Deep Learning for Dynamic Representations of Real-World Entities

by

Simon Werner

March 2024

Dissertation zur Erlangung des Doktorgrades (Dr. phil.)
Universität Trier
Fachbereich II

1. Gutachter: Prof. Dr. Achim Rettinger (Universität Trier)
2. Gutachter: Prof. Dr. Michael Färber (TU Dresden)

Abstract

Representation Learning techniques play a crucial role in a wide variety of Deep Learning applications. From Language Generation to Link Prediction on Graphs, learned numerical vector representations often build the foundation for numerous downstream tasks. In Natural Language Processing, word embeddings are contextualized and depend on their current context. This useful property reflects how words can have different meanings based on their neighboring words. In Knowledge Graph Embedding (KGE) approaches, static vector representations are still the dominant approach. While this is sufficient for applications where the underlying Knowledge Graph (KG) mainly stores static information, it becomes a disadvantage when dynamic entity behavior needs to be modelled.

To address this issue, KGE approaches would need to model dynamic entities by incorporating situational and sequential context into the vector representations of entities. Analogous to contextualised word embeddings, this would allow entity embeddings to change depending on their history and current situational factors. Therefore, this thesis provides a description of how to transform static KGE approaches to contextualised dynamic approaches and how the specific characteristics of different dynamic scenarios are need to be taken into consideration.

As a starting point, we conduct empirical studies that attempt to integrate sequential and situational context into static KG embeddings and investigate the limitations of the different approaches. In a second step, the identified limitations serve as guidance for developing a framework that enables KG embeddings to become truly dynamic, taking into account both the current situation and the past interactions of an entity. The two main contributions in this step are the introduction of the *temporally contextualized Knowledge Graph* formalism and the corresponding *RETRA* framework which realizes the contextualisation of entity embeddings. Finally, we demonstrate how situational contextualisation can be realized even in static environments, where all object entities are passive at all times. For this, we introduce a novel task that requires the combination of multiple context modalities and their integration with a KG based view on entity behavior.

Zusammenfassung

Techniken des *Representation Learnings* sind maßgeblich für die Funktion zahlreicher Anwendungen des *Deep Learnings*. Als Repräsentationen werden Abbildungen von realweltlichen Konzepten oder Dingen in den Vektorraum verstanden, welche als Eingangssignal für tiefe neuronale Netzwerke verwendet werden. Ein Beispiel sind sogenannte Wordvektoren, die als numerische Repräsentation von Worten in Techniken des *Natural Language Processing* Anwendung finden. Diese Repräsentationen sind kontextabhängig und können wie auch die zugrundeliegenden Wörter selbst verschiedene Bedeutungen in Abhängigkeit der benachbarten Wörter tragen. Auf der Ebene der numerischen Repräsentationen von Entitäten aus Wissensgraphen hingegen überwiegt der Ansatz statischer Vektorrepräsentationen. Im Hinblick auf die Speicherung statischer Informationen in Wissensgraphen ist diese Paradigma ausreichend, gerät jedoch an seine Grenzen sobald dynamische Situationen modelliert werden.

Diese Dissertation befasst sich damit, Techniken und Formalismen zu entwickeln, die es ermöglichen, kontextabhängige Vektorrepräsentationen auch für wissensgraphbasierte Anwendungen zu verwenden. Ziel ist es, mit diesen *dynamischen* oder kontextabhängigen Vektorrepräsentationen dynamische Situationen realweltlicher Entitäten zu modellieren. Im Rahmen dieser Arbeit umfasst dies Nutzer in sozialen Netzwerken, Fahrzeuge im Straßenverkehr, geopolitische Entitäten und Teilnehmer in einem Wahrnehmungsexperiment.

In diesem Rahmen wird auf Grundlage prädiktiver Fragestellungen (z.B. Welche Lokalität wird von einem Nutzer als nächstes aufgesucht?) empirisch aufgezeigt, dass die beschriebenen neuartigen Methoden den bekannten statischen Vektorrepräsentationen hinsichtlich der Modellierung dynamischen Verhaltens überlegen sind. Hierzu werden zunächst anhand empirischer Experimente die Grenzen statischer Verfahren aufgezeigt. Aufbauend hierauf wird der *temporally contextualized Knowledge Graph* Formalismus und das dazugehörige Verfahren zum *Embedding* (d.h. der Abbildung in den Vektorraum) namens *RETRA* entworfen, mit dem die geforderte Kontextualisierung der Vektorrepräsentation von Entitäten aus Wissensgraphen realisiert wird. Anschließend wird im Rahmen einer weiteren empirischen

Studie aufgezeigt, wie sich eine *statische Welt* auf das Konzept der situativen Kontextualisierung auswirkt und mit welchen Maßnahmen dem begegnet werden kann. Hierzu wird ein neuer *Task* eingeführt, für dessen Lösung es notwendig ist, verschiedene Kontextmodalitäten (hier Bild und Text) in eine wissensgraphbasierte Herangehensweise heranzuziehen.

Table of Content

List of Tables

List of Figures

1 Introduction

Representations of real-world entities that reflect some of their properties are key to the immense advances in machine learning during the past decade. Most machine learning tasks rely heavily on vector representations in some form. In image classification, the feature vectors on which the final classification is performed, are obtained by putting the original image through convolutional and pooling layers. In natural language processing, word vectors carry semantic and syntactic information that can be exploited in many tasks. Knowledge Graphs (KGs) provide structured information about real-world entities and the relations among them. Tasks like Knowledge Graph Completion are performed on Knowledge Graphs that are embedded into vector space. These techniques all have in common that they rely on vector representations of their respective inputs in order to solve a certain task. In the early approaches, one input entity, for instance a word, was assigned one static vector representation. This, however, quickly proved to be an insufficient representation paradigm. A single word's meaning might be dependent on the current context (i.e. neighboring words). Not only polysemic nouns carry ambiguity without context, but also function words that refer to different preceding words in a sentence.

With the rise of attention mechanisms and even more importantly, the transformer architecture, this problem could be solved for natural language processing tasks by the introduction of context-dependent word vectors. By contrast, in the domain of Knowledge Graph Embedding (KGE) mechanisms, the one entity - one vector paradigm to this day is still the most prevalent one. It is apparent that real-world entities like humans base their behavior on both intrinsic and extrinsic needs and changing circumstances. For instance, a person who is travelling might prioritize fulfilling the intrinsic need *hunger* over achieving mid-term goals like reaching the destination. An extrinsically motivated need might be seeking shelter due to upcoming rain. If such situations were to be modelled in a Knowledge Graph Completion application like a recommender system, moving from the constraint of static vector representations might be beneficial. These newly conceived entity representations should reflect the dynamics of changing needs by *memorizing* that

a certain need is fulfilled and by being able to adapt to changing environments. Thus, this thesis addresses these issues and explores how to transform static KGE approaches to contextualised dynamic approaches.

1.1 Motivation

Knowledge Graphs (KGs) represent structured knowledge about the world. They contain information about entities that can range from abstract concepts like word semantics to individual people in the real world. By defining relations between these entities, KGs provide means of modelling how entities relate to and interact with each other. KGs exist for a wide array of domains, from specific domains like the aforementioned semantic relations between words in WordNet[1] to more general world knowledge like Wikidata[2].

Traditionally, KGs contain static factual statements in form of triples, which contain two entities and the relation they form. Figure 1 shows an example of how the statement *John works at Company 1* could be represented as a triple in a KG.

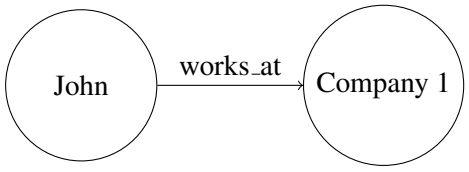

Figure 1: Knowledge graph fact

This format works well for facts that are unlikely to change, or where changes occur rarely. While it is save to assume that the statement *Mount Everest is the highest mountain on the earth* will stay true within a human conceivable time-frame, there are other facts that are more prone to change. The fact *Bonn is the capital of Germany* was true from 1949 to 1990 only, when the capital changed back to Berlin. In principal, it is possible to express this situation by introducing an additional relation *was_capital* to the KG, such that we now can express *Bonn*

[1] https://wordnet.princeton.edu/
[2] https://www.wikidata.org/

was *the capital of Germany* vs. *Berlin* ***is*** *the capital of Germany*. However, depending on the frequency of changes and whether or not it is desired to represent an order of events, adding more and more relations to express a fine granularity of changes without introducing ambiguous facts becomes a challenging problem.

A cleaner and simpler approach to this problem is the introduction of temporal information that indicates during which time a certain fact is true. The situation with changing facts could then be expressed as a Temporal Knowledge Graph (TKG) as shown in Figure 2.

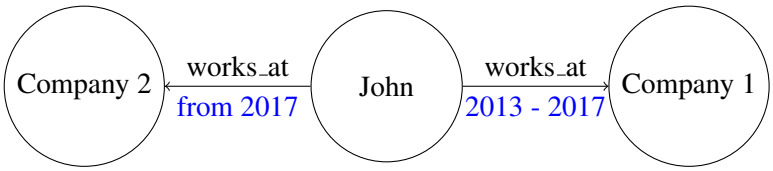

Figure 2: Temporal KG Facts

By adding temporal information to a fact, it is possible to denote the time at which the fact is valid.

Although this approach accounts for changing facts, it is not well suited for representing dynamic situations that are comprised of strongly related or even dependant events. A temporal fact can exist in isolation, while the facts that constitute a situation are dependant on each other. Consider a situation as depicted in Figure 3:

While it is possible to model this situation with tKGs in principal, some information might get lost or at least obfuscated. The situation strongly implies that the *Store* in t_2 is also the place where *Groceries* are bought in t_3. By describing the situation as a sequence of events, this connection becomes apparent, whereas a tKG can only describe a temporal proximity of the two events. Moreover, tKGs fail to show that at a given point in time t, the occurring event within a situation might be a consequence of the previous events, rather than just being correlated with a timestamp.

Another real-world phenomenon that is hard to model with existing KG-based formalisms is situational context. Situational context describes every external circumstance that might affect an entity. An example would be the weather and how

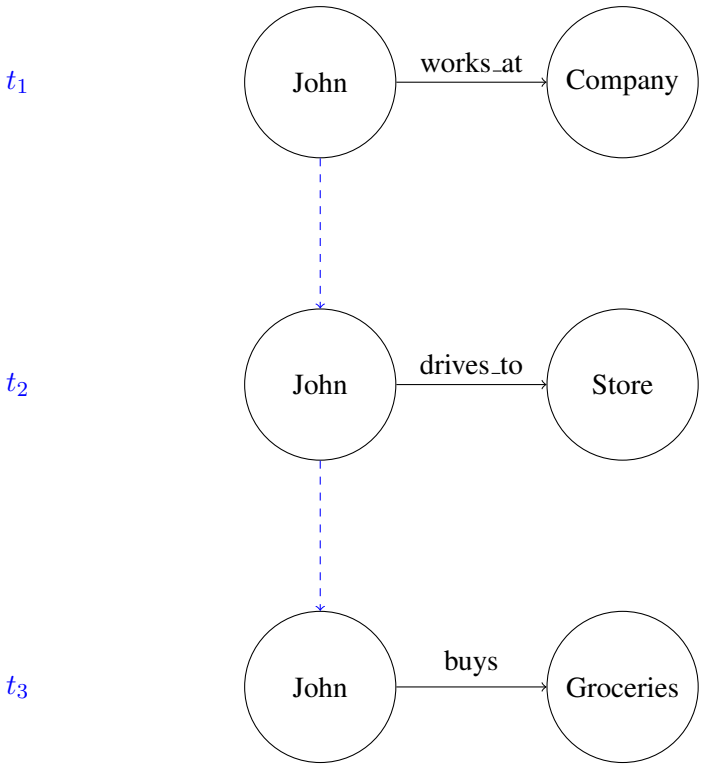

Figure 3: Dynamic situation as sequence of facts

The dashed lines connect the facts that occurred during t_n and form a sequence out of them.

it influences a situation. Consider the situations in Figure 4, where the timestamps, the individual history, the relation and the subject entity stays the same. The only differing factor is the weather, which directly affects the subject entity's decision on which activity it will pursue.

Although this situation could be modelled with hyperedges (if we assume the context to be entities), the implications are slightly different. Figure 4 shows how the situational context affects the subject entity *John* such that the object entity changes based on the context. In a hypergraph setting, the situation could be represented as by defining *trains_at* as a ternary relation. The issue with this, however, is that 1) every entity within the relation is treated the same and 2) no directional effect can be modelled. The special role of the context as something that directly

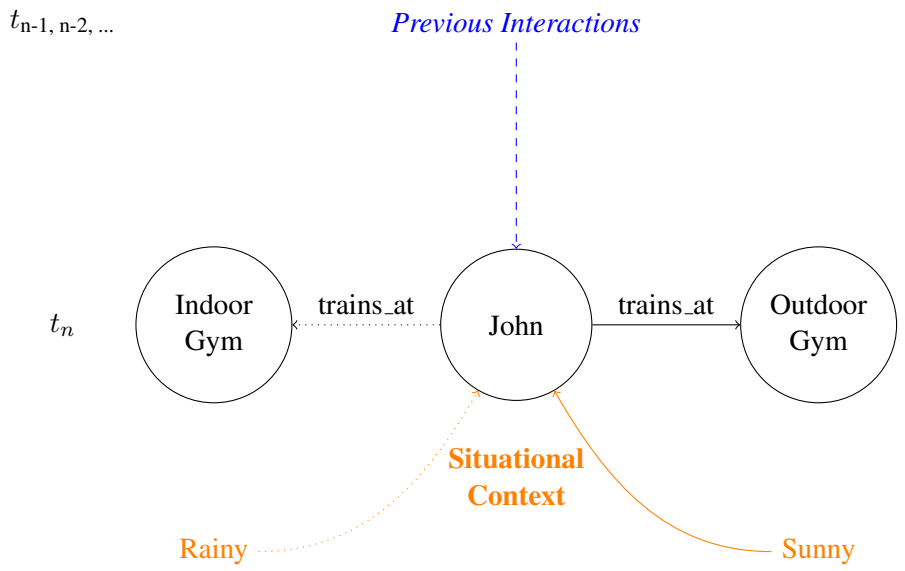

Figure 4: The influence of situational context

Depending on the situational context, the subject *John* prefers training at either *Indoor Gym* (if it is rainy), or at the *Outdoor Gym* if it is sunny.

affects the subject with regard to a decision between two locations can not be represented adequately.

In summary, there are two properties of real-world interactions that cannot be represented in their full scope with existing knowledge representation formalisms:

Sequential Context: The *memory* of an entity. How it is affected and changes by past interactions.

Situational Context: The current situational circumstances and how they influence interactions.

In this work, we attempt to bridge these gaps by introducing the necessary formalisms to fully represent the situations described above. Figure 5 shows a template for a dynamic situation with both sequential and situational context. Although parts of it can be modelled with (temporal) KGs, the important information of how the subject entity is affected by its past interactions and current context

would still be lost. The goal of this thesis is to expand the existing formalisms and their corresponding Embedding techniques towards covering the whole picture. The research questions pursued to reach this aim will be outlined in Section 1.2.

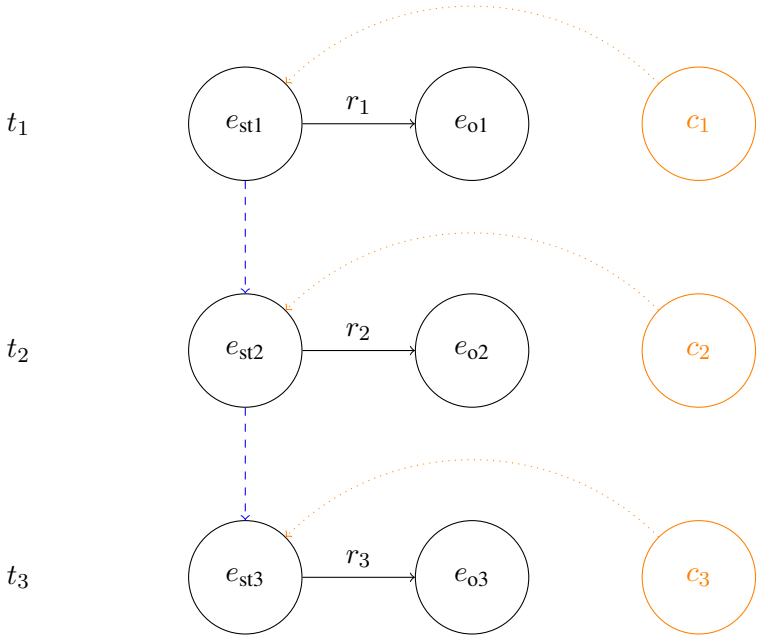

Figure 5: Entity dynamics with sequential and situational context

Modelling the dynamics of e_s across the three time steps requires taking into account the situational factors c_n at each time step and the previous state of the subject entity. If any of the previous relations r, objects e_o, or context entities c would change, then the final subject representation e_{st3} would also be slightly altered.

On a high level, there is an additional possible distinction between any two Contextualized Sequential Knowledge Graphs. The physical reality behind some knowledge graphs imposes restrictions that might prevent some entities to interact with each other. A simple example would be a traffic scenario where two vehicles simply are (yet) too far away from each other to form any relevant relation. After a few timesteps, both vehicles might cross each others path and one vehicle might give way for the other vehicle. Such scenarios, where possible interactions depend

on the history of the involved entities' actions, are called *dynamic* scenarios. On the opposite site of the spectrum, in *static* environments, there are scenarios where one acting subject interacts with completely passive objects. An example would be a person that is taking a look at the menu in a restaurant and afterwards decides which dish to have for dinner. The menu items and the menu itself do not change their state during the process, but they might affect the person's choice.

