

Link Prediction on Dynamic Attributed Knowledge Graphs for Maritime Situational Awareness

Jacques Everwyn^{1,2} Abdel-illah Mouaddib¹ Bruno Zanuttini¹ Sylvain Gatepaille² Stephan Brunessaux²

¹ Normandie Université, Université de Caen Normandie, GREYC

² Airbus Defence and Space

jacques.everwyn@unicaen.fr

Main research area: AI

July 2019

Résumé

Actuellement, les opérateurs de surveillance maritime parcourent à la main les quantités massives de données à leur disposition pour repérer les événements à surveiller. Les données maritimes viennent de sources variées et hétérogènes qui peuvent être fusionnées en un graphe de connaissance dynamique avec attributs, qui représente l'évolution d'une situation maritime. Via ce graphe, l'automatisation de la levée d'alerte revient à une tâche de prédiction de lien: étant donnés des labels venant de connaissance experte, y a-t-il d'autres situations similaires que l'on veut relever dans le graphe? Dans cet article, nous allons passer en revue plusieurs techniques de prédiction de lien dans un contexte de surveillance maritime et tirer des conclusions sur les bénéfices que pourrait apporter l'ajout d'attributs dans les modèles de graphes dynamiques pour l'exécution de cette tâche.

Keywords

Graphe de connaissance dynamique, situation maritime, attributs, apprentissage machine, prédiction de liens

Abstract

Currently, maritime surveillance operators have to monitor by hand the massive amount of data at their disposal to spot the events of interest, thus limiting their capabilities. Maritime data comes from various and heterogeneous sources, that can be merged into a dynamic attributed knowledge graph which represents an evolving maritime situation. Using this graph, the automation of alert rising comes through a link prediction task: given some labels from expert knowledge, are there similar situations of interest elsewhere in the graph? In this article, we review link prediction techniques for situation awareness in a maritime context, and draw conclusions on how the addition of attributes in a dynamic graph model could improve results on this task.

Keywords

Dynamic knowledge graph, maritime situation, attributes, machine learning, link prediction

1 Introduction

The maritime domain is the theater of many unlawful activities that may go unnoticed: terrorism, piracy, smuggling, illegal immigration... That's why Maritime Situational Awareness (MSA) is of first importance to maritime security. It is defined by NATO as "The understanding of military and non military events, activities and circumstances within and associated with the maritime environment that are relevant for current and future NATO operations and exercises, where the Maritime Environment (ME) is the oceans, seas, bays, estuaries, waterways, coastal regions and ports" [1]. MSA is often performed by surveillance operators who monitor the flow of data coming from maritime activities. This data is diverse, heterogeneous, and comes from several sources: AIS (Automatic Identification System), radars, satellites, intelligence, websites... With more than 50.000 vessels sailing the oceans each day, there is a need for automation in the detection of illicit events [2].

A maritime situation implies evolving entities: vessels, ports, countries... Such a situation can be represented by a *dynamic attributed* knowledge graph (DAKG), and understanding how its elements connect and jointly evolve gives valuable information pertaining to MSA. This task is here reduced to a link prediction problem. A *link*, or an *event*, is a relation between two entities at a given time point, for instance (Titanic ; :builtBy ; WhiteStarCompany ; 1909), and *attributed* means that entities can have attributes whose values can change over time, e.g. (Titanic ; :passengers ; 2,344 ; April 10th 1912).

Generally, link prediction is performed by learning an embedding for each entity of the graph, and predictions are

made by ranking the events in the graph using these embeddings. This can benefit to MSA in two ways:

- *data completion*: when monitoring an operational situation, the sensors and reports do not always have all the needed information at their disposal. Using link prediction, missing data can be inferred to improve MSA;
- *automated alerts*: link prediction can discover events that a human operator would not have noticed in the massive dataset. Illegal activities could also be anticipated by making prediction in the future and evaluating the risk a ship represents based on its current and past behavior.

In this article, we review (1) two models on a dynamic (but not attributed) knowledge graph, (2) the literature on static/dynamic/attributed knowledge graphs, (3) how to apply DAKGs to MSA.

2 Previous work

The previous work related to this study can be broadly divided into four categories: maritime related work, static graphs, dynamic graphs and attributed graphs.

MSA. MSA often focuses on anomaly detection [3]. It can be tackled with clustering [2], bayesian networks [4], self-organizing maps [5] and many others techniques [6]. Route estimation is also handled, e.g. with neural networks [7] or Extended Kalman filter [8]. To the best of our knowledge, this is the first attempt of using link prediction on DAKG to improve MSA.

