Received 26 May 2022, accepted 1 July 2022, date of publication 11 July 2022, date of current version 19 July 2022.
Digital Object Identifier 10.1109/ACCESS.2022.3189769

RESEARCH ARTICLE

Multiview Actionable Knowledge Graph Generation From Wikipedia Biographies

YIJUN DUAN[1], XIN LIU[1], ADAM JATOWT[2], CHENYI ZHUANG[3], HAI-TAO YU[4], (Member, IEEE), STEVEN LYNDEN[1], KYOUNG-SOOK KIM[1], AND AKIYOSHI MATONO[1]
[1]National Institute of Advanced Industrial Science and Technology Tokyo Waterfront, Koto City, Tokyo 135-0064, Japan
[2]Department of Computer Science, University of Innsbruck, 6020 Innsbruck, Austria
[3]Ant Group, Hangzhou 311121, China
[4]Faculty of Library, Information and Media Science, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan
Corresponding author: Yijun Duan (yijun.duan@aist.go.jp)

This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) under Project JPNP20006.

ABSTRACT *Actionable knowledge graph* (AKG), a specialized version of knowledge graph, was proposed recently to represent, analyze, and predict human action, thus facilitating deeper understanding of human action by robots. However, the automatic construction of AKGs from action-related corpora is still an unexplored problem. In this study, we first propose three unsupervised matrix factorization–based frameworks for AKG generation from three different perspectives: *subject*, *context* and *functionality* of action, respectively. Further, we propose a hybrid model based on neural network matrix factorization (NNMF) that considers multi-source signals simultaneously. It not only learns the latent action representations, but also learns the optimal learning objective rather than assuming it to be fixed. To quantitatively verify the utility of the constructed AKGs, we introduce a novel application, that is, predicting the most likely missing action records in Wikipedia biographies. Experimental results on a large-scale Wikipedia biography dataset show that the proposed model brings significant improvement over the baselines, which demonstrates the strong expressiveness of our generated AKGs.

INDEX TERMS Actionable knowledge graph, neural network matrix factorization, text mining, web mining.

I. INTRODUCTION

The study of human action has long attracted the interest of scientists from different fields. For example, social scientists have been interested in the structure of life events and the role of individual agency and larger social forces in shaping individual life experiences. Related topics of interest include the interplay between life events [1], the characteristics of life events that are highly transformative and iconic [2], and the systematic differences in life structure across groups [3]. Meanwhile, researchers in the information retrieval domain have found that many users use search engines not only to obtain information, but also to perform certain actions and achieve certain life goals [4]. If a search engine displays actionable information that corresponds to the potential search intent underlying the user's query, then the user directly follows the search results to achieve their set goals more effectively.

Knowledge graph (KG), as a form of structured human knowledge, is an important area of research in academia and industry. Many real-world applications, such as recommendation systems, question answering, and information retrieval, benefit from the ability to understand, represent, and reason about knowledge brought about by carefully constructed KGs. In particular, in response to the growing need to understand *actionable* knowledge, the 13th NTCIR workshop [5] introduced the novel concept of *actionable knowledge graph* (AKG), which is considered to be a specialized version of KG and contains data on possible human actions, action-related entities, and action-entity relations. The concept of AKG is a preliminary attempt to establish a common ground specification for the

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

design and analysis of actionable information and related technologies.

However, the automatic construction of AKGs from corpora related to human action (*e.g.*, biographies in Wikipedia, book-length biographies, etc.) is still an important, challenging, and unexplored problem. Few studies have focused on learning algorithms for representing human action and interaction relations in a reasonable way. The learned AKG $G = (A, R)$ (A and R are the action and relation vectors, respectively) can be used to solve many practical problems related to human action, such as predicting errors and missing actionable data in knowledge bases and improving the retrieval effectiveness of web search engines for action-related queries.

In this study, we first propose three unsupervised matrix factorization–based frameworks for AKG generation from three different perspectives: *subject*,[1] *context*[2] and *functionality* of action, respectively. Intuitively, actions performed by similar users (*e.g.*, *groups of scientists*) tend to have similar representations. Contextual information about an action includes a description of the temporal period during which it is performed. Actions that tend to occur at different times (*e.g.*, *entering high school*, *divorce*) are more likely to have different temporal representations. Finally, we studied the functionality of the action, which is interpreted as the need that a user wants to satisfy by performing such an action. On the one hand, the functionality can be embodied in the semantics of the text that describes the action; on the other hand, an action's functionality can be reflected by other actions that are complementary to it and that together satisfy a more general need of the user (*e.g.*, *entering a store* and *buying a product*, which together satisfy the user's shopping needs). Because the task of learning representations of text semantics has been intensively studied (*e.g.*, [6], [7]), this paper focuses on the functional complementarity of actions. That is, different actions that are functionally complementary to similar actions are expected to have similar vector representations.

Further, for learning AKGs, we propose a hybrid model based on neural network matrix factorization that considers multi-source signals simultaneously. The most important advantage of this ensemble model is that it not only learns the latent representations, but also learns the optimal objective function rather than assuming it to be fixed, as in general matrix factorization–based models. In addition, the matrix factorization module injects prior context knowledge into the objective function, which results in the following advantages: (1) facilitating the learning process in the solution space of the neural network, and (2) allowing for unsupervised relation extraction in a label-free learning task (such as our task). The three matrix factorization–based models and the ensemble model are explained in detail in Sections 3 and 4, respectively.

To quantitatively verify the utility of the constructed AKGs, we introduce a novel application, that is, predicting the most likely missing action records in Wikipedia biographies. Specifically, we organized three types of prediction tasks: full prediction (*i.e.*, predicting the full $\langle user, time, action \rangle$ triple), partial prediction (given one triple element, predicting the other two), and partial prediction (given two triple elements, predicting the other one). Encouraging experimental results are obtained, demonstrating that the construction of AKG is necessary for action understanding and modeling, and our proposed neural network matrix factorization ensemble model can achieve the best prediction accuracy. We also present some inter-action relation instances of diverse structures and granularity in this paper, demonstrating that the learned encoding of actions and their relationships embody deep insights.

Fig. 1 illustrates the overview of our proposed system. In summary, we summarize the main contributions of this paper as follows:

- To the best of our knowledge, this study is the *first* of its kind to discuss the construction of AKG to better represent, mine, and retrieve human action.
- We propose a novel neural network matrix factorization framework that allows unsupervised encoding of actions and underlying inter-action relationships from multiple perspectives (*context*, *subject* and *functionality*).
- We obtain the state-of-the-art experimental results on the task of predicting missing human action on a large-scale Wikipedia biographical corpus.

II. PRELIMINARY

A. INPUT

The set of event trajectories D extracted from Wikipedia biographies. Let $U = \{u_1, u_2, \ldots, u_N\}$ denote the set of people in Wikipedia. For person u_i, the corresponding event trajectory in his/her life is defined as $E(u_i) = \{(e_1, t_1) \rightarrow (e_2, t_2) \rightarrow \ldots \rightarrow (e_n, t_n)\}$, where t_i denotes the time of event e_i. From the sequence of events, we have $t_1 < t_2 < \ldots < t_n$. In addition, t_i can represent both the *absolute* time (*e.g.*, A. D. 1968) or *relative* time (*e.g.*, 24 years old). $D = \{E(u_1), E(u_2), \ldots, E(u_N)\}$.

B. OUTPUT

Actionable knowledge graph $G = (A, R)$. A is the set of all nodes in G, where each node represents an action. R is the set of all edges in G, with each edge representing an inter-action relation. G is a directed graph, which means that edges (a_i, a_j) and (a_j, a_i) may represent different relations. All elements in A and R are represented by a dense vector.

C. PRE-PROCESSING

We first transform event (e, t) in D into action (a, t). Our D contains approximately 2.3 million events, but these events can only be categorized into a few hundred event

[1] The subjects of actions are also considered as potential users of the AKG, and thus the terms *subject* and *user* are used in the same sense throughout this paper.

[2] Because Wikipedia has a more complete chronological record of human actions, this study describes the contextual semantics of actions from the *temporal* perspective.

