Embedding Taxonomical, Situational or Sequential Knowledge Graph Context for Recommendation Tasks

Simon Werner[1], Achim Rettinger[1], Lavdim Halilaj[2], and Jürgen Lüttin[2]

[1] Trier University, Trier, Germany
{werners, rettinger}@uni-trier.de
[2] Bosch Research, Renningen, Germany
{lavdim.halilaj, juergen.luettin}@de.bosch.com

Abstract. Learned latent vector representations are key to the success of many recommender systems in recent years. However, traditional approaches like matrix factorization produce vector representations that capture global distributions of a static recommendation scenario only. Such latent user or item representations do not capture background knowledge and are not customized to a concrete situational context and the sequential history of events leading up to it.

This is a fundamentally limiting restriction for many tasks and applications, since the latent state can depend on a) abstract background information, b) the current situational context and c) the history of related observations. An illustrating example is a restaurant recommendation scenario, where a user's assessment of the situation depends a) on taxonomical information regarding the type of cuisine, b) on situational factors like time of day, weather or location and c) on the subjective individual history and experience of this user in preceding situations. This situation-specific internal state of the user is not captured when using a traditional collaborative filtering approach, since background knowledge, the situational context and the sequential nature of an individual's history cannot easily be represented in the matrix.

In this paper, we investigate how well state-of-the-art approaches do exploit those different dimensions relevant to POI recommendation tasks. Naturally, we represent such a scenario as a temporal knowledge graph and compare plain knowledge graph, a taxonomy and a hypergraph embedding approach, as well as a recurrent neural network architecture to exploit the different context-dimensions of such rich information. Our empirical evidence indicates that the situational context is most crucial to the prediction performance, while the taxonomical and sequential information are harder to exploit. However, they still have their specific merits depending on the situation.

1 Introduction

Recommender systems are a mature field in research and engineering. They have been applied in many diverse applications and the approaches and data sources

used are equally diverse. Typically, specialized representation formalisms and methods are devised and optimized to exploit the specific information best. However, several applications of recommender systems in real world scenarios are faced with other challenges that should be considered in order to provide good recommendations. One factor is the consideration of context like location, time, etc. [1]. Another challenge is to deal with complex environments that are subject to greater variability and complexity of inputs to recommender systems rather than simple ratings or reviews. In some cases, the information structure that serves for recommendations is so complex that it is represented by a semantic model as a knowledge graph [28]. Similarly, some applications require more complex outputs than prioritized recommendation lists in the direction of composite or sequential recommendations.

We describe details of a concrete in-use application of a mobile location-based recommender system that makes use of semantic technologies for representing the complex information structure as well as for user information obtained from a social network. In this paper, we investigate three dimensions that provide additional information: a) symbolic background knowledge, b) situation-specific information and c) sequential information.

One illustrating example is a POI recommendation scenario, where a user's assessment of a situation depends on his preferences for a certain *type of cuisines*, situational factors like *time of day*, *weather* or *location* and on the subjective individual history and experience of this user in previous situations. For instance, a restaurant shouldn't be of interest to a user who just had something to eat.

An intuitive way of representing all this heterogeneous types of information are temporal knowledge hyper graphs that contain time-stamped hyper-edges to allow the extraction of the sequential history of previous interactions of a user in similar recommendation settings. Hereby, each concrete setting is accompanied by a list of contextual factors that are best modelled as an n-ary relation between the user and the recommendation target. Also, each entity is accompanied by symbolic background knowledge like taxonomical relations.

Knowledge Graph Embeddings (KGE) are a recent technique to transform such symbolic knowledge into predictive models, which operate on latent vector spaces. However, most current KGE methods produce exactly one embedding for each entity instance and relation type specified in a static Knowledge Graph (KG). Each embedding captures the global distributional semantic of the graph from the perspective of this entity or relation. This does not fit well to context-aware recommender systems.

In this paper, we test the hypothesis that a global KGE per entity and relation is not adequate for many recommendation tasks. Consequently, there is a need to customize static KGEs to situational and subjective contexts. More precisely we argue that most KGE models cannot generate embeddings that capture the current relational context and that contain the abstract conceptual background information as well as the subject's history of related observations.

Thus, we test different techniques to incorporate three dimensions of additional information: a) An ontology describing POIs for symbolic background

knowledge, b) n-ary relations for capturing situation-specific information and c) sequential information about an individual's previous history.

