}}; \boldsymbol{\theta^{p2}}$	The parameters of peer1 model; The parameters of peer2 model
$f(M_i, \boldsymbol{\theta})$	The detection model for block M_i in which the learned parameters of GNN model is $\boldsymbol{\theta}$
$N; N_e$	The total number of nodes in message graph G; The total number of edges in G
$G(\boldsymbol{X}, \boldsymbol{A})$	Message Graph G; Initial attribute features of the messages $\boldsymbol{X}$; The adjacency matrix $\boldsymbol{A}$
$G(\boldsymbol{X_t}, \boldsymbol{A_t})$	Constructed message Graph of teacher-language data
$G(\boldsymbol{X_s}, \boldsymbol{A_s})$	Constructed message Graph of student-language data
$G(\boldsymbol{X_{s\to t}}, \boldsymbol{A_s})$	Transformed message Graph of student-language data in teacher's semantic space
$G(\boldsymbol{X_{p1}}, \boldsymbol{A_{p1}})$	Constructed message Graph of peer1-language data
$G(\boldsymbol{X_{p1\to p2}}, \boldsymbol{A_{p1}})$	Transformed message Graph of peer1-language data in peer2's semantic space
$G(\boldsymbol{X_{p2}}, \boldsymbol{A_{p2}})$	Constructed message Graph of peer2-language data
$G(\boldsymbol{X_{p2\to p1}}, \boldsymbol{A_{p2}})$	Transformed message Graph of peer2-language data in peer1's semantic space
$\boldsymbol{h}$	The final representation of messages
$\boldsymbol{h^{stu}}; \boldsymbol{h^{tea}}$	The final representation of messages of student network; The final representation of messages of teacher network
$\boldsymbol{h^{p1}}; \boldsymbol{h^{p2}}$	The final representation of peer1-language data of peer1 network; The final representation of peer2-language data of peer2 network
$\boldsymbol{h^{p1\to p2}}$	The final representation of peer1-language data in peer2's semantic space of peer2 network
$\boldsymbol{h^{p2\to p1}}$	The final representation of peer2-language data in peer1's semantic space of peer1 network
$\mathcal{L}_t$	Triplet loss
$\mathcal{L}_{\boldsymbol{KD_f}}$	Feature-wise knowledge distillation loss
$\mathcal{L}_{\boldsymbol{KD_r}}$	Relation-wise knowledge distillation loss
$\mathcal{L}_{\boldsymbol{total}}$	Total loss

3 METHODOLOGY

This section sets out the CLKD framework for detecting events in single low-resource languages. The description begins with the overall vanilla event detection architecture (Section 3.1). Section 3.2 introduces the cross-lingual module. In Section 3.3, we describe the dedicated cross-lingual knowledge distillation (CLKD) framework in both offline and online situations in detail. Generally, the CLKD framework aims to strengthen representation learning of messages in low-resource languages by distilling useful knowledge learned from the high-resource language (i.e., English). The distillation process contains both feature-wise and relation-wise parts. Section 3.4 sets out

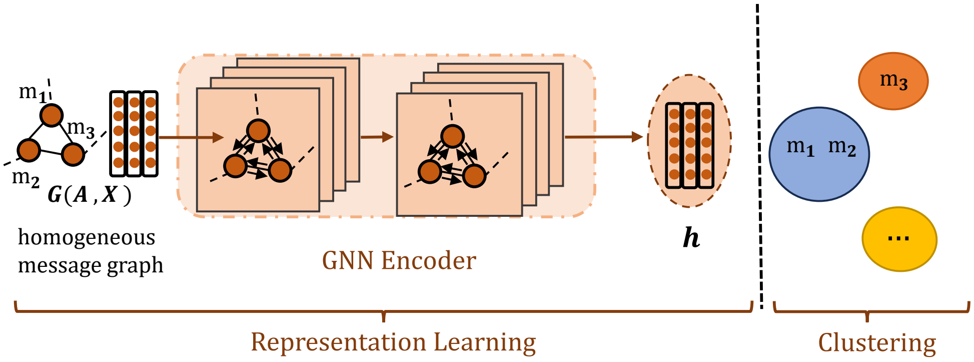


Fig. 2. **The overall vanilla detection architecture.** The whole detection architecture includes two parts: a graph-based message representation learning part and a message clustering part.

the life cycle and three stages of the CLKD framework's operation. And we conclude in Section 3.5 with a time complexity analysis.

3.1 The Overall Vanilla Detection Architecture

As shown in Fig. 2, the overall vanilla detection architecture consists of two parts: the graph-based message representation learning part and the message clustering part.

To learn complementary message representations, we first construct a message graph and then adopt a specific GNN encoder. During the graph construction process, to capture as much information as possible from the social stream, a series of useful elements, including users, named entities, and hashtags, are extracted at the data processing stage. They are leveraged to build correlations between messages. For example, if two message nodes link to the same user, entity, or hashtag, a connection is formed between these two message nodes. In this way, we obtain a handled homogeneous message graph $G(\boldsymbol{X}, \boldsymbol{A})$, where $\boldsymbol{A}$ is the adjacency matrix and $\boldsymbol{X}$ denotes the initial attribute features of the nodes, which is the average of the message's word embeddings. Now we have a homogenous message graph with initial node attribute features. The next step is to select a proper GNN encoder and adopt a scalable training strategy. Considering the powerful capability of the multi-head mechanism in fully exploiting representation subspaces, we implement a 2-layer multi-head GAT [74] network as the GNN encoder to learn message representations. Cross-entropy loss only suits the situation where the total number of events is pre-known and fixed; thus, it cannot handle stream data. We turn to use triplet loss [67] for the backpropagation of the encoder. $\mathcal{L}_t$ denotes the loss calculated by a set of triplets <anchor, positive, negative> based on true labels. The objective $\mathcal{L}_t$ is to build triplets $< m_j, m_{j+}, m_{j-} >$ and to keep the distance between the anchor's representation $\boldsymbol{h}_{\boldsymbol{m}_j}$ and the positive's $\boldsymbol{h}_{\boldsymbol{m}_{j+}}$ smaller than the distance between the anchor's $\boldsymbol{h}_{\boldsymbol{m}_j}$ and the negative's $\boldsymbol{h}_{\boldsymbol{m}_{j-}}$. Here, positive denotes a message whose label is the same as the anchor's, while negative is different from the anchor's. Note that, the performance of triplet loss heavily relies on triplet selection strategies, i.e., training with randomly selected triplets faces slow convergence while selecting hardest positive and negative samples often leads to bad local minima [34, 36, 83]. Hence, to make a balance between them, we adopt the so-called Batch Random Hard triplet selection strategy. Specifically, for each message m_j in a batch, we select a positive message m_{j+} and randomly sample a hard negative message m_{j-} within the batch whose distance to m_j is smaller than the distance between m_j and m_{j+}. The loss can be formulated as:

$$\mathcal{L}_{\mathrm{t}} = \sum_{\substack{<m_j,m_{j+},m_{j-}>\in\{\text{hard triplets}\}\\ <m_j,m_{j+},m_{j-}>\in\{m_b\}}} \max\left\{\mathcal{D}\left(\boldsymbol{h}_{m_j},\boldsymbol{h}_{m_{j+}}\right) - \mathcal{D}\left(\boldsymbol{h}_{m_j},\boldsymbol{h}_{m_{j-}}\right) + a, 0\right\}, \qquad (1)$$

where a denotes the additional margin set in the triplet loss.

After the representation learning of messages, clustering methods are employed to detect different events. Both distance-based clustering algorithms like K-Means and density-based ones like DBSCAN can be used to cluster the representations. As illustrated in Fig. 2, the resulting clusters of messages represent the detected social events.

The above-depicted architecture has a strong capability to detect events in high-resource language. To further strengthen its ability to handle low-resource language data, in this work, we further strengthen the graph-based representation learning part and propose a cross-lingual knowledge distillation framework, which integrates a cross-lingual module into the knowledge distillation process.

3.2 Cross-Lingual Module

This section describes the cross-lingual module. Different languages do not share a joint vector space, which means, the learned representations of the same thing expressed in different languages vary greatly. Also, it means sharing knowledge learned in one language with another is problematic. Considering the Limited proficiency of multilingual models such as M-BERT, XLNET, and mT5 in handling low-resource languages, we here leverage cross-lingual word embedding techniques to formulate the cross-lingual module. As shown in Fig. 3, the process begins by training a monolingual embedding model for each language considered. This could be done through any of the well-known word embedding algorithms (e.g., Word2Vec [50], GloVe [62], or fastText [9]). We use the pre-trained language models in spacy[2], which are trained by GloVe. After deriving all isolated monolingual vector spaces, the mapping between each non-English language and English pair is learned in both directions. To explore the most appropriate transformations, we try both linear and nonlinear CLWE methods.

The goal with the linear mapping is to learn a matrix $\boldsymbol{W}$ between the source space and the target space such that $\boldsymbol{W} = argmin\,||\boldsymbol{WX} - \boldsymbol{Y}||$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the embeddings for the source words and the target words respectively, see Fig. 3(b). This linear approach follows the assumption that the source and target embedding spaces are approximately isomorphic, and will likely suit languages that follow similar grammatical and vocabulary structures. We choose MUSE [42] to learn the linear mappings between all language pairs as it has yielded great results in aligning two monolingual embedding spaces. The cross-lingual word embeddings are created by Generative Adversarial Networks (GANs), where the generator learns the transformation matrix $\boldsymbol{W}$, ensuring that the transformed non-English embeddings $\boldsymbol{WX}$ approximate the English semantic embeddings $\boldsymbol{Y}$ as closely as possible. The discriminator tries to classify whether the embeddings from the English embedding distribution are real ones or transformed. Do note, however, that MUSE is not the only choice. Other linear methods like VecMap [4] would also be suitable.

For the non-linear mapping, we choose LNMAP [52]. LNMAP is a model that operates independently of isomorphic assumption. As shown in Fig. 3(c), it comprises two auto-encoders with non-linear hidden layers for each language. The auto-encoders are first trained independently in a self-supervised way to induce the latent code space of the respective languages. A small seed dictionary is then used to learn the non-linear mappings between the two learned latent spaces of the two auto-encoders. Specifically, the non-linear mappings are implemented as feed-forward neural networks with non-linear activation layers. For more details can refer to [52]. As the first non-linear cross-lingual word embedding method, LNMAP has shown outstanding performance with many language pairs including far-distance language pairs. In experiments, we use the linear and nonlinear

[2] https://spacy.io/api/annotation#section-named-entities

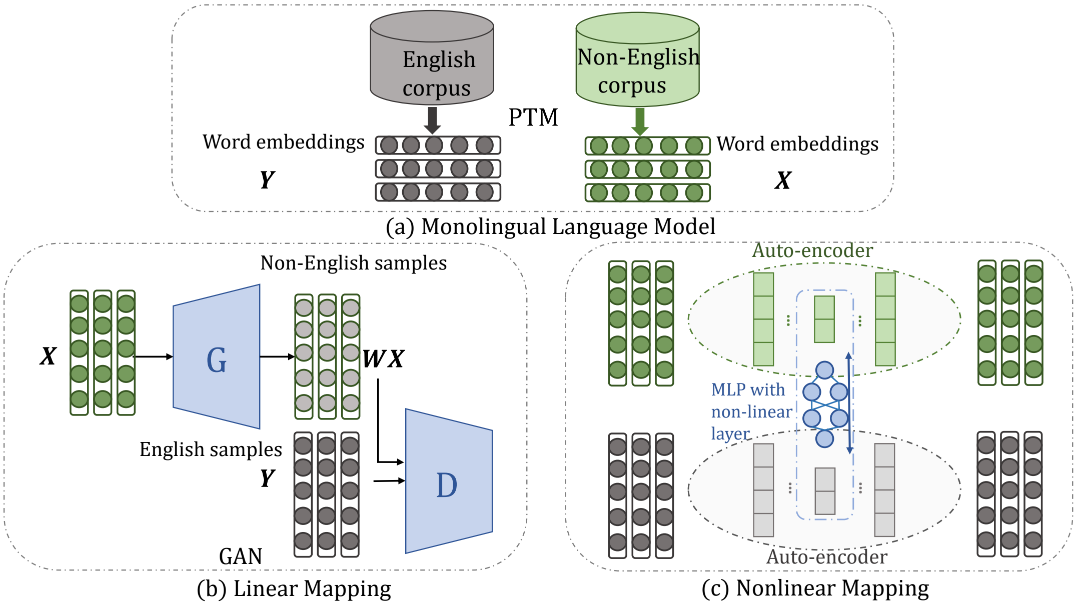


Fig. 3. **The cross-lingual module.** (a) shows the word embeddings learned by some monolingual embedding models for each language considered. (b) depicts the linear mapping between languages and (c) shows the non-linear mapping.

mappings learned by MUSE and LNMAP for each non-English-English pair to build the cross-lingual module in both directions.

3.3 Cross-Lingual Knowledge Distillation

The CLKD framework is for cases where one wishes to detect events in a low-resource language. The procedure essentially borrows knowledge learned from a high-resource language (English) and uses it to assist learning in a low-resource language. There is a large quantity of English event data that is already labeled. As shown in Fig. 4, we devise two distillation architectures – one Teacher-Student configuration, intended for offline use, and the other Mutual-Learning configuration, designed for online situations. However, the backbone of both architectures is the same GNN encoder, which is a 2-layer multi-head GAT model. And the basic training loss is the described triplet loss in Section 3.1. Meanwhile, the distillation loss designed in both architectures is the same except for the distillation direction. In the Teacher-Student configuration, the distillation flows in one direction (i.e., from teacher to student) while it goes in both directions in the Mutual-Learning configuration. To extract sufficient knowledge, and transfer it in an effective way, we propose a hybrid distillation loss that consists of both feature-wise and relation-wise distillation parts. In the following subsections, we will introduce them in detail.