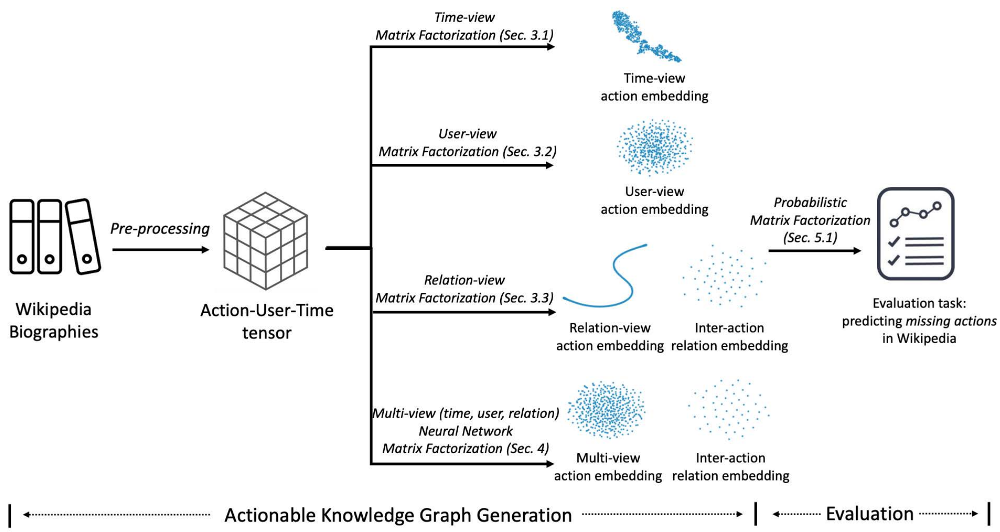


FIGURE 1. System overview.

types.[3] Obviously, studying event types instead of events is more efficient and meaningful in terms of KG construction. To convert e to a, we first use the BERT model [6] to generate e's sentence vector $\vec{e}$,[4] then cluster all $\vec{e}$ using the mini-batch K-means algorithm [10], given its high accuracy and efficiency. a is represented by the id of the category to which e belongs After conversion, the input to our models is $D = \{E(u_1), E(u_2), \ldots, E(u_N)\}$, where $E(u_i) = \{(a_1, t_1) \rightarrow (a_2, t_2) \rightarrow \ldots \rightarrow (a_n, t_n)\}$.

III. ACTIONABLE KNOWLEDGE GRAPH GENERATION

A. TIME-BASED ACTIONABLE KNOWLEDGE GRAPH GENERATION

In this section, we describe how to learn the vector representations of actions that carry temporal information. First, we construct the *action-time matrix* $G \in \mathbb{R}^{|A|\times|S|}$, where the value of entry G_{ij} in row i, and column j is the number of times that action i has been performed in time unit j. Here, $|A|$ and $|S|$ represent the number of actions and time units, respectively. j can represent either an absolute date (*e.g.*, A. D. 2008) or a relative time (*e.g.*, 25 years old). We denote the vector of action i as a_i and the vector of time unit j as s_j, and by the principle of non-negative matrix factorization [11], we have $G_{ij} \approx a_i \cdot s_j$, $a_i \geq 0$, $s_j \geq 0$.

Furthermore, it is intuitive that for any given time unit, the similarity between its vector and the vector of neighboring time units tends to be greater than the similarity between it and the vector of a time unit that is farther apart [12]. This is because people tend to have a high degree of continuity and consistency in the actions they perform in neighboring time units. For example, behaviors performed by people at ages 16 and 17 usually share a higher similarity than those performed by people at ages 16 and 56. Based on this assumption, the time vectors should satisfy $s_i \approx s_{i+1}$, where s_i and s_{i+1} denote the vectors of the adjacent time units i and $i+1$, respectively. Then, we have the following matrix factorization formula:

$$\min_{A_t, S\geq 0} \left\| G - A_t \cdot S^T \right\|^2 + \alpha \cdot \sum_{i=1}^{|S|-1} \|s_i - s_{i+1}\|^2 + \lambda \cdot (\|A_t\|^2 + \|S\|^2) \quad (1)$$

Here, $A_t \in \mathbb{R}^{|A|\times d}$ is the action vector matrix, and $S \in \mathbb{R}^{|S|\times d}$ is the time vector matrix. d is the dimension of both vector types. $|\cdot|$ represents the Frobenius norm. λ is the parameter that controls the norms of A_t and S, thus weakening the overfitting.

B. USER-BASED ACTIONABLE KNOWLEDGE GRAPH GENERATION

In this section, we describe our method for learning action representations from the user perspective. We use two data sources: *user-action matrix* and *link* information among user pages in Wikipedia. Let the user-action matrix be denoted as $H \in \mathbb{R}^{|A|\times|U|}$, where each item H_{ij} denotes the number of times action i is performed by user j. We denote the set of action vectors as $A = \{a_1, a_2, \ldots, a_{|A|}\}$, and the set of user vectors as $U = \{u_1, u_2, \ldots, u_{|U|}\}$. Similarly, there should be $H_{ij} \approx a_i \cdot u_j$, and $a_i \geq 0$, $u_j \geq 0$.

Naturally, the behavior of a certain user is often influenced by others, and different users have different influences. For example, previous studies [13], [14] have found that different

[3] As suggested by a previous study of Wikipedia biographies [8], we assume that the number of event types/actions in Wikipedia is 500, *i.e.*, $|A| = 500$.

[4] The BERT model we used is a pre-trained model trained on Wikipedia, as published by Hugging Face [9].

persons in Wikipedia have diverse historical importance.[5] In addition, people who are perceived to be of high importance tend to have Wikipedia pages connecting to other important Wikipedia articles [13]. These findings imply that users during the matrix factorization process should be paid different attention.

Furthermore, highly influential people are more likely to have more detailed and comprehensive life records in Wikipedia. Compared with the population whose many life events are "missing" in Wikipedia, the biographies of influential people in the action-user matrix H are more complete, and thus their vector representations are more reliable.

Given the link structure in Wikipedia biographies, we use the PageRank algorithm [15] to calculate the importance $I(u_j)$ for user u_j. Given the user importance, we propose the following matrix factorization objective function:

$$\min_{A_u, U \geq 0} \sum_{i=1}^{|A|} \sum_{j=1}^{|U|} I(u_j)(H_{ij} - a_i^T u_j)^2 + \lambda \cdot (\|A_u\|^2 + \|U\|^2) \tag{2}$$

Here, $A_u \in \mathbb{R}^{|A| \times d}$ and $U \in \mathbb{R}^{|U| \times d}$ are the action vector matrix and the user vector matrix, respectively, $a_i = A_u[i]$, and d is the vector dimension. The above equation allows us to learn the action vector containing user information and the user vector itself.

C. RELATION-BASED ACTIONABLE KNOWLEDGE GRAPH GENERATION

Naturally, we assume that users perform different actions to satisfy different needs. If a user performs action a_1 and then performs action a_2, it is reasonable to assume that a_2 will satisfy the requirement that a_1 cannot satisfy. Furthermore, if the functions of a_1 and a_2 are closely complementary and together satisfy the user's needs, we judge that there exists a latent relation r between a_1 and a_2 as a reflection of such complementarity. The triplet $\langle a_1, r, a_2 \rangle$ shows a higher level of understanding of action than analyzing an isolate action.

In the field of natural language processing, the relation between frequently co-occurring words is described as *selective preference* [16]. For example, the word *delicious* tends to modify the word *food*. Similarly, inter-action relations can also be understood in this way as an expression of the tendency for different actions to co-occur. Thus, we define the relation r as a tuple: $r \doteq \langle a_{head}, a_{tail} \rangle$. a_{head} and a_{tail} represent vectors of two actions that have high functional complementarity and a high tendency to co-occur.

Let the number of latent relations be m, and we define $R_h \in \mathbb{R}^{m \times d}$ as the action vector matrix in the head slot and $R_t \in \mathbb{R}^{m \times d}$ as the action vector matrix in the tail slot. Then, the relation vector matrix is $R =< R_h, R_t >$. For any action tuple (a_i, a_j), its compatibility with the lth relation is computed as $a_i \cdot R_h[l] + a_j \cdot R_t[l]$. Furthermore, given that the probability transition matrix T[6] where T_{ij} denotes the probability of action a_j being performed after action a_i is performed, and T_{ij} can also be interpreted as the sum of compatibility between (a_i, a_j) with all relations, that is, $T_{ij} = \sum_{l=1}^{m}(a_i \cdot R_h[l] + a_j \cdot R_t[l])$. When (a_i, a_j) is highly matching, T_{ij} will acquire a large value. Then, the following matrix factorization formula is proposed to learn the action and relation vector:

$$\min_{A_r, R_h, R_t \geq 0} \left\| T - A_r R_h^T Q R_t A_r^T \right\|^2 + \lambda \cdot (\|A_r\|^2 + \|R_h\|^2 + \|R_t\|^2) \tag{3}$$

Here, A_r, $[R_h, R_t]$ are the action vector matrix and relation vector matrix, respectively. Compared with the mainstream supervised relation extraction algorithms, our proposed model is unsupervised. In particular, to increase the diversity of identified relations, we introduce the constant sparsity matrix $Q = (1 - \theta) \cdot eye(m) + \frac{\theta}{m} \cdot ones(m)$ [17], [18] in the factorization objective, where $eye(m)$ and $ones(m)$ are an $m \times m$ identity matrix, and an $m \times m$ matrix with all entries of 1s, respectively. θ is the parameter controlling the sparsity of Q. The introduction of Q will lead to a smoother matrix factorization process, thus driving the resulting vectors in R_h and R_t to be diverse [17].

