MANTECH Publications

Journal of Data Mining and Knowledge Engineering

Volume 8, Issue 1, January-June 2023

Temporal Knowledge Graphs and Event Evolution Mining

Sravesh Pandey, Sobhangi Sharma, Retesh J kale

Associate Professor, Assistant Professor

Department of Computer Science

Stella Maris College, Chennai, India

Email: *Sravesh021pandey@rediffmail.com, sobhangisharmaj9@gmail.com, reteshj61k@yahoo.com*

Abstract

Temporal knowledge graphs (TKGs) extend conventional knowledge graphs (KGs) by incorporating temporal information to represent dynamic relationships over time. Event evolution mining leverages TKGs to uncover temporal patterns, causality, and sequence of events in complex systems. With applications spanning social media analysis, finance, cybersecurity, and healthcare, TKGs and event evolution mining have emerged as crucial tools for understanding evolving phenomena. This paper presents a comprehensive review of TKGs, their construction, embedding techniques, temporal reasoning models, and event evolution mining methods. We discuss challenges such as data sparsity, reasoning over long-term dependencies, and scalability, alongside current solutions. Future research directions are highlighted to address limitations and enhance the utility of temporal knowledge systems.

Keywords: *Temporal knowledge graphs, event evolution, temporal reasoning, dynamic knowledge graphs, temporal embeddings, causal inference, time-aware graph modeling.*

INTRODUCTION

Knowledge graphs (KGs) are structured representations of entities and their relationships, widely used in search engines, recommender systems, and AI applications. Conventional KGs, however, lack temporal information, treating relationships as static. Real-world data is inherently dynamic: social interactions evolve, financial markets fluctuate, and cyber events unfold sequentially. Temporal knowledge graphs (TKGs) address this limitation by encoding the temporal dimension into the graph structure, allowing models to capture how relationships change over time.

Event evolution mining focuses on extracting meaningful patterns from TKGs. By analyzing sequences of events and their dependencies, researchers can predict future occurrences, detect anomalies, and understand causal chains. This capability is increasingly important in domains such as stock market prediction, epidemic modeling, and critical infrastructure monitoring.

TEMPORAL KNOWLEDGE GRAPHS: FUNDAMENTALS

Temporal Knowledge Graphs (TKGs) extend conventional knowledge graphs by explicitly representing the temporal dimension of facts. Unlike traditional knowledge graphs, where relationships are static, TKGs capture how entities and their relationships evolve over time. This temporal information allows systems to reason about past events, detect patterns, and predict future developments.

TKGs are particularly valuable in domains where the dynamics of relationships are critical. Examples include financial markets, social networks, biomedical data, and cyber-physical systems. By integrating time into the graph, TKGs enable richer queries such as “Which companies acquired new subsidiaries in 2022?” or “Which patient treatments led to improved outcomes over time?”

STRUCTURE OF TEMPORAL KNOWLEDGE GRAPHS

Formally, a temporal knowledge graph can be defined as a quadruple:

G=(E,R,T,F)G = (E, R, T, F)G=(E,R,T,F)

Where:

- **E** is the set of entities, representing objects, persons, organizations, or abstract concepts.
- **R** is the set of relations, describing interactions or connections between entities.
- **T** is the set of timestamps, which could be discrete (days, months, years) or continuous (exact date-time).
- **F** is the set of temporal facts, each defined as F⊆E×R×E×TF \subseteq E \times R \times E \times TF⊆E×R×E×T.

A single temporal fact is represented as a quadruple (h,r,t,τ)(h, r, t, \tau)(h,r,t,τ), where:

- h∈Eh \in Eh∈E is the **head entity** (the subject of the relationship)

- r∈Rr \in Rr∈R is the **relation** type
- t∈Et \in Et∈E is the **tail entity** (the object of the relationship)
- τ∈T\tau \in Tτ∈T is the **timestamp** when the relationship holds

This formalization allows the graph to encode not only the presence of relationships but also **when they are valid**.

1. Example of Temporal Knowledge Graph Facts

Head Entity	Relation	Tail Entity	Timestamp
CompanyA	acquires	CompanyB	2023-04-12
PersonX	joins	CompanyA	2023-05-01
CompanyC	merges	CompanyD	2022-11-15
DiseaseY	affects	RegionZ	2021-07-10

Here, each row represents a temporal fact:

- The first row shows that **CompanyA acquired CompanyB** on **April 12, 2023**.
- The second row indicates that **PersonX joined CompanyA** on **May 1, 2023**.
- The timestamps are critical, as the relationship may not exist outside that time. For example, prior to 2023-04-12, the acquisition of CompanyB by CompanyA is **not valid**.