3.3.1 Feature-wise Distillation. As depicted in Fig. 4, the feature-wise distillation proposed in our work is similar to existing studies with a modification by adding a cross-lingual module. This module, see section 3.2, is designed to alleviate the discrepancy between teacher and student networks caused by different languages of training data. Generally, traditional KD methods being applied in detection tasks pay attention to distilling key information in representation space by simply matching the features in a specific intermediate layer or the output layer [15, 66]. Taking the more common Teacher-student configuration as an example, this kind of distillation

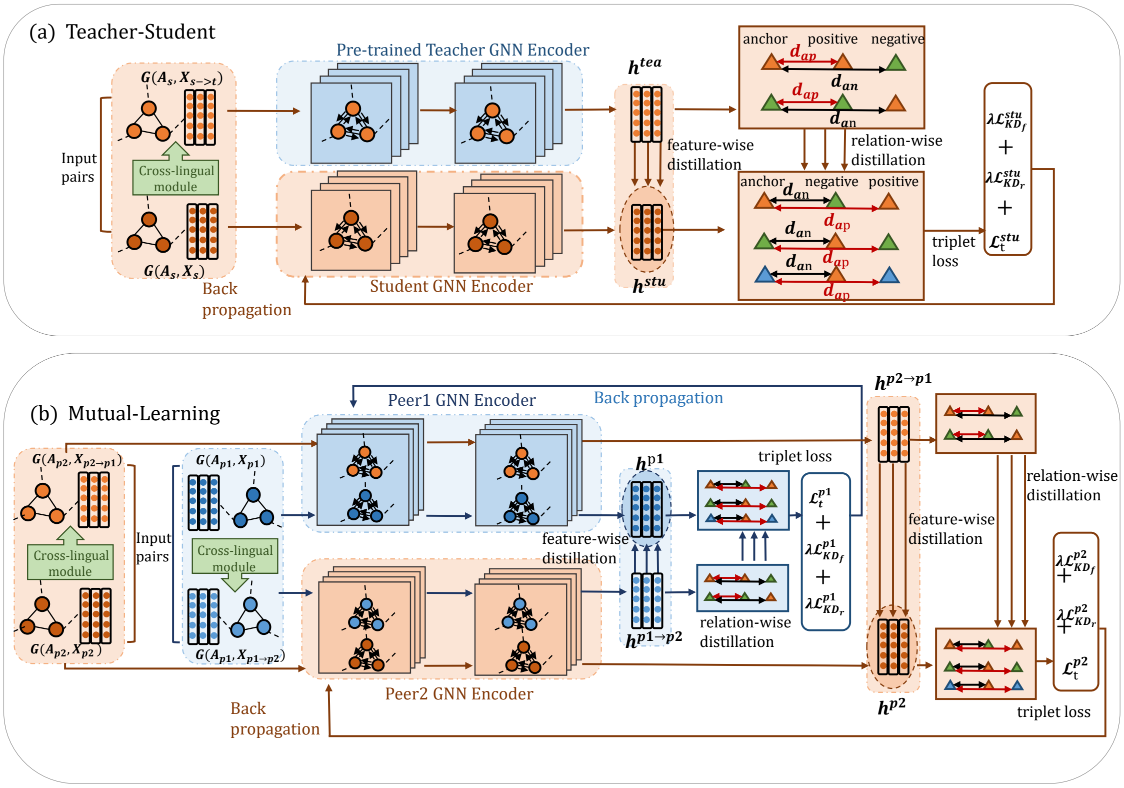


Fig. 4. **Overview of the cross-lingual knowledge distillation framework.** (a) shows the Teacher-Student distillation architecture for the offline situation; (b) shows the Mutual-Learning distillation architecture for the online situation.

can often be expressed as minimizing the following objective function:

$$\mathcal{L}^{stu}_{KD_f} = \sum_{x_i \in \mathrm{X_s}} \mathcal{D}\left(\theta^t\left(x_i\right), \theta^s\left(x_i\right)\right), \tag{2}$$

where θ^t and θ^s denote the pre-trained teacher's network and the student's network, respectively. Nevertheless, when distilling the whole feature layer in cross-modality or cross-lingual tasks, the transferred knowledge from teacher to student will inevitably contain much noise due to the dramatic domain gap or semantic gap across the training data of teacher and student. To tackle this issue and achieve more effective transfer, we add a cross-lingual module which has been described in Section 3.2 to eliminate the cross-lingual bias. The cross-lingual module can be either linear or nonlinear based on the specific language pair. The objective function now becomes:

$$\mathcal{L}^{stu}_{KD_f} = \sum_{x_i \in \mathrm{X_s}} \mathcal{D}\left(\theta^t\left(\boldsymbol{W}x_i\right), \theta^s\left(x_i\right)\right), \tag{3}$$

where $\boldsymbol{W}$ represents the cross-lingual transformation function, which as aforementioned, can be either linear or non-linear.

3.3.2 Relation-wise Distillation. The above-depicted feature-wise distillation transfers knowledge point-to-point, with the student model emulating the cross-lingual teacher model's representations. However, it may not capture the nuanced relationships between social texts, which are also crucial for precise event detection. In contrast, relation-wise distillation focuses on interdependencies and connections between social texts within events. The teacher model, specialized in event detection, understands both individual features and intricate relationships among social texts. By transferring this valuable relational knowledge, the student model develops a deeper understanding of the connections and distinctions that characterize different events, leading to enhanced accuracy in event detection. Note that relation-wise distillation successfully transfers knowledge by penalizing those higher-order structural differences. Usually, the higher the order considered in the extracted relations, the more comprehensive information can be obtained. Nevertheless, the computation cost will dramatically grow as the order gets higher [28]. Thus, we mainly focus on 2-order relational knowledge (i.e., pairwise distillation). The computation of this order is also key to the computation process of the scalable training strategy used in our work (i.e., the triplet loss). Hence, the distillation loss is consistent with the network's main training loss, which helps accelerate the training process and maintain training stability. Still taking the Teacher-student configuration as an example, the objective of the previous 2-order relation-wise distillation can be expressed as:

$$\mathcal{L}_{KD_r}^{stu} = \sum_{\left(x_i,x_j\right)\in\mathcal{X}_s^2} l_\delta\left(\mathcal{D}\left(\theta^t\left(x_i\right),\theta^t\left(x_j\right)\right),\mathcal{D}\left(\theta^s\left(x_i\right),\theta^s\left(x_j\right)\right)\right), \tag{4}$$

where $\mathcal{D}$ computes the distance of a given pair, and l_δ is a loss that penalizes structural difference between the learned feature spaces of teacher and student.

However, transferring all possible pair relations in a batch, existing relation-wise distillation approaches [28, 56, 71] ignore the mistakes made by the teacher, which severely hinders the training of student models. This problem is even more obvious and severe in the Mutual-Learning configuration. Since in early epochs, without the pre-training stage, most relations extracted by the peer model are misleading and untrustable. Forcing the students to uncritically mimic their pair models may be harmful to the model training. We solve this problem by selectively transferring beneficial relations. Recall that in Section 3.1, we introduce the triplet selection strategy in the triplet loss, i.e., the Batch Random Hard triplet selection strategy. Only those hard triplets within the same batch will be selected for backpropagation. Here we combine the relation-wise distillation method with the triplet selection strategy. The original pairwise distillation objective function (i.e., Eq. 4) is optimized by only transferring knowledge which corrects improper pair distances within the chosen negative triplets. Specifically, to guarantee that the transferred relational knowledge is indeed advantageous, we establish contrasting activation criteria for negative pairs and positive pairs in those hard triplets. In the context of a negative pair, the teacher's signal is designed to be effective only when the gap or difference between the two messages is greater in the teacher's prediction compared to the student's prediction. Conversely, when dealing with a positive pair, the teacher's signal becomes active when the gap or difference between this pair's prediction in the teacher's model is smaller than that in the student's model. This approach ensures that the margin is set in a customized and adaptive manner, taking into account the knowledge of the peer network, for each challenging negative triplet. The optimized objective function can be formulated as follows:

$$\begin{aligned}\mathcal{L}_{KD_r}^{stu} = \sum_{\substack{<m_i,m_j,m_k>\in\{\text{hard triplets}\}\\ <m_i,m_j,m_k>\in\{m_b\}\\ <m_i,m_j>\in<m_i,m_{i+}>,<m_i,m_k>\in<m_i,m_{i-}>}} &\max\left\{-\mathcal{D}\left(\theta^t\left(\boldsymbol{W}x_i\right),\theta^t\left(\boldsymbol{W}x_j\right)\right)+\mathcal{D}\left(\theta^s\left(x_i\right),\theta^s\left(x_j\right)\right),0\right\}\\ &+\max\left\{\mathcal{D}\left(\theta^t\left(\boldsymbol{W}x_i\right),\theta^t\left(\boldsymbol{W}x_k\right)\right)-\mathcal{D}\left(\theta^s\left(x_i\right),\theta^s\left(x_k\right)\right),0\right\},\end{aligned} \tag{5}$$

where x_i and x_j are the representations of a positive pair, and x_i and x_k are representations belonging to different events. This function aims to reduce the distance between those positive pairs while widely separating the negative pairs. Meanwhile, the added relation-wise distillation can also be seen as an enhanced version of the original triplet loss.

Algorithm 1: Training procedure of CLKD framework (Teacher-student)

```
Input: The original student-language dataset $(X_s, A_s)$; The transformed student-language dataset in
       teacher's semantic space $(X_{s\to t}, A_s)$; The pre-trained teacher's network $\theta^t$; Maximum training
       epoch number $E$.
Output: Trained student network $\theta^s$.
Initialization ($e$=1; Randomly initialize $\theta^s$);
while $e \leq E$ do
    Compute $\boldsymbol{h^{stu}}$ of student network;
    Compute $\boldsymbol{h^{tea}}$ of teacher network;
    Compute the triplet loss $\mathcal{L}_t(\boldsymbol{h^{stu}})$ of student network via Eq. 1;
    Compute the feature-wise distillation loss $\mathcal{L}^{stu}_{KD_f}$ via Eq. 3 and relation-wise distillation loss $\mathcal{L}^{stu}_{KD_r}$ via
      Eq. 5;
    Compute the total loss of student network $\mathcal{L}^{stu}_{total}$ via Eq. 6;
    Back-propagation to update student network $\theta^s$;
    $e = e + 1$;
end
```

3.3.3 *Teacher-student structure.* As shown in Fig. 4(a), the Teacher-Student structure is designed for offline situations. In our experiments, the network intended for French or Arabic event detection is regarded as the student and the network pre-trained on a large-scale English dataset is the teacher. It is worth noting that both the students and the teacher share the same network architecture, and the training process follows a conventional two-stage approach. Initially, we train the teacher network and keep its parameters fixed. Subsequently, we proceed to train the student network. Each student is encouraged to acquire detection knowledge from the ground-truth labels of its data using the triplet loss. Meanwhile, they are also guided to explore knowledge embedded in the output representations of the teacher through both feature-wise distillation and relation-wise distillation. Algo. 1 provides details of the procedure. Specifically, during the training process of the student, we have the processed student-language message graph $G(\boldsymbol{X_s}, \boldsymbol{A_s})$. We also utilize the corresponding student-language → teacher-language cross-lingual language models introduced in Section 3.2 to get the transformed initial features of messages, i.e., $\boldsymbol{X_{s\to t}}$. The transformed features are in English semantic space. The pair $G(\boldsymbol{X_s}, \boldsymbol{A_s})$ and $G(\boldsymbol{X_{s\to t}}, \boldsymbol{A_s})$ is the input to student and teacher networks respectively. The aim of obtaining cross-lingual attribute features $\boldsymbol{X_{s\to t}}$ is to eliminate the existing language discrepancy when French or Arabic data is input to the pre-trained teacher model specified for English. To use a real-world analogy, this process is similar to a case where there is an expert who has rich experience in social event detection but only understands English, and a student, inexperienced in social event detection, speaks another small language. The messages are converted into English before being shown to the expert. The expert can then use his or her expertise in social event detection to teach the student how to detect events by way of example. To transfer prior knowledge from the English expert to the French and Arabic students, the final message output of teacher model $\boldsymbol{h^{tea}}$ is used as an extra supervisory signal. The specific optimized functions of distillation losses adopted in the Teacher-student configuration are demonstrated in Eq. 3 and Eq. 5 which encourage the student to mimic the teacher in both point-wise and relation-wise ways. The total loss for the backpropagation of the student training network is set as a weighted sum of the triplet loss based on true labels and the optimized knowledge distillation losses:

$$\mathcal{L}_{total}^{stu} = \mathcal{L}_t(\boldsymbol{h}^{stu}) + \lambda_1 \mathcal{L}_{KD_f}^{stu} + \lambda_2 \mathcal{L}_{KD_r}^{stu}, \tag{6}$$

where λ_1 and λ_2 are the hyper-parameters that control the weights of the feature-wise and relation-wise knowledge distillation losses.

Algorithm 2: Training procedure of CLKD framework (Mutual-Learning)

```
Input: The original peer1-language dataset $(X_{p1}, A_{p1})$; The transformed peer1-language dataset in peer2's
       semantic space $(X_{p1\to p2}, A_{p1})$; The original peer2-language dataset $(X_{p2}, A_{p2})$; The transformed
       peer2-language dataset in peer1's semantic space $(X_{p2\to p1}, A_{p2})$; Maximum training epoch number
       $E$.
Output: Trained peer1 network $\theta^{p1}$ and peer2 network $\theta^{p2}$.
Initialization ($e$=1; Randomly initialize $\theta^{p1}$ and $\theta^{p2}$);
while $e \leq E$ do
    Compute $\boldsymbol{h}^{p1}$ of peer1, $\boldsymbol{h}^{p1\to p2}$ of peer2;
    Compute the triplet loss $\mathcal{L}_t(\boldsymbol{h}^{p1})$ of peer1 via Eq. 1;
    Compute the feature-wise distillation loss $\mathcal{L}_{KD_f}^{p1}$ like Eq. 3 and relation-wise distillation loss $\mathcal{L}_{KD_r}^{p1}$ of
      peer1 like Eq. 5;
    Compute the total loss $\mathcal{L}_{total}^{p1}$ of peer1 via Eq. 7;
    Back-propagation to update peer1 network $\theta^{p1}$;
    Compute $\boldsymbol{h}^{p2}$ of peer2, $\boldsymbol{h}^{p2\to p1}$ of peer1;
    Compute the triplet loss $\mathcal{L}_t(\boldsymbol{h}^{p2})$ of peer2 via Eq. 1;
    Compute the feature-wise distillation loss $\mathcal{L}_{KD_f}^{p2}$ like Eq. 3 and and relation-wise distillation loss $\mathcal{L}_{KD_r}^{p2}$
      of peer2 like Eq. 5;
    Compute the total loss $\mathcal{L}_{total}^{p2}$ of peer2 via Eq. 8;
    Back-propagation to update peer2 network $\theta^{p2}$;
    $e = e + 1$;
end
```

3.3.4 Mutual-Learning Structure. Fig. 4(b) shows the mutual distillation scheme, designed for online situations. This configuration is motivated by Authors in [87], who contend that students can learn from each other. We have formulated the structure as a cohort of two networks that exploit knowledge from each other. Hence, both networks are strengthened through the help of their peer. The training details are shown in Algo. 2. We have two processed monolingual social event data $G(\boldsymbol{X_{p1}}, \boldsymbol{A_{p1}})$ and $G(\boldsymbol{X_{p2}}, \boldsymbol{A_{p2}})$ specified for the training of peer1 network and peer2 network respectively. Similarly, to eliminate language discrepancy, we also get the transformed peer1 data in peer2-language semantic space $G(\boldsymbol{X_{p1\to p2}}, \boldsymbol{A_{p1}})$ and the transformed peer2 data in peer1-language semantic space $G(\boldsymbol{X_{p2\to p1}}, \boldsymbol{A_{p2}})$ by the corresponding cross-lingual language models. To enhance learning, during the training process, not only explicit knowledge from the true labels are leveraged, but implicit point-wise and pairwise knowledge from the peer are also used. For example, as for the training of peer1 network, the raw message representations $G(\boldsymbol{X_{p1}}, \boldsymbol{A_{p1}})$ and the transformed ones $G(\boldsymbol{X_{p1\to p2}}, \boldsymbol{A_{p1}})$ are simultaneously input into peer1 and peer2 networks respectively, with the corresponding outputs of $\boldsymbol{h}^{p1}$ and $\boldsymbol{h}^{p1\to p2}$. $\boldsymbol{h}^{p1\to p2}$ is used as an extra supervisory signal of peer1. As depicted in Fig. 4(b), the distillation loss also contains both feature-wise and relation-wise parts. By analogy, the distillation loss for the peer2 network is computed in the same way by treating the peer1 model as the guiding peer. The total loss of peer1 and peer2 is also formulated as the weighted combination of the corresponding triplet loss based on true labels and the corresponding knowledge distillation loss:

$$\mathcal{L}^{p1}_{total} = \mathcal{L}_t(\boldsymbol{h}^{\boldsymbol{p1}}) + \lambda_1 \mathcal{L}^{p1}_{KD_f} + \lambda_2 \mathcal{L}^{p1}_{KD_r}, \tag{7}$$

$$\mathcal{L}^{p2}_{total} = \mathcal{L}_t(\boldsymbol{h}^{\boldsymbol{p2}}) + \lambda_1 \mathcal{L}^{p2}_{KD_f} + \lambda_2 \mathcal{L}^{p2}_{KD_r}, \tag{8}$$

where the specific computation processes of feature-wise and relation-wise knowledge distillation losses are similar to those depicted in Eq. 3 and Eq. 5. The entire framework is trained online, and the weights of peer1 and peer2 are updated alternatively according to the combined loss. Suppose peer1 has richer training data (e.g., data in English), and the target is to improve the performance of the peer2 network (e.g., the network trained for French data). From peer1's view, the feature-wise and relation-wise knowledge distillations $L^{p1}_{KD_f}$ and $L^{p1}_{KD_r}$ provide the knowledge learned from its peer, which guides peer1 to implicitly generalize towards a more reliable direction to help detect peer2 data. In other words, with the knowledge distilled from peer2, peer1 provides better suggestions to help peer2 extract events from its data. From peer2's view, the knowledge distillation loss L^{p2}_{KD} brings additional knowledge from peer1 that serves to augment and directly enhance peer2's generalization ability. Further, an ensemble strategy is used to explore more informative and comprehensive cross-lingual knowledge in the final detection of the target peer2's data. In detail, to do the final message clustering, peer1's final representations $\boldsymbol{h}^{\boldsymbol{p2 \rightarrow p1}}$ of the transformed peer2 data $G(\boldsymbol{X}_{\boldsymbol{p2 \rightarrow p1}}, \boldsymbol{A}_{\boldsymbol{p2}})$ are concatenated with peer2's final representations $\boldsymbol{h}^{\boldsymbol{p2}}$ of the peer2 data $G(\boldsymbol{X}_{\boldsymbol{p2}}, \boldsymbol{A}_{\boldsymbol{p2}})$. This process is similar to a case with two student peers – one speaks English and the other does not. The student who speaks another language learns the message representations in her own language but also uses the suggestion of her English-speaking peer. The opinions of both students are then combined to make a more general and informed decision.