D. SUMMARY

We introduced the three methods above for generating AKG: $G(A, R)$. The time-based method (see Sec. III-A) represents $G(A, R)$ as $G(A = A_t, R = \varnothing)$, where A_t is action vector set carrying temporal information; the user-based method (see Sec. III-B) represents $G(A, R)$ as $G(A = A_u, R = \varnothing)$, where A_u contains user information; the relation-based method (see Sec. III-C) produces $G(A = A_r, R =< R_h, R_t >)$, where A_r is relation-inspired action vector set and $< R_h, R_t >$ denotes the identified relations.

However, the above methods have the following limitations: (1) The learned action vector contains only one type of information. A more general, multi-view method for AKG construction would produce more knowledge-rich representation; (2) the relation-based model restricts the relation to be represented by exactly two action vectors (*i.e.*, the 1-to-1 scenario) and cannot encode the relation among multiple actions (*i.e.*, the 1-to-N/N-to-1/N-to-N scenario); (3) the relation-based model assumes that the compatibility between action and relation is computed linearly, whereas the real valid computation method should be learned from the data.

Considering the above challenges, we propose a novel neural network matrix factorization ensemble to generate more sophisticated AKGs. The approach is described in the next section.

[5]For example, based on [13], [14], two examples of people with high historical importance are Napoleon and Albert Einstein.

[6]T can be easily obtained from the action trajectory set D (defined in Section 2).

IV. MULTI-VIEW-BASED ACTIONABLE KNOWLEDGE GRAPH GENERATION

We now represent a relation r as a vector of dimension k,[7] and define the relation matrix as $R_* \in \mathbb{R}^{m \times k}$, where m is the predefined number of relations. Then, given any two action vectors a_i and a_j, $I_{(a_i,a_j)} \in \mathbb{R}^{1\times m}$ is defined as the indexing vector of the relation between a_i and a_j. All elements of $I_{(a_i,a_j)}$ take values in the range [0, 1], which represents the fitness of the corresponding relation. In the extreme case, $I_{(a_i,a_j)}$ has only the l-th element close to one, whereas the remaining elements are close to zero, indicating that the relation between (a_i, a_j) can be approximated as the relation $R_*[l]$. Furthermore, the relation vector of (a_i, a_j) is computed as $r_{(a_i,a_j)} = I_{(a_i,a_j)} \cdot R_*$. Similarly, the relation vector of n actions $(a_1, a_2, \ldots, a_n)$ can be obtained by $r_{(a_1,\ldots,a_n)} = I_{(a_1,\ldots,a_n)} \cdot R_*$, and thus we can accommodate 1-to-N, N-to-1 and N-to-N scenarios.

For any action tuple (a_i, a_j), we assume that $I_{(a_i,a_j)}$ is influenced by all the time, user, and semantic information related to a_i and a_j. We propose the use of a feed-forward neural network to learn how these factors affect (see the left feed-forward NN in Fig. 2). Given the action-time matrix G and action-user matrix H, we have

$$I(a_i, a_j) = sigmoid(W_2 \cdot Relu(W_1 \cdot \vec{t_{ij}} + b_1) + b_2). \quad (4)$$
$$\vec{t_{ij}} = [G[i]; G[j]; H[i]; H[j]; i; j] \quad (5)$$

Here, $G[i]$, $H[i]$, and i denote the time, user, and semantic information of a_i, respectively. W_1 and b_1 are the parameters of the first layer, and W_2 and b_2 are the parameters of the second layer. n is the number of neurons in the first layer. $Sigmoid()$ maps the output of the neural network to the range of element values in the relation indexing vector $I(a_i, a_j)$: [0,1].

Instead of using linear functions to fit the compatibility between the action tuple and relation, we design another feed-forward neural network for such compatibility estimation (see the right feed-forward NN in Fig. 2). The parameters of this network can be learned in an entire data-driven manner from the probability transition matrix T. Our objective function is as follows:

$$Obj_{NN} \doteq \min_{A_*,R_*\geq 0} \sum_{(a_i,a_j)} (T_{ij} - f_{ij})^2, \quad (6)$$
$$f_{ij} = sigmoid(W_4 \cdot Relu(W_3 \cdot \vec{s_{ij}} + b_3) + b_4). \quad (7)$$
$$\vec{s_{ij}} = [a_i; a_j; I(a_i, a_j)^T \cdot R_* \cdot Q] \quad (8)$$

Here, A_* is the action vector matrix and $a_i = A[i]$. W_3 and b_3 are the parameters of the third layer. W_4 and b_4 are the parameters of the fourth layer. n is the number of neurons in the third layer. Q is a $k \times k$ constant sparsity matrix, as defined in Sec. III-C, which is used to increase the sparsity of $I(a_i, a_j)$ and the diversity of relation vector R_*.

Further, we make use of prior contextual information (G, H) to enable the neural network to learn different properties of actions. The final objective function is as follows:

$$\boxed{\begin{aligned}\min_{A_*,R_*,S,U\geq 0} \; & Obj_{NN} + \alpha \cdot (\|G - A_* \cdot S\|^2 + \|H - A_* \cdot U\|^2) \\ & + \beta \cdot (\|A_*\|^2 + \left\|R^*\right\|^2 + \|S\|^2 + \|U\|^2)\end{aligned}} \quad (9)$$

In our neural network matrix factorization ensemble, the neural network module learns the optimal objective function, whereas the matrix factorization module limits the solution space of the neural network by utilizing the prior context. Finally, the generated knowledge graph $G(A, R)$ is denoted as $G(A = A_*, R = R_*)$. We illustrate the architecture of the above ensemble model in Fig. 2.

[7] The relation vector dimension k does not need to be equal to the action vector dimension d.

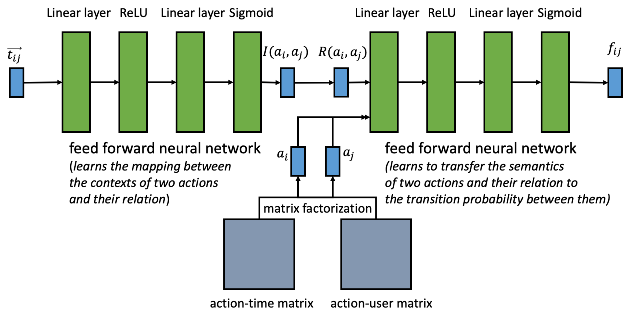


FIGURE 2. An illustration of our neural network matrix factorization model introduced in Sec. IV. $\vec{t_{ij}}$ denotes the context information (time, user, semantic) of action i and action j, whose learned vectors are represented as a_i and a_j. $I(a_i, a_j)$ and $R(a_i, a_j)$ are the relation distribution vector and the relation vector, respectively. f_{ij} is the transition probability between action i and action j.

V. EXPERIMENTS

A. EVALUATION TASK

In this section, we describe the evaluation of the quality of the generated AKGs. Specifically, we designed a novel task T: AKG-based prediction of user actions. This task can be used to quantify the effectiveness of AKGs, and can be seen as a practical application of this study.

From the viewpoint of our implementation, the task T can be described as follows: Given a dataset of users' action records X (X is the set of triples $\{u, t, a\}$, where $\{u_i, t_j, a_k\}$ indicates that user u_i performs action a_k at moment t_j), and the AKG $G = (A, R)$ is regarded as a knowledge base generated from dataset D; based on partial observable data X^o in X, the task is to predict the remaining unobservable data $X^{\neg o}$ in X. Note that D is also a collection of triples $\{u, t, a\}$, and yet the users in D may be completely different from that in X, whereas the action types in D and X are close. The G learned from D contains the latent semantics of action, which will serve as supplementary information to assist in the predictions of $X^{\neg o}$ based on X^o. The effectiveness and generalizability of G are even more convincingly demonstrated if G enables more accurate predictions of X that contain different users from D.

Algorithm 1 Generative Process for Predicting (u_i, t_j, a_k) in X

```
for each user \vec{u}_i ∈ U_X do
    draw latent user vector \vec{u}_i ~ N(0, σ_U^2 I)
end for
for each time unit \vec{t}_j ∈ T_X do
    draw latent time vector \vec{t}_j ~ N(0, σ_T^2 I)
end for
for each action \vec{a}_k ∈ A_X do
    draw latent offset vector \vec{o}_k ~ N(0, σ_A^2 I)
    draw latent action vector \vec{a}_k = \vec{o}_k + \vec{a}_k^Δ    ▷ \vec{a}_k^Δ is auxiliary action information from learned knowledge base G
end for
for each user-time-action pair (\vec{u}_i, \vec{t}_j, \vec{a}_k) ∈ X do
    draw the rating X_{i,j,k} ~ N(\vec{u}_i^T \vec{t}_j + \vec{u}_i^T \vec{a}_k + \vec{a}_k^T \vec{t}_j, σ_X^2)
end for
```

We model this prediction task using a generative process by placing Gaussian priors on latent feature vectors. The generation process of triple $\{u_i, t_j, a_k\}$ in X is simulated as follows: During the generation process, G provides *auxiliary action information*.