Static Knowledge Graph. In a static setting, each node is represented by a single vector. This field is largely covered with a broad range of techniques. Translational models evaluate a fact by measuring the distance between the two entities, generally using the relation during the translation. TransE [9] is its most known representative. Semantic matching models are similarity-based and compare the latent semantics of entities and relations embeddings. RESCAL [10] was the first to do this and has been extended multiple times [11] [12]. Neural network architectures have also been tried with NTN [13] or VGAE [14]. These models achieve great performances on static knowledge graphs but are not suited to deal with dynamic ones.

Dynamic Knowledge Graph. In a dynamic setting, each node is represented by a time series of vector modeling its evolution. This topic is emerging and has less contributions but advances have already been made. Leblay et al. [15] predict time validity for unannotated edges using side information in the learning process. Esteban et al. [16] updates the knowledge graph using an event graph to add new information, and Trivedi et al. [17] extends the bilinear model (RESCAL) with a LSTM network in order to learn non-linearly evolving entities. Jiang et al. [18] incorporate the valid time of facts using a joint time-aware infer-

ence model based on Integer Linear Programming. Self-attention networks were tried by Sankar et al. [19].

Although these models are time-aware, they do not include attribute information in the relation prediction task and we will show that they are needed when dealing with MSA.

Attributes. Lin et al. [20] can predict discrete attribute values and find correlation between them. However, they are not included during the learning of relations and relations are not included in the learning of attributes. Tay et al. [21] propose a model that jointly learns KG^R and KG^A with a neural network and predicts continuous values with a regression task. However, neither model deals with temporal data.

Li et al. [22] propose a streaming model (SLIDE) on dynamic attributed networks using a sketching matrix that summarizes the currently observed links and node attributes. They review the challenges pertaining to such networks and real-world data, but they apply it on social networks (Epinions, DBLP, ACM) that have very different kinds of attributes and only a few widely separated timesteps (~20 timesteps from one month to one year each). All these models showed that the addition of attributes improves the results on link prediction.

Table 1 compares representative models from each category and shows that no model currently fits our needs perfectly (see Part 4 for detailed requirements).

	Static	Dynamic	Attributes	Near Real-Time
TransE[9]	X			
Know-Evolve[17]		X		
MT-KGNN[21]	X		Continuous	
SLIDE[22]		X	Discrete	~

Table 1: Application domain of models comparison

3 Problem statement

The relation and attribute prediction problems are formalized in this section.

3.1 Dynamic Knowledge Graphs

Before introducing dynamicity and attributes, we recall the definition of standard knowledge graphs.

Definition 1 (standard knowledge graph) *Let*

$E = \{e_1, \dots, e_n\}$ *and* $R = \{r_1, \dots, r_k\}$ *be two finite sets, of entities and relations, respectively. A knowledge graph on* E, R *is a finite set* $KG \subseteq E \times R \times E$. *For a triple* $t = (e^s, r, e^o) \in KG$, *e^s is called the subject of* t , *r is called its relation, and* e^o *is called its object.*

We now introduce attributes and dynamicity. Note that the relation between an entity and (some value for) an attribute can be seen as a triple in a knowledge graph, but we define it differently because we want to handle them in a specific manner when predicting with knowledge graphs.

Definition 2 (frame) A frame is a quadruple $F = \langle E, R, A, D \rangle$ where E , R , and A are finite sets of elements called entities, relations, and attributes, respectively, and $D : A \rightarrow S_a$ is a function assigning a range $D(A)$ to each attribute and S_a is a set of possible values for $a \in A$ (discrete or continuous).

We write $|E|$ (resp. $|R|$, $|A|$) for the size of E (resp. of R , of A), and $|F|$ for $|E| + |R| + |A|$.

We are now in position to define a knowledge graph with attributes and time (which we simply call “knowledge graph” for simplicity).

Definition 3 (knowledge graph) Let $F = \langle E, R, A, D \rangle$ be a frame. A standard knowledge graph on F is a couple $KG = \langle KG^R, KG^A \rangle$, where

- KG^R is a finite subset of $E \times R \times E \times \tau$ with τ the set of time points,
- KG^A is a finite subset of $E \times A \times D \times \tau$ such that for all quadruples $(e, a, v, t) \in KG^A$, $v \in D(a)$ holds.