3.4 Continuous Detection Framework

To extend the framework to adapt to online (incremental) scenarios, we follow a life cycle that contains three stages: pre-training, detection, and maintenance. We only update the parameters of models in the pre-training and maintenance stages. In the detection stage, we directly adopt the latest trained models in the early block to obtain message representations and then leverage clustering techniques to detect events. Specifically, as shown in Fig. 5, in the pre-training stage, an initial message graph of each data is constructed from the first message block. Meanwhile, we also obtain the transformed message graphs in which the initial features of messages are transformed into their peer's language space. Then we utilize Algo. 2 to train the peer1 GNN encoder and peer2 GNN encoder. The pre-training stage only runs once. In the detection stage, a new graph is constructed for each coming block. We directly detect events of each coming block with the already trained model. In the maintenance stage, we continuously train the model with the newest message block, allowing the model to learn new knowledge. The training process of the maintenance stage is similar to the pre-training stage; the only difference lies in the initialization of the model's parameters. In the pre-training stage, the parameters of the peer1 GNN encoder and peer2 GNN encoder are both randomly initialized. In the maintenance stage, the initial model's parameters are those learned last time. The detection stage and the maintenance stage alternate. In this way, the model continuously adapts to incoming data. It can detect new events and update the model's knowledge. It also ensures a light training scheme as obsolete nodes in past blocks are deleted.

3.5 Time Complexity Analysis

The overall running time (except the CLKD framework in the Mutual-Learning structure) is $O(N_e)$, where N_e is the total number of edges in the message graph. In detail, the running time for constructing a monolingual

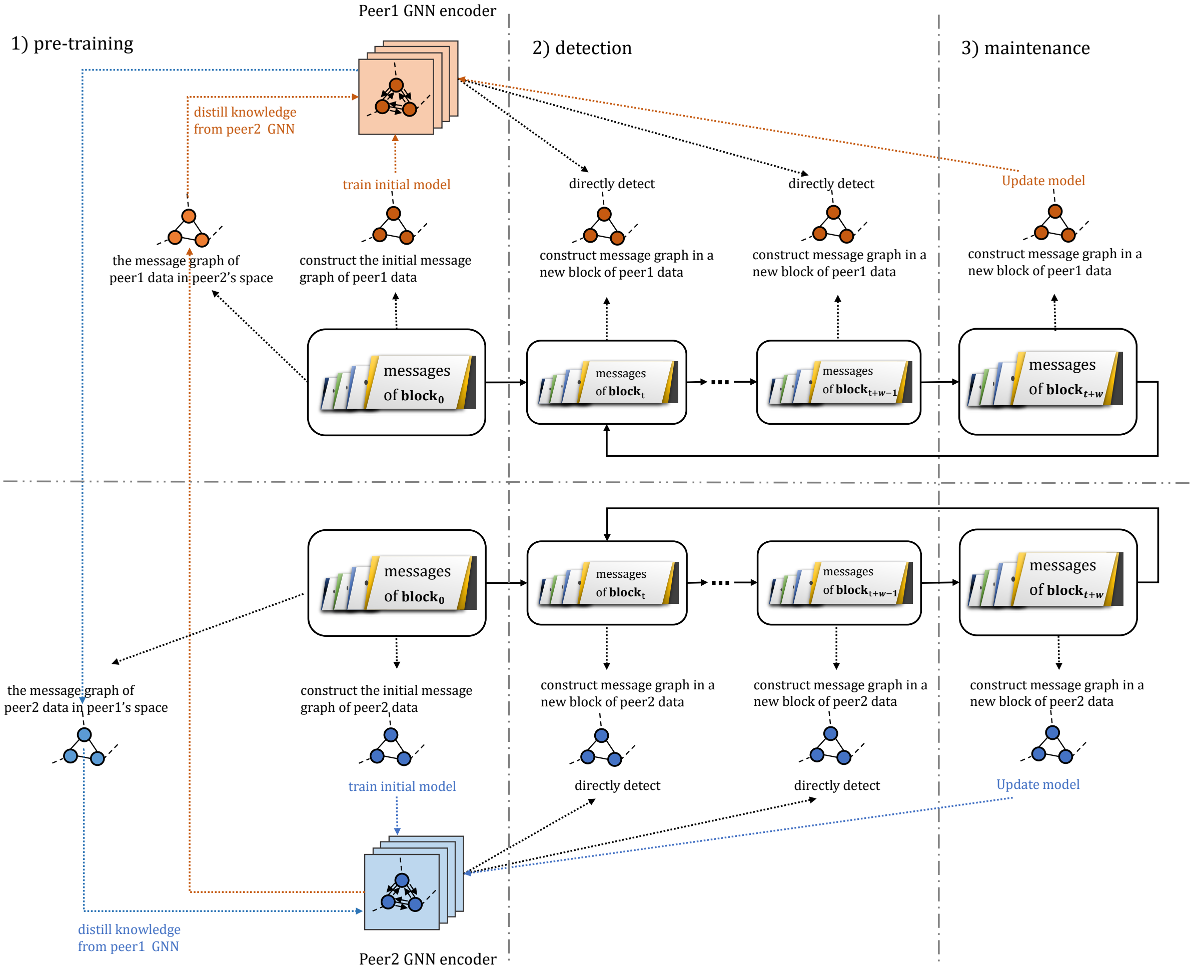
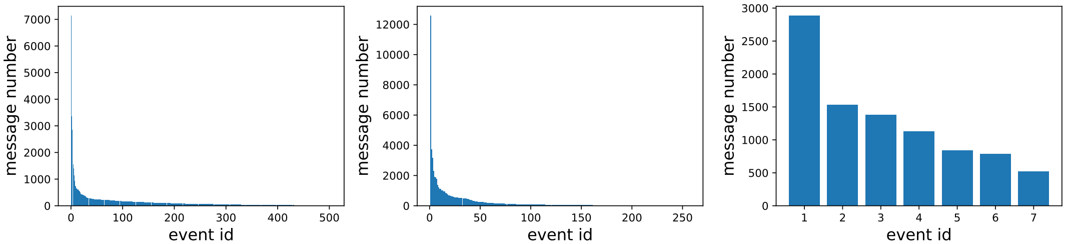

Peer1 GNN encoder
1) pre-training
2) detection
3) maintenance
distill knowledge from peer2 GNN
train initial model
directly detect
directly detect
Update model
the message graph of peer1 data in peer2's space
construct the initial message graph of peer1 data
construct message graph in a new block of peer1 data
construct message graph in a new block of peer1 data
construct message graph in a new block of peer1 data
messages of $\mathbf{block_0}$
messages of $\mathbf{block}_t$
messages of $\mathbf{block}_{t+w-1}$
messages of $\mathbf{block}_{t+w}$
messages of $\mathbf{block_0}$
messages of $\mathbf{block}_t$
messages of $\mathbf{block}_{t+w-1}$
messages of $\mathbf{block}_{t+w}$
the message graph of peer2 data in peer1's space
construct the initial message graph of peer2 data
construct message graph in a new block of peer2 data
construct message graph in a new block of peer2 data
construct message graph in a new block of peer2 data
train initial model
directly detect
directly detect
Update model
distill knowledge from peer1 GNN
Peer2 GNN encoder

Fig. 5. **The life cycle of our continuous detection framework, including pre-training, detection and maintenance.**

message graph is $O(N + N_e) = O(N_e)$, where N is the total number of messages in the message graph. Since the cross-lingual language models (mentioned in Section 3.2) can be pre-computed before training. In terms of the Teacher-Student CLKD framework (the teacher is pre-trained and fixed), we need $O(N)$ to obtain English semantic features. Propagating the GNN encoder takes $O(Ndd' + N_e d') = O(N_e)$, where d and d' are the input and output dimensions of the GNN encoder. For the loss calculation, triplet sampling takes $O(\sum_{b=1}^{B} |m_b|^2)$, where $|\{m_b\}|$ is the number of messages in the b-th batch and B is the total number of batches. Plus, another $O(\sum_{b=1}^{B} |m_b|^2)$ is required to calculate knowledge distillation loss. In reality, $O(\sum_{b=1}^{B} |m_b|^2) \ll N_e$. Thus, the total complexity is $O(N_e)$. We then analyze the overall running time of the CLKD framework in the Mutual-Learning structure. Suppose the total number of edges and nodes of the auxiliary data we select are N_e' and N'. If N_e' is in the same order of magnitude with N_e, the overall running time is $O(N_e)$. If $N_e' \gg N_e$, the overall running time is $O(N_e')$.



(a) English dataset statistics (b) French dataset statistics (c) Arabic dataset statistics

Fig. 6. **Dataset statistics.** (a), (b), (c) show the number of messages related to each event on the English dataset, French dataset and Arabic dataset, respectively.

In fact, the time of mutual learning can be roughly seen as the sum of the run-time of the auxiliary data and of the target data in the Teacher-Student structure.

4 EXPERIMENTS

To evaluate our CLKD framework, we conduct offline and online experiments with real-world datasets. The three publicly-available datasets contain Twitter messages – one each in English, French, and Arabic. As described in the Methodology, the backbone GNN encoder is configured as a two-layer multi-head GAT model, i.e., the encoder applied in KPGNN as reported in [11]. As for the cross-lingual module, we adopt MUSE in work [42] to learn the linear mappings between languages and LNMAP in work [52] for non-linear mappings.

4.1 Experimental Setup

4.1.1 Datasets. The three Twitter datasets are filtered for duplicate and unavailable tweets, leaving the following record counts:

- English Twitter dataset [49] – 68,841 manually labeled tweets relating to 503 event classes, spreading over a period of 29 days.
- French Twitter dataset [48] – 64,516 labeled tweets relating to 257 event classes over 23 days.
- Arabic Twitter dataset [2] – 9,070 labeled tweets relating to 7 catastrophic-class events over different periods.

To evaluate the CLKD framework, in the offline tests, the teacher is trained on the English Twitter dataset, and the students are trained on the French and Arabic Twitter datasets. The task's goal is to better detect events in the low-resource French and Arabic data by leveraging prior knowledge learned from the teacher network. In the online tests, we use the English and French datasets in a mutual learning scheme comprising two peer networks. The task's goal is to strengthen the model's detection capability in the French stream online. Fig. 6 shows how the three datasets are partitioned between all the events. Obviously, the numbers of messages are very unbalanced, making the detection task more difficult by design.

4.1.2 Baselines. Most models in the social event detection domain are designed for monolingual data, including general message representation learning, offline social event detection methods, and incremental ones. We select the following methods as baselines of the CLKD framework: Word2Vec [50], which uses the average of the pre-trained word2Vec embeddings of all words in the message as its representation; BERT [23], which uses the averaging BERT embeddings of all the words in a message as its representation; LDA [8], which is the most typical topic model in NLP to cluster texts; Pairwise Popularity Graph Convolutional Network (PP-GCN) [58], which is an offline fine-grained social event detection method based on GCN [41]; EventX [44], which is a fine-grained event detection method based on community detection, applicable to online scenarios; QSGNN [63], which focuses on open-set social event detection tasks and generalizes knowledge from known to unknown by leveraging the best of known samples (i.e., setting stricter constraints in the inter-class distance and direction relations) and reliable knowledge transfer (generating and selecting high-quality pseudo labels); KPGNN [11], which leverages the inductive learning ability of GNNs to efficiently represent and detect events. KPGNN has shown promising performance in social event detection tasks for both offline and online situations.

4.1.3 Experimental Setting and Implementation. For Word2Vec, we use the pre-trained 300-d language models[3, 4, 5]. For LDA, we set the topic number 50. For BERT, we use the open-source implementation[6] and adopt an average 768-d hidden-states of tokens in the last layer as the embeddings. For EventX, we follow the hyper-parameters settings suggested in the original paper. For the GNN-based methods (PP-GCN, QSGNN, KPGNN) which we select as baselines and the backbone of this work, we follow the hyper-parameter setting in KPGNN paper [11]. Specifically, we set the total number of heads to 4, the hidden embedding dimension and output dimension to 32, the total number of layers L to 2, the learning rate to 0.001, the optimizer to Adam, and training epochs to 15 with the patience of 5 for early stopping. In the data preprocessing stage, to create initial tweet representations, tweets are processed by systematically removing emojis and stripping away special characters associated with hashtags and mentions (like "#" and "@"), leaving only the textual content. The cleaned sentences are then further analyzed by calculating the average of pre-trained language model embeddings for each word in the relevant languages to generate tweet embeddings. This added step ensures that the data is suitably prepared for the next phases of analysis. Meanwhile, for the CLKD framework, we choose Euclidean distance for the distance metric in feature-wise and relation-wise knowledge distillation functions. As for the setting of hyper-parameters, the weights of the feature-wise and relation-wise knowledge distillation losses λ_1 and λ_2 are set to 0.1 and 0.5 in the Teacher-Student structure, and both to 0.1 in the Mutual-Learning structure. The mini-batch size $|\{m_b\}|$ is 2000 and the maintenance window size w is 3, which means the network is retrained every 3 message blocks. We repeat all experiments 5 times and report the mean and standard variance of the results. Note that some baselines (Word2Vec, LDA) require the number of total event classes to be pre-defined. For a fair comparison, we apply K-means clustering after obtaining the message representations from the other models and set the total number of classes to the number of ground-truth classes. DBSCAN could be used outside an experimental setting if the total number of classes was unknown, as is often the case with incremental detection.

4.1.4 Evaluation Metrics. We evaluate the performance of the models using three clustering metrics: normalized mutual information (NMI), adjusted mutual information (AMI), and adjusted Rand index (ARI). NMI measures the amount of information one can extract from the prediction distribution and has been broadly adopted in event detection method evaluations. However, NMI is not adjusted for the chance. Thus, we also select AMI, which is a more recent proposition. ARI considers all prediction label pairs and counts pairs that are assigned in the same and different clusters.

4.2 Cross-lingual Social Event Detection

This section presents the results of the CLKD framework specified for the special cases of detection in one specific low-resource language. The report begins with the offline evaluation, followed by the incremental evaluation. The percentages for training, validation, and testing are also 70%, 10%, and 20%.