In Algo. 1, U_X, T_X, A_X are the set of users, times, and actions contained in X, respectively. $N(u, \sigma^2)$ denotes the Gaussian priors. $\vec{a}_k^{\Delta}$ is the auxiliary action information derived from G. Based on the proposed models, $\vec{a}_k^{\Delta}$ can be A_t (model III-A), A_u (model III-B), A_r (model III-C) or A_* (model IV).

B. PARAMETER LEARNING

For Eq. (1), Eq. (2), Eq. (3) and Eq. (9), *stochastic gradient descent* was used to approach the optimal parameters, and *multiplicative update rules* [19] were adopted to ensure that all parameters are non-negative. To minimize Eq. (9), we alternate between updating neural network parameters while fixing the AKG vectors, and updating the AKG vectors when fixing the network parameters. To learn the parameters in task T, we first convert the generative process of T into an equivalent probabilistic matrix factorization formula [20], and then solve it using the same strategy above.

C. PREDICTION

After the optimal parameters $u_i \in U_X$, $t_j \in T_X$, and $a_k \in A_X$ in Algo. 1 are learned, we can conduct different types of predictions of unobservable data $X^{\neg o}$ in X. Specifically, we experimented with the following prediction tasks:

- *Type I: Full prediction (predicting the full triple).* In this task, we attempt to predict the most likely existing triples (u_i, t_j, a_k) in $X^{\neg o}$. The possibility score P_{ijk} for triple (u_i, t_j, a_k) is computed as follows:

$$P_{ijk} = \vec{u}_i^{\,T} \vec{t}_j + \vec{u}_i^{\,T} \vec{a}_k + \vec{a}_k^{\,T} \vec{t}_j \tag{10}$$

  For each possible combination (u_i, t_j, a_k) in $X^{\neg o}$, we compute its score and rank all triples based on such a score from high to low. The existence of top-ranked triples will be validated on ground truth data, and then the prediction effectiveness can be evaluated based on the metric (*Hits@k*), as introduced later.
- *Type II: Partial prediction (given one, predicting two).* In contrast to the previous task, in this task, we assume that for each triple (u_i, t_j, a_k) in $X^{\neg o}$, one element is known, based on which our goal is to predict the remaining two most reasonable elements. Then, we have three different cases: given u_i, predicting its associated (t_j, a_k); given t_j, predicting (u_i, a_k); and given a_k, predicting (u_i, t_j). The respective conditional probability $P_{jk|i}$ can be computed as follows, where $P_{ik|j}$ and $P_{ij|k}$ can be calculated similarly.

$$P_{jk|i} = \frac{P_{ijk}}{P_i} = \frac{\vec{u}_i^{\,T} \vec{t}_j + \vec{u}_i^{\,T} \vec{a}_k + \vec{a}_k^{\,T} \vec{t}_j}{\sum_j \sum_k (\vec{u}_i^{\,T} \vec{t}_j + \vec{u}_i^{\,T} \vec{a}_k + \vec{a}_k^{\,T} \vec{t}_j)}. \tag{11}$$

- *Type III :Partial prediction (given two, predicting one).* Similarly, there are three different subtasks: given (t_j, a_k), predicting u_i (subtask II-1); given (u_i, a_k), predicting t_j (subtask II-2); and given (u_i, t_j), predicting a_k (subtask II-3). The respective possibility $P_{i|jk}$ is computed as follows, where $P_{j|ik}$ and $P_{k|ij}$ are computed in a similar way.

$$P_{i|jk} = \frac{P_{ijk}}{P_{jk}} = \frac{\vec{u}_i^{\,T} \vec{t}_j + \vec{u}_i^{\,T} \vec{a}_k + \vec{a}_k^{\,T} \vec{t}_j}{\sum_i (\vec{u}_i^{\,T} \vec{t}_j + \vec{u}_i^{\,T} \vec{a}_k + \vec{a}_k^{\,T} \vec{t}_j)}. \tag{12}$$

  During prediction, the above computation can be further simplified. For example, when computing $P_{i|jk}$, we can calculate $\vec{u}_i^{\,T} \vec{t}_j + \vec{u}_i^{\,T} \vec{a}_k$ because the item $\vec{a}_k^{\,T} \vec{t}_j$ will be the same for each tested u_i given the same (t_j, a_k).

D. DATASET

In this study, we used an open large-scale Wikipedia biography dataset released by [8].[8] This dataset was published on January 2, 2014 and is a dump of English-language Wikipedia entries containing 242,970 biographies of people born after 1800. Each biography contained at least five events, for a total of 2,313,867 events. In addition, the data structure for each event is $\{person\_id, event\_id, year, age, terms, original\_sentence\}$, where *terms* consists of meaningful words extracted from *original_sentence* and is used to compute the event vector during pre-processing (see Sec. II). Tab. 1 shows an example of extracted event sentences from the biography of *Audrey Hepburn*.

E. DATA PREPARATION

We used 80% of the Wikipedia biography dataset [8] as the dataset D to learn the AKG G and the remaining 20% as the dataset X for the action prediction task. For X, we test

[8] The dataset can be found at http://www.cs.cmu.edu/ ark/bio/

TABLE 1. A sample of 5 of the 62 event sentences (underlined words are data after pre-processing as input to our model) from the Wikipedia biography for *Audrey Hepburn* (born 1929).

Original action sentence	Date	Age
(1) After the Germans invaded the Netherlands in 1940, Hepburn adopted the pseudonym Edda van Heemstra, because an "English sounding" name was considered dangerous during the German occupation.	1940	11
(2) By 1944, Hepburn had become a proficient ballet dancer.	1944	15
(3) For her role in Roman Holiday, Hepburn was also the first actress to win an Oscar, a Golden Globe and a BAFTA Award for a single performance in 1954.	1954	25
(4) In August 1988 , Hepburn went to Turkey on the immunization campaign.	1988	59
(5) She was awarded the Presidential Medal of Freedom in recognition of her work as a UNICEF Goodwill Ambassador in late 1992.	1992	63

for sparse and dense training settings separately. For the former, 50% of X is used for training (*i.e.*, X^o) and the remaining 50% is used for testing (*i.e.*, $X^{\neg o}$). For the latter, 80% of X was used for training and 20% was used for testing. In addition, we test the prediction performance of the analyzed models in absolute time (*e.g.*, A. D. 2000) and relative time (*e.g.*, 30 years old) separately, because they are two fairly different temporal measures. Thus, we have four experimental settings in total: (sparse, absolute time); (dense, absolute time); (sparse, relative time); and (dense, relative time). For each setting, we used 5-fold cross validation to obtain the average model performance.

F. ANALYZED METHODS

We test the performance of the following methods.

First, the model we propose consists of:

- **T-AKGG** (Time-based AKG generation) (see Sec. III-A);
- **U-AKGG** (User-based AKG generation) (see Sec. III-B);
- **R-AKGG** (Relation-based AKG generation) (see Sec. III-C);
- **F-AKGG** (Full AKG generation) (see Sec. IV).

In addition, we test the performance of the following baselines:

- **N-AKGG** (Non-actionable AKG generation). We do not use any information from the generated AKG in the task of user action prediction. We tested this method to verify the validity of the AKG generation.
- **KGPMF** (KG-based probabilistic matrix factorization [18]) The current *state-of-the-art* method for multi-view KG generation, based on a probabilistic matrix factorization model. In this method, we retain all types of information (time, user, and relation).

G. EVALUATION METRICS

In this study, we use $Hits@k (k = 10, 20, \ldots, 100)$ as our main evaluation criterion, where the meaning of $Hits@k$ is equivalent to $TPs@k$ (true positives@k, the number of correctly predicted triples of top-k ranked items). In our case, if a record of user u conducting action a at time t does not appear in the test set, there exists the possibility that *this record is not included in Wikipedia*, rather than meaning that u has not conducted a in real history. Thus, FPs (false positives) are not reliable, which makes it difficult to calculate $Precision@k$ and $F - score@k$. Because the value of $Hits@k + FNs@k$ (FNs: False negatives) for each k is the same for all analyzed methods, so computing $Hits@k$ is equivalent to computing $Recall@k = Hits@k/(Hits@k + FNs@k)$.

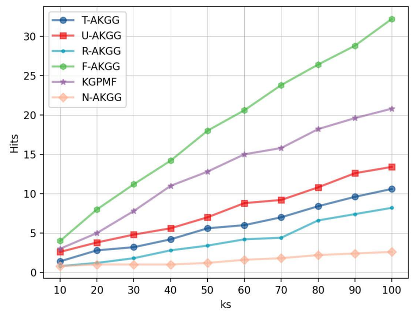
H. EXPERIMENTAL SETTINGS

First, as suggested by [8], the number of actions in Wikipedia was set to 500. Based on a grid search (range: 0-100, step size: 10), the number of relations was 50, and the dimension of the relation vector was 20. The dimension of the action/time/user vector was 50 for all methods. Inspired by [18], [21], in methods **U-AKGG**, **F-AKGG**, and **KGPMF**, we first cluster similar users into user groups (group number: 100), and use group-level user importance instead of individual importance. $learning\_rate = 0.01$ and $\lambda = 0.001$. For the ensemble model **F-AKGG**, our 5-fold cross-validation suggests $learning\_rate = \lambda = 0.005$, $number\_of\_epochs = 100$, and the hidden layer dimension is 20. For **T-AKGG**, $\alpha = 0.01$. For **R-AKGG** and **F-AKGG**, $\theta = 0.8$ (as suggested in [17]).