2. Types of Temporal Relations

Temporal knowledge graphs often include multiple types of temporal relationships:

- **Instantaneous Events:** Occur at a single point in time (e.g., a merger, a tweet).
- **Durative Relations:** Persist over a time interval (e.g., employment, membership, subscriptions).
- **Recurring Events:** Repeat periodically (e.g., seasonal sales, scheduled maintenance).

By capturing these, TKGs can answer both **static queries** (“Who is the CEO of CompanyA?”) and **temporal queries** (“Who was the CEO of CompanyA in 2021?”).

This temporal annotation allows reasoning over event sequences and evolution.

APPLICATIONS OF TEMPORAL KNOWLEDGE GRAPHS

Temporal Knowledge Graphs (TKGs) are highly versatile, enabling organizations and researchers to model **dynamic systems where relationships evolve over time**. By incorporating temporal information, TKGs provide the ability to **analyze historical trends,**

detect emerging patterns, forecast future events, and uncover causal links. Below, we elaborate on the major application domains.

1. Social Media Analysis

Social media platforms generate massive volumes of time-stamped data including posts, likes, shares, comments, and user interactions. TKGs provide a structured way to capture **how users, topics, and hashtags evolve over time**, making them invaluable for trend detection, influence analysis, and misinformation tracking.

Example:

A TKG for Twitter could encode facts like:

Head Entity	Relation	Tail Entity	Timestamp
UserA	posts	TweetX	2023-01-10
TweetX	contains_hashtag	#AI	2023-01-10
UserB	retweets	TweetX	2023-01-11

By analyzing the temporal evolution of hashtags and retweets:

- **Trend detection:** Identify emerging topics (e.g., #AI trending in January).
- **Influencer identification:** Track which users accelerate topic propagation.
- **Misinformation detection:** Trace the spread of false information and understand temporal patterns of virality.

Mini case study:

During the COVID-19 pandemic, TKGs were used to track **how misinformation about vaccines spread across platforms**. By linking user interactions with timestamps and content types, researchers could identify **critical nodes responsible for rapid dissemination** and predict future high-risk events.

Finance

Financial systems are inherently dynamic, with companies, stocks, market indices, and economic indicators evolving over time. TKGs can model **complex temporal relationships** among these entities to enable **predictive analytics, risk assessment, and portfolio**

optimization.

Example:

Head Entity	Relation	Tail Entity	Timestamp
CompanyA	acquires	CompanyB	2023-04-12
StockX	correlates_with	StockY	2023-05-01
CompanyC	partners_with	CompanyD	2022-11-15

Applications of TKGs in finance:

- **Event-driven stock prediction:** Model how corporate events (mergers, acquisitions, partnerships) impact stock prices over time.
- **Fraud detection:** Identify anomalous transactions or unusual trading patterns by observing deviations from normal temporal relationships.
- **Risk assessment:** Forecast potential systemic risks by analyzing evolving interdependencies between financial entities.

Mini case study:

A TKG representing relationships among companies, executives, and stock market movements can detect **early signs of financial stress**. For example, sudden shifts in partnerships combined with executive resignations could signal potential market volatility. Temporal embeddings in TKGs allow predictive models to anticipate such events with higher accuracy than static approaches.

TKG CONSTRUCTION METHODS

Building a Temporal Knowledge Graph (TKG) is a multi-stage process that involves extracting entities, events, and temporal information from diverse data sources, identifying temporal relations, and linking events to a canonical representation. TKG construction ensures that the resulting graph is both **accurate** and **time-consistent**, enabling downstream tasks such as event evolution mining, forecasting, and anomaly detection.

KNOWLEDGE EXTRACTION

The first step in constructing a TKG is **knowledge extraction**, which involves identifying entities, relationships, and timestamps from available data sources. Depending on the source type, extraction methods vary:

1. Structured Data Sources

Structured sources include **databases, spreadsheets, or transactional logs** where entities and relationships are explicitly recorded along with timestamps.