Normally, this information is not explicitly modelled into a Knowledge Graph, but depends on the underlying real-world scenario. This meta information can be used for a high-level description and distinction between the various scenarios that are described throughout this work. In all scenarios where a person's behavior (be it as a user of a social network, a driver of a vehicle, or a participant in a study) is modelled with that person as subject, this subject is assumed to be dynamic.

For the objects, however, different scenarios require different assumptions. On a global level, the objects are referred to as the *environment*, which can be characterized as static or dynamic. As long as all object entities are static and do not change over time, we assume the environment to be static as a whole. On the other hand, as soon as dynamic time-dependent context is expected to influence object entities in a scenario, we characterize this scenario as having a dynamic environment. The descriptions below show characterisations of potential scenarios that can be described with this terminology.

Static subject, Static environment: An ontology describing relations between entities in a closed world. All relations are fixed and do not change.

Static subject, Dynamic environment: A passive camera recording the environment. The recording does not interfere with the actions in the environment.

Dynamic subject, Static environment: An active subject interacts with completely passive objects. An example would be a person looking at a painting in a museum. The paintings are not affected by the interaction.

Dynamic subject, Dynamic environment: A traffic scenario where multiple vehicles attempt to reach their respective destinations. An interaction between the subject and any object can potentially cause both to change their status.

The present thesis focuses on the complex scenarios that require modelling dynamic subjects.

1.2 Research Questions

On the basis of these observations, we formulate the research questions below, which serve as a guideline for developing formalisms and corresponding Embedding techniques for modelling dynamics of entities in real-world scenarios. As the starting point, we explore to which extend the existing formalisms and techniques are able to represent dynamics in Knowledge Graph based representation paradigms.

RQ 1: What are the limitations of static formalisms and their corresponding embedding approaches with regard to sequential and situational context?

We perform empirical studies to support our arguments and, based on these findings, formulate in which ways the static KG paradigm requires extensions in order to enable it to capture entity dynamics.

This lays the groundwork for the next research questions, which aim at introducing a novel, extended framework for modelling dynamics in Knowledge Graphs. One of the main challenges in answering Research Questions 2 and 3 is to ensure that compatibility to static KG approaches is not limited.

RQ 2: How can the KG formalism be extended to model sequential and situational context?

RQ 3: How can static knowledge graph embeddings be transformed into contextualized representations?

Answering RQ 2 contributes to extending the formal framework of KGs to model the dynamic evolution under varying sequential and situational context. In response to RQ 3, we propose a corresponding Embedding mechanism that performs the contextualisation with respect to both the sequential and situational dimension.

The empirical studies evaluate the devised approaches that model entity dynamics in a dynamic environment.

While the sequential contextualisation of entities can be performed in similar ways in both dynamic and static environments, the situational contextualisation requires further investigation. Because the environment never changes in a static scenario, the Embedding approach for RQ 3 needs to be adapted. Research Question 4 addresses the consequences of a static environment in regard to the situational context.

RQ 4: What are the consequences of a static environment for embedding situational context?

In continuation of the observation that situational context requires novel approaches, Research Question 5 aims at integrating non-symbolic data as situational context.

RQ 5: Can situational context be expressed via non-symbolic data?

By answering these questions, we provide crucial new insights into how the KG formalism can be enabled to represent dynamic entities in varying situational contexts and diverse scenarios. The corresponding Embedding techniques that we introduce and the performed empirical studies serve as support for our argumentation.

1.3 Outline

This section provides an overview of the structure of this thesis. Starting from a high-level overview on the topic of entity dynamics in Knowledge Graphs and the related problems, we provide the necessary background information that is required to understand and tackle these issues. This is followed by the main part that consists of three chapters. The first main chapter identifies the shortcomings of the existing static formalism. Following this, the second main chapter formulates an extended formalism that captures entity dynamics in two dimensions (sequential and situational). In the third chapter, we explore which adaptations are required for these dimensions to work in static environments.

Introduction

Chapter 1 provides a high-level view on the problem of entity dynamics in Knowledge Graphs and formulates the research questions that lead to solving this problem. We establish the terms **Situational** and **Sequential Context** and how they related to Entities in Knowledge Graphs. We also introduce a level of distinction for Knowledge Graph scenarios based on the dynamics of subject entities and the environments they act in. We define four possible scenario types, of which the **Dynamic subject in a dynamic environment** and the **Dynamic subject in a static environment** are the relevant scenario types for our research.

Preliminaries

Following this, Chapter 2 provides a brief overview of research areas and concepts that are relevant for the present research. These range from techniques, notations and metrics to concrete approaches related to our work. The main focus is on topics related to Knowledge Graphs and their formal definitions. For additional chapter-specific related work, we make use of complementary related work sections in the beginning of the corresponding chapters.

Static Representations in Dynamic Scenarios

Chapter 3 is the first main chapter that serves as starting point for the subsequent research. We explore different techniques of introducing context (sequential, situational and background knowledge) to static Knowledge Graphs and study how this can contribute to representing dynamic scenarios. We conduct empirical studies on a *Location Recommendation* scenario and add the different context types to static Knowledge Graph Embedding approaches. The results are discussed and set in relation with the formal limitations of the static formalism. By this, we answer *RQ 1* and set the foundation to the developments in the subsequent chapter.

Chapter 3 is based on the conference contribution *'Embedding Taxonomical, Situational or Sequential Knowledge Graph Context for Recommendation Tasks'* which was published in the 2021 SEMANTiCS proceedings [73] and puts the experimental results in a new light with regard to modelling entity dynamics and the

limitations of static KGE approaches.

Towards Dynamic Entity Representations

On the basis of the identified limitations of static KGs for representing dynamic entities, Chapter 4 extends the existing KG formalism to *Sequentially Contextualized Knowledge Graphs* (addressing *RQ 2*) and introduces the corresponding *RETRA* embedding approach (addressing *RQ 3*). A key advantage of *RETRA* lies in its flexibility in comparison to existing approaches. It can transform pre-trained entity representations from static KGE approaches to dynamic representations that work with the same scoring functions. Alternatively, it can be used to train representations in an end-to-end fashion with any scoring function. The focus of the experimental section is on scenarios with **dynamic subjects in dynamic environments**, with each scenario covering a different aspect of dynamic representations.

Chapter 4 is based on the 2021 ESWC contribution *'RETRA: Recurrent Transformers for Learning Temporally Contextualized Knowledge Graph Embeddings'* [74].

Contextualisation in static environments

While the previous chapter handled scenarios with dynamic environments, Chapter 5 covers scenarios with **dynamic subjects in static environments**. Static environments hold certain properties which require different approaches to *situational context* in comparison to those introduced in Chapter 4. To answer *RQ 4*, these properties are identified and investigated with regard to the consequences they have on situational context. As a result of these consequences, we explore different views on situational context and integrate data from different modalities into a dynamic KG scenario (addressing *RQ 5*). The proposed techniques and approaches are evaluated on a novel dataset and a corresponding task. This data was collected during an extensive eye-tracking study and represents an entity that acts in a static environment.

Conclusion

Chapter 6 provides a summary of the empirical studies and their findings in light of the research questions addressed. In addition, directions for future research will be presented.

2 Preliminaries

This chapter provides an overview of research areas that are relevant to finding answers to the previously formulated research questions. Starting with a general overview over *Deep Learning* and *Representation learning*, we also cover the fundamental concepts of *Knowledge Graphs*, including the basics of *Machine Learning on Knowledge Graphs* and existing *KGE approaches*.

2.1 Deep Learning

Deep Learning is a subfield of Machine Learning that focuses on deep neural network architectures. In the context of this thesis, we focus on the aspect of neural networks as learnable functions.

The process of learning a function that maps some input x to an output y with a Neural Network (NN) is called training. A NN comprises an adjustable set of parameters θ that can be optimised to minimize a Loss function, given the predictions (the NN outputs y) and the corresponding targets (the supervision signal). Minimizing this Loss corresponds to adjusting the parameters θ in such way that the predictions match the targets. This is achieved through a technique based on the *Gradient Descent* procedure. For more details, refer to *Ian Goodfellow and Yoshua Bengio and Aaron Courville: 'Deep Learning'* [25].

The following overview briefly characterizes the most relevant Deep Learning architectures that are used in this thesis and for which purpose they are used.

Feed-Forward Network [25] A fully connected Neural Network with n hidden layers. Used for *classification* and for *translating* between different vector spaces.

Recurrent Neural Network [25] Neural Network with a self-loop. The outputs at a point in time t are used as inputs for $t + 1$. Used for *sequential modelling*.

Transformer Encoder [68] A Network type that implements a self-attention mechanism. Used for modelling the *context dependency* of entities.

2.2 Representation Learning

Representation learning can be described as the procedure of finding vector representations for non-numerical data to serve as inputs for deep learning applications. A crucial part for every neural network based architecture is the transformation of the raw data into numerical features [8].

On the example of Natural Language Processing, we showcase the process from symbolic to vector representations and outline the historic developments. Neural networks perform on numerical data in form of vectors and can be described as transformation matrices. Therefore, words and texts need to be translated from a symbolic form (i.e. words) to a numerical form (i.e. vectors).

Historically, the process of translating symbolic to numerical data was seen as a *feature engineering* problem. The example in Figure 6 shows how a sentence could be transformed into a vector representation using three hand-crafted feature extraction rules.

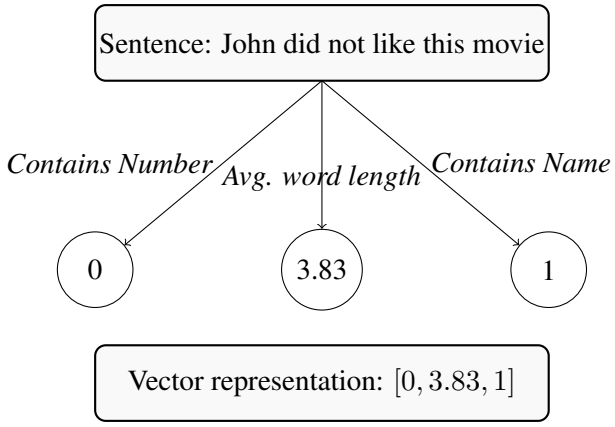

Figure 6: Feature engineering example

These hand-crafted features quickly proved to be inferior in comparison with more general-purpose representations like Bag-of-Words and TF-IDF, which are based on word frequency.

With the emergence of Word2Vec [50], approaches based on distributional se-

mantics became the prevalent paradigm. The word vectors are conditioned on a prediction task, which consists of predicting a target word based on its neighboring words. These Word embeddings have been the driving force in Natural Language Processing (NLP) in recent years. Soon after the learning of static embeddings of lexical items became popular, their drawbacks became apparent since they conflate all meanings of a word into a single point in vector space.

This limitation was recognized early and addressed by approaches that generate contextualized word representations given surrounding words in a sentence. Before the now dominant transformer [68] approach, LSTMs were used to contextualize word embeddings [32] with the integration of attention mechanisms [4].

This development, however, was to this day limited to NLP. Entities in Knowledge Graphs are still mostly represented by learnt static vectors. Therefore, the aim of this thesis is to outline an analogical path from static entity representations to contextualized representations.

2.3 Knowledge Graphs

Knowledge Graphs are directed labeled graphs that store facts as triples that consist of a subject, a predicate, and an object. Both subject and object are entities that are connected via a relation. Formally, a Knowledge Graph can be described as a tuple

$$G = (V, E, L)$$

where V is a set of vertices, $E \subseteq V \times L \times V$ a set of edges and L a set of Labels [33]. In the Knowledge Graph context, V is the set of Entities and L is the set of Relations. E contains a set of triples $(s, p, o) \in E$ that constitutes the facts in a Knowledge Graph $(s, o \in V, p \in L)$. Each fact consists of two entities, a subject and an object entity and the relation between them. Beside the (s, p, o) notation, there also exists the (h, r, t) notation, where h stands for *head*, r for the *relation* and t for *tail* entity.

Knowledge Graphs are primarily utilized as storage of structured information. With semantic technologies like *SPARQL*, they can be used from simple queries to

deductive reasoning about the entities they contain.

There are two high-level perspectives that are commonly referred to when describing KGs. In the *Closed World Assumption (CWA)*, the KG is understood as being a complete description of its domain, with no facts entering or exiting the graph. It is also assumed that all relations between the entities are known. The contrary perspective is the *Open World Assumption (OWA)*, where it is assumed that in addition to the known facts in a KG, there also exist facts that are yet to be added to the KG, in the form of establishing new links between entities or by introducing new entities. Figure 7 depicts the same KG under the two different assumptions. The red dashed nodes and relations in the left graph indicate existing facts that have not been discovered yet. The assumption for the right graph is that all facts are known and no new knowledge can be established.

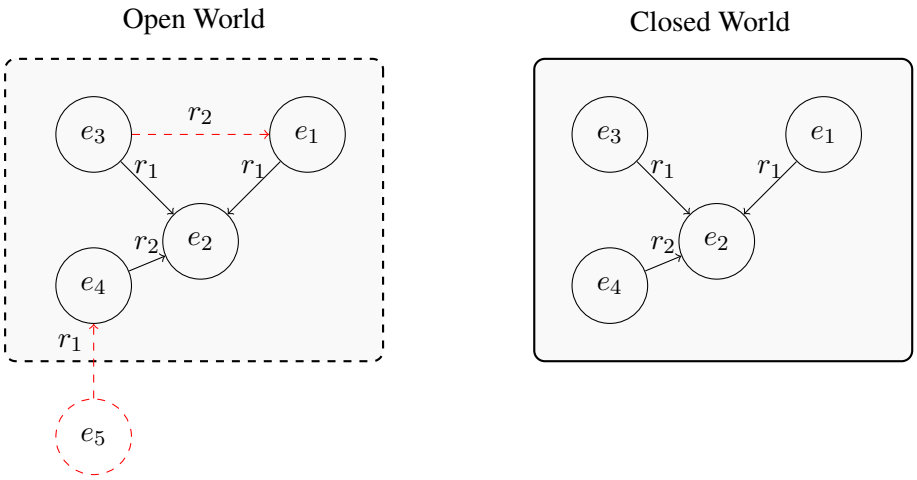

Figure 7: Open World vs. Closed World

The left side depicts a KG under the OWA with undiscovered connections (indicated in red). The right side shows a KG under the CWA, indicating that no further Links and Relations can be established.

2.3.1 Temporal Knowledge Graphs

Temporal Knowledge Graphs extend Knowledge Graphs by adding temporal information to their facts. This information is used to determine the time or time interval during which Knowledge Graph facts are valid. A real-world example would be political systems, where representatives are often elected for a fixed amount of time. The president of the United States, for example, is elected for a four year term. A fact stating that *Barack Obama* is president of the United States is only true within a certain time interval. To account for this, temporal Knowledge Graphs expands to

$$G_T = (V, E, L, T)$$

where T is a set of time stamps or time intervals. Accordingly, the facts in set $E \subseteq V \times L \times V \times T$ now contain quadruples (s, p, o, t), where $t \in T$.

2.3.2 Knowledge Hypergraphs

Knowledge Hypergraphs extend the arity of relations from binary to n-ary. Facts in Knowledge Hypergraphs are n-tuples with $(e_s, p, e_1, ..., e_n)$, where $\forall e_x : e \in V$. They can also be formalised as a tuple

$$G_H = (V, E, L)$$

with the facts expanding to $E \subseteq V \times L \times V^{|l|-1}$ with $|l|$ denoting the maximum arity in the set of relations.

The difference between temporal Knowledge Graph facts (s, p, o, t) and 3-ary Hypergraph facts (e_s, p, e_o, e_t) is that $e_t \in V$, whereas $t \in T$. The reasoning behind this is that for all $t \in T$, they are defined on an ordinal or higher scale and therefore can not be defined in the nominal scaled set V.

2.4 Machine Learning on Knowledge Graphs

Under the open world assumption, KGs are considered to be incomplete. Consequently, the question arises how the missing links (the red dashed lines in Figure 7) can be found. The corresponding machine learning task is called Knowledge Graph Completion and uses Link prediction to establish previously unknown KG facts. All contemporary Knowledge Graph Completion approaches implement at least the two following components:

1. A function f_{embed} that maps the elements of V and L to d-dimensional vector spaces:

$$f_{\text{embed}}(V, L) : V, L \rightarrow \mathbf{V}, \mathbf{L} \subset \mathbb{R}^d$$

2. A function ϕ that evaluates the plausibility of facts based on the vector representations of their elements:

$$\phi(\mathbf{s}, \mathbf{p}, \mathbf{o}) = \left\{ \begin{array}{ll} \nearrow & \text{if } (s, p, o) \in E \\ \searrow & \text{if } (s, p, o) \notin E \end{array} \right\} \text{with } \mathbf{s}, \mathbf{o} \in \mathbf{V}, \mathbf{p} \in \mathbf{L}$$

Both functions are strongly tied together in that function f_{embed} is conditioned on the function ϕ. The subsequent sections explain in more detail how this conditioning is realized and how a KGC approach is evaluated.