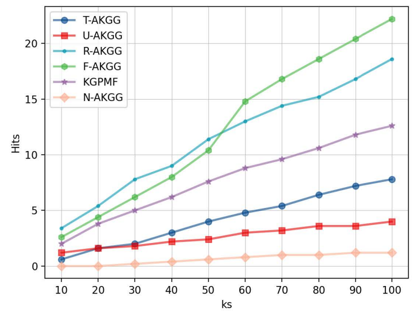
I. EXPERIMENTAL RESULTS: FULL PREDICTION (TYPE I)

Fig. 3 illustrates the performance of all methods in the full prediction task. Taken together, our NN matrix factorization ensemble model **F-AKGG** achieves the best prediction accuracy. Based on the experimental results, we perform the following detailed analysis:

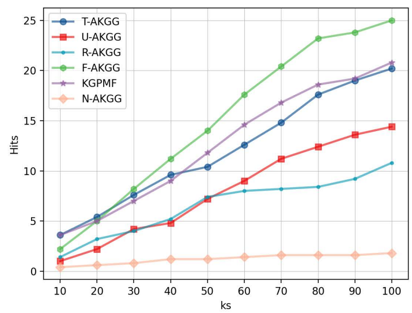
- Action prediction is a difficult and challenging task. The performance curve of **N-AKGG** shows that the average $Hits@100$ value of **N-AKGG** is only 2.2 and 0.9 in sparse and dense cases, respectively. This observation also hints at the sparseness of biographical data in Wikipedia.
- The accuracy of action prediction can be greatly improved when using AKG as a supplementary knowledge base. This finding demonstrates that the concept of AKG, as well as the study of AKG generation, is useful in helping machines understand and encode



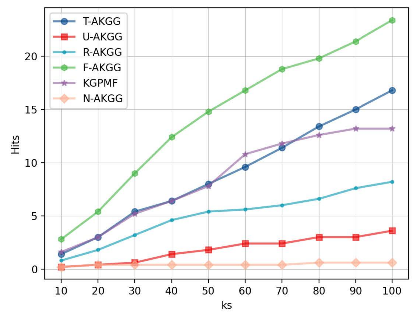
(a) absolute time, sparse setting



(b) absolute time, dense setting



(c) relative time, sparse setting



(d) relative time, dense setting

FIGURE 3. **Performance of all analyzed models for full action prediction (Type 1) in terms of *Hits@k*.**

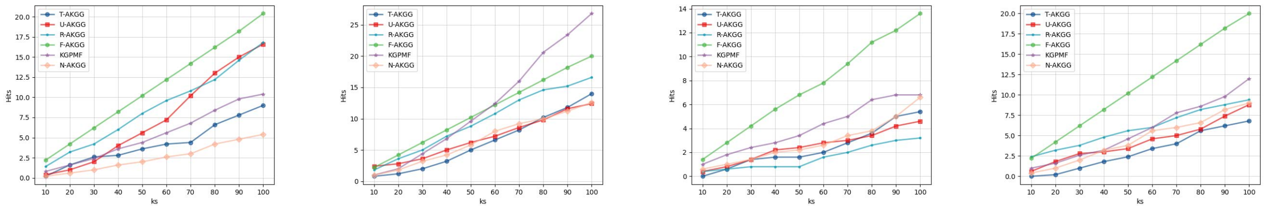
human action. In particular, **T-AKGG**, **U-AKGG**, and **R-AKGG** are on average 5.5 times, 4.7 times, 2.8 times (dense), and 11.3 times, 3.2 times, and 14.3 times (sparse) more effective than **N-AKGG**, respectively. Thus, we conclude that *time*, *user*, and *inter-action relation* are all valid signals in the AKG process.

- With more training data, **R-AKGG** performs significantly better. The average *Hits@k* value of **R-AKGG** in the dense setting increased by 54.7% compared with the sparse setting. When set to (dense, relative time) and $k \leq 50$, **R-AKGG** performs best among all methods, implying that capturing the latent relations between actions requires large amounts of training data.
- In most cases, *time* signal is more efficient and robust than *user* signal. Overall, the average *Hits@k* value of **T-AKGG** was 53.3% higher than that of **U-AKGG**. A very likely reason for this is that the temporal distribution of action in the training and test sets is much closer, whereas the difference between the user group distribution is more significant.
- The two methods that fuse all types of signals **F-AKGG** and **KGPMF** perform the best. Moreover, our proposed **F-AKGG** exhibits the best performance in almost every setting. We then conclude that it is effective to use neural networks to learn the nonlinear compatibility of actions and relations. At the same time, the training time of the neural network does not increase significantly when using matrix factorization to limit the solution space. Experiments demonstrate that our **F-AKGG** can achieve state-of-the-art performance in a full action prediction task.

J. EXPERIMENTAL RESULTS: PARTIAL PREDICTION (TYPE II)

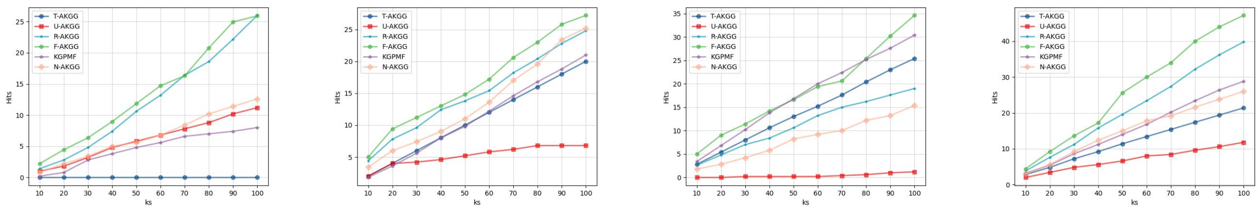
Fig. 4 to Fig. 6 show the performance of all methods in the partial prediction task (type II). Among them, Fig. 4, Fig. 5, and Fig. 6 correspond to the results of subtask 1 (given u_i, predicting (t_j, a_k)), subtask 2 (given t_j, predicting (u_i, a_k)), and subtask 3 (given a_k, predicting (u_i, t_j)), respectively. We perform the following analysis of the experimental data.

- When comparing the performance of each method in subtask 1 (see Fig. 4), we find that, first, in the two settings (c) and (d), **F-AKGG** clearly shows a more significantly superior performance (2.79 and 2.36 times, respectively, as the average performance of the other methods). In setting (a), the performance of **F-AKGG** is better but relatively less significant (1.29 as effective as the second-best method). In setting (b), **F-AKGG** is the second-best performing method after **KGPMF** when k is large ($60 \leq k \leq 100$). In general, as k increases, the number of missing actions correctly predicted by **F-AKGG** shows an obvious linear trend. For every 10 increases in k, the number of hits achieved by **F-AKGG** approximately increases by 2.02 (a), 1.98 (b), 1.36 (c), and 1.98 (d) hits on average.
- In subtask 2, we find that although **F-AKGG** can still be considered as the best method (shows the



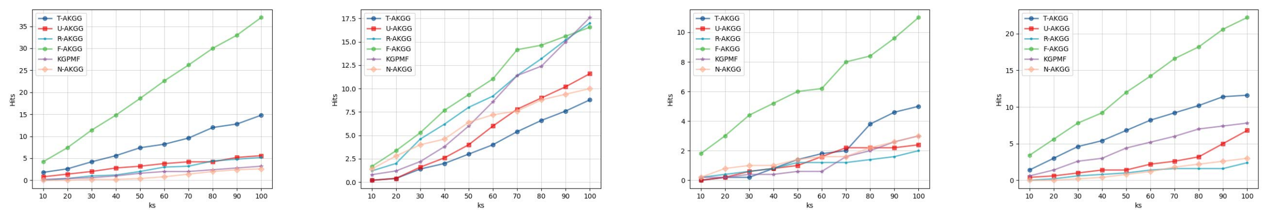
(a) absolute time, sparse setting (b) absolute time, dense setting (c) relative time, sparse setting (d) relative time, dense setting

FIGURE 4. **Performance of all analyzed models for subtask 1 (given u_i, predicting (t_j, a_k)) in partial action prediction (type II) in terms of *Hits@k*.**



(a) absolute time, sparse setting (b) absolute time, dense setting (c) relative time, sparse setting (d) relative time, dense setting

FIGURE 5. **Performance of all analyzed models for subtask 2 (given t_j, predicting (u_i, a_k)) in partial action prediction (type II) in terms of *Hits@k*.**



(a) absolute time, sparse setting (b) absolute time, dense setting (c) relative time, sparse setting (d) relative time, dense setting