[3] https://spacy.io/models/en-starters#en_vectors_web_lg
[4] https://spacy.io/models/en-starters#fr_vectors_web_lg
[5] https://github.com/bakrianoo/aravec
[6] https://huggingface.co/models

Table II. Pre-trained teacher network and knowledge prior. "KPGNN-Tea" means the model trained on English Twitter data. "KPGNN-Tea-LLA" means first transforming non-English data into English semantic space in a linear way and then inputting the data into the trained teacher model. "KPGNN-Tea-NLA" is similar to "KPGNN-Tea-LLA" with the only modification of transforming semantic space in a nonlinear way.

Model	Metrics	English Data	French Data	Arabic Data
KPGNN-Tea	NMI	.70±.02	.50±.01	.39±.01
	AMI	.63±.01	.42±.00	.20±.01
	ARI	.23±.01	.11±.01	.19±.02
KPGNN-Tea-LLA	NMI	-	.52±.01	.42±.01
	AMI	-	.44±.01	.22±.01
	ARI	-	.16±.01	.20±.02
KPGNN-Tea-NLA	NMI	-	.58±.01	.40±.03
	AMI	-	.51±.01	.26±.02
	ARI	-	.20±.01	.21±.03

4.2.1 Offline Evaluation. Recall that, in offline situations, the CLKD framework follows a two-stage training strategy – first the teacher network is trained, and then the student networks. Hence, in the first stage, we train a teacher network on the English Twitter dataset. Then we train the student networks on the non-English datasets (i.e., French and Arabic) while fixing the teacher network's parameters. To explore how much knowledge the trained English teacher network could lend to the student networks, we experiment with directly inputting the student datasets into the pre-trained teacher network, adding the cross-lingual module, and recording these results separately. The results are shown in Table II. Here "KPGNN-Tea" means the model trained on English Twitter data. "KPGNN-Tea-LLA" means first transforming non-English data into English semantic space in a linear way and then inputting the data into the trained teacher model. "KPGNN-Tea-NLA" is similar to "KPGNN-Te-LLA," with the only modification being that it transforms semantic space in a nonlinear way. With 0.70±0.02 NMI, 0.63±0.01 AMI, and 0.23±0.01 ARI, the teacher network on English produces outstanding results. On the French and Arabic datasets, directly inputting datasets to the teacher network returns 0.50±0.01 NMI and 0.39±0.01 NMI, respectively. Table III shows these values are better than most baselines, which suggests that the teacher does hold prior knowledge that could be transferred to the student networks. Furthermore, when adding the linear and nonlinear language transformations, the results are even better. For example, if we map the French data and Arabic data into English semantic space in a nonlinear way, the teacher network returns 0.58±0.01 NMI and 0.41±0.03 NMI respectively, which greatly surpasses the results with no semantic mapping. This also demonstrates the effectiveness of the designed cross-lingual module in our CLKD framework.

Table III shows further validation of the effectiveness of the CLKD framework. Here, we train the French and Arabic student networks under feature-wise and relation-wise supervision from the teacher but without making language alignment, i.e., without the cross-lingual module, denoted as CLKD-w/o LA. To demonstrate the effects of different level orders on knowledge distillation, we further remove the relation-wise supervision from the teacher, i.e., CLKD-w/o LA-w/o RKD. CLKD-LLA denotes the implementation of having both feature-wise and relation-wise knowledge distillation with the linear cross-lingual module MUSE, and CLKD-NLA is with the non-linear cross-lingual module LNMAP. From a careful review of these results, we observe the following: (1) The GNN-based methods (PP-GCN, KPGNN, QSGNN) achieve much better results than the other baselines, which is due in large part to their ability in aggregating message attribute features and structural information.

Table III. Offline Evaluation on French and Arabic datasets. The bolded results represent the optimal outcome.

Methods	French Data			Arabic Data		
	NMI	AMI	ARI	NMI	AMI	ARI
Word2Vec [50]	.26±.00	.23±.00	.04±.00	.03±.00	.03±.00	.02±.00
BERT [23]	.33±.01	.24±.00	.04±.00	.14±.00	.14±.00	.07±.00
LDA [8]	.26±.00	.16±.00	.01±.00	.03±.00	.02±.00	.02±.00
PP-GCN [58]	.56±.02	.48±.01	.17±.01	.33±.02	.32±.02	.31±.02
EventX [44]	.52±.00	.58±.00	.01±.00	.28±.00	.03±.00	.03±.00
QSGNN [63]	.67±.02	.63±.02	.30±.03	.78±.02	.77±.01	.79±.02
KPGNN [11]	.63±.01	.59±.01	.27±.03	.75±.02	.74±.01	.75±.02
CLKD-w/o LA-w/o RKD	.64±.01	.59±.01	.27±.02	.76±.02	.77±.03	.78±.03
CLKD-w/o LA	.69±.02	.65±.02	.39±.04	.79±.02	.79±.02	.81±.03
CLKD-LLA	**.71±.02**	**.66±.01**	**.43 ±.02**	.80±.03	.80±.02	.82±.03
CLKD-NLA	.70±.02	.66±.01	.38±.02	**.81±.02**	**.81±.02**	**.84±.02**
promotion	↑ 4%	↑ 3%	↑ 13%	↑ 3%	↑ 4%	↑ 5%

Table IV. The statistics of French social stream.

Blocks	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
# of messages	5356	3186	2644	3179	2662	4200	3454	2257
# of events	22	19	15	19	27	26	23	25
Blocks	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
# of messages	3669	2385	2802	2927	4884	3065	2411	1107
# of events	31	32	31	29	28	26	25	14

Specifically, QSGNN gains better performance than KPGNN while the adopted backbone GNN model is the same. We owe this improvement to the stricter constraints of setting larger inter-class distance and orthogonal inter-class feature direction in QSGNN, which greatly helps distinguish different events. (2) Generally, the four variants of the Teacher-Student CLKD framework gain equal or better results than the original KPGNN model and the powerful QSGNN model. This is because, under the supervision of the pre-trained teacher network, prior knowledge that helps detect events is transferred to the students to augment their training. Specifically, the better result of CLKD-w/o LA-w/o RKD compared to KPGNN validates the effectiveness of transferring feature-wise knowledge from the teacher. This kind of point-wise information helps the students encode great representations for messages in low-resource languages. In CLKD-w/o LA, except for the feature-wise knowledge extracted from the teacher, the higher-order relation-wise knowledge is also transferred to students. As can be seen in Table III, with the help of beneficial relation-wise knowledge, the model's ability to distinguish different events gets greatly enhanced. That is because, by adding the relation-wise information, students are capable of setting larger inter-class distances and smaller intra-class distances for those hard negative triplets. Meanwhile, further comparing the promotions of adding feature-wise supervision (i.e., the promotion of CLKD-w/o LA to KPGNN)

Table V. Incremental evaluation NMIs on the French dataset. The underlined and bolded results represent the optimal outcome.

Blocks	M_1	M_2	M_3	$M4$	M_5	M_6	M_7	M_8
Word2Vec [50]	.22±.00	.22±.00	.25±.00	.28±.00	.48±.00	.33±.00	.35±.00	.37±.00
BERT [23]	.32±.00	.32±.00	.31±.00	.33±.00	.47±.00	.36±.00	.41±.00	.44±.00
LDA [8]	.20±.00	.09±.00	.13±.00	.10±.00	.24±.00	.22±.00	.12±.00	.24±.00
PP-GCN [58]	.49±.01	.45±.00	.56±.03	.54±.03	.54±.02	.52±.02	.56±.04	.56±.03
EventX [44]	.34±.00	.37±.00	.37±.00	.39±.00	.53±.00	.44±.00	.41±.00	.54±.00
QSGNN [63]	**.57±.01**	.58±.01	.57±.01	**.58±.03**	.61±.02	.60±.01	.64±.01	.57±.02
KPGNN [11]	.54±.01	.56±.02	.52±.03	.55±.01	.58±.02	.59±.03	.63±.02	.58±.02
CLKD-w/o LA-w/o RKD	.54±.01	.57±.01	.54±.02	.57±.02	.61±.02	.61±.03	.64±.02	.60±.01
CLKD-w/o LA	.55±.02	.57±.02	**.63±.03**	.55±.02	.60±.02	.67±.02	**.65±.02**	**.62±.01**
CLKD-LLA	.57±.02	.58±.01	**.63±.01**	.57±.02	**.62±.02**	.68±.01	.63±.03	.61±.02
CLKD-NLA	.56±.02	**.59±.02**	.58±.02	.52±.01	.57±.00	**.69±.02**	.63±.02	.59±.02
promotion	–	↑ 1%	↑ 6%	↓ 1%	↑ 1%	↑ 9%	↑ 1%	↑ 4%

Blocks	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
Word2Vec [50]	.33±.00	.46±.00	.41±.00	.40±.00	.22±.00	.36±.00	.41±.00	.28±.00
BERT [23]	.38±.00	.42±.00	.45±.00	.48±.00	.31±.00	.43±.00	.39±.00	.34±.00
LDA [8]	.16±.00	.17±.00	.22±.00	.28±.00	.19±.00	.24±.00	.33±.00	.07±.00
PP-GCN [58]	.54±.02	.56±.06	.59±.03	.60±.02	.61±.01	.60±.02	.57±.03	.53±.02
EventX [44]	.45±.00	.52±.00	.48±.00	.51±.00	.44±.00	.52±.00	.49±.00	.39±.00
QSGNN [63]	.52±.02	.60±.01	**.60±.01**	.61±.02	.59±.04	.68±.02	.63±.02	.51±.03
KPGNN [11]	.48±.02	.57±.01	.54±.01	.55±.04	.60±.02	.66±.01	.60±.01	.52±.02
CLKD-w/o LA-w/o RKD	.55±.02	**.66±.02**	.59±.01	.64±.01	**.65±.00**	**73±.00**	.66±.02	.51±.02
CLKD-w/o LA	.57±.04	.64±.02	**.60±.02**	.70±.01	.62±.01	71±.02	.70±.02	.50±.02
CLKD-LLA	.55±.01	.62±.02	**.60±.02**	**.71±.01**	.63±.02	**.73±.01**	.72±.02	**.54±.03**
CLKD-NLA	**.63±.01**	.59±.02	.57±.02	**.71±.01**	.62±.02	.68±.02	**.73±.03**	.53±.04
promotion	↑ 9%	↑ 6%	–	↑ 10%	↑ 4%	↑ 5%	↑ 10%	↑ 1%

and adding relation-wise supervision (i.e., the promotion of CLKD-w/o LA to CLKD-w/o LA-w/o RKD), it is obvious that adding the relation-wise supervision achieves more improvements. Because instead of the specific message representations, the key to recognising different messages is their relations. (3) On both the French and Arabic datasets, the variants with the cross-lingual module perform best, verifying the merits of this approach. On the French dataset, the variant with the linear cross-lingual module (CLKD-LLA) delivers the best result, while, on the Arabic dataset, the variant with the non-linear cross-lingual module (CLKD-NLA) is the best. This linear variant may have been better with French because the French-English language pair comes closer to an

Table VI. Incremental evaluation AMIs on the French dataset. The underlined and bolded results represent the optimal outcome.

Blocks	M_1	M_2	M_3	$M4$	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
Word2Vec [50]	.21±.00	.21±.00	.23±.00	.27±.00	.46±.00	.31±.00	.33±.00	.34±.00	.30±.00	.42±.00	.38±.00	.37±.00	.20±.00	.34±.00	.38±.00	.25±.00
BERT [23]	.28±.00	.31±.00	.32±.00	.30±.00	.44±.00	.33±.00	.36±.00	.38±.00	.28±.00	.35±.00	.34±.00	.44±.00	.26±.00	.40±.00	.39±.00	.27±.00
LDA [8]	.19±.00	.06±.00	.11±.00	.08±.00	.20±.00	.19±.00	.10±.00	.20±.00	.12±.00	.11±.00	.18±.00	.25±.00	.17±.00	.21±.00	.30±.00	.02±.00
PP-GCN [58]	.48±.00	.44±.02	.55±.03	.54±.04	.53±.02	.50±.03	.55±.04	.55±.02	.48±.03	.55±.04	.57±.02	.58±.02	.59±.02	.59±.01	.55±.03	.52±.02
EventX [44]	.11±.00	.12±.00	.11±.00	.14±.00	.24±.00	.15±.00	.12±.00	.21±.00	.16±.00	.19±.00	.18±.00	.20±.00	.15±.00	.22±.00	.22±.00	.10±.00
QSGNN [63]	**.56±.01**	.57±.01	.56±.02	**.57±.03**	**.59±.01**	.59±.01	**.63±.01**	.55±.02	.46±.02	.58±.01	**.59±.02**	.59±.02	.58±.03	.67±.02	.61±.00	.50±.03
KPGNN [11]	.54±.01	.55±.01	.55±.02	.55±.01	.57±.01	.57±.02	.61±.02	.57±.02	.46±.02	.56±.02	.53±.01	.56±.02	.60±.02	.65±.00	.58±.02	.50±.01
CLKD-w/o LA-w/o RKD	**.56±.00**	.56±.01	.52±.02	.56±.01	.58±.02	.55±.02	.62±.02	.58±.02	.49±.02	**.63±.02**	.57±.02	.59±.01	**.64±.01**	**.72±.01**	.61±.02	.50±.02
CLKD-w/o LA	.55±.02	.56±.02	**.62±.02**	.53±.01	.58±.02	.64±.02	**.63±.02**	**.60±.02**	.52±.05	.61±.02	.57±.02	.67±.01	.62±.01	.71±.02	.66±.02	.50±.02
CLKD-LLA	**.56±.02**	.57±.00	.60±.02	.56±.03	**.59±.02**	**.65±.01**	.62±.02	.58±.02	.49±.02	.60±.02	.58±.01	**.68±.02**	.63±.02	**.72±.02**	.68±.02	.51±.03
CLKD-NLA	.55±.01	**.59±.02**	.56±.01	.50±.02	.55±.01	.64±.02	.62±.02	.57±.01	**.58±.01**	.57±.01	.55±.02	.66±.02	.60±.02	.69±.01	**.70±.01**	**.52±.01**
promotion	–	↑ 2%	↑ 6%	↓ 1%	–	↑ 6%	–	↑ 3%	↑ 10%	↑ 5%	↓ 1%	↑ 9%	↑ 4%	↑ 5%	↑ 9%	–

isomorphic assumption. In other words, their vector spaces have a more similar geometric structure. The opposite is true of the English-Arabic pairs, so the non-linear variant is better suited to this task.

4.2.2 Incremental Evaluation. To create an online testing environment, we split the French and English datasets by dates to construct two parallel social streams. The Arabic dataset is not included in this incremental experiment because it is not collected based on a continuous period but, rather, on past catastrophic events in different time periods. For both selected datasets (i.e., the English dataset and the French dataset), we use the messages of the first week to form an initial message block M_0 and the messages from the remaining days to form the following message blocks. Recall that the French data spans a period of 23 days, and the English spans 29 days, and the

Table VII. Incremental evaluation ARIs on the French dataset. The underlined and bolded results represent the optimal outcome.