2.4.1 Training Procedure

The function f_{embed} is normally realized as parameterized function with adjustable parameters θ and can be trained by applying a negative sampling strategy. By creating a batch that contains one true fact $(s, p, o) \in E$ and n false facts $(s, p, o') \notin E$, we know that the score $\phi(\mathbf{s}, \mathbf{p}, \mathbf{o})$ should be higher than $\phi(\mathbf{s}, \mathbf{p}, \mathbf{o'})$. This can be used as the training signal to minimize a suitable Loss like the Cross-Entropy Loss.

Triples are corrupted by taking a true fact from a KG and replacing either the subject or object entity by another entity $\in V$ such that the resulting fact $\notin E$, thus creating a false fact. Following this routine, a number of $n + 1$ facts (n corrupted,

1 true fact) can be assembled and evaluated. Since it is known which facts are true and which are false, the expected outcome is a distribution of scores, where:

$$\forall fact : \phi(fact_{\text{false}}) \ll \phi(fact_{\text{true}})$$

Finally, the Cross-Entropy Loss between the expected and the predicted distribution is calculated and the resulting Error is backpropagated to adjust the parameters θ of f_{embed}. In formal terms, training is considered as minimizing the Cross Entropy Loss L w.r.t. the parameters θ of f_{embed}.

2.4.2 Evaluation Metrics

Knowledge Graph Embedding models are most commonly evaluated on three metrics for Link Prediction, namely Mean Rank (MR), Mean Reciprocal Rank (MRR) and Hits@k [71]. These metrics are computed on KG facts that were not used for training. For a given incomplete triple $(s, p, ?)$ or $(?, p, o)$, every entity $e \in V$ is inserted and the score is calculated. In the next step, all entities are ranked according to their score. Based on the entity rankings and the ground truth, the rank of the expected entity can be determined. Mean Rank (MR) describes the averaged achieved rank of the ground truth entity for a given set E of facts:

$$MR = \frac{1}{|E|} \sum_{n=1}^{|E|} rank_n$$

The desired result for the Mean Rank is when $MR = 1$, indicating that for all facts in the set E, the ground truth entity was ranked first. Since the upper limit of the MR equals the number of entities $|V|$ in the Graph $|G|$, this metrics allows only for limited comparison of approaches across graphs. A way of mitigating this issue is by reporting the Mean Reciprocal Rank, which is defined as follows:

$$MRR = \frac{1}{|E|} \sum_{n=1}^{|E|} rank_n^{-1}$$

By calculating the average of the inverse rank, the MRR always results in a value between 0 and 1, disregarding the number of entities in a Graph. The best possible value for the MRR is when $MRR = 1$, with lower values indicating a worse performance. Hits@k describes metrics that indicate whether the ground truth entity was ranked among the k highest ranking entities. The most commonly reported k values are 1, 3 and 10. In formal terms, Hits@k reports the proportion of incomplete facts for which the ground truth entity was ranked among the top k:

$$Hits@k = \frac{1}{|E|} \sum_{n=1}^{|E|} \left\{ \begin{array}{ll} 1 & \text{if: } rank_n \leq k \\ 0 & \text{if: } rank_n > k \end{array} \right\}$$

The Hits@k results in values between 0 and 1, with values close or equal to 1 indicating a good performance. In most cases, MR, MRR and Hits@k are reported together.

2.5 Knowledge Graph Embedding Approaches

In recent years, Knowledge Graph Embedding (KGE) has been a very vibrant field in Machine Learning and Semantic Technologies, especially in the area of Representation Learning (see [34] for a survey). Numerous methods for embedding knowledge graphs have been proposed and even more adaptations have been published. In general, KGE methods can be 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 [66]) or Hypercomplex $\mathbb{H}^d$ (cmp. [86]) have been used as well.

Standard KGE methods do not take into account temporal information nor contextual factors that influence the plausibility of a fact. However, there have been attempts to address each limitation, as outlined in the following.

2.5.1 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 context. Such approaches from Statistical Relational Learning are based on graphical models and tensor factorization [58]. A more recent approach extends the current KGE method SimplE [35] to hypergraphs [21] but does not take into account temporal or sequential information.

2.5.2 Temporal Knowledge Graph Embeddings

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 [39], [18]. A more sophisticated approach is proposed in [46]. A more entity-centric perspective is taken in [65] which attempts to model the temporal evolution of entities. Approaches like TeRo [78] and DyERNIE [30] explore non-euclidean embedding spaces for modelling temporal patterns in tKGs.

3 Static Representations in Dynamic Scenarios

Assessing the limitations of static KGE approaches with regard to sequential and situational context (addressing *RQ 1*) requires a suitable dynamic scenario in which both types of information play an important role. A POI (point of interest) recommendation scenario is an excellent match, because not only the history of previously visited POIs is important, but also the current situational context. As an example, the current weather (situational context) might affect whether an individual would rather pursue an indoor or an outdoor activity. Additionally, an individual's history of visited POIs also has an influence on which POI the individual choses next.[3]

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 [69]. Similarly, some applications require more complex outputs than prioritized recommendation lists in the direction of composite or sequential recommendations.

An 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 should not be of interest to a user who just had something to eat.

[3] Parts of this chapter have been accepted as a paper to 2021 SEMANTiCS conference and have been only altered minimally. See [73]

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. With this, 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.

KGEs allow to transform such symbolic knowledge into predictive models that operate on latent vector spaces. However, static 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 section, we address *RQ 1* and test the hypothesis that a global vector representation 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.

To support the hypothesis, different techniques to incorporate three dimensions of additional information are tested: 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.

In addition to the empirical experiments, the different dimensions are described on a formal level in accordance with the formalisms established in the previous sections.

The contributions can be summarized as follows:

- We provide 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.

3.1 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 in the following. 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 [5, 87] 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 [49].

An early approach for POI recommendation based on models for human mobility and their dynamics in social networks is described in [17]. Another early approach for context-aware recommendation that considers social network information, personal preferences and POI popularity is presented in [85]. Nousal et al. [53] 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. [7] 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, however, method-wise not directly related to our ap-

proach, which is focused on knowledge graph embedding methods.

An approach presented by Baral et al. [6] 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 [16]. Feng et al. [23] 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 [44]. 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 [61]. Another knowledge graph embedding based approach [76] 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 [51]. 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 [84]. [81] propose LBSN2Vec, a hyper graph embedding approach designed specifically for LBSN data which we also use in our experiments.

3.2 Capturing Taxonomical, Contextual and Sequential Information for Recommendations

The goal of this section is to investigate the potential of three different types of information, namely taxonomical, contextual and sequential, for their use in embedding-based recommender systems and how their use is limited by static embeddings. 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. This section describes the POI Categories (POICa) ontology used for representing information about POIs mainly by exploiting their hierarchical relationships.

3.2.1 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[4] 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.

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.

[4] https://developer.foursquare.com/docs/build-with-foursquare/categories

- *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.

As depicted in Figure 8, POICa ontology comprises a number of subcategories distributed in various levels, which are highlighted in different colors for a better readability.

3.2.2 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*[5]. For instance, in order to represent geo-spatial information for a given POI the following concepts from *DCTerms*,

[5]https://schema.org/, http://xmlns.com/foaf/0.1/, http://dbpedia.org/ontology#, http://purl.org/dc/terms/, https://cutt.ly/QhQrFxv#

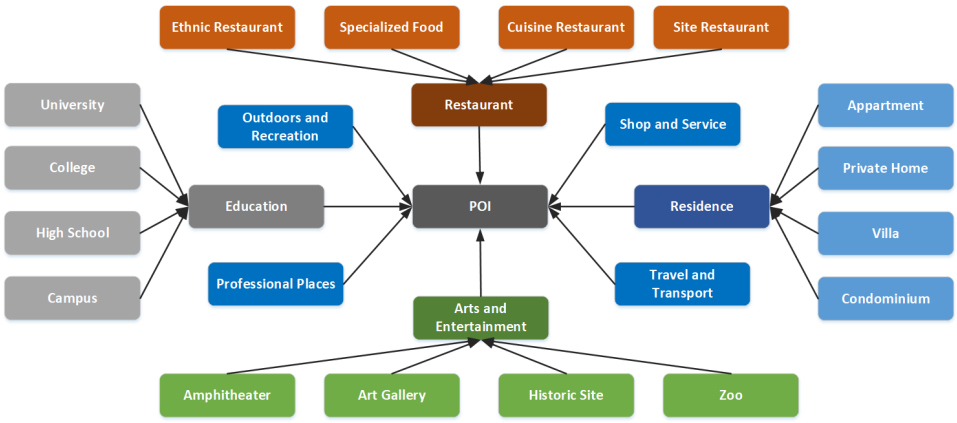

Figure 8: POICa-breadth view.

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. For this study, our focus was to describe the core concepts that form the basis to understand the conducted work from the taxonomic point of view.

3.2.3 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 [21], a recently introduced hypergraph embedding approach that showed promising results on other tasks and allows for easy adap-

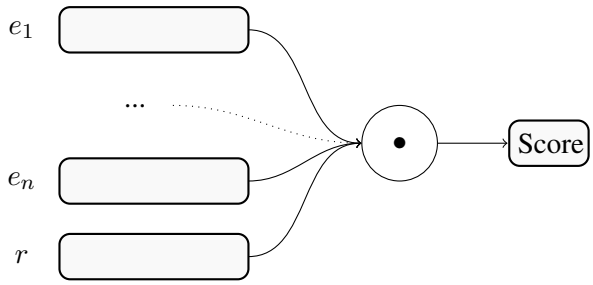

Figure 9: HypE architecture with scoring function.

tion 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 $\phi(\mathbf{r}(\mathbf{e_1}, ..., \mathbf{e}_{|\mathbf{r}|}))$, and describes the sum of the element-wise product of the corresponding embedding vectors (cmp. Fig. 9).

3.2.4 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:

Cold start: In many situations, the system encounters a new user or cannot 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 cannot be re-trained after each new data point arrives.

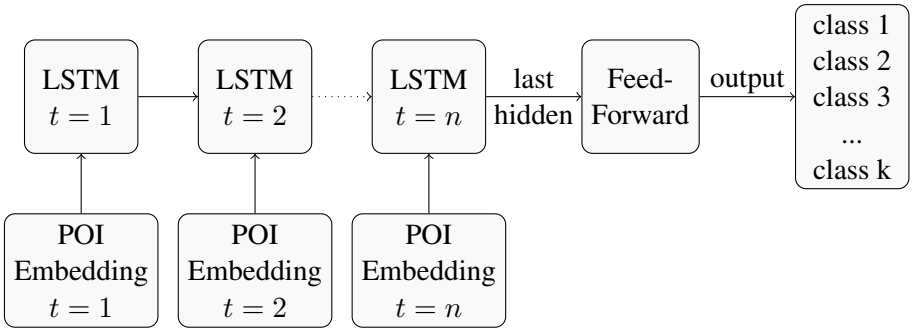

Figure 10: The architecture of the LSTM approach.

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 [32] 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. 10).

3.3 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 [81] 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 [29]
- Prediction based on a hypergraph approach [21]
- Prediction based on sequential modelling

The first type of experiments were run on existing implementations[6] 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[7]. Our implementation and experimental settings can be found in the repository[8] 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 behavior and therefore only consider a "hit" when the exact POI for this sequence is recommended.

[6] https://github.com/thunlp/OpenKE
[7] https://github.com/ElementAI/HypE
[8] https://github.com/siwer/TaxonomicalKGE

3.3.1 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[66], Distmult[80], Hole[52], Simple[35] and Transe[10]. 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.

3.3.2 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 (refer to Figure 11 for a visualisation). The hour of day and the day of week are derived from the timestamps in the original data. To achieve the results presented here, 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 an 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

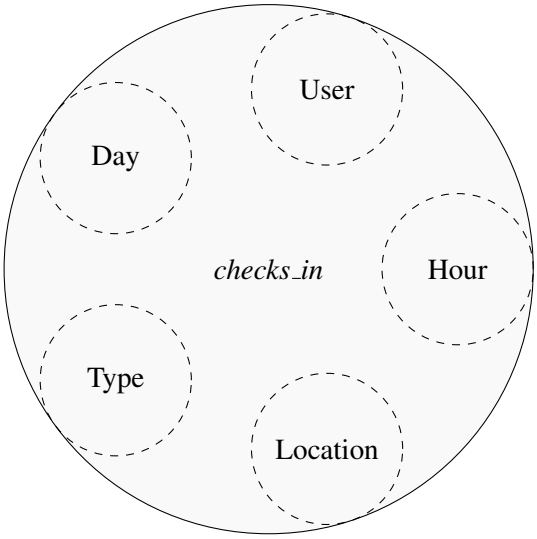

Figure 11: The *checksIn* relation as a hypergraph fact

This example shows a hypergraph fact where all entities have the same weight without any notion of order for the *checks_in* relation.

layer) learns to project from the ontology space to the general feature space. Figure 12 shows architecture 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 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 behavior is also consistent across the other observed metrics. In Tables 2 and 3, the '+ Ont' approaches denote $\lambda > 0$ for the HypE approach. As with the binary approaches, we also present the results from the best runs.

λ	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 λ on MR

3.3.3 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 it into a discrete sequence of check-ins. This, however, is not an assumption that would reflect real-word behavior, 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

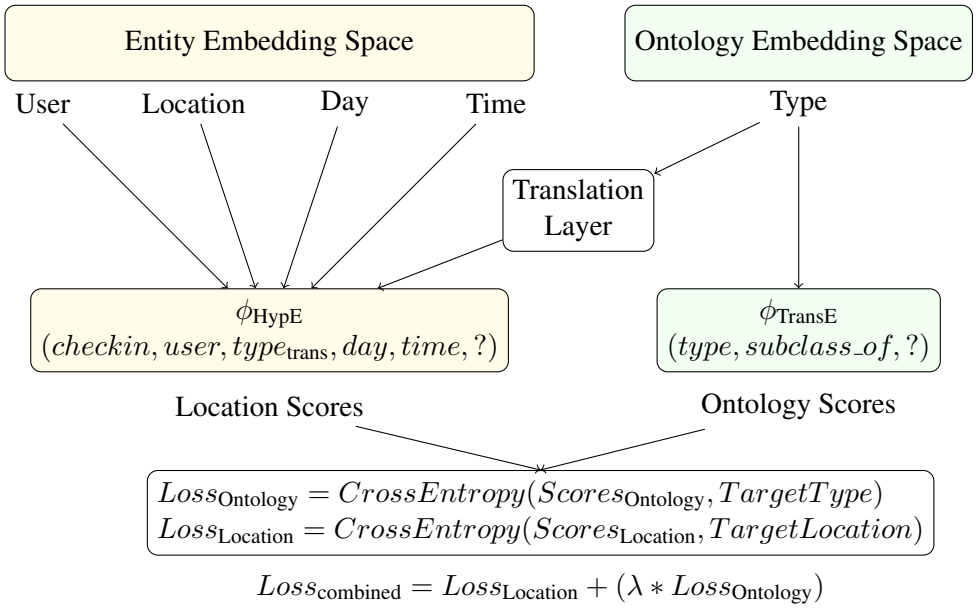

Figure 12: Architecture of the combined approach.

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.

3.3.4 Discussion of Results

Tables 2 and 3 provide an overview of the best hits@10 results for each setup described above. For the KGE approaches, also results for adding information modeled in the POICa ontology (see Sec. 3.2) are reported. First, the results reported in [81] 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 did not make a significant difference.

This is likely due to the naive way of introducing just the relational information

Approach	Jakarta	Jakarta + Ont	EmbDim
LBSN2Vec [81]	0.08	-	128
Complex	0.041	0.040	100
Distmult	0.045	0.045	150
HolE	0.031	0.030	100
SimplE	0.047	0.046	100
TransE	0.064	0.064	200
HypE	**0.742**	**0.771**	130/50
LSTM	0.085	-	100

Table 2: Best hits@10 results and corresponding embedding dimensions (Jakarta subset)

Approach	NYC	NYC + Ont	EmbDim
LBSN2Vec [81]	0.11	-	128
Complex	0.033	0.032	100
Distmult	0.035	0.039	100
HolE	0.024	0.023	100
SimplE	0.035	0.036	100
TransE	0.044	0.045	150
HypE	**0.722**	**0.738**	130/25
LSTM	0.080	-	100

Table 3: Best hits@10 results and corresponding embedding dimensions (NYC subset)

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.

3.4 Limitations of Static Approaches

Based on the empirical findings, we discuss how the underlying KG formalisms are fundamentally limited with regard to describing dynamic situations. At first, it appears that all of the three observed dimensions can be incorporated within or with the help of KGE approaches.

3.4.1 Ontological Information

Conceptually, adding the ontological information to (Hyper-) Knowledge Graphs in a Recommendation Scenario is equal to extending the KG by adding more facts that provide further detail about how different locations are related with each other. If, for instance, we establish that a location loc_1 and a location loc_2 belong to the same type $type_1$, it becomes easier to infer that a user $user_i$ who has a preference for $type_1$ is more likely to visit locations of this type, instead of a different location loc_3 that is of type $type_2$. While this additional information could only partly (if at all) be exploited in the binary KGE approaches, it has shown to be of more use in the n-ary scenario.

For the n-ary scenario, a slightly different approach was chosen. By embedding the ontological information into its own space and projecting the type information from there to the hypergraph, we were able to control its influence through the λ hyperparameter (see Figure 12). A value of $\lambda = 0$ corresponds to simply providing the type of a location as additional information, whereas $\lambda = 1$ corresponds to fully include all ontological information. By observing different values for λ, we can conclude that the ontology provides information beyond additional context in form of the location type (see Table 1).