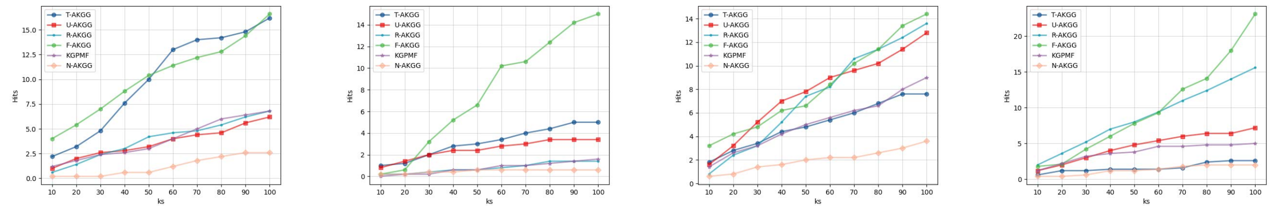
FIGURE 6. **Performance of all analyzed models for subtask 3 (given a_k, predicting (u_i, t_j)) in partial action prediction (type II) in terms of *Hits@k*.**

strongest predictive ability in three scenarios (a), (b), and (d) and is one of the best two methods in scenario (c)), the performance difference between it and the second-best performing method (on average of all settings, 13.3% above) is not as significant as in subtask 1. We observe that the purely relation-based method **R-AKGG** can be regarded as the overall second-best method. In contrast, the user-based method **U-AKGG** performs poorly. Thus, it can be concluded that the introduction of user information for some prediction tasks, including subtask 2, can even decrease prediction performance. Furthermore, the signal used for AKGG is task-dependent and should be selected carefully.

- We find that **F-AKGG** is well suited for subtask 3. In most settings ((a), (c), (d)) **F-AKGG** exhibits an obvious performance boost (1.36, 4.50, and 1.61, respectively, as the average performance of the other methods). In setting (b), it still takes the lead when $10 \leq k \leq 90$. Intuitively, given an action type, the task of predicting user groups who are likely to perform the action type and the corresponding performing time requires the prediction model to understand and model the action from both time and user perspectives simultaneously, which naturally coincides with our idea when designing **F-AKGG**. Therefore, it is clear why **F-AKGG** shows such a good performance in this task. It is also interesting to note that the temporal signal is more effective in this task compared with subtasks 1 and 2, which again demonstrates our previous observation of the task-dependent signal importance.
- In summary, we conclude that **F-AKGG** has the best overall performance in the type II prediction task. Among the 12 subtasks, **F-AKGG** is the undisputed best method in 10 of them, and achieves a significant performance lead in five of them. We once again validate the effectiveness of introducing AKG for the action prediction task and the plausibility of the action representation learned by **F-AKGG**.

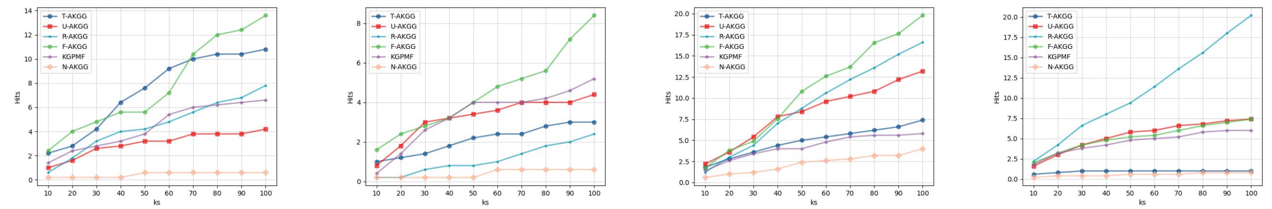
K. EXPERIMENTAL RESULTS: PARTIAL PREDICTION (TYPE III)

We illustrate the performance of all methods in the partial prediction task (Type III) in Fig. 7, Fig. 8, and Fig. 9, where they correspond to the results of subtask 1 (given (t_j, a_k), predicting u_i), subtask 2 (given (u_i, a_k), predicting t_j), and



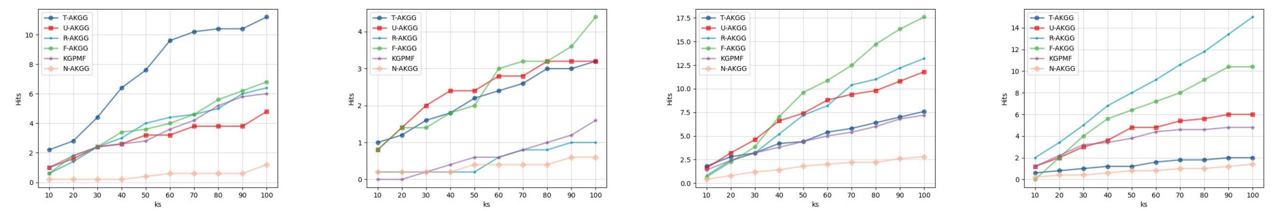
(a) absolute time, sparse setting (b) absolute time, dense setting (c) relative time, sparse setting (d) relative time, dense setting

FIGURE 7. Performance of all analyzed models for subtask 1 (given (t_j, a_k), predicting u_i) in partial action prediction (type III) in terms of *Hits@k*.



(a) absolute time, sparse setting (b) absolute time, dense setting (c) relative time, sparse setting (d) relative time, dense setting

FIGURE 8. Performance of all analyzed models for subtask 2 (given (u_i, a_k), predicting t_j) in partial action prediction (type III) in terms of *Hits@k*.



(a) absolute time, sparse setting (b) absolute time, dense setting (c) relative time, sparse setting (d) relative time, dense setting

FIGURE 9. Performance of all analyzed models for subtask 3 (given (u_i, t_j), predicting a_k) in partial action prediction (type III) in terms of *Hits@k*.

subtask 3 (given (u_i, t_j), predicting a_k), respectively. The following is our analysis of the experimental results.

- In subtask 1, **F-AKGG** can be considered as the most competitive method in general, but has no obvious advantage in three of the four settings. In setting (a), **T-AKGG** performs equally well; in setting (c), **U-AKGG** and **R-AKGG** perform similarly to **F-AKGG**; in setting (d), **F-AKGG** performs significantly better than **R-AKGG** only when $70 \leq k \leq 100$. Only in setting (b), **F-AKGG** clearly outperforms the others, where its average performance is 2.46 times as that of the second-best method: **T-AKGG**.
- In subtask 2, it can be observed that **F-AKGG** does not achieve a significant performance lead in any setting. In particular, in setting (d), **F-AKGG** is much inferior to **R-AKGG** (it only achieves 47.4% of the average predictive ability of **R-AKGG**), and close to **U-AKGG** and **KGPMF**. Nevertheless, **F-AKGG** still exhibited strong stability under diverse experimental conditions. In settings (a), (b), and (c), it remains one of the first echelons of all methods. When k takes values within specific ranges (*e.g.*, $k \geq 60$ in (b), and $k \geq 50$ in (c)), the best performance can be obtained. We note that in the two settings where relative time is adopted, **R-AKGG** performs well.
- **F-AKGG** performs relatively poorly in subtask 3. In setting (a), the average performance is only 51.9% of the best method **T-AKGG**. In setting (d), the average performance is 74.2% of the best method **R-AKGG**. In (b), the predictive ability of all methods is bifurcated, where **F-AKGG**, **T-AKGG**, and **U-AKGG** are significantly better than the others. (c) is the only scenario where **F-AKGG** shows the performance lead.
- In summary, in the series of experiments of type III, we find that **F-AKGG** is not as dominant as it is in type I and type II experiments. However, its performance is more robust compared with other methods in various settings, and therefore, it can still be judged as the best model from a global perspective. Another interesting finding is that **T-AKGG** performs relatively well when absolute time is used, and **R-AKGG** achieves good performance when relative time is used. Therefore, we conclude that the dynamic patterns of actions are

TABLE 2. Example actions and their corresponding descriptive terms.

Action	Most probable terms in action class	Action	Most probable terms in action class
A1	married, wife, personal, college, family, personal life, girlfriend, born	A12	contract, signed, announced, released, season, free-agent, left, years
A2	married, daughter, met, three, son, john, children, actress	A13	band, joined, began, formed, guitar, drummer, group, music
A3	birth, gave, born, family, son, ceremony, family, church	A14	released, number, announced, album, release, single, hit, new album
A4	married, divorce, marriage, years, divorced, ended, later, separated	A15	released, song, studio album, single, hit, recorded, love, top
A5	abuse, crime, murder, police, sexual, allegations, accused, sex	A16	award, album, best, year, band, nominated, records, received
A6	arrested, police, arrest, killed, shot, car, death, -year-old	A17	school, college, high school, education, attended, entered, class, years
A7	court, sentenced, guilty, years, prison, convicted, sentence, federal	A18	graduated, college, bachelor, university, degree, educated, Oxford, earned
A8	championships, championship, national, junior, u.s., u-, under-, champion	A19	university, law, school, studied, study, law school, degree, law degree
A9	draft, nhl draft, round, drafted, selected, professional career, nhl, pick	A20	degree, thesis, university, received, phd, dissertation, ph.d., doctorate
A10	season, show, scored, promotion, scoring, league, division, captain	A21	college, professor, university, became, appointed, lecturer, faculty, physics
A11	signed, million, season, minor league contract, contract, new, released, re-signed		

more meaningful from the absolute time perspective, whereas the interaction relations make more sense from the relative time perspective. The only method that does not rely on AKG, **N-AKGG**, is the worst performer, implying the necessity of AKG in type III tasks.