Blocks	M_1	M_2	M_3	$M4$	M_5	M_6	M_7	M_8
Word2Vec [50]	.08±.00	.10±.00	.16±.00	.11±.00	.25±.00	.13±.00	.15±.00	.18±.00
BERT [23]	.09±.00	.14±.00	.17±.00	.12±.00	.22±.00	.12±.00	.16±.00	.18±.00
LDA [8]	.01±.00	.04±.00	.01±.00	.06±.00	.02±.00	.02±.00	.05±.00	.04±.00
PP-GCN [58]	.27±.03	.21±.01	.38±.03	.35±.05	.30±.01	.27±.02	.38±.05	.38±.04
EventX [44]	.01±.00	.01±.00	.01±.00	.01±.00	.03±.00	.01±.00	.01±.00	.02±.00
QSGNN [63]	.30±.01	.38±.02	.36±.02	.36±.01	.36±.02	.36±.01	.38±.02	.35±.02
KPGNN [11]	.29±.02	.37±.01	.39±.04	.36±.04	**.37±.02**	.35±.04	.37±.02	.38±.02
CLKD-w/o LA-w/o RKD	.28±.04	.38±.04	.31±.01	.35±.03	**.37±.02**	.31±.02	.37±.02	.37±.03
CLKD-w/o LA	.27±.01	.37±.01	.48±.02	.37±.03	.34±.02	.37±.02	.37±.04	.39±.01
CLKD-LLA	**.31±.02**	**.39±.02**	.43±.03	**.40±.04**	.35±.03	.42±.02	.35±.04	**.40±.03**
CLKD-NLA	.29±.02	.38±.03	**.50±.05**	.33±.03	.33±.04	**.45±.02**	**.39±.02**	.39±.02
promotion	↑ 1%	↑ 1%	↑ 11%	↑ 4%	–	↑ 9%	↑ 1%	↑ 2%
Blocks	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
Word2Vec [50]	.10±.00	.20±.00	.13±.00	.19±.00	.07±.00	.17±.00	.20±.00	.11±.00
BERT [23]	.10±.00	.13±.00	.10±.00	.24±.00	.08±.00	.18±.00	.17±.00	.11±.00
LDA [8]	.01±.00	.02±.00	.03±.00	.05±.00	.01±.00	.03±.00	.10±.00	.01±.00
PP-GCN [58]	.32±.04	.37±.04	**.37±.04**	.39±.03	.39±.01	.39±.06	.40±.06	**.26±.03**
EventX [44]	.01±.00	.02±.00	.01±.00	.02±.00	.01±.00	.03±.00	.02±.00	.01±.00
QSGNN [63]	.30±.04	.37±.02	.24±.03	.40±.04	.33±.03	.47±.03	.32±.02	.25±.02
KPGNN [11]	.23±.02	.38±.02	.25±.02	.46±.02	.36±.05	.47±.03	.37±.02	**.26±.02**
CLKD-w/o LA-w/o RKD	.27±.03	**.45±.05**	.24±.03	.44±.01	.38±.01	.58±.02	.44±.01	.22±.03
CLKD-w/o LA	.35±.05	.39±.05	.25±.02	.58±.01	.39±.01	.54±.02	.53±.03	.20±.03
CLKD-LLA	.30±.02	.37±.03	.24±.03	.52±.02	.36±.03	**.60±.03**	**.59±.04**	.22±.04
CLKD-NLA	**.42±.03**	.33±.02	.28±.01	**.60±.02**	**.40±.01**	.50±.03	.57±.02	.22±.01
promotion	↑ 10%	↑ 7%	↓ 9%	↑ 14%	↑ 1%	↑ 13%	↑ 19%	↓ 4%

goal of the task is to detect events in the French message blocks $M_1, M_2, ..., M_{16}$. The corresponding English message blocks $M_1, M_2, ..., M_{16}$ act as assistants, and blocks $M_{17}, M_{18}, ..., M_{22}$ are treated as redundant data in this incremental experiment.

As shown in Fig. 4(b) and described in Algo. 2, armed with mutual guidance, the two networks are updating in a synchronous way. For the French network trained with message block M_i on the French dataset, its peer is the network trained with the corresponding message block M_i on English data. Note that PP-GCN is an offline-only method, so it is not directly applicable to this dynamic setting. Therefore, to form a proxy, we train a new PP-GCN model from scratch for each message block, using the previous block as the training set and making predictions

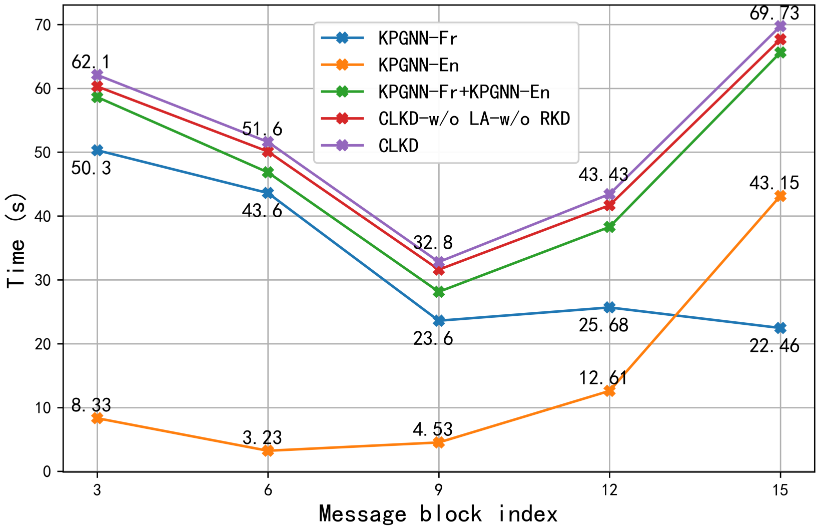


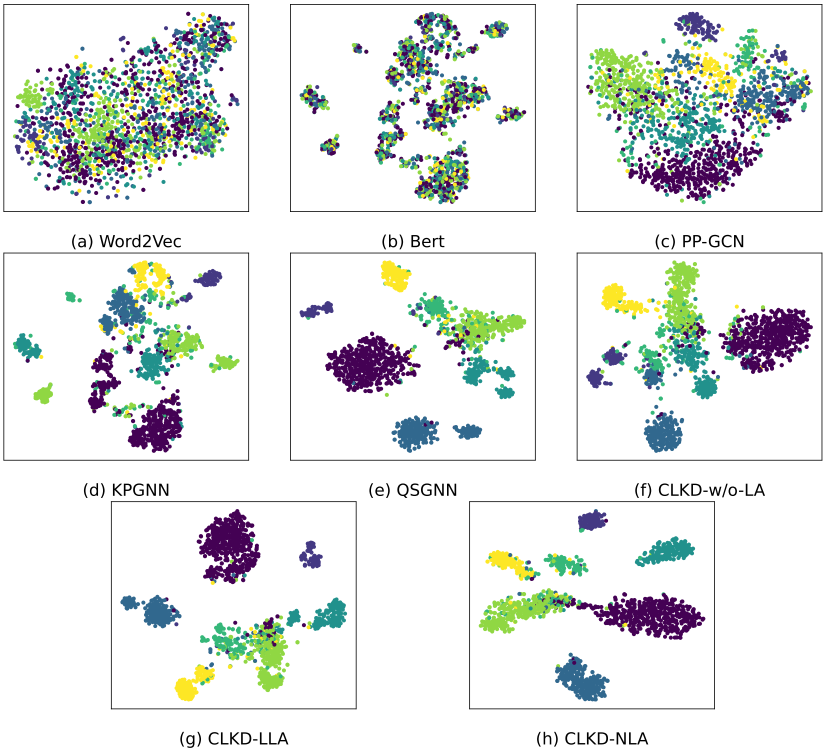
Fig. 7. **Time Consumption. (We have added some specific values and adjusted the position of the legend.)**

on the current block. As for QSGNN, we follow the same unsupervised training method as in the original paper, i.e., first using the original model to generate quality-aware pseudo labels and then updating the model with those generated labels every three blocks.

The results for the French dataset are summarized in Table V, VI, and VII (NMI, AMI, and ARI). The CLKD framework yields better results than KPGNN and the other baselines in almost all message blocks, which demonstrates the advantage of using English prior as a training assistant to low-resource languages. Further analyzing the results, we can see that the linear or non-linear cross-lingual module performs best for most message blocks. This is consistent with our observation in the offline experiments that adding the cross-lingual module further improves the model performance. Meanwhile, we notice that a linear transformation is more appropriate for English-French pairs, which is also consistent with our offline experimental results. For some blocks, like M_9, a linear transformation is not the best choice. This is reasonable and acceptable because not every language pair is fully isomorphic. Although French and English are very close, the pre-trained linear mapping is likely to suit most cases but "most" does not mean "all".

4.3 Time Consumption

To demonstrate that the proposed CLKD framework is practical and can be used for scalable training, we conduct some time trials of the CLKD on the French dataset in an online setting. Fig. 7 shows the results. We use a mini-batch sampling strategy during the training process, recording the time for one mini-batch. As the cross-lingual language models are pre-trained, the transformed attribute features are obtained at the data processing stage. Thus, the time consumption for the three variants with knowledge distillation is the same. The time consumption of CLKD-w/o LA-w/o RKD is used to represent the time needed for all three variants and the time consumption of CLKD is the total time consumption for the whole framework. For a more intuitive comparison, we also record the time KPGNN needed to process a single French message block and its corresponding English message block, denoted as KPGNN-Fr and KPGNN-En, respectively. As shown in Fig. 7 and consistent with the time complexity analysis in Section 3.5, the time needed for the CLKD framework to process the combined data stream is almost



(a) Word2Vec

(b) Bert

(c) PP-GCN

(d) KPGNN

(e) QSGNN

(f) CLKD-w/o-LA

(g) CLKD-LLA

(h) CLKD-NLA

Fig. 8. **Visualization of the learned message representations on the Arabic dataset.**

equal to the sum of KPGNN-Fr and KPGNN-En. Moreover, as discussed in Section 3.4, we only use the newest message block data for training in the maintenance stage. Hence, obsolete nodes are deleted and the training regime stays light. Overall, these results prove that the CLKD framework can be used for scalable training.

4.4 Visualization

To give a more intuitive comparison, and to further show the extent to which knowledge distillation and the cross-lingual module help the event detection process, we prepare a visualization of the Arabic Twitter dataset. For this, we plot the representations of the test set using t-SNE [73] using calculated message embeddings from Word2Vec and BERT; the output embeddings from the last layer of PP-GCN, KPGNN, and QSGNN; and the same from our variants with knowledge distillation, i.e., CLKD-w/o LA, CLKD-LLA, and CLKD-NLA. The results are shown in Fig. 8. Noticed that in the Arabic Twitter dataset, there are 7 catastrophic events in total: *the Jordan floods, Kuwait floods-18, Hafr Albatin floods-19, the Cairo bombing, the Dragon storms, the Beirut explosion, and Covid-19.* Each color in Fig. 8 represents an event. Comparing Fig. 8(a) and (b) with (c), (d), and (e), it is obvious that the results are worse in (a) and (b). Thus, we conclude that GNN-based methods are better than the methods that rely purely on learning message representations, i.e., Word2Vec and BERT. As explained, this is largely due to the expressive power of GNNs in capturing structural information and rich semantics at the same time. Meanwhile, if we compare Fig. 8(d) with (e), it is obvious that inter-class representations learned by QSGNN are better split than the ones in KPGNN. That is because QSGNN adds a stricter inter-class distance demand as

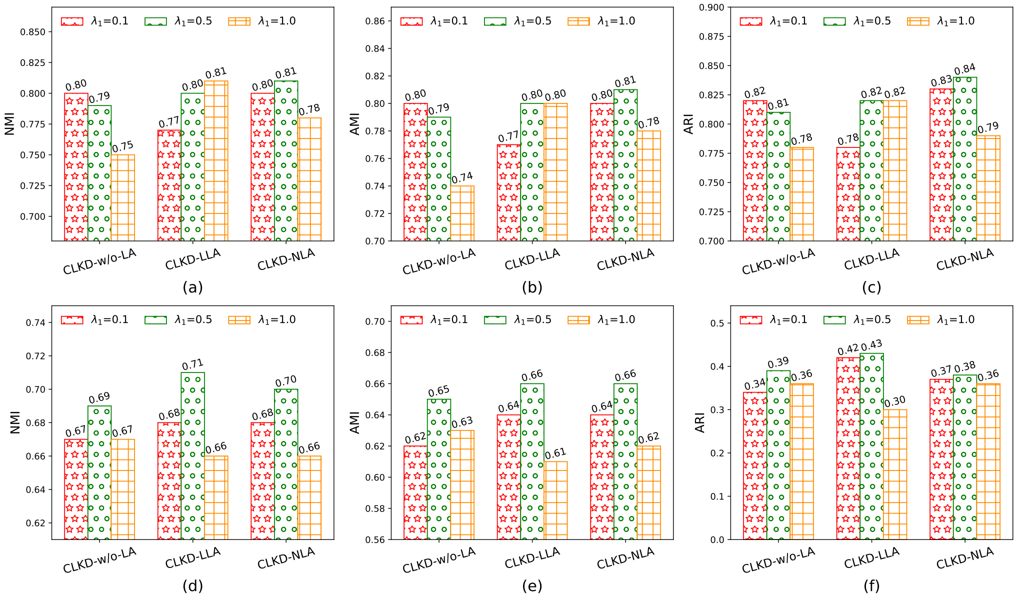


Fig. 9. **Analysis of λ_1 in the offline situation.** (a), (b), (c) show the results on the Arabic dataset; and (d), (e), (f) show the results on the French dataset.

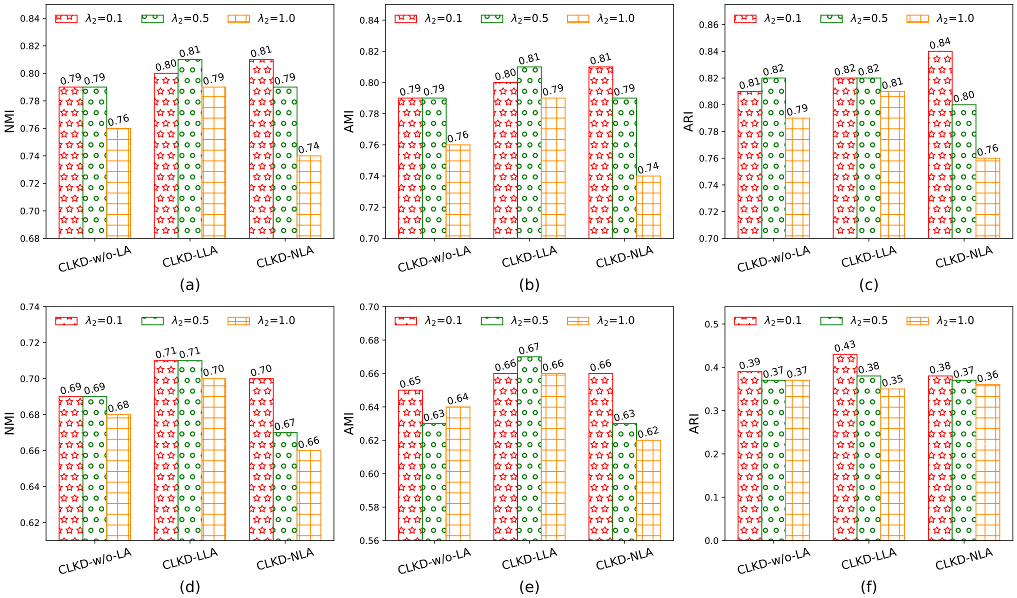
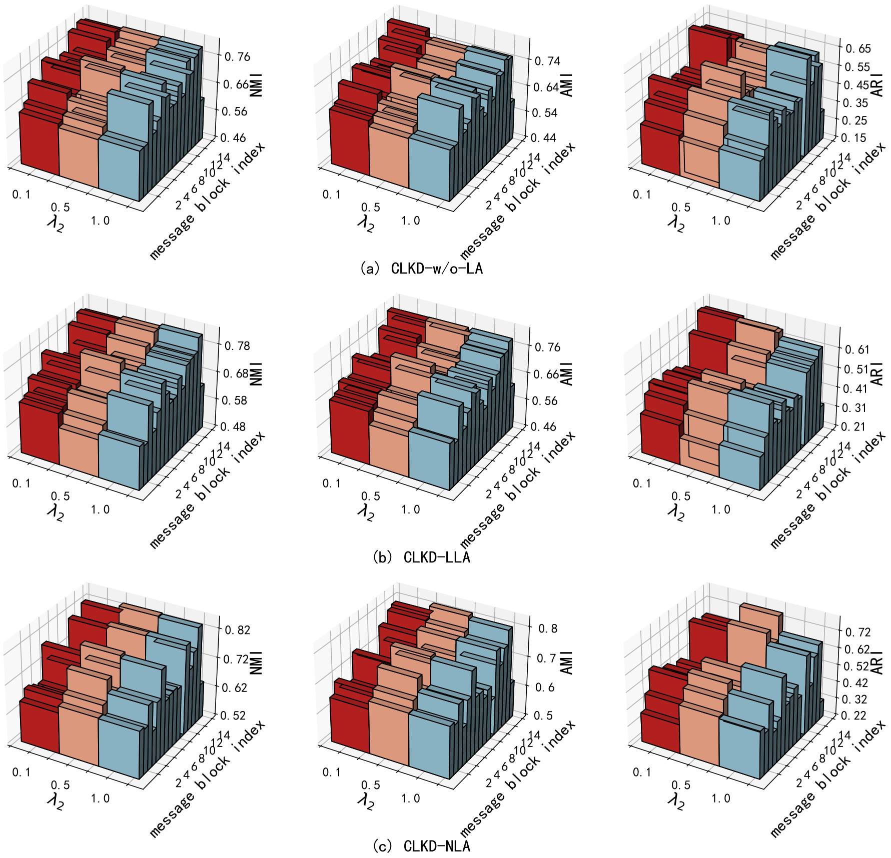


Fig. 10. **Analysis of λ_2 in the offline situation.** (a), (b), (c) show the results on the Arabic dataset; and (d), (e), (f) show the results on the French dataset.