For $KG = \langle KG^R, KG^A \rangle$, KG^R is called the relational part of KG , and KG^A is called its attributional part. The last component of each tuple in KG^R or KG^A is called its timestamp or time point ($t \in \tau$), the time at which the attribute’s value or entity’s relation is valid.

Intuitively, (e^s, r, e^o, t) is read “entity e^s is in relation r with entity e^o at time t ”, and (e, a, v, u) is read “entity e has value v for attribute a at time u ”. Given a knowledge graph KG , we always write KG^R (resp. KG^A) for its relational (resp. attributional) part. Figure 1 is an example of the previously defined KG .

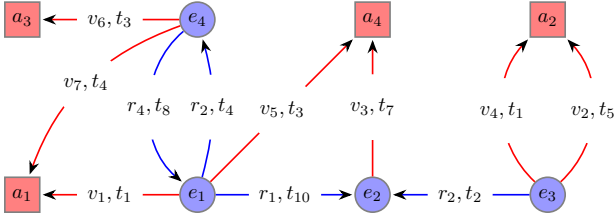


Figure 1: Example of knowledge graph KG on a frame $F = \langle E, R, A, D \rangle$. The nodes $e_i \in E$ are entities and the nodes $a_j \in A$ are attributes. The relations $r_n \in R$ are annotated on edges between two entities. The values v_k , annotated on the edges between two attributes, belong to the range $D(A)$ of the attribute they are attached to, and the t_i s are the timestamps of the edges. Attributes of entities can change their value over time (e_3 and a_2) and two entities can have common attributes (e_4 , e_1 and a_1) but not necessarily with the same value. The blue nodes and edges are KG^R and the red edges with all the nodes are KG^A .

3.2 Prediction Problems with Knowledge Graphs

We are interested in predicting the missing relations between entities and values of attributes in knowledge graphs. We focus on the case where, for some timestamp, they can be predicted from the values of a subset of the relations (R') and attributes (A') at previous timestamps.

The relation r^* (resp. attribute a^*) is said to be determined by $KG^{R', A', \leq t^*}$ if for all timestamps t^* , there is a function $f(\cdot)$ such as $f(KG^{R', A', \leq t^*})$ outputs a relational (resp. attributional) quadruple comprised of r^* (resp. a^*) at time t^* that exists in KG , where $KG^{R', A', \leq t^*}$ denotes the restriction of KG to quadruples with a relation in R' or an attribute in A' , and with a timestamp until t^* . This is the determined relation (resp. attribute) problem.

With this in hand, the learning problem which we tackle is the following. Intuitively, for a given knowledge graph KG^* , we are given all the information just before timestamp t^* together with some information at timestamp t^* , and the problem is to induce some target relation r^* (or attribute a^*) at time t^* .

4 Application to MSA

“Real-world” datasets often have more constraints than the academic ones (YAGO [23], Wikidata...) because of their specificities. Maritime datasets are no exception and the following challenges must be overcome.

4.1 Evolution of attributes

A maritime situation is a fast evolving world with very little time between two events. For instance, the event databases ICEWS [24] and GDELT [25] respectively have a temporal granularity of one day and fifteen minutes. In MSA, a good evolutionary model is needed for change detection and the granularity depends on the task. For a change in the position/course/speed of a vessel (dynamic attributes), the information must be given within minutes (e.g., rapid response needed in case of piracy). But to detect a change in a vessel particulars (identifier, name...), the granularity needed can be in hours or days. When modeling the evolution of a vessel’s attributes, they can be divided into two categories [26]:

- Static attributes: related to static information about a given vessel (name, flag, length...). They are not supposed to change but their evolution must nevertheless be monitored to report modifications (e.g. change of owner) or anomalies (e.g. identity fraud).
- Dynamic attributes: these can be divided into two sub-categories:
 - Kinematic attributes that refers to location, speed, course...
 - Non-kinematic attributes such as passengers, cargo, crew...

Both types of attributes must be handled in the DAKG. Note that they can be discrete (e.g. flag) or continuous (e.g. position).

4.2 Event and threat detection

An event (or quadruple) represents a new relation between two entities or an abovementioned attribute evolution. A maritime relation can be proximity between two vessels, an exchange of goods, harbouring in a port, an attack on another ship... Such events find their roots in both KG^R and KG^A . For instance, two cargo ships from allied countries stopped at the same position are likely to be performing a transshipping (proximity, speed, flag). Currently, most of these events are found using rule-based systems. Using knowledge graphs and machine learning, it could be possible to find events using latent features that cannot be perceived by a human or a rule.