Given such a formalization of contextualized observations over time, our goal is to learn embeddings that go beyond binary pre-trained KGEs by taking into account an ontology and the sequential history of contextualizing factors. We attempt that by a hypergraph- and a taxonomy embedding technique and recurrent neural networks. We make the following contributions:

- We propose a formal ontology for modeling abstract background knowledge in recommenation scenarios (addressing dimension a) and feed it into Knowledge Graph Embedding (KGE) methods.
- We apply a hypergraph embedding approach to include the situational context (addressing dimension b).
- We model the temporal context of an individual with a recurrent neural network
- We evaluate these methods on a context aware POI recommendation task to gain insights for the individual benefits of the dimensions to the recommendation performance.

2 Related Work

In this section we first survey previous work on the task of POI recommendation. Some more recent approaches rely on knowledge graph embeddings, which we also do in this work. Consequently, we discuss the fundamentals related to this area in more detail next.

2.1 Recommender Systems for Location Based Social Networks

Context information is particularly important for location based recommender systems where context like location, time, weather, or trip purpose has a large influence on the POI to recommend. Recommender systems based on location based social networks (LBSN) have been the subject of intensive recent research activities, see [2, 35] for recent surveys. In-vehicle recommender systems provide even more context information such as vehicle sensor based information about occupants and driver, vehicle state, or surrounding traffic [19].

An early approach for POI recommendation based on models for human mobility and their dynamics in social networks is described in [7]. Another early approach for context-aware recommendation that considers social network information, personal preferences and POI popularity is presented in [33]. Nousal et al. [22] analysed simple measures such as popularity, category preference, temporal preference, social filtering, with supervised learning using linear regression model or decision trees for next place prediction. Baral et al. [4] propose a hierarchical contextual POI sequence recommender that formulates user preferences as hierarchical structure and exploits contextual trend to generate personalized POI sequences. Those works are method-wise not directly related to our approach, which is focused on knowledge graph embedding methods.

An approach presented by Baral et al. [3] describes a contextualized location sequence recommender that generates contextually coherent POI sequences relevant to user preferences exploiting recurrent neural networks (RNN) and extended Long-short term memory (LSTM) networks. A method based on matrix factorization to embed personalized Markov chains and localized regions for successive personalized POI recommendation is used in [6]. Feng et al. [10] propose a personalized ranking metric embedding method (PRME) which jointly models the sequential information and individual preferences. A fourth-order tensor factorization-based ranking methodology that captures long- and short term preferences simultaneously has been reported in [17]. We also investigate methods in this directions by using an LSTM-based approach in one of our experiments.

Even more closely related to methods investigated in this paper is a knowledge graph embedding method that learns semantic representations of both entities and paths between entities for characterizing user preferences described in [24]. Another knowledge graph embedding based approach [29] jointly captures the sequential effect, geographical influence, temporal effect and semantic effect by embedding four corresponding knowledge graphs (POI-POI, POI-Region, POI-Time and POI-Word) into a shared low-dimensional space. A state-of-the-art deep learning recommendation model has been reported in [20]. Categorical features are represented by an embedding vector, generalizing the concept of latent factors used in matrix factorization. A Spatial-Aware Hierarchical Collaborative Deep Learning model (SH-CDL) that jointly performs deep representation learning for POIs from heterogeneous features and hierarchically additive representation learning for spatial-aware personal preferences is presented in [32]. [31] propose LBSN2Vec, a hyper graph embedding approach designed specifically for LBSN data which we also use in our experiments.

2.2   Knowledge Graph Embedding

In recent years, Knowledge Graph Embedding (KGE) has been a very vibrant field in Machine Learning and Semantic Technologies, specifically in the area of Representation Learning (see [13] for a survey). Numerous methods for embedding knowledge graphs have been proposed and even more adaptations have been published. KGE methods can be roughly characterized by the representation space and the scoring function.

The vector representations of entities and relations are traditionally Euclidean $\mathbb{R}^d$, but many different spaces like Complex $\mathbb{C}^d$ (e.g., in [26]) or Hypercomplex $\mathbb{H}^d$ (cmp. [34]) have been used as well.

Standard KGE methods don't take into account temporal information or contextual factors that influence the plausibility of a fact. However, there have been attempts to address each limitation, as outlined next.