3.4.2 Contextual Information

Inserting contextual information by extending the binary $checks\_in(user, location)$ relation to a 5-ary $checks\_in(user, location, hour, day, type)$ relation is relatively straight-forward. As it is shown empirically, the performance of this approach surpasses the binary approaches at ease. By providing detailed situational context, it becomes easy to identify patterns that correspond to a daily (and weekly) routine of a user. The correlation between certain times and weekdays and location types like bars (mostly visited in the evening and weekends) vs. workplaces (mostly visited during daytime and working days) is apparent.

Although extending the $checks\_in$ relation to a higher arity is a formally legitimate way of describing the situation, it is unsatisfactory with regard to the underlying real-world situation. HypE takes the direction of a relation into account by

applying position dependent filters and assumes that the underlying Hypergraph edges consist of an ordered sequence of nodes[22][21]. In an example relation $sells(seller, buyer, item)$, the position of an entity e_1 indicates its role as either buyer or seller. However, in the $checks\_in$ relation described above, $hour$ and day provide situational context of potentially same importance, whereas $type$ can be seen as a property of $location$. The underlying interactions could be described as: *Given the current day and time, the user checks in at a location of a certain type*. In this situation, the context influences the user's decision. Interactions like this cannot be naturally described by order alone. In conclusion, although the Hypergraph approach performs well empirically, it is not an adequate description formalism for this kind of situations.

3.4.3 Sequential Information

The presented approach to include sequential information is a combination of the learnt HypE embeddings and a LSTM network. While the empirical results suggest that the sequence of check-ins contains valuable information for prediction, there are a few limitations to this procedure. Firstly, the entity representations are not directly conditioned on predicting the next location based on a sequence, but on a prediction task based on situational context as described in the section above. Secondly, since the embeddings are obtained through the HypE approach, the underlying representation formalism is not suited for expressing sequential information. Ideally, instead of relying on one static embedding for each entity, the vector representation should reflect the slight changes within the entity based on the previous interactions. An obvious example would be a person that is hungry and visits a restaurant. Afterwards, the hunger is satisfied, which should be reflected in an updated entity embedding. In the presented approach, this change is to some degree realized in the hidden state of the LSTM network. In every step, the hidden state gets updated which could be interpreted as interactions influencing the current entity. However, there is no explicit user representation involved. A user representation would ideally capture both general and situation-specific information, whereas with the current approach, the representation is only situation-specific.

3.5 Conclusions

In this chapter, we obtained empirical evidence for how state-of-the-art latent recommendation approaches can exploit ontological, situational and sequential information with static entity representations. 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. In addition to 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. It can then 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 with static representations is non-trivial. With regard to the first Research Question, the limitations of the tested approaches could be summarized as follows:

RQ 1: What are the limitations of static formalisms and their corresponding embedding approaches with regard to sequential and situational context?

- Representing situational context as a Hypergraph does not reflect the complex inter-entity interactions sufficiently. Different entities play different roles in a given situational context. Given the $checks\_in()$ relation from before, it becomes apparent that *hour* and *day* directly affect the *user*. They do not influence the *location* or its *type*. The *type* is a passive and - within a reasonable timeframe - immutable description or attribute of the *location*. Al-

though it is formally possible to depict such a situation as a Hypergraph, the real-world relations between the different entities are not well represented.

- There are two main limitations of the presented approach for representing sequential context. The first is that the entity embedding approach and its underlying formalism is decoupled from the sequential contextualisation that happens in the recurrent neural network. The second limitation is that there is no explicit *user* information passed in this network. As a result, the concrete sequence of actions is not directly connected to a global *user* representation.

Establishing these limitations points out a way towards truly dynamic entity representations. It becomes apparent that the ability to undergo changes should be reflected in a given representation. This means that the representation $\mathbf{u}$ of a user should be different under different circumstances, while still carrying some global information like personal preferences. These different circumstances could be of both sequential and situational nature. Ideally, this ability should be reflected in the underlying representation formalism, aswell as in the corresponding embedding mechanism. The following chapter explains how the binary KG formalism can be extended with regard to situational and sequential context and additionally proposes a corresponding embedding approach.

4 Towards Dynamic Entity Representations

Based on the observations from the previous chapter, we devise strategies to overcome the identified limitations of static KG-based formalisms and embedding approaches. The key limitation can be described as the unability to reflect how interactions can have a (lasting) effect on entities.[9]

We have described two main sources of influence under which an entity acts. The first is the situational context that, for a given point or period in time t, affects an entity. This could be phenomena like environmental factors such as the weather, the current day, or the presence of other entities. The second source is the specific history of an entity. Past interactions can influence the current internal state of an entity. For an animate entity like a human, this state could be of internal physiological nature, like being hungry or tired, or a result of external actions like the disposition towards an other entity.

In traditional KGs and corresponding embedding approaches, there is no option to express contextual influences for entities. An entity e is always the same, independent of the factors by which it is affected in different contexts. This is a rather non-intuitive approach for modelling real-world behavior, since different situations require specific adaptations. A simple example would be a person which acts differently in a professional context vs. a private context. Consequently, a single static representation per entity is not adequate for many real-world scenarios. Instead, entities need to be put into context by factors specific to the current situation and their subjective history.[10] This requires a different entity embedding for each situation, not just one that attempts to be universally valid.

By extending the KG formalism to incorporate situational and sequential context, we address *RQ2*. Additionally, corresponding to *RQ3*, we present an Embedding approach for the extended formalism and test it empirically on three diverse scenarios. The contributions of this chapter with regard to the research questions

[9] Parts of this chapter have been accepted as conference paper to ESWC 2021 and have been only altered minimally. See [74]

[10] In psychology and neuroscience, this distinction might be referred to as semantics vs. episodic memory. See [67]

are summarized below:

- We propose *temporally contextualized KG facts* (tcKG facts) as a modelling template for situation-specific information in a KG. This adds a temporal sequence of hyper-edges (time-stamped subject-relation-object triples where the relation is n-ary in order to capture n contextualizing factors) to an existing static KG (see Sec. 4.2).
- We introduce the deep learning framework RETRA, which transforms static global entity and relation embeddings into temporally contextualized embeddings, given corresponding tcKG facts. This situation-specific embedding reflects the role an entity plays in a certain context and allows to make situational predictions (see Sec. 4.3). RETRA uses a novel *recurrent architecture* and a *constrained multi-headed self-attention layer* that imposes the relational structure of temporally contextualized KG facts during training (see Sec. 4.4).

In order to demonstrate how broadly applicable tcKG and RETRA are, we apply and test them in three diverse scenarios, namely location recommendation, event prediction and driving-scene classification. Our empirical results indicate that contextualizing pre-trained KGEs boosts predictive performance in all cases (see Sec. 4.5).

4.1 Dynamic Entity Representations

The previously introduced KG formalisms allow for flexible modelling of real-world entities and their interactions in diverse scenarios. As established in the previous chapters, none of them facilitate modelling real-world situation *dynamics*, where the current state of an entity is directly influenced by previous interactions and additional external context that is not defined within a relation. While Knowledge Hypergraphs and temporal Knowledge Graphs are capable of integrating contextual and temporal information into their facts, they both have the fundamental limitation that the entities and - in an Embedding scenario - their vector

representations are fixed. In the example of a person, there would be one situation-independent representation only, ignoring any previous interactions that might still affect this person. If, for instance, we know that this person already had lunch, we can assume that in the next interactions, the underlying need *hunger* will not have great effect on this person's behavior. To fully capture situation dynamics, a model should be able to take into account both the influence of previous interactions and the current contextual information.

4.2 Modeling Subjective Temporal Context

This section outlines how Knowledge Graphs can be easily extended to additionally model both the contextual and the sequential information. The design idea behind the extensions is that the original formalism does not have to be changed in order to be compatible with the newly introduced extensions. To address *RQ2*, we start from the assumption that static KG facts act as background knowledge but the inference task depends on the situational context and the subject's memory. Consequently, we need to extend triples as follows:

KG facts are defined as a triple (e^s, r, e^o) where $e^s, e^o \in \{e^1, ..., e^{n_e}\}$ is from the set of n_e entity instances and $r \in \{r^1, ..., r^{n_r}\}$ from the set of n_r relation types. KG facts constitute subject-predicate-object statements that are assumed as being static and stable background knowledge.

tKG facts are quadruples (e^s, r, e^o, t) where $t \in \mathbb{N}$ indicates a point in a sequence when the fact occurred. In many scenarios, t is obtained from discretizing timestamps and thus creates a globally ordered set of facts, where n_t is the total number of points in time (cmp. [65]). Note that temporal KGs have mostly been using t to model the point in time when a fact is being observed. Here, we are taking a slightly different perspective by modeling in which point in time a subject e^s makes an experience in relation to interactions it has made at previous points in time. The main purpose of t in our use-case is to order the sequence of facts.

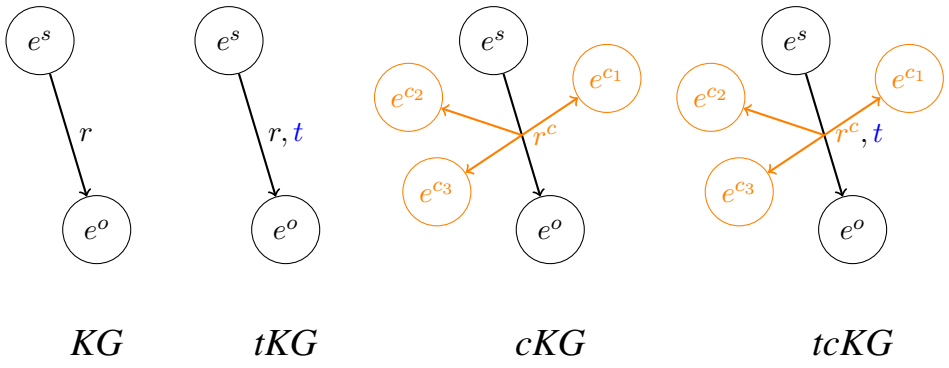

Figure 13: From KG triples to tcKG facts.

From subject-predicate-object KG triples, to temporal tKG facts which occur at time t, to contextualized cKG facts which allow to model influencing factors as an n-ary relation, to tcKG facts which contextualize the KG observation at a certain time. This models the observations of a sequence of relations r^c the subject e^o is involved in.

cKG facts are $(n+1)$-tuples $(e^s, r^c, e^o, e^{c_1}, ..., e^{c_{n_c}})$ that allow to model situational context as an n_c-ary relation. $e^{c_1}, ..., e^{c_{n_c}}$ are the n_c context entities influencing the relation r^c between subject e^s and object e^o. One key point is that as opposed to Hypergraphs, the relation r^c still keeps a clearly defined subject e^s and object e^o entity.

tcKG facts are $(n+2)$-tuples $(e^s, r^c, e^o, t, e^{c_1}, ..., e^{c_{n_c}})$ which represent sequentially contextualized KG facts by combining the features of tKGs and cKGs. Intuitively, they capture a specific situation which subject e^s is experiencing at time t. e^c are influencing factors towards e^s's relation to object e^o.

The evolution from KG facts to tcKG facts is depicted in Figure 13. With these facts as additional building blocks, we can now model task-specific temporally contextualized KGs as temporal hypergraphs (i.e., relations are potentially n_c-ary and potentially associated with timestamps t).

This subjective temporal context is input to RETRA as follows:

1. Given an n_c-ary relation r^c we first define one entity participating in r^c as

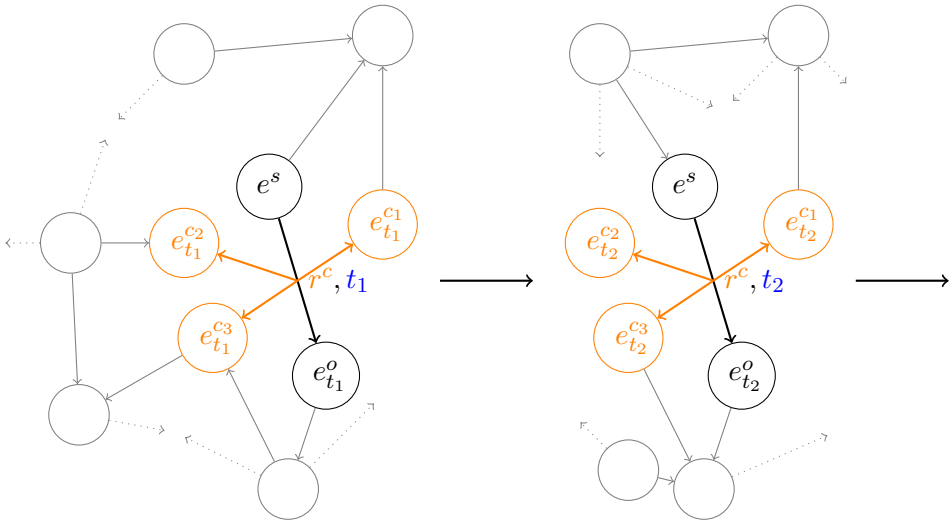

Figure 14: Temporally unrolled tcKG facts.

tcKG facts (shown in black) and their relation to static background knowledge (KG and cKG facts, shown in gray) as a temporally unrolled KG.

the subject e^s whose perspective is represented in respect to an object e^o.[11]

2. The contexts $c_1, ..., c_{n_c}$ are given by the remaining entities involved in the n_c-ary relation. They define the influencing factors in a concrete situation.

3. If available, the context can be extended by entities deterministically dependent on e^o. Non-deterministically dependent context of e^o or e^c or facts that are not specific to a certain point in time t are not explicitly modelled (see gray edges and gray nodes in Fig. 14).

4. Finally, the temporal context is modeled by n_c relation instances of r^c that involve e^s as the subject. Sorted by time-stamp t, r^c defines the sequential context from the perspective of e^s towards its relation to e^o (see black edges and black nodes in Fig. 14). Note, that the context entities e^c do change in

[11] Note, that this is a deliberate modeling choice that is not due to technical limitations. RETRA can model any sets of subjects and objects since transformers allow variable numbers of inputs and can mask any subset during training. We chose this restriction since this is pragmatically the most common pattern and avoids a cluttered notation.

Application	Subject e^s	Relation r^c	Object e^o	Contexts $c_1, ..., c_{n_c}$
Location recommendation	user	checksIn	location	time of day, weather , day of week, location type, ...
Event prediction	source actor	eventType	involved target organizations	target country, source country, sector, ...
Driving-scene classification	ego vehicle	involvedIn	conflict-type	ego lane, foe road users, foes' lanes, signaling, acceleration, speed, ...

Table 4: Illustrating examples of instantiated tcKG patterns for three applications.

every step, as does e^o, thus $e^o_{t_1} \neq e^o_{t_2}$. Consequently, all the facts associated to e^c and e^o do change in every step (gray edges and gray nodes in Fig. 14). Only e^s and the relation-type of r^c stay fixed as defined above.

With the above selection procedure, we obtain a sequence of tcKG facts from the KG by filtering for relations r^c with subject e^s. Consequently, such a model of dynamic context consists of a sequence of (e^s, r^c)-tuples with varying sequence-length n_t. In each step r^c has an varying object e^o and is characterized by n_c contextual factors e^c.[12]

For illustration purposes, Table 4 shows instantiations of the tcKG modelling pattern according to our three applications domains *location recommendation*, *event prediction* and *driving-scene classification* (see Sec. 4.5 for details).

4.3 Embedding Subjective Temporal Context

So far, the tcKG modelling pattern provides an explicit representation of dynamic context of a subject and a relation-type as a sequence of sub-graphs (see Fig. 14). The second contribution of this chapter, addressing RQ3, is a machine learning method that captures this information in two embeddings, the sequential context $\mathbf{e}^s$

[12]Note, that the arity n_c does not need to be fixed in each step and for each e^s. Variable-length context, unknown or missing e^cs can be modelled and handled efficiently with RETRA, since transformers can handle variable input lengths.

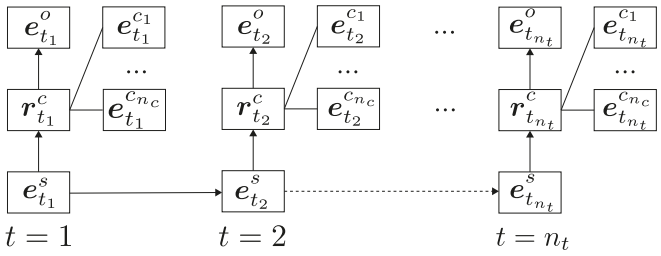

Figure 15: Sequential context for subject and relation

Sequential context for subject and relation embedding $\mathbf{e^s}$ and $\mathbf{r}$. The object $\mathbf{e}^o$ and contextual factors $\mathbf{e}^c$ refer to a different symbolic KG entity e^o and e^c in every step. They are given by (pre-trained) static KGEs. In contrast $\mathbf{e}^s$ and $\mathbf{r}$ represent the same symbolic KG node e^s and hyper-edge r, regardless of time and context. However, the embedding is customized with a situation-specific contextualized embedding, depending on the temporal and relational context.

and the situational context $\mathbf{r}^c$.[13] Once we obtain those embeddings we then can use any embeddings-based scoring function, e.g., for contextualized link prediction.