If event mining extracts raw facts, threat detection is a task highly related to its context and definition. A nation will not consider a transshipping between two fishing vessels as a threat since they are more likely to exchange fish than warheads, but an NGO for ocean conservation can suspect illicit fishing of an endangered species. Performing this task still requires either expert knowledge or labeled events.

4.3 Streaming

MSA requires a constant monitoring of maritime areas, meaning that the model must deal with a continuous flow of data. Even if the model does not change after training, the representations of entities and relations must be updated regularly with the incoming information to keep an up-to-date view of the situation. Recent work on the subject can be found in the literature [22, 27].

4.4 Uncertainty

Maritime data often results from hard (sensors) and soft (websites, intelligence) data fusion. However, this data is not always 100% certain: an intelligence report may have a typo, sensors have a range and precision (e.g. +/- 500 meters), or collisions may happen when satellites receive signals. Errors and approximations are inherent to real-world data and the uncertainty of facts must be taken into account when making link prediction [28, 29].

4.5 Explainability

Link prediction models are often black boxes when it comes to the origin of the prediction. However, a surveillance operator needs to know why a prediction was made in order to understand it and justify any upcoming response to an event. Because operators still do not trust AI-based systems to take decisions, explainability is needed to take DAKG-based decisions for MSA [30].

An illustration of all these concepts can be found in Figure 2.

5 Reviewed models

A dynamic and a static link prediction methods are presented in this section. They use embeddings to represent elements of the graph i.e. continuous vector representations for entities, attributes and sometimes relations ($[c_1, \dots, c_n]$ with $\forall i \in [1, n], c_i \in \mathbb{R}$ and n the dimension of the embedding). Algorithm 1 shows the high level mechanisms of the two following models.

Algorithm 1: High-level learning algorithm

Result: Up-to-date embeddings

Input: training set S (triples/quadruples), entities E , relations R , number of iterations nb_it

Initialization

```

| Initialize embeddings;
for  $i \leftarrow 0$  to  $nb\_it$  do
|   Sample batch from  $S$ ;
|   Update embeddings of  $E$  (and  $R$  if relations have
|     embeddings) using score function;
|   Update model parameters (if any) using score
|     function;

```

end

5.1 Know-Evolve

Proposed by Trivedi et al. [17], this model uses a **temporal point process** framework for temporal reasoning over dynamically evolving knowledge graphs that models the **occurrence of a fact**. They propose a novel deep learning architecture that evolves over time based on availability of new facts. The dynamically evolving network (Recurrent Neural Network) ingests the incoming new facts, learns from them and **updates the embeddings of involved entities** based on their recent relationships and temporal behavior. Their model can predict the occurrence of a fact, but also the time when a fact may potentially occur. It supports the Open World Assumption and can predict over unseen entities.

The point process is characterized by the following conditional intensity function:

$$\lambda_r^{e_s, e_o}(t|\bar{t}) = \exp(g_r^{e_s, e_o}(\bar{t})) * (t - \bar{t}) \quad (1)$$

$\lambda_r^{e_s, e_o}(t|\bar{t})$ represents intensity of event involving triplet (e^s, r, e^o) at time t given previous time point $\bar{t}$ when either e^s or e^o was involved in an event. The $\exp$ function ensures that intensity is positive and well-defined, and the model is learned by minimizing the joint negative log likelihood of intensity function.

The relational score function $g_r^{e_s, e_o}$ is computed using a bilinear formulation as follows:

$$g_r^{e_s, e_o} = v^{e_s}(t-)^T \cdot R_r \cdot v^{e_o}(t-) \quad (2)$$

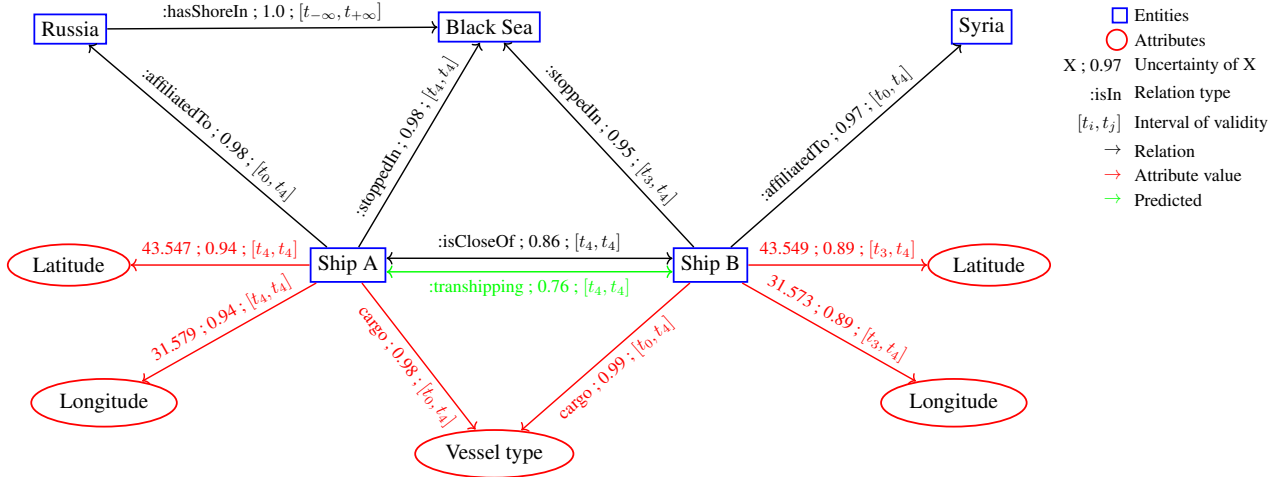


Figure 2: Example of KG at time t_4 (best viewed in color). Events in red represent KG^A and the other events KG^R . Ship A and Ship B are vessels from different countries, moving in the Black Sea. They have a static attribute (vessel type) setting them as cargos and dynamic attributes (latitude and longitude) revealing their positions. Before t_4 , the two vessels were moving in the Black Sea and had different positions, but now (t_4) they have stopped and are close to each other. A possible link prediction from $KG^{<t_4}$ is that the two vessels are performing transshipping. Note that the $:isStopped$ relation can be deduced from the *speed* attribute going to zero, not represented here for the sake of clarity. If relations such as Russia having a shore on the Black Sea are 100% sure, some are more uncertain: the position of Ship B is only 89% sure because the signal was picked up by a satellite in an area with high ship density. More, the uncertainty of predicted relations ($:transshipping$) depends on the uncertainty of the root events. Finally, to predict the transshipping action in time, the model must be updated with the root causes as soon as they are available, hence the need for streaming link prediction.

with $v^{e^s}, v^{e^o} \in \mathbb{R}^d$ the latent feature embeddings of entities, $R_r \in \mathbb{R}^{d \times d}$ the relationship weight matrix and t represents time point just before time t .

In Know-Evolve, events are included in KG^R and the model partially solves the determined relation problem (only using KG^R). KG^A is not included as Know-Evolve does not consider attribute nodes.

The authors proposed a more recent model [31] that improves the first one with an attention mechanism; however in the absence of source code, it is Know-Evolve that is evaluated here¹.

5.2 TransE

TransE [9] is the most representative translational distance model and now has many extensions [32]. It represents both entities and relations as vectors in the same space.

Given an event (e^s, r, e^o) , the relation is interpreted as a **translation vector r between e^s and e^o** so that r connects the two embedded entities with low error, i.e., $e^s + r \approx e^o$ when (e^s, r, e^o) holds. The scoring function is then defined as the (negative) distance between $e^s + r$ and e^o , i.e.,

$$f_r(e^s, e^o) = -\|e^s + r - e^o\|_{1/2}$$

where $_{1/2}$ refers to the L_1 or L_2 norm. The score is expected to be large if (e^s, r, e^o) holds. But this method has problems dealing with 1-to-N, N-to-1 and N-to-N relations, and can not process a temporal graph nor KG^A . It

can learn over new relations but not over new entities. In our study, it only tackles the determined relation problem without t and KG^A . The evaluation was performed using OpenKE [33]².

6 Experiments

In these experiments, the performed task is the prediction of the position of a vessel at the next time points. Obviously, there are many better fitted methods to do this (like regression or a Kalman filter), but the ultimate goal is to use the full capacities of the knowledge graph i.e. exploit all the relationships, events and attributes in the maritime surveillance ecosystem to perform better link predictions. Position prediction is just a reduction of this task to test knowledge graphs capabilities on MSA. As we could not find a method handling both time and attributes that can be tested on our data, the attribute *:location* is replaced by a relation *:isLocatedIn* between a vessel and an area.

6.1 Dataset

In the absence of publicly available maritime knowledge graph, we created our own in order to evaluate the models.