2.3 Contextual Knowledge Graph Embeddings

From the Knowledge Graph perspective, hypergraphs with n-ary relations and hyper-relational graphs with meta information encoded on the relations are exploited for modeling the context. Such approaches from Statistical Relational Learning are based on graphical models and tensor factorization [23]. A more recent approach extends the current KGE method SimplE [15] to hypergraphs [9] but does not take into account temporal or sequential information. This approach was used as the basis for our hypergraph embedding experiments. More details on our adaptions can be found in section 4.2.

Embedding temporal dynamics of a knowledge graph and thus tackling *(Lim2)* has received much attention recently. Knowledge graphs in which facts only hold within a specific period and where the evolution of facts follows a sequence have become increasingly available. This also increased the interest in learning embeddings that take the temporal information into account.

Basic approaches to temporal KGE model facts as temporal quadruples. They are optimized for scoring the plausibility of (unkown) facts at a given point in time [16], [8]. A more sophisticated approach is proposed in [18]. It even checks the temporal consistency given contextual relations of the subject and object. Besides the inability of those models to model n-ary sequential context, we are also taking a different focus by using the temporal dimension to model the history of experiences of a subject. A more entity-centric perspective is taken in [25], which attempts to model the temporal evolution of entities, where [14] take a relation-specific perspective instead. Similar to our approach, [27] proposes an LSTM-based approach, which exploits relation-specific embedding of entities.

3 Capturing Taxonomical, Contextual and Sequential Information for Recommendations

The goal of this paper is to investigate the potential of three different types of information, namely taxonomical, contextual and sequential, for their use in embedding-based recommender systems. We chose a knowledge graph as the underlying data structure, since it allows to include all those information types in one representation formalism. We first show how to model taxonomical information, before including situational context and the sequential history.

3.1 Modeling Taxonomical Information

This section describes the POI Categories (POICa) ontology used for representing information about POIs mainly by exploiting their hierarchical relationships.

Conceptualizing and Formalizing The main objective of the POICa ontology is focused on representing: 1) *taxonomic knowledge*, encoding hierarchical information between different POIs, and 2) *auxiliary knowledge*, which comprises information for a specific check-in of a user in a particular POI including

geo-spatial and temporal data, i.e. the location of the POI and the timestamp information about the check-in action. The underlying structure of the POICa ontology is built on top of Foursquare Categories[3] where the core concept is the *POI*. Several object and datatype properties describe a particular POI with respect to its attributes and relationships with other concepts.

As depicted in Figure 1, POICa ontology comprises a number of subcategories distributed in various levels, which for the sake of better readability are highlighted with different colors. The first level under the *POI* concept includes subcategories described in the following:

- *Art and Entertainment* - is the category for representing places related to art, culture, music, exhibitions, etc.
- *Education* - are entities which provide education-related services and learning environments.
- *Professional Places* - groups places which are involved or perform business activities.
- *Outdoors and Recreation* - are places where the recreation is commonly realized in natural settings.
- *Residence* - used to group POIs that mainly serves as living places.
- *Restaurant* - groups all types of restaurants split on various criteria, such as cuisine.
- *Shop and Service* - used to group POIs which are dedicated for selling goods or services.
- *Transportation* - containing POIs which enable carrying of people and goods from one place to another.

Each of these subcategories is further specialized utilizing *subClassOf* axiom in order to provide a detailed classification based on the shared characteristics, such as the type of the activity they perform combining with regional information. Several additional classes such *EthnicRestaurant*, *SiteBasedRestaurant*, *SpecializedFoodRestaurant* are introduced with the aim of grouping restaurants based on ethnicity or cuisine, style and flavour, respectively.

Alignment with and reuse of external ontologies In order to ensure interoperability with other information from different sources, we reused a number of concepts from external ontologies such as *Schema.org*, *FOAF*, *DBpedia*, *DCTerms* and *Weather*[4]. For instance, in order to represent geo-spatial information for a given POI the following concepts from *DCTerms*, *Schema.org* and *DBpedia*: *dct:Location*, *schema:PostalAddress* and *dbo:City* are reused.

The current version of the POICa ontology contains 953 classes, 8 object properties, 12 datatype properties and 4 annotation properties. In this paper, our focus was to describe the core concepts that form the basis to understand the conducted work from the taxonomic point of view.