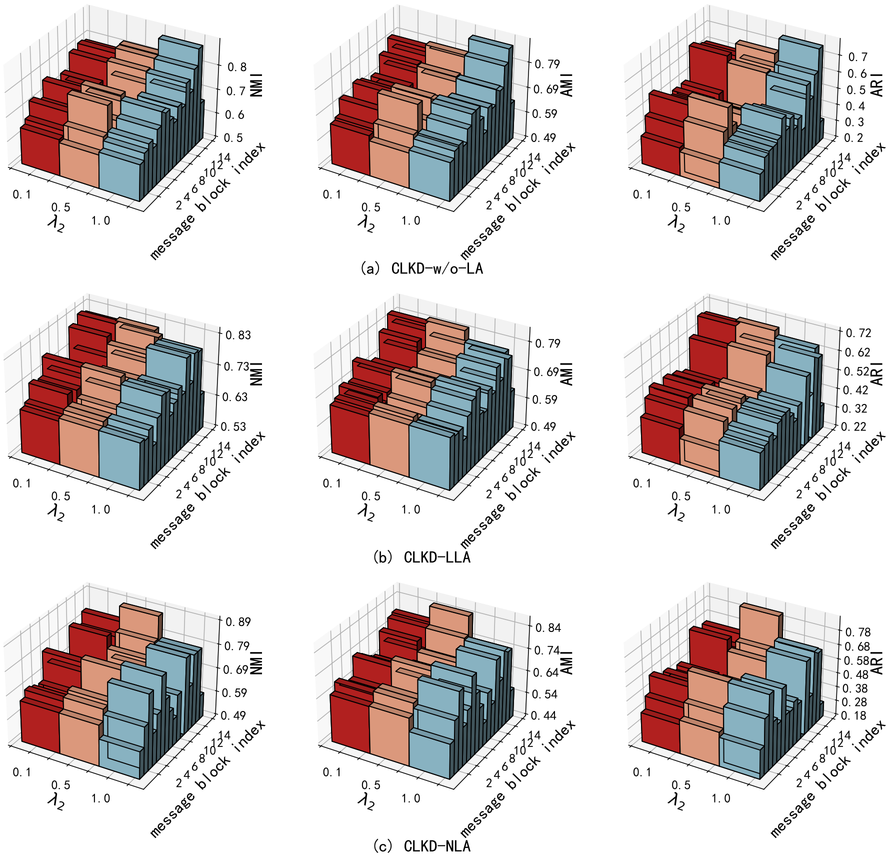
(a) CLKD-w/o-LA

(b) CLKD-LLA

(c) CLKD-NLA

Fig. 11. **Analysis of λ_1 in the online situation.** Noticed that the x-axis denotes message block index. (a), (b), and (c) show the results of the CLKD-w/o LA variant, CLKD-LLA variant, and CLKD-NLA variant on the French dataset in an incremental setting, respectively.

well as an orthogonal constraint in inter-class feature direction. Comparing Fig. 8(d) with (f), (g), and (h), we can see that the three variants that learn with prior knowledge do better. This demonstrates the effectiveness of knowledge distillation. What's more, the variants that incorporate the cross-lingual module, Fig. 8 (f) and (g), even have more distinct boundaries than (e) (i.e., the powerful QSGNN model), which speaks to the importance of the cross-lingual module. In essence, the cross-lingual feature-wise and beneficial relation-wise supervision from the teacher can also be seen as adding stricter distance constraints to inter-class and intra-class pairs.



(a) CLKD-w/o-LA

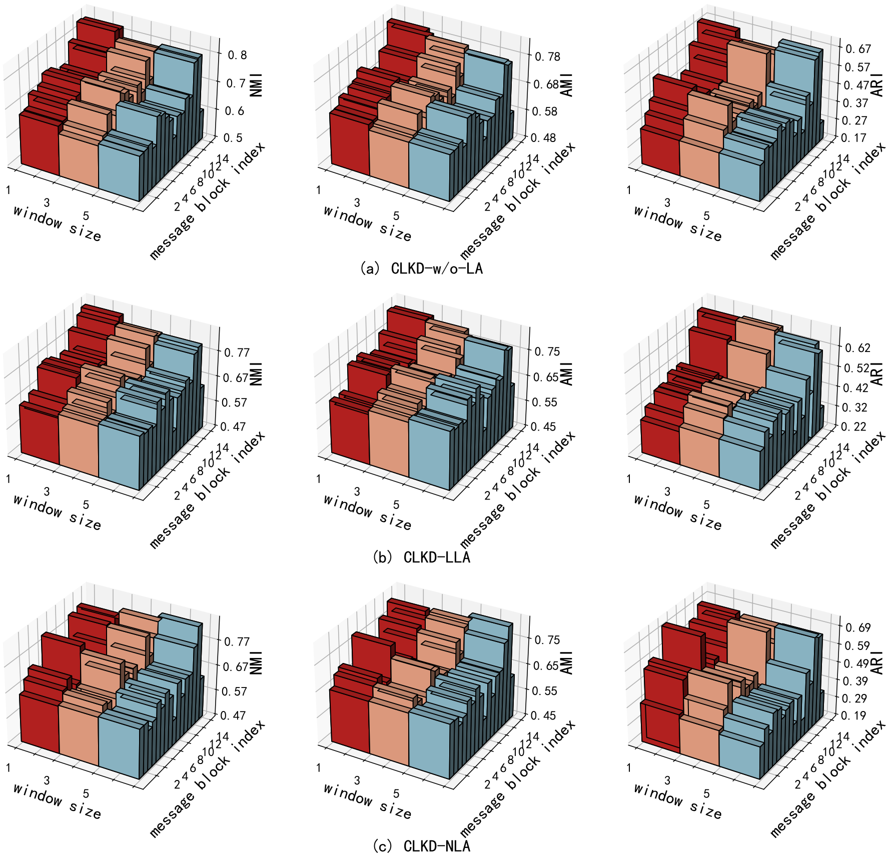
(b) CLKD-LLA

(c) CLKD-NLA

Fig. 12. **Analysis of λ_2 in the online situation.** Noticed that the x-axis denotes message block index. (a), (b), and (c) show the results of the CLKD-w/o LA variant, CLKD-LLA variant, and CLKD-NLA variant on the French dataset in an incremental setting, respectively.

4.5 Hyper-parameter Analysis

Analysis of the weight of feature-wise knowledge distillation loss λ_1. Note that λ_1 and λ_2 are the most important hyper-parameters to tune in the objective function of the CLKD framework. Fig. 9(a), (b), and (c) demonstrate how the performance of the CLKD framework (Teacher-Student) is affected by the choice of λ_1 on the Arabic dataset in the offline situation, and Fig. 9(d), (e), and (f) show the performance on the French dataset. We validate different values of λ_1 ranging from 0.1 to 1.0 ({0.1, 0.5, 1.0}), noting that the choice of λ_1 determines to which degree the student's learning features are affected by the teacher's. With a large value, the feature-wise



(a) CLKD-w/o-LA

(b) CLKD-LLA

(c) CLKD-NLA

Fig. 13. **Analysis of w in the online situation.** (a), (b), and (c) show the results of the CLKD-w/o LA variant, CLKD-LLA variant, and CLKD-NLA variant on the French dataset in an incremental setting respectively.

Knowledge distillation loss will play a more important role, which means representations learned by students are forced to be more similar to the teachers. Considering the pre-trained teacher network with a cross-lingual module has been validated to gain great results in detecting events from student language data, therefore, a relatively large weight can be chosen in offline situations. We set λ_1 to 0.5. This is also borne out in the result that the model achieves the best results when λ_1 is set to 0.5. For example, on the French dataset, when $\lambda_1 = 0.5$, with 0.71 NMI, 0.66 AMI, and 0.43 ARI, CLKD-LLA provides the best performance. On the Arabic dataset, the same is true but of CLKD-NLA, with 0.81 NMI, 0.81 AMI, and 0.84 ARI. However, regarding CLKD-w/o LA, larger λ_1 brings worse results on the Arabic dataset. This is reasonable since CLKD-w/o LA has not considered cross-lingual bias. Thus blindly making the representations of the teacher and students similar may hinder the model training. Fig. 11 shows the results for the French dataset in a mutual learning scheme with varying λ_1. As shown in Fig. 11, for most message blocks, the choice of λ_1 doesn't make much difference to the NMI, AMI, and ARI results. The three lines basically coincide. Hence, we can conclude that the CKLD framework in the mutual learning structure shows robust performance at all the selected λ_1 values on the French dataset.

Analysis of the weight of relation-wise knowledge distillation loss λ_2. Similar to λ_1, we also choose the value of λ_2 from $\{0.1, 0.5, 1.0\}$ and plot the results in Fig. 10 and Fig. 12. Specifically, Fig. 10(a), (b), and (c) demonstrate how the performance of the CLKD framework (Teacher-Student) is affected by the choice of λ_2 on the Arabic dataset and (d), (e) and (f) show the performance on the French dataset. As can be seen from the figure, the model with small λ_2 gains better results. For example, on the Arabic dataset, when λ_2 is set to 0.1, CLKD-NLA performs best with 0.81 NMI, 0.81 AMI, and 0.84 ARI. The performance of the model on French data shows a similar pattern. With 0.71 NMI, 0.66 AMI, and 0.43 ARI, CLKD-LLA achieves the best results when λ_2 is set to 0.1. However, setting λ_2 to a large value (i.e., $\lambda_2 = 1$) will decrease the model performance. As aforementioned, adding relation-wise knowledge distillation can also be seen as adding a stricter distance constraint to the inter-class and intra-class pairs. When λ_2 is large, the model may focus too much on the hardest positive and negative samples, resulting in bad local minima. As for the online situation, Fig. 11 shows the results for the French dataset with varying λ_2. Still, for most message blocks, the choice of λ_2 doesn't make much difference to the final results. This validates that the Mutual-Learning structure is more insensitive to parameter λ_2.

Analysis of the maintaining window size w. w determines the model update frequency when dealing with online social streams. Fig. 13 charts CLKD's performance in the Mutual-Learning structure with different sizes of w. Generally, performance is better with a small w, such as $w = 1$ and $w = 3$. For example, with window sizes $w = 1, 3, 5$, the average NMI results of CLKD-LLA are 0.64, 0.63, and 0.62, respectively. That is because smaller w means more frequent model updates. Specifically, when $w = 1$, the model is updated in each message block by continuously training with the current block data. In this way, the model is fully adapted to the social stream data. However, a small w also requires more training time. For a good balance between performance and efficiency, we select $w = 3$ in our experiments.

5 EXTENSION,IMPLICATION AND LIMITATION

In this section, we test the CLKD framework on other pre-trained language models and discuss its limitations.

Extension of CLKD to other pre-trained language models, graph neural network encoders and cross-lingual mappings: Note that in the above experiments, the initial representations of messages are obtained by GloVe [62], and the cross-lingual module (i.e., the linear mapping model MUSE and the non-linear mapping LNMAP) is also trained by the word embeddings learned from GloVe. Here we test another static pre-trained language model fastText [9]. Specifically, we use the open-source pre-trained 300-d language models[7] trained on Common Crawl and Wikipedia using fastText to learn message representations and language mappings. We record the offline evaluation on French and Arabic datasets (see Table. VIII) and the online evaluation on French dataset (see Table. IX, X, and XI). Similarly, our CLKD framework with linear or non-linear language mappings learned from fastText also obtains the best results, which validates the robustness of our CLKD framework. However, carefully watching the promotions by distilling only point-wise information (CLKD-w/o LA-w/o RKD) to the original KPGNN, we find that using the representations learned from Glove achieves more improvements. That may be because the learning process of Glove focuses more on word pair relations. Thus the calculated point-wise representation contains richer knowledge.

It is also essential to emphasize that while the proposed Cross-Lingual Knowledge Distillation (CLKD) framework utilizes established components like the pre-existing KPGNN network structure and cross-lingual mappings

[7] https://fasttext.cc/docs/en/crawl-vectors.html

such as MUSE and LNMAP, its core innovation lies in integrating and extending these components with novel distillation techniques. These techniques include cross-lingual feature-wise distillation and selective relation-wise distillation, alongside experimental validations. The KPGNN encoder employed in the CLKD serves primarily an illustrative purpose. To demonstrate the versatility and robustness of the CLKD framework, we conduct extensive experiments by substituting the specific GNN encoder in KPGNN with a temporal-aware graph neural network encoder from another study, as cited in [64]. The results, shown in Table XII, reveal that incorporating a GNN encoder that integrates temporal data further improves event detection, especially in the Arabic dataset, where events span various periods. Note that in all versions of CLKD listed in Table XII, the temporal GNN encoder is utilized. The enhanced performance of the complete CLKD framework with the alternative temporal-aware GNN encoder (i.e., CLKD-LLA, CLKD-NLA), as compared to the vanilla versions (i.e., CLKD-w/o LA-w/o RKD, CLKD-w/o LA), underscores the effectiveness of the proposed feature-wise and relation-wise distillation techniques in CLKD. Furthermore, to demonstrate the scalability of our CLKD framework, we also evaluate it using other cross-lingual mappings such as VecMap [4], in place of MUSE and LNMAP. The comparison of different mapping strategies, presented in Table XIII, shows that employing VecMap for cross-lingual transformation within the CLKD framework also improves detection capabilities compared to the version without language alignment (i.e., CLKD-w/o LA), further affirming the scalability and effectiveness of the CLKD framework.

The implication of CLKD: This work proposes the Cross-Lingual Knowledge Distillation (CLKD) framework to address social event detection problems in low-resource languages. By combining cross-lingual transformation techniques with knowledge distillation, our framework successfully achieves cross-lingual transfer learning and greatly strengthens detection models in low-resource languages. This approach has broader implications beyond event detection and can improve other natural language processing applications, such as sentiment analysis and question answering systems, in settings where data scarcity is a concern. Overall, our CLKD framework offers a promising solution for enhancing NLP tasks in low-resource language environments.