One way to capture sequential context in a single embedding is to represent the history of sequential information in a latent state. As common in Hidden Markov Models or Recurrent Neural Nets, all $t-1$ previous contextualized observation are reduced to one embedding capturing the latent state up to this point. In our model, this memory is captured in the $\mathbf{e}^s$ embedding. Thus, we define the probability P of the contextualized relation representation $\mathbf{r}^c$ as being conditioned on $P(\mathbf{r}^c|\mathbf{e}^s, \mathbf{r}, \mathbf{e}^o, \mathbf{e}^{c_1}, ..., \mathbf{e}^{c_{n_c}})$. The subjective context representation $\mathbf{e}^s$ depends on $\mathbf{r}^c$ but also on the previous experience $\mathbf{e}^s_{t-1}$ in similar situations: $P(\mathbf{e}^s_t|\mathbf{e}^s_{t-1}, \mathbf{r}^c_t, \mathbf{e}^o_t)$. These conditional dependencies are visualized in Figure 15.

In the context of traditional KGE methods, a minimal Knowledge Graph Embedding approach requires two functions, one for mapping the entities and relations in V to their vector representation in a d-dimensional space, and one for scoring the plausibility of a triple:

$$f_{embed} : e \longrightarrow \mathbf{e} \,|\, e \in V, \mathbf{e} \in \mathbb{R}^d$$

[13] We indicate embedding vectors for nodes $\mathbf{e}$ and relations $\mathbf{r}$ with bold symbols to contrast them to symbolic nodes e and relations r from the KG.

$$\phi : \mathbf{s}, \mathbf{p}, \mathbf{o} \longrightarrow score \,|\, \mathbf{s}, \mathbf{p}, \mathbf{o} \in \mathbb{R}^d, score \in \mathbb{R}$$

The scoring function serves two main purposes. The first one is to ensure that the vector representations are able to capture the Ground Truth of the Knowledge Graph. Secondly, the scoring function is used for Link Prediction in a Knowledge Graph Completion task. In case of a parameterized scoring function, both scoring and embedding function are jointly optimized. In this section, we build upon this minimal approach and extend it to embed the modelling formalisms that were introduced before.

To extend the Embedding approach to capture situational context in r_{context}, a contextualisation function has to be added that transforms a current relation representation and the current context representations to a new contextualized relation representation:

$$f_{context} : \mathbf{r}, \mathbf{c_1}, ..., \mathbf{c_n} \longrightarrow \mathbf{r_{context}} \,|\, \mathbf{r}, \mathbf{c}, \mathbf{r_{context}} \in \mathbb{R}^d$$

Similarly, the extension to the sequential Knowledge Graph formalism requires an additional sequentialising function that lets the subject entity *memorize* the past interaction:

$$f_{sequence} : \mathbf{e_t^s}, \mathbf{r_t}, \mathbf{e_t^o} \longrightarrow \mathbf{e_{t+1}^s} \,|\, \mathbf{e} \in \mathbb{R}^d$$

Both functions are parameterized by adjustable weights θ, which are optimized with regard to a Cross Entropy Loss based the results of ϕ.

Both extensions are compatible with each other, such that a Contextualised Sequential Knowledge Graph Embedding approach can be described by the four functions f_{embed}, ϕ, $f_{context}$ and $f_{sequence}$. All functions can be realized as neural networks which can then be optimized for solving the given Embedding problem. The scoring function, however, is historically not realized as a neural network, but mostly as pre-defined vector-matrix operations.

4.4 RETRA: The Recurrent Transformer

Learning customized embeddings based on subjective sequential and situational context requires a novel Neural Network (NN) architecture that implements the functions for embedding, scoring, sequentialising and contextualisation.

4.4.1 The RETRA Architecture

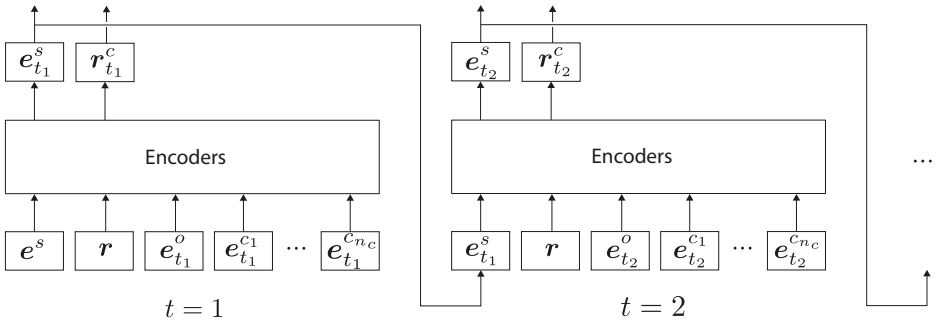

Figure 16: Recurrency in the RETRA architecture.

In the first step, e^s is not temporally contextualized but a static KGE embedding. In $t = 2$ the contextualized $e^s_{t_1}$ is used as input to generate the temporally contextualized $e^s_{t_2}$

Our model is inspired by the encoder stack of transformers [68] and Recurrent Neural Networks (RNNs) and can thus be called a Recurrent Transformer (RETRA). We cannot use common RNN architectures, like LSTMs [32] since they do not handle multiple variable length inputs per step in a temporal sequence.

Regarding input and output, RETRA receives the result of the potentially pre-trained f_{embed} function that outputs static embeddings $\mathbf{e}^s, \mathbf{r}, \mathbf{e}^o, \mathbf{e}^{c_1}, ..., \mathbf{e}^{c_{n_c}}$. The output is the situationally contextualized $\mathbf{r}^c$. In addition, e^s's previous subjective memory $\mathbf{e}^s_{t-1}$ is passed on to generate the temporally contextualized embedding $\mathbf{e}^s_t$ for the current step. Thus, the only non-pre-trained embedding passed on to the next step is $\mathbf{e}^s_{t-1}$ (cmp. Fig. 16).

The final crucial building block to transform $\mathbf{r} \longrightarrow \mathbf{r}^c$ and $\mathbf{e}^s_{t-1} \longrightarrow \mathbf{e}^s_t$ is handled inside the encoder stack. Similar to [68], we use a stack of encoder layers, each consisting of a self-attention layer followed by a feed forward network. We

adapt each attention head in the self-attention layer to resemble the structure of the relations defined by a tcKG. Thus, we do not need to calculate the pairwise attention for all inputs to the encoder, but can attend $\mathbf{r}^c$ only to $\{\mathbf{e}^s, \mathbf{r}, \mathbf{e}^o, \mathbf{e}^{c_1}, ..., \mathbf{e}^{c_{n_c}}\}$. Similarly, we can constrain the attention of $\mathbf{e}_t^s$ to $\{\mathbf{e}_{t-1}^s, \mathbf{r}_t^c, \mathbf{e}_t^o\}$ only. This is displayed by the diagonal arrows inside the first encoder layer in Figure 17.[14]

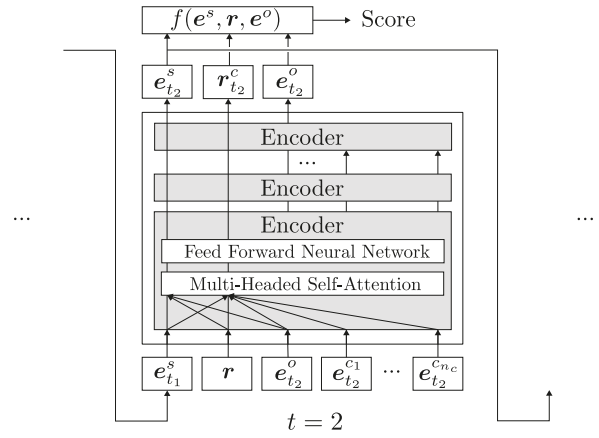

Figure 17: Inside the Encoders in the RETRA architecture

cmp. Fig. 16: Stacked encoder-layers, each with constrained multi-headed self-attention, followed by the scoring function which returns a scalar score for $(\mathbf{e}^s, \mathbf{r}, \mathbf{e}^o)$-triple.

In summary, the function $f_{context}$ is realized as multiheaded restricted self-attention and returns the contextualized relation embedding $\mathbf{r^c}$ that contains the situational context. The sequential context is contained in the contextualized subject embedding $\mathbf{e_t^s}$ by realizing the function $f_{sequence}$ as a recurrent neural network.

4.4.2 Training RETRA

To optimize the weight matrices in the feed-forward and self-attention layers using backpropagation, we need to measure the plausibility of predicted facts (using a scoring function ϕ) and its deviation from known facts given in the training data (using a loss function). In principal RETRA is independent of the choice of the scoring function and training objective (see Table 5 in [34] for an overview of

[14]The constraining inside the attention heads is an engineering choice and acts as an inductive bias. Any other constraining is possible including no constraining.

state-of-the-art KGE models and their scoring functions). Any scoring functions and training objectives can be plugged into RETRA as long as it allows to calculate a gradient.[15] We settle on the most common KGE training objective, namely link prediction, which we in "transformer terms" refer to as "object masking". The training target is to correctly predict e^o_{t+1} given e^s_t and r^c_t, where e^o_{t+1} is unknown. Thus, the weights need to be adjusted, such that the scoring function $\phi(\mathbf{e}^s_t, \mathbf{r}^c_t, \mathbf{e}^o_t)$ outputs a high score for the correct e^o_{t+1} from the training data (and a low score for all other entities). Using a soft-max function on the predicted scores for each e^os allows to calculate the cross-entropy loss against the correct triple and backprop the error.

We are only interested in optimizing the embeddings regardless of the scoring function provided. Thus, we compare the predictive performance of a given KGE model, including its scoring function, to using the same scoring function but transforming the embeddings to temporally contextualized KGEs.

4.5 Implementation and Empirical Testing

This section provides implementation details and reports empirical results from three diverse application domains. An overview of the features selected in each domain as tcKG facts is provided in Table 4. The SUMO dataset and the parameters used for training in the following experimental section, as well as the code to run the experiments, are available in our repository[16] on Github.com.

The proposed RETRA approach is implemented based on PyTorch's[17] Transformer Encoder layer, which provides an internal self-attention layer. As seen in Figure 17, we use an embedded triple $(\mathbf{e}^s_{t_1}, \mathbf{r}_{t_2}, \mathbf{e}^o_{t_1})$ plus its contexts $\mathbf{e}^{c_1}, ..., \mathbf{e}^{c_{n_c}}$ as input and assume the output to contain the contextualized embeddings $\mathbf{e}^s_{t_2}$ and $\mathbf{r}^c_{t_2}$. Of those embeddings, the contextualized subject embedding replaces or complements the global subject embedding in the next time-step. This is repeated over the whole sequence of experiences of e^s. By doing so, the subject embedding alters

[15] Also, many other self-supervised training objectives are possible. Starting from relation masking to temporal subject masking (mask subject e^s_t and condition the prediction on e^s_{t-1}).

[16] https://github.com/siwer/Retra

[17] https://pytorch.org/

based on the history of previous inputs and its current context.

One key feature of RETRA is its complementarity to existing KGE methods. In the use-cases we present here, we use the static KGEs and their respective scoring functions from three established baseline KGE techniques, namely TransE [10], SimplE [35] and HolE [52]. The implementations of the baseline models were acquired using the OpenKE[18] framework, which offers fast implementations of various KGE approaches. The focus of this work is not to obtain the best overall predictive performance but to show how temporal contextualization can improve existing KGEs. For that reason, we chose three basic and established baselines.

There are two possible setups to realize the RETRA model formulation. The first setup is a two-step approach that consists of pre-training the global embeddings, which are the fed into the RETRA model, whereas the second setup learns the global embeddings during training of RETRA.

4.5.1 Location Recommendation

For location recommendation, we re-use the New York City dataset[19], which was created and used for a different recommendation scenario before [82]. The data consists of check-ins from Foursquare[20], which is a location based social network. Every check-in consists of various pieces of information, including user, location, location type, country and time. The recommendation target is a *location* or *Point of Interest* (POI) for a particular user, given background knowledge about the locations and the history of *visits* or *checkIns* of users at POIs. The ranking is done by using a scoring function $f(\mathbf{e}^s, \mathbf{r}, \mathbf{e}^o)$ provided by the baseline KGE approaches. The result of a forward step in this scenario is a tensor containing the scores for every potential location in the data. This information plus the information about the known target location given in the training data serves for calculating the cross-entropy loss.

[18] https://github.com/thunlp/OpenKE
[19] https://github.com/eXascaleInfolab/LBSN2Vec
[20] https://foursquare.com

Location Recommendation - Experimental Design:

We consider the 'raw' setting provided in the data set, since we are treating every check-in as one distinct time-step. The data set contains 104,991 distinct check-ins, 3,626 distinct locations, 3,754 distinct users and 281 distinct types. For training and testing we were using a random 80 - 20 split of our data.[21] Users with only one check-in were not considered because a sequence of at least two check-ins is needed for contextualization. In addition to the input of a triple (e^s, r, e^o), we explicitly passed the preceding location and the current location's type as context. The check-ins are not uniformly distributed over users. There are many users with only one or two check-ins, and few users with a lot of check-ins (up to 4,069). The same pattern can be identified with the locations. This extreme imbalance makes this a very challenging task, since we assume that a longer sequence provides more information on a certain user's behavior than a short sequence would do.

Basic KGE approaches are unable to incorporate the inherent sequential and n-ary relational information provided by such a dataset and are thus fundamentally limited for this task. For both approaches, we have chosen the default number of dimensions (130) as the embedding size.

Location Recommendation - Experimental Results:

It can be seen in Table 5 that all baseline approaches have performance issues, which we attribute to the skewed distribution. "Imp" refers to the relative percentage of performance change compared to the corresponding baseline metric. Still, we use these approaches as our global baseline embeddings to see if it is possible to incorporate more information by modelling the sequence and the context information and thus obtain an increase in performance. When the baseline KGEs are combined with RETRA, we indeed obtain a considerable relative performance increase. Numbers in bold indicate the best results. While the overall performance is still low, the results show that the usage of sequential and contextual information for enhancing entity embeddings can improve the performance of standard KGE approaches by a factor of up to 15.

[21] Note that the splits and embedding dimension are different from the setup in Section 3

Approach	Hit@10	Hit@3	Hit@1
TransE	0.0100	0.0017	0.0004
SimplE	0.0077	0.0035	0.0013
HolE	0.0038	0.0004	0.0
RETRA + TransE	0.0203	0.0005	0.0001
Improvement	*103%*	*-70%*	*-75%*
RETRA + SimplE	**0.0592**	**0.0521**	**0.0194**
Improvement	*668%*	*1388%*	*1392%*
RETRA + HolE	0.0209	0.0005	0.0
Improvement	*450%*	*25%*	*0%*

Table 5: Metrics for the best runs of the baseline and combined approaches.

When testing different combinations of model parameters, we can observe that the learning rate has the strongest influence on the performance. Changing the number of transformer layers does not seem to have a big impact in general. Apparently, the interactions between features is not complex enough to require several attention layers. For all tested scoring functions, the combination with RETRA led to an improvement in performance.

4.5.2 Driving Situation Classification

Much progress has been made towards automated driving. One challenging task in automated driving is to capture relevant traffic participants and integrated prediction and planning of the next movement by considering the given context and possible interactive scenarios. Here, we define the problem as predicting the driving maneuver (e.g. following, merging, overtaking) of a vehicle given the current state of the driving scene. According to [40], approaches for vehicle motion prediction can be grouped into physics-based, maneuver-based and interaction-based. Interaction-based methods extend maneuver-based methods by modelling the dependencies between pairs of vehicles. Related work based on different deep neural network approaches and feature combinations for trajectory prediction has been

described in [41] in which surrounding vehicles and their features are extracted from fixed grid cells. In comparison, our approach uses relational data between the ego and foe vehicles. Our motivation is that explicit representation of triples might lead to improved modelling of interactions between vehicles.

Driving Situation Classification - Experimental Design:

We use SUMO[22] (Simulations of Urban Mobility), an open source, highly portable, microscopic and continuous multi-modal traffic simulation package to generate driving data. More than $50,000$ driving scenes of a motorway were generated. The vehicle parameters as well as driving styles were varied widely in order to simulate a large variety of vehicles and driving behaviors. This resulted in situations such as risky driving situations, abandoned driving maneuvers, unexpected stops and even accidents. We have developed a knowledge graph to represent the simulated data by entities (e.g. scene, situation, vehicle, scenario), relations between entities (e.g. isPartOf, occursIn, type) and their associated features (e.g. speed, acceleration, driving direction, time-to-collision). This resulted in more than 900 million RDF-triples with around 2 million scenes which comprise more than 5 million *Lane Change* and *Conflict* situations, respectively. It represents a valuable benchmark data-set for driving situation analysis. More information on the design and creation process of the data-set is available in [28].