---

[3] https://developer.foursquare.com/docs/build-with-foursquare/categories
[4] https://schema.org/,   http://xmlns.com/foaf/0.1/,   http://dbpedia.org/ontology#, http://purl.org/dc/terms/, https://cutt.ly/QhQrFxv#

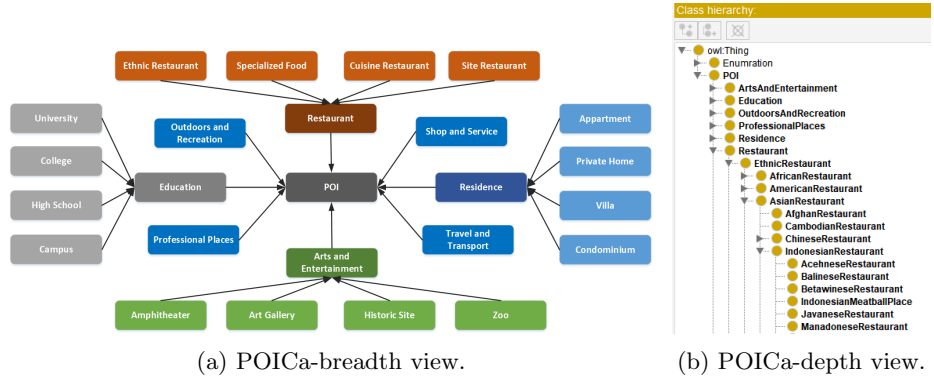(a) POICa-breadth view.   (b) POICa-depth view.

Fig. 1: **POICa ontology**. Main concepts representing different categories within the ontology: a) depicts all categories within the first level of the hierarchy including some important ones from the second level; b) depicts depth of a particular category i.e. *Restaurant*.

3.2 Context-aware Hypergraph-Embeddings

In traditional user-item-recommender systems, there is only one binary relation indicating which user interacted with which item. However, this cannot capture the multi-relational background knowledge described above and also cannot include situational context that describes the conditions when and how this interaction took place.

Thus, representing recommendation scenarios by only using binary relations can cause an information loss that might lead to poor performance on a recommendation task. To make full use of all the contextual information like day of the week and current time, that are contained in the dataset, the binary relations need to be extended to n-ary relations.

We therefore build on HypE [9], a recently introduced hypergraph embedding approach that showed promising results on other tasks and allows for easy adaption to our recommendation use-case. HypE uses a multilinear scoring function and additionally uses learnt convolutional filters to model the different importance of entities in different relations. The recommendation itself is made through computation of a score, given n entities (depending on the arity of the relation) and the relation. As an example, given a context (i.e. the weather, day, time, proximity), all potential POIs can be ranked by computing the score for each and choosing the POI with the highest score as the recommendation. The scoring function of HypE is defined as $\phi(r(e_1, ...e_{|r|}))$, and describes the sum of the element-wise product of the corresponding embedding vectors (cmp. Fig. 2).

3.3 Sequence-aware Recurrent Neural Nets

Having access to the full information and relying on a system that is constantly learning from new data is often an unrealistic assumption. Common issues are:
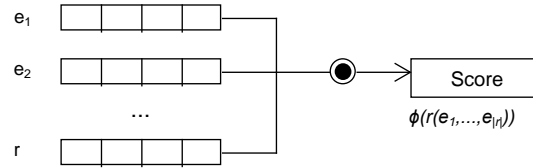
Fig. 2: HypE architecture with scoring function

.

Cold start: In many situations the system encounters a new user or can't identify the current user and thus does not have access to the user's history and preferences.

Missing context: Often the full context of the recommendation situation is not available. The system still has to produce a recommendation without contextual factors.

Online Machine Learning: Most machine learning methods learn from (mini) batches and can't be re-trained after each new data point arrives.

A more realistic scenario is that a large data set of a LBSN is available for off-line training, but recommendations still have to be generated for new users without contextual information. Based on the assumption that a user chooses a POI not only based on contextual information, but also based on the last POI he visited, a personalized recommendation might even be possible for short individual histories. For example, this captures that a user would typically not visit a restaurant right after returning from lunch and therefore should also not be recommended doing so.

To capture the sequential nature of such a scenario, we propose to use an LSTM network [12] that receives a sequence of check-ins without additional contextual information as input and predicts the next location in the sequence. We use the off-line trained HypE-Embeddings of the locations as our POI representations and minimize a Cross-Entropy-Loss to learn the next location in the sequence. As a proof-of-concept, we chose a simple network architecture, using an LSTM layer for modelling the sequential information followed by a fully connected layer for the prediction (see Fig. 3).