The limitation of CLKD: Here we acknowledge the limitation of our CLKD framework. Certainly, assuming a universal distribution of events across all languages may not universally hold true due to cultural disparities. However, despite this challenge, valuable insights can still be gained. Many human behaviors are indeed universal across different countries, allowing for useful cross-lingual knowledge extraction to enhance each other for many language pairs. For instance, our experiments with English-French and English-Arabic pairs demonstrate tangible benefits in distilled event detection knowledge. As shown in Table II, directly inputting French and Arabic datasets into the detection model trained on the English dataset yields NMI scores of 0.50±0.01 and 0.39±0.01, respectively. In regions with limited cultural, historical, and economic similarities, CLKD may not perform optimally. Conducting experiments with validation data beforehand may provide insights into its effectiveness in those language pairs. Specifically, applying a high-resource language-trained model (e.g., the English-trained model) directly to detect events in other languages offers a preliminary assessment of its potential efficacy. We can then adjust the distillation language pairs based on this assessment. This pragmatic approach facilitates practical evaluation and adjustment based on real-world performance, ensuring adaptability and refinement in cross-lingual event detection. Another potential limitation of this work is the challenge of ensuring high-quality cross-language conversions. Note that most cross-lingual transformation studies assume that languages share similar underlying structures and semantics. However, languages can exhibit significant divergence in syntax, vocabulary, and cultural context. This language divergence can limit the effectiveness of transferring knowledge from high-resource to low-resource languages, especially when the languages in question are vastly different. Non-linear language alignment techniques can partially address this issue, but they still cannot guarantee high-quality transformations for significantly different languages.

Table VIII. Offline Evaluation on French and Arabic datasets using representations and language mappings learned from fastText.

Methods	French Data			Arabic Data		
	NMI	AMI	ARI	NMI	AMI	ARI
KPGNN [11]	.63±.02	.59±.01	.28±.03	.76±.01	.74±.02	.76±.02
CLKD-w/o LA-w/o RKD	.63±.01	.60±.01	.28±.02	.76±.03	.75±.02	.77±.03
CLKD-w/o LA	.66±.04	.63±.03	.35±.04	.78±.02	.79±.02	.79±.02
CLKD-LLA	**.70±.02**	**.65±.01**	**.39 ±.02**	.80±.03	**.81±.03**	.81±.03
CLKD-NLA	**.70±.02**	.64±.01	.38±.02	**.81±.02**	.80±.02	**.83±.02**

Table IX. Incremental evaluation NMIs on the French dataset using representations and language mappings learned from fastText.

Blocks	M_1	M_2	M_3	$M4$	M_5	M_6	M_7	M_8
KPGNN [11]	.55±.01	.55±.02	.54±.02	.56±.02	.58±.02	.60±.02	.63±.02	.59±.02
CLKD-w/o LA-w/o RKD	.54±.01	.56±.01	.54±.02	.56±.02	.60±.02	.60±.03	.63±.02	.60±.01
CLKD-w/o LA	.56±.02	.57±.02	.57±.03	**.57±.03**	.60±.02	.61±.02	**.64±.02**	.60±.01
CLKD-LLA	**.58±.02**	**.58±.01**	**.62±.01**	**.57±.02**	**.62±.02**	.64±.02	**.64±.02**	.61±.02
CLKD-NLA	.57±.02	**.58±.02**	.60±.02	.56±.03	.61±.02	**.65±.02**	.63±.02	**.62±.02**

Blocks	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
KPGNN [11]	.47±.02	.59±.02	.57±.02	.56±.03	.60±.02	.66±.02	.61±.01	.53±.02
CLKD-w/o LA-w/o RKD	.50±.02	.60±.02	.57±.01	.61±.01	.63±.01	67±.02	.62±.02	.52±.02
CLKD-w/o LA	.51±.02	.62±.02	.58±.02	.64±.01	.63±.01	69±.02	.65±.02	.50±.02
CLKD-LLA	.53±.02	**.63±.02**	**.59±.02**	.67±.01	**.67±.02**	**.71±.01**	.70±.02	**.54±.02**
CLKD-NLA	**.55±.02**	**.63±.02**	.58±.02	**.68±.01**	.64±.03	.68±.02	**.71±.02**	.53±.04

6 RELATED WORK

Social Event Detection. According to survey [33], based on their objectives, social event detection can be broadly categorized as either feature-pivot (FP) [27] or document-pivot (DP) [81] methods. When separated by the techniques they use, social event detection methods can be divided into incremental clustering [1, 38, 54, 85], community detection [25, 44–46, 69, 84] and topic modeling [7, 16, 20, 59, 89]. More recently, with the great success of Graph Neural Networks [47], there has been a move towards GNN-based social event detection [11, 21, 58, 60, 61, 64]. Peng et al. [58] use Graph Convolutional Network as their event categorization model. Ren et al. [64] incorporate temporal information into the message-passing scheme of GNN and propose ETGNN. To further extend ETGNN to imbalanced social event detection tasks, they design an uncertainty-guided contrastive loss [65] to regulate the representation learning of GNN. However, the methods mentioned above are only applicable to closed-set settings. Scaling to incremental setting, Cao et al. [11] leverage inductive GNNs to extract information. Further, Peng et al. [61] extend the GNN encoder in [11] by using reinforcement learning techniques to assign different thresholds for different relations. Later, Cao et al. [12] approach the social event detection task from a

Table X. Incremental evaluation AMIs on the French dataset using representations and language mappings learned from fastText.

Blocks	M_1	M_2	M_3	$M4$	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
KPGNN [11]	.54±.02	.54±.01	.54±.02	.56±.01	.57±.01	.57±.02	.62±.02	.58±.02	.46±.03	.58±.02	.55±.01	.55±.04	.59±.02	.65±.02	.60±.01	.52±.02
CLKD-w/o LA-w/o RKD	.55±.00	.55±.01	.55±.02	.56±.01	.57±.02	.56±.02	.62±.02	.57±.02	.50±.02	.59±.02	.56±.02	.58±.01	.61±.02	66±.02	.61±.02	.51±.02
CLKD-w/o LA	.55±.02	.56±.02	.57±.02	.55±.01	.58±.02	.59±.02	**.63±.02**	.58±.02	.51±.02	.61±.02	.58±.02	.62±.01	.62±.01	68±.02	.64±.02	.51±.02
CLKD-LLA	**.56±.02**	.57±.00	**.59±.02**	**.57±.03**	**.59±.02**	**.62±.02**	.62±.02	.59±.02	.54±.01	**.62±.02**	**.60±.02**	**.66±.01**	**.63±.02**	**.70±.02**	.69±.02	**.53±.03**
CLKD-NLA	**.56±.01**	**.58±.02**	.58±.01	.55±.03	.58±.01	.61±.02	.62±.02	**.61±.02**	**.55±.01**	.61±.02	.59±.02	**.66±.01**	.62±.02	.68±.02	**.70±.02**	**.53±.03**

Table XI. Incremental evaluation ARIs on the French dataset using representations and language mappings learned from fastText.

Blocks	M_1	M_2	M_3	$M4$	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
KPGNN [11]	.29±.02	.37±.02	.40±.02	.37±.04	.36±.02	.36±.04	.38±.02	.38±.02	.24±.02	.39±.02	.27±.02	.49±.02	.33±.05	.48±.03	.36±.02	.24±.02
CLKD-w/o LA-w/o RKD	.29±.03	.38±.04	.36±.03	.36±.03	.36±.02	.33±.03	.37±.02	.38±.02	.27±.03	.40±.05	.24±.03	.47±.03	.34±.01	.50±.02	.43±.01	.22±.03
CLKD-w/o LA	.29±.01	.39±.02	.41±.02	.37±.03	**.37±.02**	.35±.02	.38±.03	.39±.01	.35±.05	.39±.05	.25±.02	.52±.01	.36±.01	.54±.02	.49±.05	.23±.04
CLKD-LLA	**.31±.02**	.39±.02	.43±.03	**.40±.04**	.35±.03	.42±.02	.37±.04	**.40±.03**	.39±.04	**.41±.03**	**.28±.04**	.56±.02	**.39±.03**	**.58±.03**	**.57±.04**	**.25±.04**
CLKD-NLA	.30±.02	**.40±.05**	**.45±.04**	.36±.03	.36±.04	**.45±.02**	**.39±.02**	.39±.02	**.40±.04**	**.41±.04**	.27±.01	**.60±.02**	.38±.01	.57±.04	.56±.02	**.25±.03**

novel perspective by fully exploring message correlations through graph structure entropy minimization. Li et al. [43] propose a multi-relational prompt-based pairwise message learning mechanism to simultaneously utilize structural and content information. Compared to early studies, GNN-based methods show their superiority in knowledge acquisition and preservation. However, although GNN-based models achieve fairly high accuracy, their application to date has been heavily restricted to high-resource monolingual data – especially English. Only a few works focus on non-English languages. For example, in [48], a french corpus is annotated for event detection tasks. In [6], SVM is adopted for detection on an Indian dataset. There is still no effective deep learning

Table XII. Offline Evaluation on French and Arabic datasets using the Temporal-aware GNN encoder.

Methods	French Data			Arabic Data		
	NMI	AMI	ARI	NMI	AMI	ARI
KPGNN [11]	.63±.02	.59±.01	.28±.03	.76±.01	.74±.02	.76±.02
Temporal-aware GNN	.66±.03	.61±.02	.32±.02	83±.02	.79±.02	.84±.02
CLKD-w/o LA-w/o RKD	.65±.02	.62±.02	.33±.02	.83±.03	.80±.02	.84±.02
CLKD-w/o LA	.70±.02	.63±.03	.36±.02	.84±.02	.81±.02	.84±.03
CLKD-LLA	**.72±.02**	**.68±.01**	**.41 ±.03**	.85±.03	.84±.02	.84±.03
CLKD-NLA	.71±.02	.67±.02	.39±.03	**.87±.02**	**.86±.02**	**.87±.02**

Table XIII. Offline Evaluation on French and Arabic datasets using different cross-lingual mappings.

Methods	French Data			Arabic Data		
	NMI	AMI	ARI	NMI	AMI	ARI
CLKD-w/o LA	.66±.04	.63±.03	.35±.04	.78±.02	.79±.02	.79±.02
CLKD (with MUSE)	.70±.02	**.65±.01**	.39 ±.02	.80±.03	**.81±.03**	.81±.03
CLKD (with LNMAP)	.70±.02	.64±.01	.38±.02	**.81±.02**	.80±.02	**.83±.02**
CLKD (with VecMap)	**.72±.02**	**.65±.01**	**.40±.03**	.80±.01	.80±.02	.82±.02

framework generalized for event detection tasks with any low-resource languages. Also, existing multilingual event detection methods [46] are not capable of fully utilizing the rich information in social streams. Hence, in this work, we propose a novel multilingual social graph construction strategy that makes GNN models compatible with multilingual social media data and propose the CLKD framework for segment detection for low-resource languages.

Knowledge Distillation. Knowledge distillation is initially adopted for model compression [10], to learn a compact student model from a larger teacher model [35] by letting the student imitate the output of the teacher. Two key factors here are what knowledge to learn and how to effectively transfer it. Regarding the former, popular supervision extracted from the teacher includes the forms of class posterior probabilities [35], feature representations [5, 66], and inter-layer flows (the inner product of feature maps) [82]. As for the knowledge transfer process, to penalize the difference between student and teacher, the distillation loss can be considered as cross-entropy loss, Kullback-Leibler (KL) divergence loss, etc. Noticed that classical Teacher-Student distillation methods follow an offline training strategy that involves two phases of training. Zhang et al. [87] overcome this limitation with a one-phase online training regime that distils knowledge between two student models acting as peers to each other. Recently, there has been a rising interest in cross-modality knowledge distillation which transfers supervision across modalities, see [31, 40, 70]. Li et al. [40] devise a mutual knowledge distillation scheme that exploits prior knowledge learned from a source modality to improve the performance in the target modality - the goal being to overcome annotation scarcity. We share a similar philosophy but treat different languages as different modalities. Due to its strong capability in transferring knowledge, knowledge distillation has also been widely adopted in applications in the natural language processing (NLP) domain [13, 32, 55]. For instance, Gupta et al. [32] adopt knowledge distillation approaches to train Answer Sentence Selection (AS2) models for low-resource languages. Zheng et al. [13] utilize the "dark knowledge" acquired from large teacher models and adaptive hints to address domain differences between teacher and student models. Their approach achieves successful application in Question Answering (QA) systems. Pan et al. [55] introduce Meta-KD, a framework for meta knowledge distillation designed for compressing language models. Besides, given the recent success of self-supervised learning, which allows models to learn from extensive unlabeled data by generating substitute supervisory signals, a series of studies have integrated self-supervised learning into the knowledge distillation process [14, 88]. For example, Chen et al. [14] propose a stratified distillation strategy to address bias in knowledge distillation that comes from teacher-provided soft labels. This method partitions items into multiple groups based on popularity and extracts ranking knowledge within each group. Zheng et al. [88] develop a cross-level knowledge distillation approach to tackle cross-domain few-shot classification by incorporating a small portion of unlabeled images from the target domain during training. More recently, with the development of GNN, a series of methods [26, 80] have applied knowledge distillation on GNNs to achieve high efficiency. They distil graph knowledge extracted by high-capacity teacher GNN models to lightweight students via penalizing soft logit differences between teachers and students. Xia et al. [78] further consider the over-smoothing problem of GNN and adopt representation recalibration to address it. Later, considering GNNs are often shallow, Wang et al. [75] propose a novel multi-teacher KD method to learn combined knowledge from multiple teachers in a parallel way. Guo et al. [30] instead propose BGNN which also transfers knowledge from multiple GNNs into a student GNN but in a sequential way. Unlike these methods which focus on extracting complementary knowledge from multiple teachers, our CLKD framework prioritizes the maximum utilization of knowledge from a single teacher GNN, making it a more cost-effective approach.

Relational-Knowledge based methods. Unlike classical knowledge distillation methods which transfer point-wise information such as soft logits, feature representations and so on, a group of methods instead explore relational knowledge. These approaches [28, 56, 71, 76], also named relation distillation, attempt to extract higher-order structure information from the teacher's output space. For example, Park et al. [56] simultaneously compare both angle-wise and distance-wise relations between features. In [72], it has been observed that semantically similar inputs tend to elicit similar activation patterns. Therefore, a similarity-preserving knowledge distillation method is designed. It demands the student to preserve the pairwise similarities instead of simply mimicking original features from the teacher. Peng et al. [57] further expand pair-wise similarities to correlations among multiple instances and propose a correlation congruence knowledge distillation method. In recent years, it is worth noting that relational knowledge distillation has been widely employed in both the computer vision domain and natural language domain. Work [39] utilizes cluster methods to learn visual word vocabulary and leverages this vocabulary to quantize each location of feature maps. Work [24] proposes a hierarchical relational knowledge distillation to compress pre-trained language models. Inspired by the effectiveness of relational knowledge distillation, we design a hybrid knowledge distillation method which considers both point-wise and relation-wise information. Meanwhile, considering the existence of many improper relations made by the teacher, when transferring relation-wise information, we only distil beneficial relational knowledge.

Cross-Lingual Word Embeddings. Cross-Lingual Word Embeddings (CLWE) methods learn a shared word vector space, where words with similar meanings result in similar vectors regardless of the language they are originally expressed in. Early CLWE methods have been dominated by projection-based methods [51]. These approaches learn a linear projection by minimizing the distance between translation pairs in a training dictionary. The requirement of dictionary is later reduced with self-learning [3], and then removed via unsupervised initialization heuristics [4, 37] and adversarial learning [42, 86]. Almost all these methods inherently assume that the embedding spaces of the different languages are approximately isomorphic, i.e., that they are similar in geometric structure. However, this simplified assumption has been questioned by researchers recently. [53, 68] attribute the performance degradation of existing CLWE methods to the strong mismatches in embedding spaces caused by the linguistic and domain divergences. Due to this, Mohiuddin et al. [52] propose a novel non-linear

CLWE method based on two auto-encoders. In this work, we pay attention to the use of proper transformations between language pairs in social event detection tasks.