L. EXPERIMENTAL RESULTS: SUMMARY

It can be summarized from the above analysis of the experimental results for all prediction tasks (type I, type II, and type III) that the construction of AKG is necessary for action understanding and modeling (because **N-AKGG** is nearly the worst model in all tasks). Furthermore, the ensemble model **F-AKGG** achieves the globally best predictive ability, demonstrating that our idea of multi-view encoding of actions and their relations indeed leads to the AKG of better quality. Moreover, **F-AKGG** exhibits different levels of fitness for different types of tasks (type I > type II > type III). In total, among all 28 settings (seven subtasks and four settings for each of them), **F-AKGG** is clearly the best performer for 15 of them, in the first echelon for 10 of them, and relatively weak for the remaining three.

M. CASE STUDY: INTER-ACTION RELATION EXAMPLES

In this section, we present some inter-action relation instances of diverse structures learned by **F-AKGG** in Tab. 2 and Tab. 3, which exhibits the best performance in the quantitative evaluation.

- By looking at the words in Tab. 2, we infer that relation R1 reflects the transition relationship: *get married→divorce*. Interestingly, there is more than one action expressing the same semantic meaning: *get married*, probably because they occur at different stages of life (*e.g.*, 20s, 30s, 40s), and **F-AKGG** can understand each of them to have a similar relationship to *divorce*, thus learning a "N-to-1" pattern instead of many "1-to-1"s.
- R2 can be understood as a *crime→sentence* relationship.
- R3 reflects another "N-to-N" relationship from the early stages of an athlete's career (*championships in youth*, *draft*) to some later stages (*contract signing*, *contract renewal*, etc.). Similar to R1, although *contract signing* may occur repeatedly throughout an athlete's career, they are viewed by our approach as playing similar roles within a broader relation.

TABLE 3. N-to-N inter-action relation instances. A1∼A21 represents Action 1∼Action 21, whereas R1∼R6 represents Relation 1∼Relation 6.

Relation	Head slot	Tail slot
R1	A1, A2, A3	A4
R2	A5, A6	A7
R3	A8, A9	A10, A11, A12
R4	A13	A14, A15, A16
R4	A17, A18	A19, A20
R5	A20	A21

- R4 represents part of the career trajectory of a popular band (from *band formation* to *album releases* and *award receiving*).
- R5 implies the life transition of people from *high school education* and *college education* to *graduate school education*. After that, some of them get a faculty position in academia, as reflected by R6. We can naturally combine R5 and R6 to portray typical stages in the life of a scientist.

The above case studies show that our approach performs well in terms of learning semantic relations among actions of diverse structures and granularity, which can bring novel insights to human-robot interaction tasks.

N. CASE STUDY: GENERATED PARTIAL BIOGRAPHY FOR YURIKO NAKAMURA

In this section we present predictions made by our generated AKG about the life of *Yuriko Nakamura*, who at the time of writing does not have a biographical entry in the English Wikipedia. This is a double partial triple prediction task (see Sec. 5.3): given subject u, predict action a and occurrence time t. Since *Yuriko Nakamura* was not included in our dataset, we set her vector representation as the average of the vectors of three similar Japanese female musicians[9] who we manually select from our dataset. Tab. 4 shows the top-5 predicted actions in the life of *Yuriko Nakamura* in chronological order predicted by **F-AKGG**. Note that to increase prediction accuracy we use a relatively wide temporal granularity: 10 years. The predicted actions were

[9]These musicians are *Akiko Yano* (https://en.wikipedia.org/wiki/Akiko_Yano), *Ringo Sheena* (https://en.wikipedia.org/wiki/Ringo_Sheena) and *Toshiko Akiyoshi* (https://en.wikipedia.org/wiki/Toshiko_Akiyoshi).

TABLE 4. Top-5 predicted actions in the life of *Yuriko Nakamura* made by our generated AKG. Due to space limitation for each action we only show up to 8 descriptive terms.

Action + Date	Most probable terms in action
music study (20s)	study, studied, conservatory, composition, school, piano, graduated
music performing (30s)	concert, held, live, music, song, performed, tour, show
album recording (30s)	jazz, recorded, recording, album, band, blues, played, records
album releasing (30s)	released, number, announced, album, release, single, hit, new album.
media interview (40s)	interview, music, said, work, wrote, critic, film, magazine

validated using her official homepage[10] and confirmed to have actually occurred based on her personal profile.

VI. RELATED WORKS

To the best of our knowledge, the concept of AKG was first introduced at the NTCIR-13 workshop [5] in 2017. Its organizers designed two AKG-related tasks: action mining (AM) and query-based AKG generation (Q-AKGG). The former task was designed to search for related actions (*e.g.*, visit a temple) associated with the query entity (*e.g.*, Tokyo), whereas the latter task was designed to generate relevant actions (*e.g.*, live in a flood area) and entity types (*e.g.*, thing, event) for the given query (*e.g.*, consequences of flood). For the first task, Rahman *et al.* [22] proposed a probabilistic model based on Bayes' theorem, and Kang *et al.* [23] proposed a solution based on grammatical tree rules. The best performing model was proposed by the designers of the task [5], which is another Bayesian network model. For the second task, Lin *et al.* [24] designed a solution using a language model based on Dirichlet smoothing. Different from the existing research which mainly focuses on action search, we tackle the task of generation and embedding of AKG.

Technically, this study is close to the knowledge graph embedding (KGE) domain. Existing KGE models mainly include the following categories: (1) translational models (*e.g.*, [25]), (2) sensor decomposition models (*e.g.*, [26]), (3) neural models (*e.g.*, [27], [28]), (4) language models (*e.g.*, [29]), and (5) entailment-aware models (*e.g.*, [30]). Different from existing KGE models, our model fuses multi-source heterogeneous data in an unsupervised manner. Due to space limitation, we cannot provide a complete review of all important works of KGE. More detailed and comprehensive reviews of KGE can be found in [31]–[33].

This work also lies in the field of biography mining. Related topics of interest include the interplay between life events [1], the characteristics of life events that are highly transformative and iconic [2], and the systematic differences in life structure across groups [3], etc. In particular, Wikipedia has been used extensively, for the task of disambiguition of named entities [34], [35], the recognition of biographical sentences [36], the identification of latent biographical structure [8], and the summarization of typical life trajectories and events [37], [38], etc.

[10]https://yurikopia.com/disco/

VII. CONCLUSION

From the time the concept of an *AKG* was introduced in NTCIR-13 [5], the use of techniques from the KG domain to enhance the robot's usage of human biographical data and understanding of human action has gradually attracted the interest of researchers in multi-disciplinary fields. In this study, we, for the first time, propose an unsupervised model for the automatic multi-view generation of AKG. A new neural network matrix factorization framework is proposed to learn meaningful action representations and extract latent inter-action relations from multiple perspectives: *subject*, *context*, and *functionality*. The proposed model brings significant improvement over the baselines in a novel application for predicting missing human action records in Wikipedia.

Our research can not only assist social scientists and historians in the analysis of human action, but also inject new technical innovations into the field of KG generation and relation extraction. In the future, we will focus on exploring ways to improve the expressiveness of AKG in human-robot interaction scenarios, with the goal of building a smart society.

VIII. STATEMENTS AND DECLARATIONS

- **Conflicts of Interest** There are no conflicts or competing interests. We ensure that this manuscript has a novel contribution and has not been published or submitted to any other publisher before.
- **Author Contributions** Methodology: Yijun Duan and Chenyi Zhuang. System implementation / experiments / original draft preparation: Yijun Duan. All authors contributed to the study conception, revised and approved the manuscript.
- **Consent for Participation & Publication** There is the consent of all authors.
- **Data Availability** The dataset used in this study is available at http://www.cs.cmu.edu/ ark/bio/.

REFERENCES

[1] L. A. Lillard and L. J. Waite, "A joint model of marital childbearing and marital disruption," *Demography*, vol. 30, no. 4, pp. 653–681, Nov. 1993.
[2] D. P. Hogan and N. M. Astone, "The transition to adulthood," *Annu. Rev. Sociol.*, vol. 12, no. 1, pp. 109–130, 1986.
[3] C. Wagner, D. Garcia, M. Jadidi, and M. Strohmaier, "It's a man's Wikipedia? Assessing gender inequality in an online encyclopedia," 2015, *arXiv:1501.06307*.
[4] A. Broder, "A taxonomy of web search," *ACM SIGIR Forum*, vol. 36, no. 2, pp. 3–10, 2002.
[5] R. Blanco, H. Joho, A. Jatowt, H. Yu, and S. Yamamoto, "Overview of NTCIR-13 actionable knowledge graph (AKG) task," in *Proc. NTCIR Conf.*, 2017, pp. 1–6.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[7] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*.