4   Experimental Setup and Results

The following section describes the experiments on POI recommendation based on the knowledge graphs described in the previous sections. We use two data sets that were introduced in a different knowledge graph embedding approach [31] for a POI recommendation scenario and use the reported hit@10 value from the same paper as a baseline to compare our results to. The experiments can be divided into three different sets of runs:

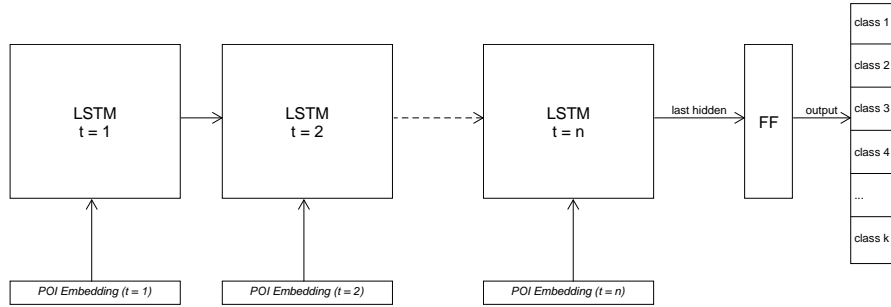– Prediction based on binary KGE approaches [11]

Fig. 3: The architecture of the LSTM approach
.

- Prediction based on a hypergraph approach [9]
- Prediction based on sequential modelling

The first type of experiments were run on existing implementations[5] that were adapted for a more convenient usage without altering the core of the implementations. For the third set of experiments, we used a simple LSTM network that receives a sequence of visited POIs as input and outputs a prediction of the next POI in the sequence. The combined approach was built using HypE[6]. Our implementation and experimental settings can be found in the repository[7] on GitHub.

The datasets in use consist of 104,997 and 376,077 data points, which represent the check-ins at locations in New York City and Jakarta over the course of two years. The larger Jakarta set contains 8,805 distinct POIs and 6,183 distinct users, while the NYC set contains 3,626 distinct POIs and 3,573 users. Since the original data represents a hypergraph, it had to be adjusted for usage with binary relations. The information loss in this procedure led to smaller datasets for the binary KGE approaches in comparison to the hypergraph approach. To make sure that the results are still comparable, splitting the data into test, validation and training set was the first step in data preparation, before the data was prepared for usage in the different settings.

As the results, we report the 'filtered' values for the binary and 'raw' values for the n-ary approaches. The filtered setting counts a "hit" as long as the the predicted value is an element of the ground truth, whereas the raw setting only considers the current sample value as a true result. In the third case (LSTM), we report the raw setting only, because we only want to model the sequential behaviour and therefore only consider a "hit" when the exact POI for this sequence is recommended.

---

[5] https://github.com/thunlp/OpenKE

[6] https://github.com/ElementAI/HypE

[7] https://github.com/siwer/TaxonomicalKGE

4.1   Binary knowledge graph-embedding approaches

The first task was limited to represent the data as triples, consisting of subject, predicate and object. As there is no relation information that we can directly take from the original data, we introduced two different relations which we considered to be carrying most information. The first relation is checksIn(user, POI) and the second one is typeOf(POI, category). For the setting that incorporates the ontological data, we introduced an additional relation subclassOf(category, category) which is only present in the training data and is meant to provide further information for the recommendation task. Based on the available implementations we conducted a series of experiments using a large variety of binary KGE approaches, including Complex[26], Distmult[30], Hole[21], Simple[15] and Transe[5]. As for the parameter settings, we tested across different embedding dimensions and left the other options to default values. We only report the best results for each method.

4.2   Knowledge hypergraph embedding

In preparation for the HypE approach, we defined one relation checksIn(user, hour of day, day of week, type, location) to represent the data. The hour of day and the day of week are derived from the timestamps in the original data. To achieve the results presented in table 2, we used a slightly different implementation of the HypE approach. The scoring function including the convolutions is still the same, but we made a few adaptions for faster runs on our dataset. We also slightly altered the training objective; instead of scoring against a fixed number of negative samples, we always scored against all possible locations. We only consider the 'raw' setting for evaluation.