AIS data. The dataset used in our experiments is based on real maritime data: AIS messages transmitted by vessels. AIS is a short range (37-74km) ship-to-ship and ship-to-shore navigational data exchange system. It is currently

¹<https://github.com/rstriv/Know-Evolve>

²<https://github.com/thunlp/OpenKE>

Dataset	#Vessels	#Areas	#Events	Train	Test
Gibraltar 1M	2,545	1,556	955k	720k	235k

Table 2: Dataset composition

the main source of information available in support of maritime surveillance. The satellite version of AIS (S-AIS) gives a broader range (~5000km) but the transmissions are less regular and more subject to signal collision [2]. AIS provides the following non-exhaustive list of information about ships:

- the unique identifier of the vessel (called MMSI),
- its longitude/latitude,
- its speed and course,
- the timestamp of the report,
- the type of ship,
- the destination.

AIS to KG. A knowledge graph can be built using these AIS messages, where vessels are entities with attributes. Other entities can be added like nations (flag of the ship) or ports. However, the reviewed methods can only handle time, not attributes, hence the need to consider attributes as entities. In our work, the focus is on the evolution of the positions of vessels. Positions being continuous values, they need to be discretized to be casted as entities in the graph. Therefore, the studied area is converted into a grid made of $1\text{km} \times 1\text{km}$ squares and each square is an entity (further referred to as "areas").

Moreover, AIS messages are on average received every three minutes so it can be a reasonable choice to separate each time point by three minutes, instead of having a time point every second as it happens in the data (different events can be attached to the same time point). Finally, as the chosen models can not always handle entities or relations not encountered during the training phase, the test set is filtered to remove any event involving an entity or relation not present in the train set. Note that only one relation type is considered here: "vessel :isLocatedIn area" ($|R| = 1$) and each event is represented by a quadruple $(e^s, r, e^o, t) \in KG^R$.

To summarize, we build the knowledge graph consisting of entities = {vessels, areas} and relation = {isLocatedIn} over one month, we divide it into train/test sets and run the methods to predict the relation "isLocatedIn" between vessels and areas.

The dataset covers the Gibraltar Strait from February 2nd, 2017 to March 2nd, 2017 and the test set is comprised of the eight last day of the studied period. It means that we are predicting positions at time t given all positions at times $< t$. More information is given in Table 2 and Figure 3 illustrates the trajectories recorded during the considered period.

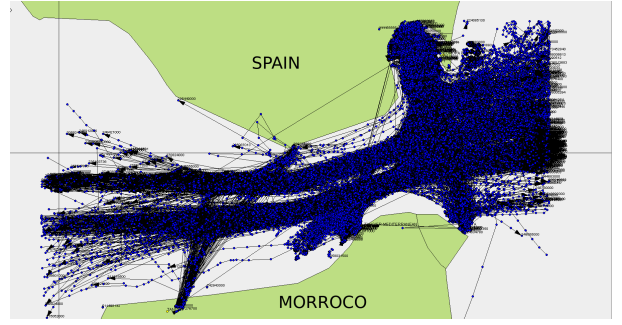


Figure 3: One week of maritime traffic in the Gibraltar Strait (best viewed in color)

6.2 Evaluation task

Link prediction. The evaluation is performed on the link prediction task: given a quadruple (e^s, r, e^o, t) , e^o is replaced by every possible entity and the resulting quadruple is evaluated by the model. All the quadruples are then ranked in descending order of plausibility and we record the Mean Average Rank (MAR) and the @Hits10 measure (one of the 10 best ranked quadruples is the true one). A lower rank means that the quadruple is classified better (the best rank being 1 and the worst the number of entities) and @Hits10 is expressed in percentage of correctly ranked quadruples i.e. higher is better. The filtering method of TransE [9] is applied, i.e. the quadruple is not ranked against corrupted quadruples that are true.

Sliding window evaluation. The performance is tested using the sliding window evaluation from Know-Evolve. We divide the test set into 8 different slides, each slide including one day of time (Know-Evolve uses 12 slides of two weeks each). This method is said to "help to realize the effect of modeling temporal and evolutionary knowledge" [17].

Static method on dynamic data. As it is a static method, the evaluation of TransE required some modifications of the dataset. All the timestamps t are removed and as a result, multiple occurrences of the same triples (e^s, r, e^o) appear. Those are removed in order to have a unique representant for each triple and the dataset is then comprised of 102,470 (train) and 16,807 (test) events. The test set still only contains entities seen during training.