Examples:

- Financial databases recording mergers, acquisitions, and stock trades.
- Healthcare EHRs containing diagnoses, treatments, and timestamps.
- Sensor logs from IoT devices in industrial or transportation systems.

Extraction Process:

1. Identify entity tables (e.g., Patients, Companies).
2. Identify relationship tables (e.g., treatment_given, acquisition).
3. Extract timestamp fields corresponding to each relationship.
4. Convert each record into a temporal fact (h,r,t,τ)(h, r, t, \tau)(h,r,t,τ).

Illustration:

PatientID	Disease	Treatment	Date
P001	Flu	DrugA	2023-01-10

Becomes a temporal fact: (P001, treated_with, DrugA,2023−01−10)(P001, treated_with, DrugA, 2023-01-10)(P001,treated_with,DrugA,2023−01−10).

2. Semi-Structured Data Sources

Semi-structured data sources, such as **Wikipedia, Wikidata, or DBpedia**, contain both structured elements (tables, infoboxes) and natural language text. These sources often include temporal annotations for historical or factual events.

Example:

- Wikipedia infobox: CompanyA | Founded: 2005 | Acquired CompanyB: 2018
- Wikidata entry: (Q1234, acquisition, Q5678, 2018)

Extraction Methods:

- **Template Parsing:** Extract key-value pairs from infoboxes.
- **SPARQL Queries:** Directly query Wikidata for temporal facts.
- **Pattern Matching:** Detect temporal expressions in text fields.

Semi-structured sources bridge the gap between fully structured and unstructured data, providing rich temporal information with moderate preprocessing.

3. Unstructured Data Sources

Unstructured sources are **text-heavy datasets** such as news articles, scientific papers, social media posts, and blogs. Extracting temporal knowledge from these requires **Natural Language Processing (NLP)** techniques.

Extraction Workflow:

- **Entity Recognition:** Identify entities (persons, organizations, locations) using Named Entity Recognition (NER).
- **Event Detection:** Detect actions, occurrences, or relationships in sentences.
- **Temporal Tagging:** Extract explicit dates ("on April 12, 2023") or relative expressions ("last Monday").
- **Fact Construction:** Convert detected entities, relations, and timestamps into temporal facts (h,r,t,τ)(h, r, t, \tau)(h,r,t,τ).

Example Sentence:

"On 12th April 2023, CompanyA acquired CompanyB."

Extraction Result:

(CompanyA, acquires, CompanyB, 2023−04−12)(CompanyA, acquires, CompanyB, 2023-04-12)(CompanyA, acquires, CompanyB, 2023−04−12)

TEMPORAL RELATION IDENTIFICATION

After entities and events are extracted, the next step is to **identify and classify the temporal relationships** between them. This is critical for building a graph that reflects the evolution of relationships over time.

1. Rule-Based Approaches

Rule-based methods rely on **predefined linguistic or logical patterns** to detect temporal relations.

- **Temporal patterns:** Phrases like "before," "after," "on [date]," or "during [period]."
- **Regular expressions:** Detect date formats or recurring patterns in text.
- **Event sequencing rules:** Define relative order of events (e.g., a hire event always follows a job posting).

Pros: Simple, interpretable, works well for well-defined domains.

Cons: Poor generalization, limited to predefined patterns.

2. Machine Learning Approaches

Machine learning methods treat temporal relation extraction as a **sequence labeling or classification problem**:

- **Conditional Random Fields (CRFs):** Label tokens in a sentence with temporal tags (e.g., EVENT, DATE, BEFORE, AFTER).
- **BiLSTM-CRF Models:** Capture sequential dependencies in text, improving extraction of multi-token events and temporal expressions.

Example:

Sentence: *"CompanyA acquired CompanyB on April 12, 2023."*

Model Output:

- CompanyA → HEAD_ENTITY
- acquired → RELATION
- CompanyB → TAIL_ENTITY
- April 12, 2023 → TIMESTAMP

3. Transformer-Based Models

Pretrained transformer models, such as **BERT, RoBERTa, or T5**, can be fine-tuned for **temporal relation extraction** with high accuracy:

- Handle complex language and long-range dependencies.
- Can be trained on domain-specific corpora (finance, healthcare, social media).
- Often outperform traditional ML models in noisy or unstructured data scenarios.

Workflow:

1. Input text is tokenized and encoded using BERT.
2. Transformer layers capture contextual information for each token.

3. A classification head predicts temporal relations and timestamps.

Example: Fine-tuning BERT to extract (CompanyA, acquires, CompanyB, 2023-04-12) from financial news.