   For integration of background knowledge we implemented a model that combines the HypE approach for n-ary relations together with a binary approach (TransE) to embed the ontological information. The underlying idea is that the ontological information (in this case the POI categories) will be embedded in their own ontology space, while the other information (users, locations, etc.) will be embedded in a separate space. A translation layer (implemented as a feed-forward layer) learns to project from the ontology space to the general feature space. Algorithm 1 below shows the training procedure of our approach. We implemented a hyperparamer λ to control the influence of the ontological information during training. The training objectives are now to predict the location given (user, type, time, day) and to predict the superclass of a type given the provided ontology. For evaluation we still only consider the location prediction task. Across all experimental runs in different configurations, the results with $\lambda > 0$ outperformed the ones where $\lambda = 0$. Table 1 shows an example of the influence of λ on the training for both NYC and JAK data. The results shown there are averaged over runs with varying ontology space dimension (130,75,50,25). The general entity space dimension is fixed at 130 over those runs. As indicated by our empirical results, the most beneficial values for λ lie between 0.2 and 0.8. This behaviour is also consistent across the other observed metrics. In table 2,

the '+ Ont' approaches denote $\lambda > 0$ for the HypE approach. As with the binary approaches, we also present the results from the best runs.

Lambda	0.0	0.2	0.4	0.6	0.8	1.0	Params
MR	23.66	**20.67**	21.76	20.95	21.06	20.90	NYC, lr = 0.01
MR	21.48	20.25	20.24	20.38	**19.98**	20.07	NYC, lr = 0.005
MR	21.73	20.52	19.60	20.18	**19.55**	20.21	JAK, lr = 0.01
MR	17.22	16.43	**15.76**	15.93	16.05	15.88	JAK, lr = 0.005

Table 1: Influence of Lambda on MR

Algorithm 1: Training of the combined approach

input : batch = $\{checkinRelation, subClassRelation, user, time, day, type\}, targets = \{location, superType\}, weightFactor = \lambda, Parameters = \Theta$

typeVector = $ontologySpace_\Theta(type)$;
entityVectors = $entitySpace_\Theta(user, time, day)$;
relationVectors = $relationSpace_\Theta(checkinRelation, subClassRelation)$;
for *all i in types* **do**
  $\mid$ ontologyScores $\leftarrow \phi_{transE}$ (type, subClassOf, types$_i$)
end
translatedType = TranslationLayer$_\Theta$(type);
for *all i in locations* **do**
  $\mid$ locScores $\leftarrow \phi_{hypE}$ (checkin, user, translatedType, day, time,
  $\mid$ locations$_i$);
end
lossOntology = $crossEntropy(ontologyScores, superType)$;
lossLocation = $crossEntropy(locScores, location)$;
combinedLoss = $lossLocation + (\lambda * lossOntology)$;
$\Theta \leftarrow update(\Theta, backProp(combinedLoss))$;

4.3 LSTM-based Sequence-aware Recommendations

As the basis for experiments with the LSTM network, we use the location embeddings that were acquired in the experiments from the section above. Thus, some global contextual information is captured in the embeddings, however, the LSTM is not aware of any situational context, nor of the personalized history of the user, beyond a few previous check-ins. We chose the best performing HypE models for both datasets to provide the location representations.

Since sequential information is used, the original data had to be transformed to represent the check-in sequence(s) of a user. The extreme case would be assuming one sequence per user, i.e. taking all interactions of one user and transform

Approach	Jakarta	Jakarta + Ont	EmbDim	NYC	NYC + Ont	EmbDim
LBSN2Vec [31]	0.08	-	128	0.11	-	128
Complex	0.041	0.040	100	0.033	0.032	100
Distmult	0.045	0.045	150	0.035	0.039	100
HolE	0.031	0.030	100	0.024	0.023	100
SimplE	0.047	0.046	100	0.035	0.036	100
TransE	0.064	0.064	200	0.044	0.045	150
HypE	**0.742**	**0.771**	130/50	**0.722**	**0.738**	130/25
LSTM	0.085	-	100	0.080	-	100

Table 2: Best hits@10 results for different approaches with the corresponding embedding dimensions. For the KGE approaches, also results for adding information modeled in the POICa ontology (see Sec. 3.1) are reported.

it into a discrete sequence of check-ins. This, however, is not an assumption that would reflect real-word behaviour, because it is unlikely that a location which a user visited a month ago would influence a decision of today. To capture this, we assumed a new sequence after 6 hours passed between two check-ins. As a result, there are now 12,781 sequences in the NYC training set and 1,605 sequences in the NYC test set (For Jakarta: 56,670 and 5,319). Therefore, we consider at least two check-ins within a 6 hour window as a sequence. The choice of the duration after which a new sequence is assumed has a large influence on the final training data. A window of 24 hours would lead to fewer, but longer sequences, while a 4 hour window would yield more very short sequences. To ensure the relatedness of check-ins in the sequences, a shorter window is favorable, although at the cost of having shorter sequences. In the end, around 70% of the obtained sequences had a length of 2. Since we are interested in testing the performance also for cold start problems this is a suitable setup.