[8] D. Bamman and N. A. Smith, "Unsupervised discovery of biographical structure from text," *Trans. Assoc. Comput. Linguistics*, vol. 2, pp. 363–376, Dec. 2014.

[9] T. Wolf *et al.*, "HuggingFace's transformers: State-of-the-art natural language processing," 2019, *arXiv:1910.03771*.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[11] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[12] Z. Yao, Y. Sun, W. Ding, N. Rao, and H. Xiong, "Dynamic word embeddings for evolving semantic discovery," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 673–681.

[13] Y. Takahashi, H. Ohshima, M. Yamamoto, H. Iwasaki, S. Oyama, and K. Tanaka, "Evaluating significance of historical entities based on tempo-spatial impacts analysis using Wikipedia link structure," in *Proc. 22nd ACM Conf. Hypertext Hypermedia*, 2011, pp. 83–92.

[14] A. Jatowt, D. Kawai, and K. Tanaka, "Predicting importance of historical persons using Wikipedia," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 1909–1912.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 1999-66, Nov. 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[16] T. Van de Cruys, "A neural network approach to selectional preference acquisition," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 26–35.

[17] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsNMF)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 403–415, Mar. 2006.

[18] C. Zhuang, N. J. Yuan, R. Song, X. Xie, and Q. Ma, "Understanding people lifestyles: Construction of urban movement knowledge graph from GPS trajectory," in *Proc. IJCAI*, Aug. 2017, pp. 3616–3623.

[19] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 556–562, 2001.

[20] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.

[21] Y. Wu, X. Liu, M. Xie, M. Ester, and Q. Yang, "CCCF: Improving collaborative filtering via scalable user-item co-clustering," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, Feb. 2016, pp. 73–82.

[22] M. M. Rahman and A. Takasu, "TLAB at the NTCIR-13 AKG task," in *Proc. NTCIR Conf.*, 2017, pp. 1–4.

[23] X. Kang, Y. Wu, and F. Ren, "TUA1 at the NTCIR-13 actionable knowledge graph task: Sampling related actions from online searching," NTCIR, Tokyo, Japan, Tech. Rep. 2, 2017.

[24] X. Lin, W. Lam, and S. Sharma, "CUIS team for NTCIR-13 AKG task," in *Proc. 13th NTCIR Conf.*, 2017, pp. 1–3.

[25] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[26] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, nos. 1–4, pp. 164–189, 1927.

[27] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," 2017, *arXiv:1707.01476*.

[28] R. Wang, B. Li, S. Hu, W. Du, and M. Zhang, "Knowledge graph embedding via graph attenuated attention networks," *IEEE Access*, vol. 8, pp. 5212–5224, 2020.

[29] P. Ristoski and H. Paulheim, "RDF2VEC: RDF graph embeddings for data mining," in *Proc. Int. Semantic Web Conf.* Kobe, Japan: Springer, 2016, pp. 498–514.

[30] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 192–202.

[31] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.-C. N. Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, "Knowledge graphs," 2020, *arXiv:2003.02320*.

[32] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," *Electronics*, vol. 9, no. 5, p. 750, May 2020.

[33] F. Lu, P. Cong, and X. Huang, "Utilizing textual information in knowledge graph embedding: A survey of methods and applications," *IEEE Access*, vol. 8, pp. 92072–92088, 2020.

[34] S. Cucerzan, "Large-scale named entity disambiguation based on Wikipedia data," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn. (EMNLP-CoNLL)*, 2007, pp. 708–716.

[35] R. Bunescu and M. Paşca, "Using encyclopedic knowledge for named entity disambiguation," in *Proc. 11th Conf. Eur. Chapter Assoc. Comput. Linguistics*. Trento, Italy: Association for Computational Linguistics, Apr. 2006. [Online]. Available: https://www.aclweb.org/anthology/E06-1002

[36] M. Conway, "Mining a corpus of biographical texts using keywords," *Literary Linguistic Comput.*, vol. 25, no. 1, pp. 23–35, Apr. 2010.

[37] Y. Duan, A. Jatowt, and K. Tanaka, "Discovering typical histories of entities by multi-timeline summarization," in *Proc. 28th ACM Conf. Hypertext Social Media*, Jul. 2017, pp. 105–114.

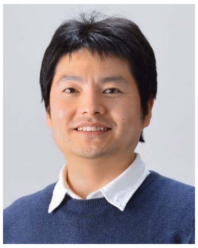
[38] Y. Duan, A. Jatowt, and K. Tanaka, "History-driven entity categorization," in *Proc. Asia–Pacific Web (APWeb) Web-Age Inf. Manage. (WAIM) Joint Int. Conf. Web Big Data*. Chengdu, China: Springer, 2019, pp. 349–364.

YIJUN DUAN received the bachelor's degree in control engineering from Nanjing University, Nanjing, China, and the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan. He is currently a Postdoctoral Researcher with the Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Japan. His research interest includes time-stamped data mining.

XIN LIU received the M.Sc. degree in computer science from Wuhan University, and the Ph.D. degree in computer science from the Tokyo Institute of Technology. He is currently a Senior Researcher with the Artificial Intelligence Research Center (AIRC), National Institute of Advanced Industrial Science and Technology (AIST). His research interests include graph & network analytics, data mining, machine learning, and deep learning.

ADAM JATOWT received the Ph.D. degree in information science from The University of Tokyo. He is currently a Full Professor with the Department of Computer Science and the Digital Science Center, University of Innsbruck, Austria. Before moving to Europe, he worked as an Assistant Professor and an Associate Professor with the Department of Social Informatics, Kyoto University. Then, he worked as a Postdoctoral Researcher with the National Institute of Information and Communications Technology (NICT), Japan, in 2005, before moving to Kyoto University. His research interests include information retrieval, natural language processing, digital libraries, and digital history.

CHENYI ZHUANG received the B.S. degree in SE from Nanjing University, in 2011, and the M.S. and Ph.D. degrees in informatics from Kyoto University, in 2014 and 2017, respectively. He joined Ant Group as a Senior Algorithm Expert, in June 2019. Prior to that, he was a Researcher with the Artificial Intelligence Research Center (AIRC), AIST, in Japan. From 2015 to 2018, he was also serving as a Young Scientist in Japan Society for the Promotion of Science (JSPS). His current research interests include structured data mining, recommendation, and urban computing.



HAI-TAO YU (Member, IEEE) received the Ph.D. degree from the Faculty of Engineering, University of Tokushima, Japan, in 2013. He is currently working with the University of Tsukuba as a Tenured Associate Professor and leading the Information Intelligence Research Group. His research interests include information retrieval, knowledge graph, and machine learning. He has published a number of papers on top international conferences (e.g., WSDM, CIKM, SIGIR, WWW, ECIR, and AAAI) and journals (e.g., *Information Processing and Management*, and *Information Retrieval Journal*). From February 2015 to January 2017, he held a visiting faculty position with the IR Group, School of Computing Science, University of Glasgow, U.K. In 2017, he has been awarded the Excellent Young Faculty Award of University of Tsukuba. He is a member of ACM. He is the Co-Organizer of the NTCIR tasks, such as Temporalia-2 and AKG. He has been serving as a Program Committee Member for a number of top international conferences (e.g., WSDM, CIKM, SIGIR, and ECIR). For more information visit the link (https://ii-research-yu.github.io).

STEVEN LYNDEN received the Ph.D. degree in computer science from Cardiff University, in 2004. Following this, he was a Postdoctoral Researcher with Newcastle University and Manchester University, before moving to Japan to work at the National Institute of Advanced Industrial Science and Technology (AIST), where he is currently a Senior Researcher. His research interests include linked open data, web data management, and distributed query processing.

KYOUNG-SOOK KIM received the Ph.D. degree in computer science from Pusan National University, South Korea, in 2007. She worked as a Researcher with the National Institute of Information and Communications Technology (NICT), Japan, from November 2007 to March 2014. She has been working with the National Institute of Advanced Industrial Science and Technology (AIST), Japan, since April 2014, where she is currently the Deputy Director of the Artificial Intelligence Research Center. Her research interests include big data analysis, spatiotemporal data platforms, and location-based services. She currently contributes to international standardization related to big data and AI technologies in ISO, IEC, JTC1, and OGC. She is a member of ACM SIGMOD, DBSJ, and IEICE.

AKIYOSHI MATONO received the Ph.D. degree from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2005. From 2005 to 2007, he worked as a Postdoctoral Researcher with the National Institute of Advanced Industrial Science and Technology (AIST), Japan. From 2007 to 2020, he worked as a Researcher at AIST, where he is currently working as the Team Leader of Data Platform Research Team with the Artificial Intelligence Research Center. His research interests include database technology, parallel processing, and query processing. He is a member of DBSJ, IPSJ, and IEICE.