Driving Situation Classification - Experimental Results:

We conducted two sets of experiments on the SUMO data, which both aimed at predicting the type of a conflict. We needed to make this distinction since the baseline KGE methods cannot use context and thus have to make predictions based on the situation-ID. Instead, RETRA learns a dedicated situation embedding based on the context and previous driving scenes. When experimenting with the different baseline KGE scoring functions, we noticed that a fully connected feed forward layer (FF) as a trainable scoring function performs better. The results are shown in the first four rows of Table 6. Since various SimplE implementation we tried did

[22] https://www.eclipse.org/sumo/

Approach	Sequence Length	Hit@3	Hit@1	MR	MRR
HolE	0	0.9366	0.5235	1.76	0.72
TransE	0	0.7668	0.2729	2.56	0.53
SimplE	0	-	-	-	-
RETRA+FF	0	**0.9946**	0.8060	1.23	0.89
RETRA+FF	5	0.9731	0.8212	1.23	0.90
RETRA+FF	10	0.9672	0.8382	1.17	0.91
RETRA+FF	15	0.9858	0.8455	1.17	**0.92**
RETRA+FF	20	0.9871	**0.8469**	**1.16**	**0.92**

Table 6: Results for the SUMO driving situation classification data set.

not scale to the size of this data set, we cannot report any results. The task was to predict the correct situation type, given surrounding traffic. All performance metrics (hit@k, mean rank (MR) and mean reciprocal rank (MRR)) indicate that context is crucial and more previous observations information improve the performance. Obviously, RETRA+FF considerably outperforms the baselines, even as a non-recurrent version. This is mostly due to its ability to contextualize a situation embedding which avoids the need for explicit situation-IDs.

Since the previous steps in time leading up to the current situation are potentially important in driving scenes, we specifically investigated the influence of previous situations on the predictive performance. The last four rows of Table 6 show how RETRA handles different numbers of recurrence steps by feeding in the preceding 5 - 20 driving situations leading up to the current point in time. It can be observed that the result improvements grow proportionally with the sequence length. This confirms the assumption that the history is important in driving situations and RETRA is able to exploit it.

4.5.3 Event Prediction

The *Integrated Crisis Early Warning System* [11] contains information on geopolitical events and conflicts and is a widely used benchmark for both static and

	Contextualized			Non-Contextualized		
Metric	TransE	SimplE	HolE	TransE	SimplE	HolE
Hits@1	0.519	**0.570**	0.537	0.264	0.264	0.299
Hits@3	0.691	**0.739**	0.703	0.398	0.398	0.463
Hits@10	0.821	**0.843**	0.822	0.532	0.532	0.623
Hits@100	**0.941**	**0.941**	0.940	0.775	0.775	0.840
MR	152.92	91.02	**88.00**	311.85	311.85	193.13
MRR	0.625	**0.669**	0.638	0.358	0.358	0.409

Table 7: Contextualized vs. non-contextualized KGE for different scoring functions on the ICEWS event prediction data set.

temporal KGE approaches. We specifically use this dataset to showcase how contextualizing can improve and generalize binary KGE approaches.

Event Prediction - Data Set:

For our experiments, we use the 2014 subset[23] of the ICEWS data as described in [24] as a basis, and add contextual information that we take from the original 2014 data[24]. In addition to the triples consisting of *Source*, *Event Text* and *Target*, we use the entities *Source Sector*, *Source Country*, *Target Country* and *Intensity* to contextualize the *Source*.

Event Prediction - Experimental Results:

The target is to predict the *target entity*, typically the organization involved in the event, given the *source entity*, aka actor, and the relation. In both setups, we optimize a cross-entropy loss by calculating scores for all possible triples in a query $(s, r, ?)$. The target is to produce the highest score for the original triple given in the ground-truth. In addition to using only the information presented in triples, we also consider contextual information for our training. This is achieved by passing all information through RETRA and using the contextualized subject entity for the query $(s_c, r, ?)$. In this way, the embeddings are learnt in such a manner that they

[23] https://github.com/nle-ml/mmkb/tree/master/TemporalKGs

[24] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/28075

contribute to the contextualizing given a binary scoring function from our baseline KGE methods. As shown in Table 7, using the contextual information results in a huge improvement in performance for all tested baseline scoring functions and evaluation metrics. This, again, indicates that context is crucial and RETRA is able to exploit it, regardless of the KGE scoring function used.

4.6 Conclusions

In this chapter, we propose the modeling template **tcKG** for temporally contextualized KG facts (addressing *RQ2*) and **RETRA**, a Deep Learning model intended to transform static Knowledge Graph Embeddings into temporally contextualized ones, given a sequence of tcKG facts (addressing *RQ3*). With RETRA we tackle two limitations of current KGE models, namely their lacking ability of taking situational context into account and capturing the sequential evolution of an entity embedding, given its subjective history of similar previous events. Our experimental results on three data sets from diverse application domains indicate that existing KGE methods for global embeddings can benefit from using RETRA to contextualize their embeddings. We could also demonstrate that both, situational and temporal context, enhance performance considerably.

This section discusses and summarizes the ideas and approaches that were introduced in response to research questions 2 and 3.

RQ 2: How can the KG formalism be extended to fully capture sequential and situational context?

RQ 3: How can static knowledge graph embeddings be transformed into contextualized representations?

A key concept of the introduced formalism and the corresponding embedding mechanism is that the change of an entity over time can be modelled. The underlying idea is that there exists a global representation of an entity that alters its state depending on past interactions. This temporal or sequential context reflects the concept of the *episodic memory* introduced in [67].

In the implementation of RETRA for the empirical experiments, the current situational context is expressed within the contextualized relation r^c. This is one possible perspective on situational context. Here, the idea is that the current context affects only the current interaction and is not carried on to the next interaction. If it is desired to carry on this situational information too, $f_{context}$ could be also applied to e^s instead of r. The situational context can be interpreted as the *semantic memory* component [67].

Our proposed approach for answering RQ 2 is to introduce the **tcKG** formalism. Extending the triples to n-tuples that in addition to (e^s, r, e^o) also contains context entities e^c and a timestamp t allows for modelling both situational and sequential context. Note that in our implementation of RETRA (referring RQ 3), the timestamp t is treated differently than it is normally in temporal KGE approaches. The common perspective is that t is embedded in its own vector space which makes it possible to use in a temporal scoring function. In our approach, t is used to create an ordered set of interactions, which is processed recurrently, such that $e^s_{\text{t1}} \neq e^s_{\text{t2}}$. In most temporal KGE approaches, an entity e and its vector representation is always the same, such that $e_{\text{t1}} = e_{\text{t2}}$.

In contrast to Hypergraph approaches, where the scoring function ϕ is defined over an n-tuple with n as the arity of a relation r, our approach is compatible with binary scoring functions. This is achieved by introducing the f_{context} function which takes as inputs the context entities $\mathbf{e^c}$, a relation $\mathbf{r}$ and returns a contextualized relation $\mathbf{r^c}$. Doing this allows us to utilise binary scoring functions like SimplE or TransE even in scenarios where more than two entities are involved.

Finally, we briefly discuss the experiments and their focus with respect to different aspects of tcKG and RETRA. Although all three experiments show the elements of the newly introduced formalisms, there is always a focus on a certain aspect.

- In the *Location Recommendation* scenario, the focus lies on demonstrating the ability to contextualize pre-trained embeddings. As an example, this means that the SimplE embeddings are optimized independently from $f_{context}$ and $f_{sequence}$. Instead of training the embedding and the context

and sequence jointly, training RETRA in this case means that only the parameters of those two functions are updated. This is an important proof-of-concept for the underlying ideas, because the observed improvement is not achieved by conditioning f_{embedd} on $f_{context}$ and $f_{sequence}$, but rather by leveraging pre-learnt entity semantics and their contexts.

- The focus of the *Driving Situation Classification* experiments is on the importance of sequential context. Although we make full use of the ability of RETRA to capture situational context as well, we explored to which degree a longer history is exploitable in this scenario. For this, we artificially limited the amount of available previous interactions and compared the classification performance with respect to the length of the interaction sequence. Across 3 out of 4 observed metrics, there is a clear trend that a longer sequence leads to an increased performance.

- Eventually, the *Event Prediction* scenario focuses on how situational context can be used within binary KGE approaches. By applying $f_{context}$, rich situational context can be aggregated within either the relation or the subject entity of a triple. This has the advantage that additional information can be easily exploited without changing to a scoring function of higher arity. This is especially useful in scenarios where situational context is not available for all triples, such that a fixed arity relation is unsuitable to capture all contexts.

The next chapter applies the newly introduced contextualisation mechanisms in a scenario where a dynamic subject entity acts in a static environment and demonstrates their application in this scenario. This includes the usage of non-relational data and alternative views on situational context. Additionally, these views and their implementations are tested in an empirical study based on the new scenario.

5 Contextualisation in Static Environments

In the previous chapter, the tcKG formalism was introduced and used to model subject dynamics in three different scenarios. They all had in common that the *environment* in these scenarios could be characterised as dynamic. This implies that for every given fact e^s, r^c, e^o and timestamp t, the context entities in an otherwise similar triple might vary:

$$t_1 : e_{t_1}^{c_1}, ..., e_{t_1}^{c_{nc}} \neq t_2 : e_{t_2}^{c_1}, ..., e_{t_2}^{c_{nc}}$$

In a scenario where a dynamic entity is acting in a static environment, the opposite is true, since the only change is happening within the subject entity which interacts with static objects:

$$t_1 : e_{t_1}^{c_1}, ..., e_{t_1}^{c_{nc}} = t_2 : e_{t_2}^{c_1}, ..., e_{t_2}^{c_{nc}}$$

This property of static environments has implications for the previously introduced view on situational context. We can speak of a static environment when all object entities are passive, meaning that the only actor is a subject entity that interacts with these passive object entities. The sequential context in such a scenario is defined by the order in which the subject entity interacts with the passive objects, and the frequency of these interactions. For instance, if a subject entity e_s interacts with the object entities in $seq_1 = (e_{o1}, e_{o5}, e_{o2}, e_{o3}, e_{o4})$ (in this order), the sequentially contextualised $e_{s_{seq1}}$ vector representations would be different than $e_{s_{seq2}}$, which interacted with the sequence $seq_2 = (e_{o1}, e_{o4}, e_{o2}, e_{o3}, e_{o5})$. Computationally, this sequential contextualisation can be realised by the approaches introduced in the previous chapter.

Unlike with sequential context, computing the situational context requires to make different assumptions. Chapter 4 roughly considered situational context as potentially anything that happens outside the primary *subject - object* interaction. These could be phenomena like the current day of the week, current affiliation of the object entity, or simply the proximity of other entities. Some of these, like

the current day, are obviously undergoing frequent changes, whereas others are less prone to changes. The key concept, however, is that 1) the situational context can change and 2) that the situational context influences the current interaction. None of these observations hold in a static environment. Once the stimuli in our example are created, they never change. Therefore, the stimulus can be understood as a closed world. The consequences for situational context can be summarized as follows:

- In a static environment, the situational context for a given *subject, relation, object* triple is always the same, independent of t.
- Due to this, situational contextualisation always leads to the same contextualized relation, basically reducing it to a static Embeddding.
- This makes situational contextualisation for each time step redundant.

Based on these observations, we explore how situational context can be understood in a closed world, how the interplay between sequential and situational context can be modelled (addressing **RQ 4**) and how it is possible to expand the situational information by using representations from different modalities (addressing **RQ 5**).

From the perspective of a dynamic KGE approach like the previously introduced RETRA, the KG serves as an intermediate representation of the underlying situation in the physical world in terms of the involved entities and their relations. In the previous chapters, context (both sequential and situational) was understood as additional information that is modelled within the KG formalism. Although this view showed to be appropriate based on the conducted experiments, situations are possible where situational context is only available in modalities different from a KG format. These modalities could in principal reach from images, over texts to sounds, as long as they represent the same situation. See Figure 18 for an illustration of this view.

Evaluating these ideas requires a suitable real-world scenario where all object entities are passive, and a corresponding task that can be solved as a link prediction problem. For this, we propose the novel task of Perception-guided Crossmodal

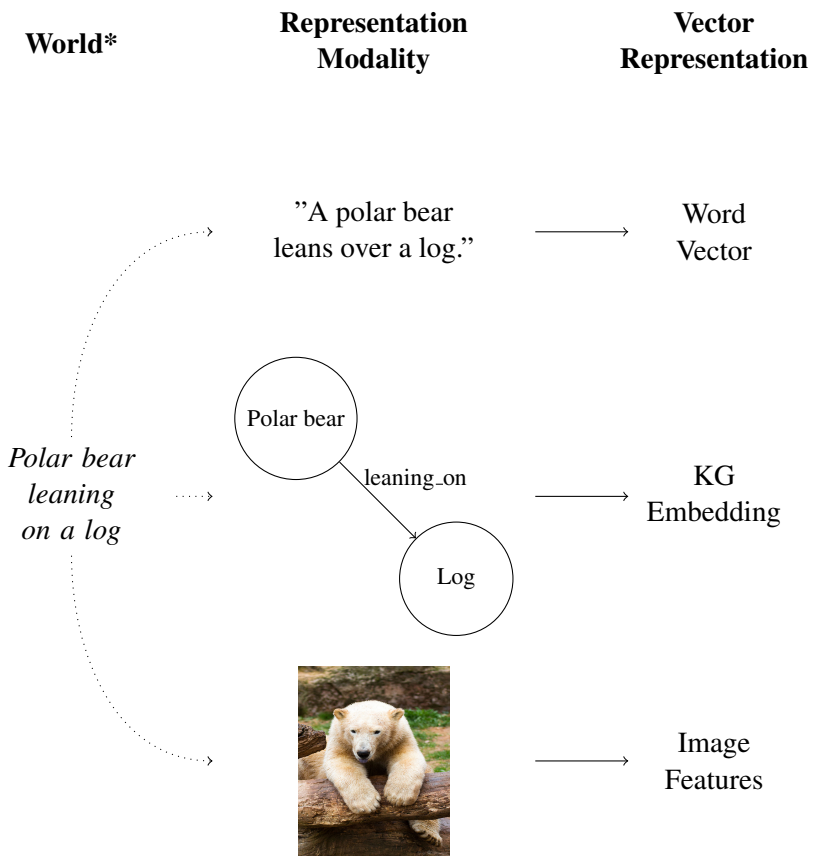

Figure 18: Representation modalities

*In substitution of an *actual* polar bear, a textual representation should suffice here.

Entailment (PCE) which attempts to predict how a test person assesses the coherence of different modalities of a multimodal stimulus (e.g., an image with captions) given the test person's fixation sequence. We collect the first benchmark data set for PCE by tracing a test person's eye movements while the person is judging if the central visual objects are mentioned in the image's captions. In our empirical studies, we model the participants' perception and final decision in the tcKG formalism by assuming the participant as the subject entity that interacts with the contents of the stimulus that we assume as the objects. Since the objects and their contexts in

this scenario never change, this setup is well suited for studying the implications of a static environment for embedding situational context.

Before we describe the empirical studies in detail, we first introduce closely related work which has not been covered in the previous chapters. This is followed by a detailed description of the data collection process and a section that motivates the usage of non-KG modalities in the experimental sections.

5.1 Related Work

There are multiple related research areas that have to be taken into consideration. A large and active area with recent contributions is the general field of multi-modal (in our case visual-linguistic) representation learning. Since we borrow techniques from this field to enable the usage of non-relational data in Knowledge Graph contexts, the main focus of this section will be on this area. Other related work will be covered in the following subsection, where we discuss previous work on eye tracking in psychology and its relevance for machine learning and our work.

Because the present study does not cover other modalities, we refer to [79] for a detailed overview over models that operate on modalities that go beyond vision and text. The most prevalent ones are video [60] and speech [2], but there are also models that process graph structures [14] or music [43] as additional modalities.

5.1.1 Foundation Models

There are two key techniques on which most recent language models are built on: the attention mechanism [4] and the transformer architecture [68]. Despite their impressive capabilities, there are certain disadvantages that come along with models like LaMDA [64], GPT-3 [12] and beyond. Firstly, the amount of resources needed to train a modern language model in both time and storage is tremendous. Secondly, there are ongoing arguments that question the abilities of such models to solve tasks that appear trivial to humans, but are hard to solve for machines [26]. At the same time, there is work which shows the potential benefits of inductive bias in multi-modal environments [57] and work that supports the argument that there is a similarity in human and machine perception [13].

5.1.2 Visual-Linguistic Transformers and Tasks

A multitude of recent approaches in visual-linguistic representation learning are available. Most of them have in common that they combine the BERT [20] architecture with a vision module. Models that fall in this category are, for instance, VilBert [48], VL-Bert [59] and Visual Bert [42].

Others, like LXMERT [62], Uniter [15], SimvLM [72] and, most recently, OFA [70] use other transformer-based architectures. They all have in common that they are evaluated at least partly on one or more of the benchmark tasks provided by datasets like Visual Genome [38], MS-Coco [45], or Flickr30k [56]. Normally, the tasks are divided into two sub-categories which are briefly described here.

Pre-training tasks: Masked Language Modeling, Masked Region Modeling, Textual Grounding

Downstream tasks: Visual Commonsense Reasoning, Visual Question Answering, Natural Language for Visual Reasoning, Region to Phrase Grounding, Visual Entailment

This list is not exhaustive, but contains tasks most commonly referred to in visual-linguistic representation learning. The task with the highest similarity to the Perception-guided Crossmodal Entailment task that we propose in this work is Visual Entailment [77]. In Visual Entailment, an image text combination functions as a premise/hypothesis pair and the task is to classify whether the hypothesis is entailed in the premise, contradicts the premise, or is neutral with regard to the premise.

5.1.3 Semantic and Episodic Memory

A concept that we only mentioned very briefly in Chapter 4 is the idea of semantic and episodic memory. The current chapter expands on these ideas and interprets them in the light of modelling a participant in the PCE task. Here, we focus on certain aspects of the aforementioned memory types that overlap with the model architectures we introduce. For a more in-depth view, refer to [67] and [3].