We modelled the neural network architecture as a classification problem, where the last hidden state of the LSTM is used as the input for the classification feed-forward layer. Due to the different dataset sizes, the NYC set has 3,626 classes (distinct locations) and the Jakarta set has 8,805 classes.

4.4   Discussion of results

Table 2 provides an overview of the best hits@10 results for each setup described above. First, the results reported in [31] are shown for comparison. Then, results of all binary KGE methods are reported and compared to when information from the POICa ontology is added. Throughout all experiments, the ontological information didn't make a significant difference. This is likely due to the naive way of introducing just the relational information from the ontology into the graph, without considering their semantics, like that of a taxonomical relation. Apparently, this adds more complexity than it provides valuable learning signals. We assume that a more sophisticated approach to exploit the ontology, as done in the HypE approach, can improve the results considerably.

All binary KGE approaches clearly show an inferior performance to LBSN2Vec. This is likely due to their inability to exploit contextual information. This observation becomes clear when looking at the HypE results. Like LBSN2Vec, HypE does exploit n-ary relation and thus the full situational context, however, their embedding techniques are fundamentally different. HypE's results are a quantum leap when compared to any other approach we tested. Since HypE is based on years of KGE research and optimized for use-cases with rich situational context an improvement was expected, but this extend was still surprising. As opposed to the naive approach of just adding the taxonomical information to the training data, the approach of jointly training embeddings for the prediction task and the ontology yielded a measurable increase in prediction performance.

Finally, the LSTM results based on sequential information show that its performance is below LBSN2Vec, specifically for the NYC data set. It is still noteworthy that such a result is obtained after only seeing one previous POI check-in without additional user-specific or contextual information. On the one hand, this seems reasonable since the POI embeddings from HypE are used as input and thus some global context of each POI is provided to the LSTM. On the other hand, there seems to be a valuable signal in the previously visited POI, that is not exploited by the other methods.

5   Conclusions and Future Work

In this paper, we obtained empirical evidence for how well state-of-the-art latent recommendation approaches can exploit ontological, situational and sequential information in a POI recommendation task. Our empirical evidence indicates that the situational context is most crucial to the prediction performance, while the taxonomical and sequential information are harder to exploit. As we have shown with the experiments based on HypE, a beneficial exploitation of ontological information requires a more sophisticated approach than just augmenting the knowledge graph with relations from the ontology. In our approach, we learn an additional dedicated ontology embedding space and train a translation layer to fuse both spaces. Besides of our approach, materializing implicit knowledge or deducing additional positive and negative training data might be another step in this direction. The LSTM approach seems to be an interesting option for cold start scenarios or whenever online learning is computationally not feasible. Also, this approach only initially requires KGE embeddings trained on the full information. Then it can be trained on sequence information only, without situational context, and applied to novel sequences of unknown users, again without situational context.

Summing up, this work shows that the different dimensions each provide separate benefits, but exploiting all of them is non-trivial. Thus, promising future steps with great potential are methods for a tight integration of expressive formal ontologies with latent machine learning as well as deep learning architectures for a joined embedding of multi-ary knowledge graphs with sequential information.