EVENT LINKING

Once entities and temporal facts are extracted, **event linking** ensures that the same real-world entity is represented consistently across the TKG.

1. Purpose

- Avoids redundancy (e.g., “Apple Inc.” vs. “Apple”).
- Maintains temporal consistency across different sources.
- Facilitates reasoning over long-term event sequences.

2. Techniques

- **Entity Resolution:** Match textual mentions to canonical entities in a knowledge base (e.g., Wikidata IDs).
- **Coreference Resolution:** Identify multiple mentions of the same entity within or across documents.
- **Temporal Alignment:** Ensure that timestamps are normalized (e.g., “last Monday” → exact date) and conflicts are resolved.

Example:

- Mentions: “Apple acquired Beats” in a news article, “Apple Inc. bought Beats Electronics” in Wikipedia.
- After linking: (Apple, acquires, Beats, 2014-08-01)

Event linking ensures that queries and temporal reasoning across the TKG produce **accurate and coherent results**.

TEMPORAL KNOWLEDGE GRAPH EMBEDDING

Embedding models project entities and relations into continuous vector spaces for downstream tasks like link prediction and event forecasting.

1. Static vs. Temporal Embeddings

Embedding Type	Description	Use Case
Static	Ignore temporal info	Traditional KG tasks
Temporal	Encode timestamp info	Event prediction, evolution mining

2. Temporal Embedding Methods

1. **Translational Models:** Extends TransE with time-aware scoring functions (e.g., TTransE).
2. **Recurrent Models:** Use RNNs or LSTMs to capture sequential dynamics in entity relations.
3. **Graph Neural Networks (GNNs):** Incorporate temporal edges in message passing frameworks (e.g., TGAT, TGN).

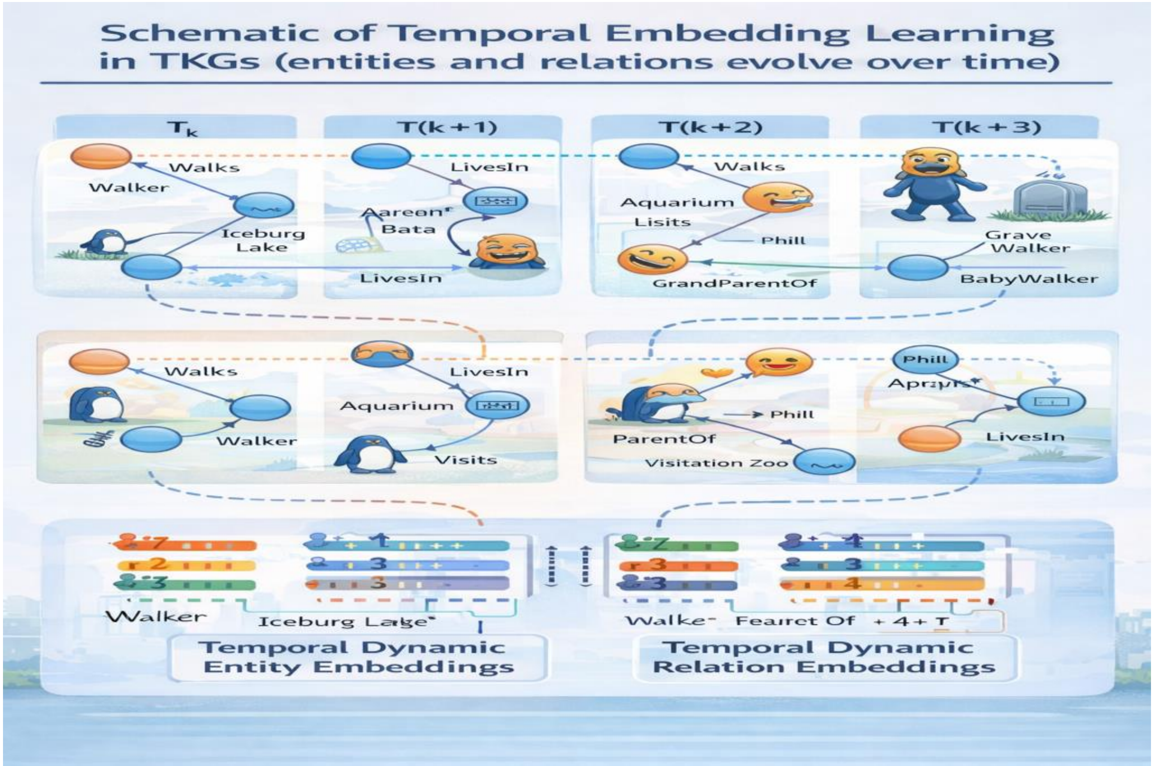

Figure 1: Schematic of temporal embedding learning in TKGs (entities and relations evolve over time).