Tulving introduces the concept of an episodic memory that exists next to a semantic Memory (a term that was introduced by 1966 (Quillian)). Semantic memory can be understood as a *network of concepts, words and images, capable of making inferences and comprehending language*[67]. It exists next to the Episodic memory, in which *information about temporally dated episodes or events and the temporal-spatial relations among these events*[67] are stored. In summary, semantic memory stores knowledge that goes beyond personally experienced unique episodes, whereas episodic memory focuses on the order of events. These concepts provide a fitting analogy for the models we devise in this section, since we are aiming at representing human perception processes.

5.1.4 Machine Learning on Eye Tracking Data

Historically, eye tracking data has been most commonly used as a tool in the psychology of (visual) perception, where it has been used to determine how visual perception works both physiologically and psychologically [27]. In other lines of psychology-related work, eye tracking data is used to to train statistical models for predicting behavior in binary choice settings[36] [37] [63] [9]. In contrast to our work, the approaches described above use directly derived information from the eye tracking sequences (i.e. dwell time, revisits) [55] as input for their statistical models, while we focus on symbolic representations of multi-modal AOIs that we use as input for deep learning models.

In summary, there are important aspects in which our work differs from both the line of Visual-Linguistic Transformer research and the line of eye tracking based research in psychology. In contrast to Visual Entailment tasks, we have an additional behavioral aspect that has to be addressed in our models. In Visual Entailment, there is always a ground truth for a given image/text pair, whereas in our case, different participants might have opposing evaluations of a given image/text pair, which is why our models have to reflect this aspect in their decisions. On the other hand, the models in Psychology do account for individual behavior, but are purely statistical models (as opposed to our deep learning based approaches), use the data differently (feature engineering on the basis of some properties in the eye

tracking sequence, vs. representation learning techniques on our end) and do not account for multi-modality.

Most importantly, no previous work attempts to fuse the two research areas in order to derive perception-guided machine learning models.

5.2 Perception-Guided Crossmodal Entailment

This section describes in detail how we obtain the data for both the perception-guided ML model based on eye tracking data, as well as the content based transformer models. For both models, the identical prediction targets are used, namely the task of (Perception-guided) Crossmodal Entailment (PCE).

As basis for the eye tracking study, we selected a set of multi-modal documents from the Visual Genome data set [38]. Visual Genome (VG) consists of 108,249 images that are annotated on several layers, from regions to objects, represented in a graph structures. We leverage the underlying scene graph to rank entities depicted in an image by their centrality degree. Based on this ranking, we chose regions that are connected by their contained entities and retrieve the associated caption. We extract up to three regions for each picture which, due to the procedure just described, are connected. The region captions then serve as textual description of the picture. The region descriptions combined with the image constitute the multi-modal document used for further annotation.

5.2.1 Crossmodal Entailment Task Selection

To ensure that the final eye tracking data is useful for further experimentation, a pre-test was conducted to determine the inter-annotator agreement with regard to our Crossmodal Entailment task. Concretely, we chose to show the question *"Are the central objects in the image mentioned in the caption?"* to the participants since it requires crossmodal reasoning and helps to assess human perception. 250 candidate documents were created and tested with three annotators (research assistants). Out of the 250 candidates, 153 received a perfect inter-annotator agreement and were thus chosen as the final eye tracking data set. For the eye tracking experiments, all stimuli were further annotated with AOIs, which later serve as basis

for the mapping between the extracted transition matrices and the input for the machine learning architectures. In the caption, all tokens were assigned one respective AOI, whereas the annotation of the images followed the annotated objects. Since the object annotations are rectangular and sometimes overlapping, not all visual AOIs could be drawn exactly over the objects. In such cases, the main objective was to capture the entity. Before the eye tracking study started, all stimuli were again checked for errors in the captions and object annotations. If spelling mistakes were found, they were corrected and documented. Objects with unintuitive VG-annotations were removed. Again, this was documented so that all steps could be reconstructed afterwards.

5.2.2 Eye Tracking and Human Assessment Recording

Overall, 109 participants in the age range of 19-61 years participated in the study, which took place in the eye tracking laboratory of the Media Studies department at our institution. The average age was M_{age} = 25.4 years, the majority of the participants were female (F= 77, 70.64 %; M= 30, 27.52 %). They were recruited via several open calls on university newsletter sites, via online advertisements on social media (Facebook, Instagram), and on printed posters on campus. The experiments were conducted between 5 May and 14 July, 2022. A small participant remuneration was offered (5 € per participant). Participants were mainly students and staff of Trier University. Participation requirements included basic knowledge of English to ensure they understood the textual stimuli in the experimental task.

A total of 153 text-image-stimuli were selected to be shown to the participants. To ensure that the length of an eye tracking session was not too long to retain the participant's attention, the participants were divided into three groups: group A ($N_{participants}$=37; $N_{stimuli}$=50), group B ($N_{participants}$=35; $N_{stimuli}$=50) and group C ($N_{participants}$=37; $N_{stimuli}$=53). With exception of the different stimuli, all other experiment settings were kept constant: a short instruction slide was shown followed by 50-53 blocks consisting of a text-image-stimulus and a following question slide. The blocks of stimulus and question slides were shown in a randomized order. For the study, the experiment was programmed and analyzed in the software iMotions

Figure 19: Visualisation of human attention

Solving the task: *Are the central objects in the image mentioned in the caption?* The red areas are showing the focus of human attention.

(Version 9.1.5). The total duration of the experimental task was approximately eight minutes (excluding the time for initial calibration).

5.2.3 Symbolic Fixation Sequence Extraction

From the eye tracking study, we obtain the participant-specific fixation sequences for all stimuli. An aggregated example representation of the perception process is shown in Figure 19. In addition to the information about the participant and stimulus, each fixation contains information about its location (x and y coordinates), its duration, dispersion and the associated annotated region. For instance, Table 8 lists how the sequence in Figure 26 is represented. All fixations are ordered by the time of their occurrence, meaning that they are ordered chronologically.

Note that in this raw data export from iMotions, there are artifacts like in row

Table 8: Fixation sequence for participant EWCX and stimulus ID 2412873.

Row Nr.	AOI	Coordinates	Duration	Dispersion
1	vis_wall	865.44, 346.46	100.00	0.23
2	txt_wall	709.13, 29.92	158.32	0.16
3	off	709.13, 29.92	158.32	0.16
4	txt_Rock	658.71, 39.78	191.67	0.12
5	txt_wall	760.30, 32.51	258.33	0.18
...	...	...	...	...
18	vis_ground	976.25, 859.95	166.66	0.28
19	vis_zebra	718.85, 823.35	166.66	0.26
20	vis_wall	988.11, 431.43	250.00	0.23

numbers 2 and 3, where the same fixation is assigned to two different AOIs. In cases like this, we remove the non-specific entry for AOI *off*, which normally is used to describe a fixation that is on none of the annotated AOIs. The coordinates represent fixation centers calculated by iMotions, which is why they are floating point numbers instead of integers, as one would expect for pixel values.

Apart from the already mentioned issues at the annotation level, there are other potential sources for noise in the data, that could not be avoided fully:

Overlapping annotation: Due to the manual annotation process, adjacent AOIs may overlap. If a fixation is placed on a coordinate with two (or more) AOIs, the same fixation will occur in the data once for each annotated AOI on this position.

Participant always picks the first answer: If this co-occurs with very short fixation sequences and duration, this might be an indicator that the participant does not seriously attempt to solve the given task.

Participant shows unusual patterns: If multiple revisits of a textual element are recorded, this might be an indicator that the word in question is unknown to the participant.

Imprecise calibration: Although the eye tracking setup is individually calibrated for every participant, too much head movement can lead to imprecise eye-

movement measurements. This in turn may lead to misplaced fixation centers.

In summary, the data we obtain consist of the created stimuli (images, annotated AOIs and corresponding captions) and a participant specific fixation sequence for each stimulus. Overall, the acquired data set contains 5,400 unique fixation sequences, where each sequence represents a perception pattern of a participant generated when parsing the stimulus. The fixation sequences contain a total of 148,100 identified fixations, on average 27.42 fixations per stimulus exposure.

5.3 Non-Relational Data in Knowledge Graphs

This section briefly describes the perspective on using non-relational data in KG contexts. In addition to the obtained symbolic fixation sequence, our data also contains the corresponding image regions and the textual description from the caption. Symbolic representations of the image regions and words from the textual description can be directly used in a dynamic KG to model how the participant perceives the stimulus. This KG can then be embedded into vector space so that tasks like Link Prediction can be performed. The potential issue with this approach is the loss of information when reducing an image to a simple symbolic representation.

The alternative approach and focus of this section consists of using modality-specific embedding methods and translating the resulting vectors to a shared space with the KG embeddings. The intuition behind this lies in the rich semantics that are contained in an image feature vector or a word vector. These vector representations carry notions of similarity, class membership and further implicit world-knowledge. The following sections showcases this on the example of ResNet [31] image features and BERT [20] word vectors. These are the example models that are used in the later empirical study. The reason for choosing them are their accessibility as pre-trained models and proven reliability. Figure 20 shows the alternative views on how to integrate non-relational data into a KG perspective on the example of the image modality.

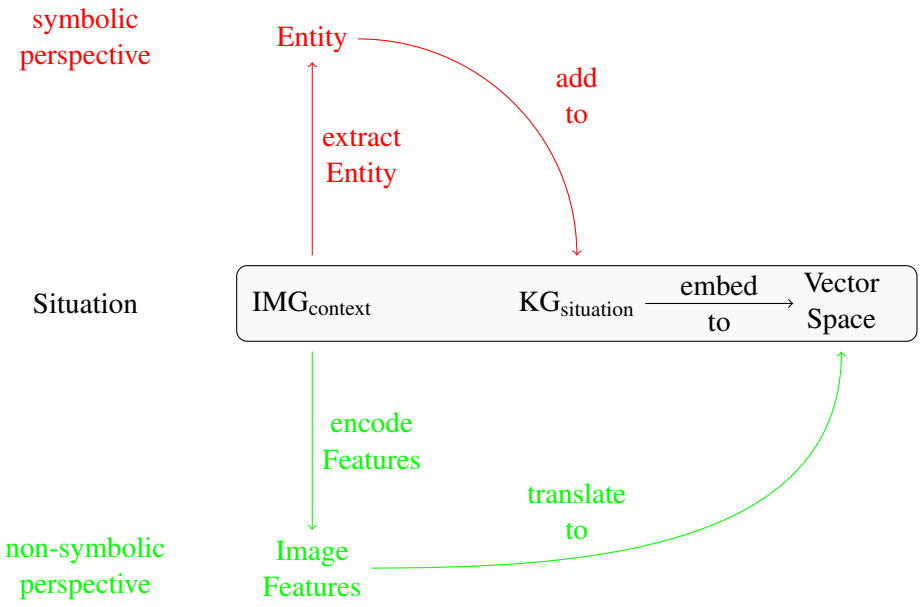

Figure 20: Non-relational data for KGs

In the symbolic perspective (red), an entity is extracted from the context image and simply added to the existing $KG_{situation}$. In the alternative non-symbolic perspective (green), the image is encoded into features and translated to a common vector space with the Graph Embeddings.

5.3.1 ResNet Image Features

Originally, ResNet is a model architecture for image classification and is trained to correctly classify an image in one of the 1,000 ImageNet [19] classes. The classes range from different animals, over specific dog breeds to more broader objects like cars or trees. It can be repurposed to function as an image feature encoder by ignoring the final classification layer and instead assume the last internal image representation as output. These representations have the advantageous property that a strong similarity between two feature vectors indicates that both underlying images belong to the same class. This means that these image features already contain an implicit notion of class membership in one single vector representation. In a KG embedding, this information would need to be learned from an explicit fact like $typeOf(entity, class)$.

5.3.2 BERT Word Vectors

In short, BERT is a transformer based language model that is trained on a masked language modelling task. The goal of this task is to correctly predict a masked out word in a sentence on the basis of its neighboring words. The key advantage of BERT over other (static) word representations is that every word vector is contextualized by its neighboring words. The resulting word representations carry semantic and to some extend even syntactic information. Additionally, as a result of the contextualization, the vector representations of words with multiple meanings are disambiguated.[83]

The motivation for incorporating non-relational data into KG scenarios are as follows:

Lack of KG data: There often is no or no sufficient contextual knowledge available in the KG modality. The formalisms and approaches from the previous chapters rely on the availability of contextual information in a KG format. The proposed approaches modify the commonly used triple-based approaches by introducing the possibility of contextualising entities. However, these contextualisation functions require symbolic entities (or their respective vector representations). If such symbolic entity representations are not available, contextualisation is not possible. With the presented perspective on non-relational data in KGs, contextualisation is possible on the basis of different modalities.

World Knowledge: Pre-trained models that produce vector representations for different modalities are widely accessible and easy to deploy. These vector representations often carry rich semantics that go beyond symbolic entities. In KGs, nodes are similar if they have similar relations with other entities. By using pre-trained encoders for modalities like image and text, the notion of similarity between entities is already incorporated in their respective vector representation.

5.4 Experimental Setup I

To evaluate our approaches, three different models with focus on different aspects of the data were implemented (see Table 9 for an overview). The focus of the *LSTM* [32] model is on the aspect of sequential context. The participant as subject entity interacts with the symbolic representation of the stimulus content as object entities, as indicated by the eye-tracking data. This experiment serves as baseline to which we compare the models that contain non-relational data. In contrast to this symbolic view, the focus of the *Transformer* model is on non-relational situational context in form of pre-trained feature representations of the stimulus. The participant as subject entity interacts with all stimulus contents, but no notion of order is taken into account. The goal is again to train a model that can predict a participant's evaluation of the *caption - image* combination. Finally, as a combination, we train an *Ensemble* model that learns a combined view.

The task for all models is to predict each participant's evaluation of the text-image combinations. We understand this problem as contextualising a subject entity (the participant) given a stimulus (in varying modalities) such that the subject entity evaluates the *caption - image* combination as one of the following outcomes:*1)* ***yes****, the caption mentions the central objects in the image*, *2)* ***no****, central objects in the image are not mentioned* and *3)* ***unclear***.

The data is divided into training, evaluation and test sets in a random 80/10/10 split. All the models that are described in the following sections are optimized with AdamW [47] optimizer on a Cross-entropy loss and trained for a maximum of 30 epochs. The final model is chosen as the one with minimum loss on the evaluation split. If not stated otherwise, we use the PyTorch [54] (version 1.11) implementations for all building blocks in our models. We conduct all experiments on the GPU server at our institution that is equipped with Nvidia RTX 2080 (11 GB) GPUs and V100 (32 GB) GPUs. Depending on the batchsize and model family, training and evaluation of one model takes up to a maximum of about 40 minutes. All models fit on the 11 GB of memory that is provided by the RTX 2080. The hyperparameter ranges that were tested to find the best performing Ensemble and best performing Perception-guided Transformer are given in Table 10. All

Table 9: Overview of the implemented models and their respective focus.

Model	Focus	Representation
LSTM	Sequential Context	Symbolic representation of AOIs
Transformer	Non-relational Situational Context	Pre-trained image and word embeddings
Ensemble	Sequential and Situational Context	Symbolic and pre-trained features

Table 10: Tested hyperparameters and their ranges.

Parameter	Range
Learning Rate	1e-04, 5e-04, 1e-05, 5e-05
FF-Dim	32, 64, 128, 256
Embedding Dim	8, 16, 32
Batch size	16, 64, 128

possible combinations were evaluated.

Although we model the PCE task as a KG problem, we use the more general classification problem metrics *Accuracy* and the *F1 score* for the evaluation. The possible outcomes of the participant's decision are restricted to only three different outcomes, which makes the common Link Prediction metrics (MR, MRR, Hits@k) not useful for this task.[25] Instead, F1 is reported to account for the uneven distribution of decision outcomes across the data. Additionally, because of the rarity of the *unclear* class, we test all models on two different settings, one with all classes present (3 classes) and one with the *unclear* class excluded from evaluation.

5.4.1 Sequential Contextualisation

Sequence: Conceptual View

In the episodic view, the subject entity (the participant) evolves over the perception process of the symbolic AOI representations in a stimulus. The initial dashed

[25] Note that Accuracy and Hits@1 are equivalent

subject entity denotes the participant in a *neutral state*. In the last time step, the sequentially contextualized participant decides if the caption fits the image. The $decides()$ relation objects are restricted to the three possible outcome scenarios. The sequentialisation function f_{seq} is realised as an LSTM model and the scoring function ϕ as a feed-forward layer. Figure 21 shows the sequential contextualisation process as a KG representation. This view resembles the *episodic memory* of the participant, where the focus is on the order of events.