References

1. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Proceedings of the ACM Conference on Recommender Systems, RecSys. pp. 335–336 (2008)
2. Bao, J., Zheng, Y., Wilkie, D., Mokbel, M.F.: Recommendations in location-based social networks: a survey. GeoInformatica **19**, 525–565 (2015)
3. Baral, R., Iyengar, S.S., Li, T., Balakrishnan, N.: Close: Contextualized location sequence recommender. In: Proceedings of the 12th ACM Conf. on Recommender Systems. p. 470–474. RecSys '18, Association for Computing Machinery (2018)
4. Baral, R., Iyengar, S.S., Li, T., Zhu, X.: Hicaps: Hierarchical contextual poi sequence recommender. In: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. p. 436–439. SIGSPATIAL '18, Association for Computing Machinery (2018)
5. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. p. 2787–2795. NIPS'13, Curran Associates Inc., Red Hook, NY, USA (2013)
6. Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: Successive point-of-interest recommendation. In: IJCAI (2013)
7. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: KDD (2011)
8. Dasgupta, S.S., Ray, S.N., Talukdar, P.: Hyte: Hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2001–2011 (2018)
9. Fatemi, B., Taslakian, P., Vázquez, D., Poole, D.: Knowledge hypergraphs: Extending knowledge graphs beyond binary relations. CoRR **abs/1906.00137** (2019), http://arxiv.org/abs/1906.00137
10. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: IJCAI (2015)
11. Han, X., Cao, S., Xin, L., Lin, Y., Liu, Z., Sun, M., Li, J.: Openke: An open toolkit for knowledge embedding. In: Proceedings of EMNLP (2018)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
13. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: Representation, acquisition and applications. preprint arXiv:2002.00388 (2020)
14. Jiang, T., Liu, T., Ge, T., Sha, L., Chang, B., Li, S., Sui, Z.: Towards time-aware knowledge graph completion. In: Proceedings of COLING, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 1715–1724 (2016)
15. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS 2018. pp. 4289–4300 (2018)
16. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: Companion Proceedings of the The Web Conference 2018. pp. 1771–1776 (2018)
17. Li, X., Jiang, M., Hong, H., Liao, L.: A time-aware personalized point-of-interest recommendation via high-order tensor factorization. ACM Transactions on Information Systems (TOIS) **35**, 1 – 23 (2017)
18. Liu, Y., Hua, W., Xin, K., Zhou, X.: Context-aware temporal knowledge graph embedding. In: International Conference on Web Information Systems Engineering. pp. 583–598. Springer (2019)

19. Luettin, J., Rothermel, S., Andrew, M.: Future of in-vehicle recommendation systems @ bosch. Proceedings of the 13th ACM Conf. on Recommender Sys. (2019)
20. Naumov, M., Mudigere, D., Shi, H.J.M., Huang, J., Sundaraman, N., Park, J., Wang, X., Gupta, U., Wu, C.J., Azzolini, A.G., Dzhulgakov, D., Mallevich, A., Cherniavskii, I., Lu, Y., Krishnamoorthi, R., Yu, A., Kondratenko, V., Pereira, S., Chen, X., Chen, W., Rao, V., Jia, B., Xiong, L., Smelyanskiy, M.: Deep learning recommendation model for personalization and recommendation systems (2019)
21. Nickel, M., Rosasco, L., Poggio, T.A.: Holographic embeddings of knowledge graphs. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 1955–1961. AAAI Press (2016)
22. Noulas, A., Scellato, S., Lathia, N., Mascolo, C.: Mining user mobility features for next place prediction in location-based services. 2012 IEEE 12th International Conference on Data Mining pp. 1038–1043 (2012)
23. Rettinger, A., Wermser, H., Huang, Y., Tresp, V.: Context-aware tensor decomposition for relation prediction in social networks. Social Network Analysis and Mining **2**(4), 373–385 (2012)
24. Sun, Z., Yang, J., Zhang, J., Bozzon, A., Huang, L.K., Xu, C.: Recurrent knowledge graph embedding for effective recommendation. Proceedings of the 12th ACM Conference on Recommender Systems (2018)
25. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 3462–3471. JMLR. org (2017)
26. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Balcan, M., Weinberger, K.Q. (eds.) Proceedings of the 33nd International Conference on Machine Learning, ICML 2016. vol. 48, pp. 2071–2080. JMLR.org (2016)
27. Wang, H., Kulkarni, V., Wang, W.Y.: Dolores: deep contextualized knowledge graph embeddings. arXiv preprint arXiv:1811.00147 (2018)
28. Wang, H., Zhao, M., Xie, X., Li, W., Guo, M.: Knowledge graph convolutional networks for recommender systems. The World Wide Web Conference (2019)
29. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (2016)
30. Yang, B., tau Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases (2014)
31. Yang, D., Qu, B., Yang, J., Cudre-Mauroux, P.: Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach. In: The World Wide Web Conference. p. 2147–2157. Association for Computing Machinery (2019)
32. Yin, H., Wang, W., Wang, H., Chen, L., Zhou, X.: Spatial-aware hierarchical collaborative deep learning for poi recommendation. IEEE Transactions on Knowledge and Data Engineering **29**, 2537–2551 (2017)
33. Ying, J.J.C., Lu, E.H.C., Kuo, W.N., Tseng, V.S.: Urban point-of-interest recommendation by mining user check-in behaviors. In: UrbComp '12 (2012)
34. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. In: Advances in Neural Information Processing Systems. pp. 2731–2741 (2019)
35. Zhao, S., King, I., Lyu, M.R.: A survey of point-of-interest recommendation in location-based social networks. CoRR **abs/1607.00647** (2016)