6.3 Results

Experimental settings. We used the settings reported in [17] to run Know-Evolve. For TransE, we set batch size=200, learning rate = 0.001 and embedding dimension = 64.

Quantitative Analysis. Figure 4 show the results of the reviewed models over the Gibraltar1M dataset. Know-Evolve, being a temporal model, performs way better than TransE which struggle in @Hits10 prediction despite being not so far from Know-Evolve in Mean Rank. The reason is

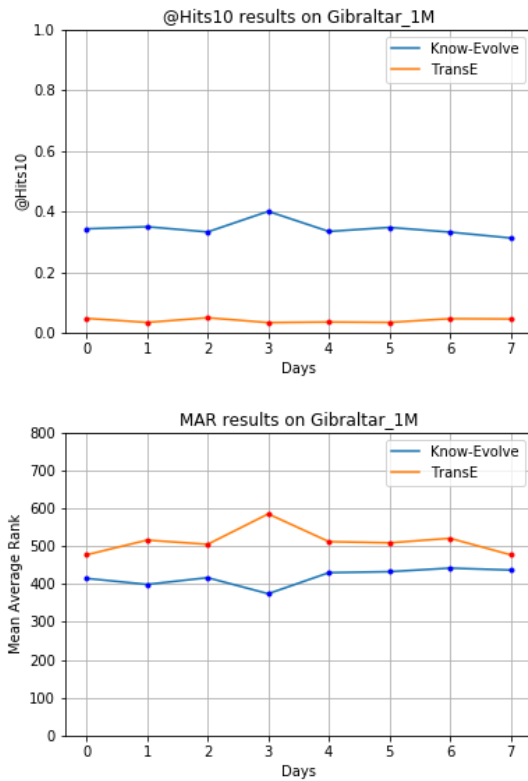


Figure 4: @Hits10 and MAR results of tested models

that TransE depend only on static entity embeddings to perform prediction. With an average @Hits10 of 34%, Know-Evolve captured the relationship between the vessels and the areas better than TransE but do not excels at the task.

Contextual analysis. This evaluation was made on a single task: predict in which area a vessel will be next. This task is made harder by the discretization of the positions: areas are independant and the graph does not tell if areas are close to each other or not. The only way to extract proximity is the analysis of a vessel's track (the succession of relations with area entities), meaning that two areas having a relation with a vessel in a short timespan may be close. More, a proximity relationship between two vessels in the same area could not be established because areas are too wide to consider two vessels as close (e.g. enough to perform an exchange of goods). At last, areas not seen in training cannot be predicted as next location due to the limitations of TransE. Know-Evolve somehow managed to find some connections between vessels and areas but the results are very unsatisfactory: a Mean Rank of 400 means that the correct area is on average ranked 400th, against $MR = 20$ on ICEWS [17]. Despite the difficulty induced by the discretization, position prediction is a simple task and the models performed poorly: they are not adequated to address this problem. The use of positions as continuous attributes could solve the abovementioned issues and

improve the results on position prediction with knowledge graphs.

7 Conclusion and future work

In this article, we reviewed two link prediction techniques for a task: the evolution of the positions of vessels using a dynamic knowledge graph for Maritime Situational Awareness. We showed that relational data (KG^R) is not sufficient for modelling the movement of a vessel and that attributional information should be used (KG^A). We also exhibited the challenges that need to be overcome to apply DAKGs on MSA, and formalized the relation and attribute value prediction problem.

We foresee several tasks for future work: (1) make the prediction task more realistic by adding more entity and relation types in the dataset, such as ships going in and out of ports, or encounters between ships, (2) find a model that can handle both KG^R and KG^A for link and attribute prediction in a temporal setting, (3) perform threat and/or anomaly detection on DAKGs. These are the three requirements to fully evaluate the use of DAKGs on operational maritime data.

8 Acknowledgements

This work was partially supported by the French National Association for Research and Technology and by Airbus Defence and Space.