7 CONCLUSION

In this study, we focus on social event detection in low-resource languages. We have devised a novel cross-lingual knowledge distillation framework (CLKD) to borrow prior knowledge learned from the high-resource English language. Specifically, to extract more comprehensive information, the distillation contains both feature-wise and relation-wise parts. Furthermore, during feature-wise knowledge transfer, an additional cross-lingual module is combined to eliminate language bias. As for transferring relation-wise knowledge, we carefully select beneficial relations to avoid distraction caused by misjudgments made by the teacher. The merits of this approach are demonstrated in experiments with several real datasets. A particularly interesting future research direction would be cross-lingual event propagation.

ACKNOWLEDGEMENT

This work is supported by National Key R&D Program of China through grant 2022YFB3104703, NSFC through grants 62322202, U21B2027, 61972186, and 62266028, Beijing Natural Science Foundation through grant 4222030, Guangdong Basic and Applied Basic Research Foundation through grant 2023B1515120020, Yunnan Provincial Major Science and Technology Special Plan Projects through grants 202302AD080003, 202202AD080003 and 202303AP140008, General Projects of Basic Research in Yunnan Province through grant 202301AS070047, 202301AT070471, and the Fundamental Research Funds for the Central Universities. Jia Wu was supported by the Australian Research Council (ARC) Project DP230100899.

REFERENCES

[1] Charu C Aggarwal and Karthik Subbian. 2012. Event detection in social streams. In *SDM*. 624–635.

[2] Alaa Alharbi and Mark Lee. 2021. Kawarith: an Arabic Twitter Corpus for Crisis Events. In *WANLP*. 42–52.

[3] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *ACL*. 451–462.

[4] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *ACL*. 789–798.

[5] Jimmy Ba and Rich Caruana. 2014. Do Deep Nets Really Need to be Deep?. In *NIPS*. 2654–2662.

[6] Fazlourrahman Balouchzahi and H Shashirekha. 2020. An Approach for Event Detection from News in Indian Languages using Linear SVC. In *FIRE*. 25–28.

[7] Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond trending topics: Real-world event identification on twitter. In *AAAI*, Vol. 5. 438–441.

[8] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3 (2003), 993–1022.

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

[10] Cristian BUCILA, Rich CARUANA, and Alexandru NICULESCU-MIZIL. 2006. Model compression. In *KDD*. 535–541.

[11] Yuwei Cao, Hao Peng, Jia Wu, Yingtong Dou, Jianxin Li, and Philip S. Yu. 2021. Knowledge-Preserving Incremental Social Event Detection via Heterogeneous GNNs. *WWW* (2021), 3383–3395.

[12] Yuwei Cao, Hao Peng, Zhengtao Yu, and S Yu Philip. 2024. Hierarchical and incremental structural entropy minimization for unsupervised social event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8255–8264.

[13] Cen Chen, Chengyu Wang, Minghui Qiu, Dehong Gao, Linbo Jin, and Wang Li. 2021. Cross-domain Knowledge Distillation for Retrieval-based Question Answering Systems. In *WWW 2021*. 2613–2623.

[14] Gang Chen, Jiawei Chen, Fuli Feng, Sheng Zhou, and Xiangnan He. 2023. Unbiased Knowledge Distillation for Recommendation. In *WSDM 2023*. 976–984.

[15] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. 2017. Learning efficient object detection models with knowledge distillation. *NIPs* 30 (2017), 742–751.

[16] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. 2014. Btm: Topic modeling over short texts. *IEEE TKDE* 26, 12 (2014), 2928–2941.

[17] Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving Multilingual Models with Language-Clustered Vocabularies. In *EMNLP*. 4536–4546.

[18] Giorgio Ciano, Alberto Rossi, Monica Bianchini, and Franco Scarselli. 2021. On Inductive-Transductive Learning with Graph Neural Networks. *IEEE TPAMI* (2021), 758–769.

[19] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *ACL*. 8440–8451.

[20] Mário Cordeiro. 2012. Twitter event detection: combining wavelet analysis and topic inference summarization. In *Doctoral symposium on informatics engineering*, Vol. 1. 11–16.

[21] Wanqiu Cui, Junping Du, Dawei Wang, Feifei Kou, and Zhe Xue. 2021. MVGAN: Multi-View Graph Attention Network for Social Event Detection. *ACM TIST* 12, 3 (2021), 1–24.

[22] Xing Dai, Zeren Jiang, Zhao Wu, Yiping Bao, Zhicheng Wang, Si Liu, and Erjin Zhou. 2021. General instance distillation for object detection. In *CVPR*. 7842–7851.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL* (2018), 4171–4186.

[24] Chenhe Dong, Yaliang Li, Ying Shen, and Minghui Qiu. 2021. HRKD: Hierarchical Relational Knowledge Distillation for Cross-domain Language Model Compression. In *EMNLP*. 3126–3136.

[25] Mateusz Fedoryszak, Brent Frederick, Vijay Rajaram, and Changtao Zhong. 2019. Real-time event detection on social data streams. In *KDD*. 2774–2782.

[26] Kaituo Feng, Changsheng Li, Ye Yuan, and Guoren Wang. 2022. Freekd: Free-direction knowledge distillation for graph neural networks. In *SIGKDD*. 357–366.

[27] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S. Yu, and Hongjun Lu. 2005. Parameter free bursty events detection in text streams. In *VLDB*. Citeseer, 181–192.

[28] Shiming Ge, Kangkai Zhang, Haolin Liu, Yingying Hua, Shengwei Zhao, Xin Jin, and Hao Wen. 2020. Look one and more: Distilling hybrid order relational knowledge for cross-resolution image recognition. In *AAAI*, Vol. 34. 10845–10852.

[29] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. 2021. Distilling object detectors via decoupled features. In *CVPR*. 2154–2164.

[30] Zhichun Guo, Chunhui Zhang, Yujie Fan, Yijun Tian, Chuxu Zhang, and Nitesh V Chawla. 2023. Boosting graph neural networks via adaptive knowledge distillation. In *AAAI*, Vol. 37. 7793–7801.

[31] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. 2016. Cross modal distillation for supervision transfer. In *CVPR*. 2827–2836.

[32] Shivanshu Gupta, Yoshitomo Matsubara, Ankit Chadha, and Alessandro Moschitti. 2023. Cross-lingual knowledge distillation for answer sentence selection in low-resource languages. In *ACL Findings 2023*.

[33] Mahmud Hasan, Mehmet A Orgun, and Rolf Schwitter. 2018. A survey on real-time event detection from the twitter data stream. *Journal of Information Science* 44, 4 (2018), 443–463.

[34] Alexander Hermans, Lucas Beyer, and Bastian Leibe. 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737* (2017).

[35] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[36] Elad Hoffer and Nir Ailon. 2015. Deep metric learning using triplet network. In *SIMBAD*. Springer, 84–92.

[37] Yedid Hoshen and Lior Wolf. 2018. Non-Adversarial Unsupervised Word Translation. In *EMNLP*. 469–478.

[38] Linmei Hu, Bin Zhang, Lei Hou, and Juanzi Li. 2017. Adaptive online event detection in news streams. *Knowledge-Based Systems* 138 (2017), 105–112.

[39] Himalaya Jain, Spyros Gidaris, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. 2020. QuEST: Quantized embedding space for transferring knowledge. In *ECCV*. Springer, 173–189.

[40] Shujun Wang Kang Li, Lequan Yu and Pheng-Ann Heng. 2020. Towards Cross-Modality Medical Image Segmentation with Online Mutual Knowledge Distillation. In *AAAI*. 775–783.

[41] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*. 1–14.

[42] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *ICLR*. 1–14.

[43] Pu Li, Xiaoyan Yu, Hao Peng, Yantuan Xian, Linqin Wang, Li Sun, Jingyun Zhang, and Philip S Yu. 2024. Relational Prompt-based Pre-trained Language Models for Social Event Detection. *arXiv preprint arXiv:2404.08263* (2024).

[44] Bang Liu, Fred X Han, Di Niu, Linglong Kong, Kunfeng Lai, and Yu Xu. 2020. Story Forest: Extracting Events and Telling Stories from Breaking News. *ACM TKDD* 14, 3 (2020), 1–28.

[45] Fanzhen Liu, Shan Xue, Jia Wu, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Jian Yang, and Philip S. Yu. 2020. Deep Learning for Community Detection: Progress, Challenges and Opportunities. In *IJCAI*. 4981–4987.

[46] Yaopeng Liu, Hao Peng, Jianxin Li, Yangqiu Song, and Xiong Li. 2020. Event detection and evolution in multi-lingual social streams. *Frontiers of Computer Science* 14, 5 (2020), 1–15.

[47] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu. 2021. A Comprehensive Survey on Graph Anomaly Detection with Deep Learning. *IEEE TKDE* (2021), 1–1. https://doi.org/10.1109/TKDE.2021.3118815

[48] Béatrice Mazoyer, Julia Cagé, Nicolas Hervé, and Céline Hudelot. 2020. A french corpus for event detection on twitter. In *LREC*. 6220–6227.

[49] Andrew J McMinn, Yashar Moshfeghi, and Joemon M Jose. 2013. Building a large-scale corpus for evaluating event detection on twitter. In *KMIS*. 409–418.

[50] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*. 1–12.

[51] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* (2013).

[52] Tasnim Mohiuddin, M Saiful Bari, and Shafiq Joty. 2020. LNMap: Departures from Isomorphic Assumption in Bilingual Lexicon Induction Through Non-Linear Mapping in Latent Space. In *EMNLP*. 2712–2723.

[53] Aitor Ormazabal, Mikel Artetxe, Gorka Labaka, Aitor Soroa, and Eneko Agirre. 2019. Analyzing the Limitations of Cross-lingual Word Embedding Mappings. In *ACL*. 4990–4995.

[54] Ozer Ozdikis, Pinar Karagoz, and Halit Oğuztüzün. 2017. Incremental clustering with vector expansion for online event detection in microblogs. *Social Network Analysis and Mining* 7, 1 (2017), 1–17.

[55] Haojie Pan, Chengyu Wang, Minghui Qiu, Yichang Zhang, Yaliang Li, and Jun Huang. 2021. Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains. In *ACL/IJCNLP*. 3026–3036.

[56] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *CVPR*. 3967–3976.

[57] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. 2019. Correlation congruence for knowledge distillation. In *ICCV*. 5007–5016.

[58] Hao Peng, Jianxin Li, Qiran Gong, Yangqiu Song, Yuanxing Ning, Kunfeng Lai, and Philip S. Yu. 2019. Fine-grained event categorization with heterogeneous graph convolutional networks. *IJCAI* (2019), 3238–3245.

[59] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *WWW*. 1063–1072.

[60] Hao Peng, Jianxin Li, Yangqiu Song, Renyu Yang, Rajiv Ranjan, Philip S. Yu, and Lifang He. 2021. Streaming social event detection and evolution discovery in heterogeneous information networks. *ACM TKDD* 15, 5 (2021), 1–33.

[61] Hao Peng, Ruitong Zhang, Shaoning Li, Yuwei Cao, Shirui Pan, and Philip S. Yu. 2023. Reinforced, Incremental and Cross-Lingual Event Detection From Social Messages. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 1 (2023), 980–998.

[62] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.

[63] Jiaqian Ren, Lei Jiang, Hao Peng, Yuwei Cao, Jia Wu, Philip S. Yu, and Lifang He. 2022. From Known to Unknown: Quality-aware Self-improving Graph Neural Network For Open Set Social Event Detection. In *CIKM*. 1696–1705.

[64] Jiaqian Ren, Lei Jiang, Hao Peng, Zhiwei Liu, Jia Wu, and Philip S. Yu. 2022. Evidential Temporal-aware Graph-based Social Event Detection via Dempster-Shafer Theory. In *ICWS*. IEEE, 331–336.

[65] Jiaqian Ren, Hao Peng, Lei Jiang, Zhiwei Liu, Jia Wu, Zhengtao Yu, and S Yu Philip. 2023. Uncertainty-guided Boundary Learning for Imbalanced Social Event Detection. *TKDE* 01 (2023), 1–14.

[66] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. Fitnets: Hints for thin deep nets. In *ICLR*. 1–13.

[67] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*. 815–823.

[68] Anders Søgaard, Ivan Vulić, Sebastian Ruder, and Manaal Faruqui. 2019. Cross-lingual word embeddings. *Synthesis Lectures on Human Language Technologies* 12, 2 (2019), 1–132.

[69] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Di Jin, et al. 2022. A comprehensive survey on community detection with deep learning. *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–21.

[70] Fida Mohammad Thoker and Juergen Gall. 2019. Cross-modal knowledge distillation for action recognition. In *ICIP*. IEEE, 6–10.

[71] Jialin Tian, Xing Xu, Zheng Wang, Fumin Shen, and Xin Liu. 2021. Relationship-preserving knowledge distillation for zero-shot sketch based image retrieval. In *MM*. 5473–5481.

[72] Frederick Tung and Greg Mori. 2019. Similarity-preserving knowledge distillation. In *ICCV*. 1365–1374.

[73] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008), 2579–2605.

[74] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. (2018).

[75] Kai Wang, Yu Liu, Qian Ma, and Quan Z Sheng. 2021. Mulde: Multi-teacher knowledge distillation for low-dimensional knowledge graph embeddings. In *WWW*. 1716–1726.

[76] Kai Wang, Yifan Wang, Xing Xu, Xin Liu, Weihua Ou, and Huimin Lu. 2022. Prototype-based Selective Knowledge Distillation for Zero-Shot Sketch Based Image Retrieval. In *MM*. 601–609.

[77] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. 2019. Distilling object detectors with fine-grained feature imitation. In *CVPR*. 4933–4942.

[78] Lianghao Xia, Chao Huang, Jiao Shi, and Yong Xu. 2023. Graph-less collaborative filtering. In *WWW*. 17–27.

[79] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *NAACL*. 483–498.

[80] Cheng Yang, Jiawei Liu, and Chuan Shi. 2021. Extract the knowledge of graph neural networks and go beyond it: An effective knowledge distillation framework. In *WWW*. 1227–1237.

[81] Yiming Yang, Tom Pierce, and Jaime Carbonell. 1998. A study of retrospective and on-line event detection. In *SIGIR*. 28–36.

[82] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*. 4133–4141.

[83] Baosheng Yu, Tongliang Liu, Mingming Gong, Changxing Ding, and Dacheng Tao. 2018. Correcting the triplet selection bias for triplet loss. In *ECCV*. 71–87.

[84] Weiren Yu, Jianxin Li, Md Zakirul Alam Bhuiyan, Richong Zhang, and Jinpeng Huai. 2017. Ring: Real-time emerging anomaly monitoring system over text streams. *IEEE Transactions on Big Data* 5, 4 (2017), 506–519.

[85] Kuo Zhang, Juan Zi, and Li Gang Wu. 2007. New event detection based on indexing-tree and named entity. In *SIGIR*. 215–222.

[86] Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *ACL*. 1959–1970.

[87] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *CVPR*. 4320–4328.

[88] Hao Zheng, Runqi Wang, Jianzhuang Liu, and Asako Kanezaki. 2023. Cross-Level Distillation and Feature Denoising for Cross-Domain Few-Shot Classification. In *ICLR 2023*.

[89] Xiangmin Zhou and Lei Chen. 2014. Event detection over twitter social media streams. *The VLDB journal* 23, 3 (2014), 381–400.