EVENT EVOLUTION MINING

Event evolution mining involves analyzing sequences of events to identify patterns, causal relationships, and predict future events.

1. Event Sequence Modeling

- **Markov Models:** Represent events as states with transition probabilities.
- **Recurrent Neural Networks:** Capture longer-term dependencies beyond Markov assumptions.
- **Temporal Point Processes:** Model event occurrences as stochastic processes over time.

2. Causal Inference in TKGs

Understanding causality in evolving systems is crucial. Techniques include:

- Granger causality adapted to graph structures.
- Counterfactual reasoning using temporal embeddings.
- Probabilistic graphical models incorporating temporal dependencies.

3. Applications

Domain	Example	Benefits
Finance	Predicting mergers and acquisitions	Early investment insights
Healthcare	Tracking disease progression	Early detection and intervention
Social Media	Detecting misinformation spread	Proactive moderation

CHALLENGES AND SOLUTIONS

1. Data Sparsity

Temporal knowledge graphs often have sparse temporal facts. **Solutions:** data augmentation, knowledge graph completion models.

2. Scalability

TKGs can grow rapidly with time and entities. **Solutions:** temporal sampling, hierarchical embedding's, distributed GNNs.

3. Long-Term Dependency Modeling

Some events influence distant future occurrences. **Solutions:** Transformer-based temporal models (e.g., Time-aware Graph Transformers).

4. Noise and Uncertainty

Event extraction from unstructured data is prone to errors. **Solutions:** uncertainty-aware embedding's, probabilistic reasoning, ensemble models.

FUTURE DIRECTIONS

- **Explainable Temporal Reasoning:** Generating human-readable explanations for predictions.
- **Integration with Multi-Modal Data:** Incorporating images, text, and sensor data in TKGs.
- **Federated TKGs:** Privacy-preserving knowledge sharing across organizations.
- **Real-time Event Evolution Mining:** Streaming TKG updates for immediate decision-making.

CONCLUSION

Temporal knowledge graphs and event evolution mining offer powerful frameworks for modeling dynamic systems. By incorporating temporal dimensions, TKGs enable richer reasoning and predictive capabilities. Despite challenges such as sparsity, scalability, and long-term dependency modeling, recent advances in embedding techniques, graph neural networks, and probabilistic reasoning have significantly improved performance. Future research should focus on explainability, multi-modal integration, and real-time analysis to fully realize the potential of TKGs in complex domains.

REFERENCES

1. Wu, Y., et al. “Temporal Knowledge Graph Embedding for Event Forecasting.” *IEEE Transactions on Knowledge and Data Engineering*, 2021.
2. Jin, W., et al. “Recurrent Event Networks for Temporal Knowledge Graphs.” *AAAI Conference on Artificial Intelligence*, 2020.
3. Zhang, M., et al. “Temporal Graph Neural Networks for Dynamic Knowledge Graphs.” *NeurIPS*, 2020.
4. Goel, A., & Sharma, P. “Event Evolution Mining in Social Media.” *Journal of Data Mining*, 2019.
5. Trivedi, R., et al. “Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs.” *ICML*, 2017.
6. Li, J., et al. “TGAT: Temporal Graph Attention Networks for Temporal Knowledge Graphs.” *WWW*, 2020.
7. Wang, X., et al. “Event Forecasting on Temporal Knowledge Graphs Using Temporal Point Processes.” *KDD*, 2019.
8. Pan, S., et al. “Survey on Knowledge Graph Embeddings.” *IEEE Transactions on*

Knowledge and Data Engineering, 2019.
9. Chen, L., et al. "Causal Discovery in Temporal Knowledge Graphs." *AAAI Workshop on Causality*, 2021.
10. Das, P., et al. "Challenges and Opportunities in Temporal Knowledge Graphs." *Information Systems*, 2022.