References

- [1] North Atlantic Treaty Organisation, "MC MSA draft definition", 2007
- [2] N. Le Guillarme and X. Lerouvreux, "Unsupervised Extraction of Knowledge from S-AIS Data For Maritime Situational Awareness", *FUSION*, 2013
- [3] M. Riveiro, G. Pallotta and M. Vespe, "Maritime anomaly detection: A review", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2018
- [4] S. Mascaro, A. E. Nicholso, and K. B. Korb, "Anomaly detection in vessel tracks using Bayesian networks", *International Journal of Approximate Reasoning*, vol. 55, no. 1, pp. 84–98, 2014.
- [5] M. Riveiro, F. Johansson, G. Falkman, and T. Ziemke, "Supporting maritime situation awareness using self-organizing maps and gaussian mixture models", *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, vol. 173, p. 84, 2008
- [6] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, G.-B. Huang, "Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey", *CoRR*, abs/1606.00981, 2016

- [7] D. Zissis, E. K. Xidias, and D. Lekkas, "Real-time vessel behavior prediction", *Evolving Systems*, vol. 7, no. 1, pp. 29-40, 2016
- [8] L. Perera, C. Guedes Soares, "Ocean Vessel Trajectory Estimation and Prediction Based on Extended Kalman Filter", *2nd International Conference on Adaptive and Self-adaptive Systems and Applications*, 2010
- [9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data", *NIPS*, 2013
- [10] M. Nickel, V. Tresp, and H.-P. Kriegel, "A Three-Way Model for Collective Learning on Multi-Relational Data", *ICML*, 2011
- [11] M. Nickel, L. Rosasco, and T. Poggio, "Holographic Embeddings of Knowledge Graphs", *AAAI*, 2016
- [12] H. Liu, Y. Wu, and Y. Yang, "Analogical Inference for Multi-relational Embeddings", *ICML*, 2017
- [13] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning With Neural Tensor Networks for Knowledge Base Completion", *NIPS*, 2013
- [14] T. N. Kipf and M. Welling, "Variational Graph Auto-Encoders", *NIPS Workshop on Bayesian Deep Learning*, 2016
- [15] J. Leblay and M. W. Chekol, "Deriving Validity Time in Knowledge Graph", in *Companion of the The Web Conference 2018 on The Web Conference*, pp. 1771-1776, 2018
- [16] C. Esteban, V. Tresp, Y. Yang, S. Baier, and D. Krompaß, "Predicting the Co-Evolution of Event and Knowledge Graphs", *19th International Conference on Information Fusion*, 2016.
- [17] R. Trivedi, H. Dai, Y. Wang, and L. Song, "Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs", *ICML*, 2017
- [18] T. Jiang et al., "Towards Time-Aware Knowledge Graph Completion", *International Conference on Computational Linguistics*, 2016
- [19] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "Dynamic Graph Representation Learning via Self-Attention Networks", *arXiv:1812.09430*, 2018
- [20] Y. Lin, Z. Liu, and M. Sun, "Knowledge Representation Learning with Entities, Attributes and Relations", *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 2016
- [21] Y. Tay, L. A. Tuan, M. C. Phan, and S. C. Hui, "Multi-Task Neural Network for Non-discrete Attribute Prediction in Knowledge Graphs", *CIKM*, 2017
- [22] J. Li, K. Cheng, L. Wu, and H. Liu, "Streaming Link Prediction on Dynamic Attributed Networks", in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM, Marina Del Rey, CA, USA*, pp. 369-377, 2018
- [23] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia", *Artificial Intelligence*, vol. 194, pp. 28-61, 2013
- [24] E. Boschee, J. Lautenschlager, S. O'Brien, S. Shellman, J. Starz, M. Ward, "ICEWS Coded Event Data", 2015
- [25] K. Leetaru and P. A. Schrod, "GDELT: Global Data on Events, Location and Tone", *ISA Annual Convention*, 2013
- [26] J. Roy, "Anomaly detection in the maritime domain", *Proceedings of SPIE - The International Society for Optical Engineering*, 2008
- [27] P. Zhao, C. Aggarwal, and G. He, "Link prediction in graph streams", in *IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, Finland*, pp. 553-564, 2016
- [28] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models", in *International Conference on Machine Learning*, pp. 1278-1286, 2014
- [29] M. W. Chekol, G. Pirrò, J. Schoenfish and H. Stuckenschmidt, "Marrying Uncertainty and Time in Knowledge Graphs", *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017
- [30] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)", *IEEE Access*, vol. 6, pp. 52138-52160, 2018
- [31] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "Representation Learning over Dynamic Graphs", *To be published in ICLR*, 2019
- [32] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge Graph Embedding: A Survey of Approaches and Applications", *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724-2743, 2017
- [33] X. Han et al., "OpenKE: An Open Toolkit for Knowledge Embedding", *EMNLP*, 2018