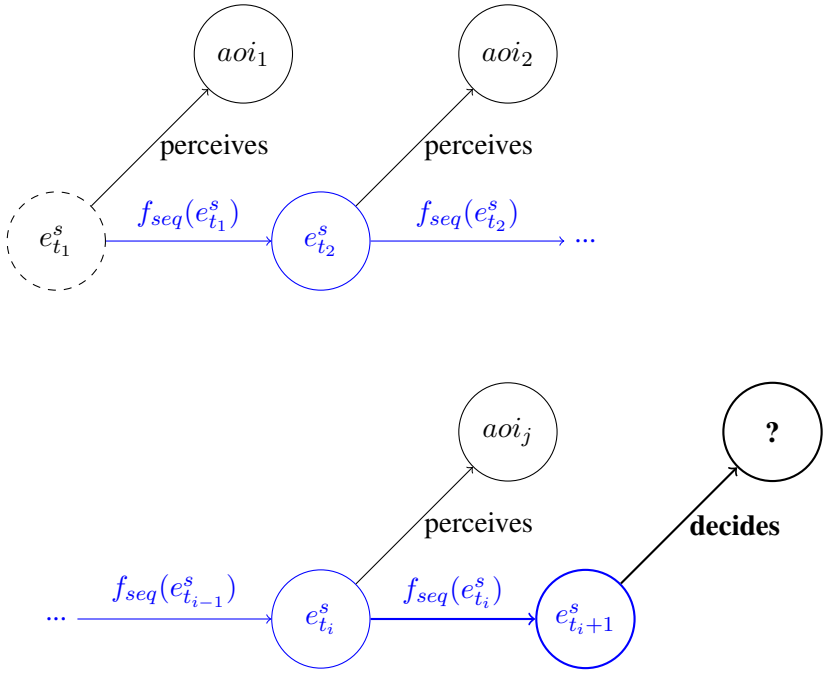

Figure 21: Episodic view

The blue colored subject entity *memorizes* which object entity has been observed. This resembles the concept of *episodic memory*, where the emphasis is on the order of events, rather than on the meaning of perceived entities. Note that current the object entity and relation are also passed through f_{seq} at every t and are only omitted due to space restrictions in the figure.

Sequence: Implementation Details

For this setup, we implement a unidirectional LSTM model that only receives symbolic representations of the current participant and the AOIs as a sequence. Here,

the LSTM is a realisation of the sequential contextualization function f_{seq}, which models the evolution of the participant entity during the perception process. If we take the sequence from Table 8 as an example, each row corresponds to one step in the LSTM. In this example, the symbol denoted *vis_wall* would correspond to the input in the first step t_1, *txt_wall* to t_2, and so on until the last element of the sequence *vis_wall* is reached[26]. Before feeding the sequence into the LSTM, we put the one-hot encoded elements through an embedding layer that outputs a dense low-dimensional vector representation of the elements in the AOI sequence. Additionally, we apply the same technique to create embeddings for the participants. As a final step we concatenate each AOI sequence element with the current participant embedding and put it through a feed-forward layer to receive a combined representation, capturing *Who is looking at which AOI*. This final representation is then used as the input for the LSTM, of which we take the final hidden state and perform the classification by running it through a final feed-forward layer which serves as the scoring function. During training, the Embedding layers are also updated, so that the resulting representations are fitted to the task. Overall, there are 837 unique AOI identifiers (i.e. the text representations like "vis_wall" or "txt_zebra") in the dataset. The symbolic AOI representations are not stimulus specific, but rather on a conceptual level, which is why the "vocabulary" is relatively small.

5.4.2 Situational Contextualisation

Situation: Conceptual View

In the semantic view, the participant as the subject entity is contextualized given the stimulus AOIs which are interpreted as situational context. The situational contextualisation function f_{sit} is realised as a transformer encoder with its self-attention. The final link prediction setup is the same as in the sequential view. Figure 22 denotes how the context entities affect the participant embedding, which functions as the subject entity. This view resembles the concept of *semantic memory*, where the participant's knowledge about the world is represented by the *meaning* of the

[26]Note that we do not add any additional information to the sequence, so that representations are purely symbolic

available context entities. This meaning is encoded in the feature vectors of the corresponding modalities (either text or image).

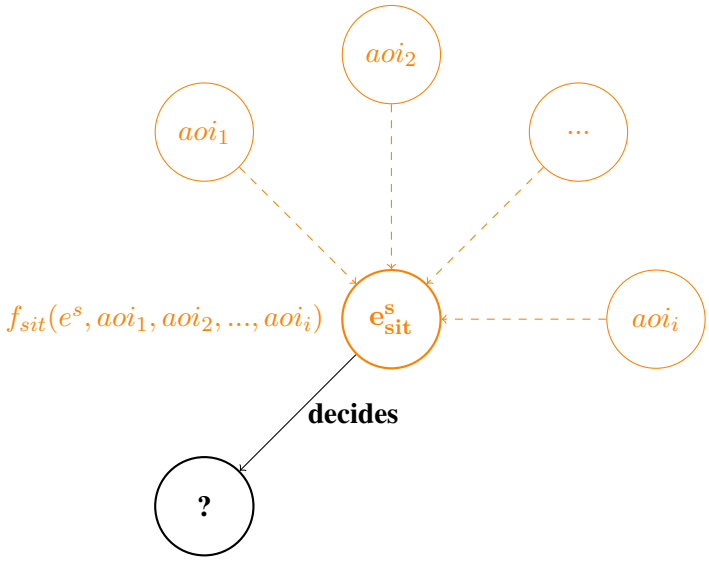

Figure 22: Semantic view

The situational context and the affected subject entity are colored in orange. This view resembles the *semantic memory*, where the focus lies on the meaning of the perceived entities.

Situation: Implementation Details

The fundamental building block for this baseline is a transformer encoder, which comprises multiple attention heads and layers. Figure 23 shows the baseline model, which can be extended and adapted to different tasks by adding additional layers on top. We use pre-trained BERT[27] [20] embeddings that are obtained from the *Huggingface Transformers* Package [75] as representations for the linguistic inputs (the captions) and a pre-trained ResNet50[28] [31] for obtaining image features of the pre-defined regions. The reason for using pre-defined regions instead of applying an object detector as in most state-of-the-art approaches is to ensure a one-to-one mapping between the symbolic eye tracking representation and the non-symbolic

[27] https://huggingface.co/docs/transformers/v4.28.1/en/model_doc/bert
[28] https://pytorch.org/vision/main/models/resnet.html

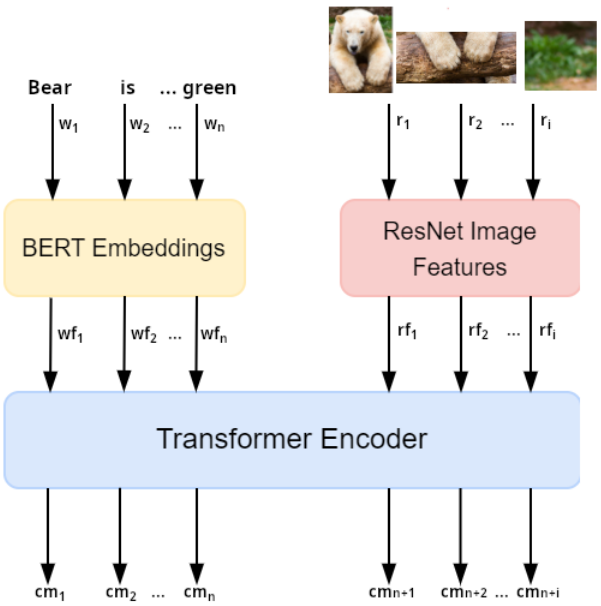

Figure 23: Baseline multimodal transformer

Textual input is referred to as w and image regions as r, respectively. The outputs marked as cm are cross-modal embeddings carry both visual and linguistic features

contextual input. Note that the ResNet and BERT weights are not updated during training.

For the experiments described below, we replaced the classification token by a participant embedding. The main function of this is to let the model know who is watching the current stimulus and model how the stimulus contents influence the participant's decision. There is also a single feed-forward layer between the ResNet image features and the transformer encoder, which is used for downsizing and translating the image feature dimension to match with the BERT Embedding dimension.

For the final decision, the participant embedding is put through a single feed-forward layer with three output dimensions that correspond to the three possible outcomes. This represents the scoring function in a link prediction task, where the

missing node in the triple

$$evaluates\_stimulus(participant, ?)$$

has to be predicted correctly. Conceptually, this view models how the stimulus contents affect the participant by providing situational context for the aforementioned relation.

5.4.3 Ensemble Model

Ensemble: Concept

On a conceptual level, this model reflects how the situational and sequential contextualisation both contribute to the final evaluation. The sequentially contextualized subject entity represents the process of perceiving the stimulus contents, whereas the situationally contextualized subject reflects the broader meaning of the stimulus contents. Figure 24 visualizes the decision process in the Ensemble model.

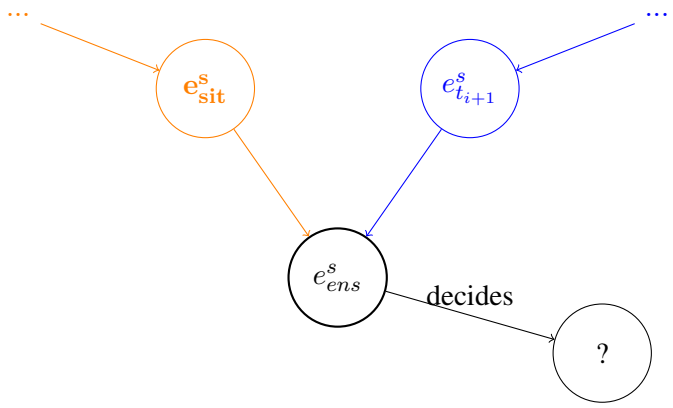

Figure 24: Ensemble concept

The situationally contextualized entity representation e^s_{sit} and the sequentially contextualized entity representation $e^s_{t_{i+1}}$ are jointly trained and combined into one entity representation that makes the final decision.

Ensemble: Implementation

In addition to the perception-guided approach of the LSTM and the content-based approach of the Transformer model, we introduce a third model as a combination of both. In this Ensemble model, we simply combine both of the models described above and use the average of both classifications as a final classification. This has been shown to be sufficient in our case, where we train the Ensemble parts in a joint fashion and do not require to introduce additional trainable parameters. By training both the LSTM and Transformer based models in a jointly fashion, we assume that the model learns how to combine the episodic and semantic view in order to make a correct prediction.

5.5 Experimental Setup II

To avoid training two models, we propose an alternative to the Ensemble model that only adapts the existing Transformer model. We experiment with the direct application the fixation sequences in form of transition matrices and adding them to the attention weights in the multimodal Transformer model. The intention behind this is to find an alternative way of combining the situational and the sequential data in one model. We call this a Perception-Guided Multimodal Transformer.

Unified Perspective: Conceptual View

In the unified view, the participant representation (e^s_{ctx}) is contextualized by the stimulus contents, with an additional notion of sequence between the AOI representations. The sequence information is realised as the Transition Matrix T, which denotes the amount and direction of the Transitions of the participant's focus during perception. Accordingly, the contextualisation function $f_{ctx}()$ takes the situational context and the Transition Matrix T as Input. It is implemented as a weighted self-attention within the transformer encoder and represents a united view on sequential and situational context. The final scoring is again realized with a feed-forward layer as scoring function ϕ. Figure 25 illustrates the concept of this view in terms of contextualising the subject entity.

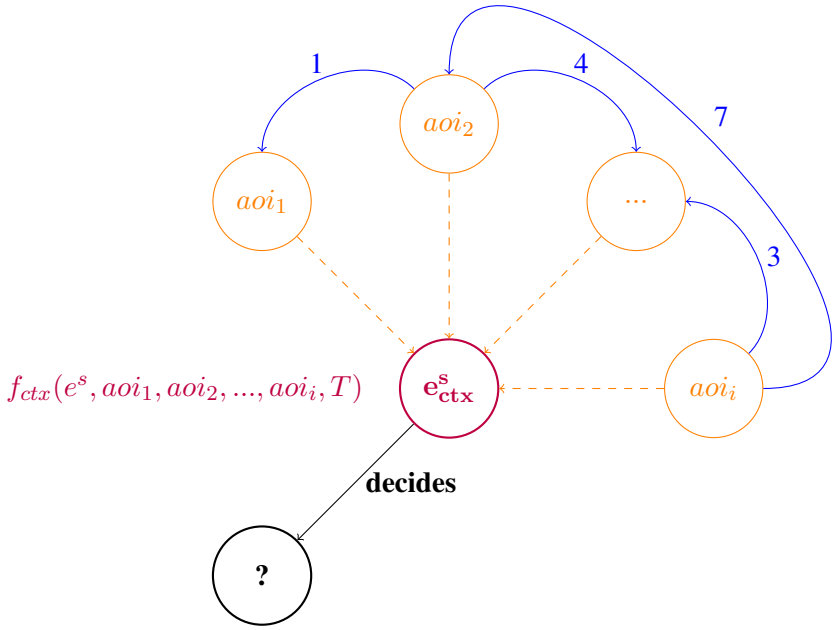

Figure 25: Unified view

Blue denotes sequential contextualisation and orange situational context. The combined contextualisation is marked in purple.

Unified Perspective: Implementation Details

The advantage of the above described procedure is that it does not require an increase in the number of trainable parameters when compared to the vanilla transformer approach. This section covers in detail how we process and inject this sequential information to the transformer model.

The bias injection process that we chose is straight-forward. We use the *src_mask* parameter in the PyTorch Transformer Encoder to add the transition matrix to the attention weights. Note that although the parameter is named src_mask, the docu-

Table 11: Transition matrix (partial) for participant EWCX and stimulus ID 2412873.

	vis_wall	**txt_wall**	**off**	**txt_rock**	**...**
vis_wall	0	1	0	0	...
txt_wall	0	0	1	0	...
off	0	0	0	1	...
txt_rock	0	1	0	0	...
...	...	...	...	...	...

mentation states that *If a FloatTensor is provided, it will be added to the attention weight*[29]. The attention weight refers to the term in brackets in equation 1 [68].

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Due to the sparsity of the matrices and the final softmax application that happens after adding the matrices to the weights, it might be beneficial to scale the weights to ensure that the perception information is not vanishing during calculations. This is enabled by the manual tuning parameter λ. We found it also beneficial to transpose the matrix along the diagonal and add the result to the original matrix. Equation 2 shows this amplification function. An example based on the matrix in Table 11 is shown below.

$$\text{Amplify}(M_{\text{Transition}}, \lambda) = \lambda(M_{\text{Transition}} + M_{\text{Transition}}{}^T) \tag{2}$$

When we input the matrix from Table 11 into equation 2 and set the weight variable λ to 5, we get Equation 3:

$$\text{Amplify}(M_{\text{Transition}}, 5) = \begin{bmatrix} 0 & 5 & 0 & 0 \\ 5 & 0 & 5 & 5 \\ 0 & 5 & 0 & 5 \\ 0 & 5 & 5 & 0 \end{bmatrix} \tag{3}$$

[29] https://pytorch.org/docs/1.12/generated/torch.nn.Transformer.html

Table 12: Experiment I: Results for the PCE task across the different models. Naive refers to a baseline that always predicts the most frequent class.

Setting	Metric	LSTM	Transformer	Ensemble	Naive
3 Classes	Accuracy	.7466	.7238	**.7728**	.6715
	F1	.4849	.5240	**.5628**	.2678
2 Classes	Accuracy	.8494	.8611	**.8742**	.7639
	F1	.7767	.7879	**.8112**	.4330

This operation reduces the sparsity of the matrix and in combination with the upscaling lead to the best results.

5.6 Empirical Results

Setup I: Results

In the first set of experiments, we investigated how the different kinds of information (content vs. sequence) contribute to solving the PCE task. For a fair comparison, all three models are trained using the same hyperparameters that are based on the best-performing Ensemble model. We use 6 attention heads and layers for the transformer, 256 dimensions for the transformer feed-forward network and 32 embedding dimensions (for both the participant and AOI embedding) in the LSTM network. All three models are trained using a batch size of 128 and a learning rate of 0.0001. We report the prediction accuracy and the F1 score (averaged over all classes) on the test split as evaluation metrics for all conducted experiments. Because it shows that the rare *unclear* outcome is problematic for purely sequential models, we also report Accuracy and F1 scores that exclude samples with this class. In addition to our models, we report a naive baseline that always predicts the most frequent label. Table 12 shows that the combination of sequential and contextual information in an ensemble leads to the best results.

A deeper look in the classification results show that the LSTM model never predicts *unclear*. We therefore conclude that the sequence alone does not contain sufficient information to make a confident prediction for this outcome.

Table 13: Experiment I: Ablation Results for the PCE task without participant embedding. *Trans* and *Ens* refer to the Transformer and Ensemble models.

Setting	Metric	LSTM	Trans	Ens
3 Class	Accuracy	.6944	.7189	**.7271**
		(-6.9 %)	(-0.6 %)	(-5.9 %)
	F1	.3849	**.5146**	.4956
		(-20.6 %)	(-1.8 %)	(-11.9 %)
2 Class	Accuracy	.7899	**.8254**	.8242
		(-7.0 %)	(-4.1 %)	(-5.7 %)
	F1	.6144	**.7742**	.7429
		(-20.9 %)	(-1.7 %)	(-8.4 %)

Setup I: Ablation

In support for the view of the problem as a KG based task, we empirically determine the extend to which the individual participant representations contribute to the model performance.

For this experiment, we use the same hyperparameters that were used to achieve the results in Table 12. The only difference is that here we removed all individual participant representations from the models. We report the same evaluation metrics as before and show the relative change in percent. It becomes apparent that the subject-centric view with participant representation that we incorporate into our models is a very reasonable approach. Table 13 shows how the model performance deteriorates without explicit entity representation.

Setup II: Results

When evaluated on the test data, the Perception-Guided Transformer achieves an on-par performance with the Ensemble model in both settings. Table 14 shows that the approach we chose leads to competitive results, with the advantage of needing less trainable parameters.

We used 128 feed-forward dimensions in the transformer, a batch size of 16 and a learning rate of 0.0001. In this setting, the transition matrices are weighted by a λ of 500, which we empirically determined to be most effective in this scenario.

Table 14: Experiment II: Results for the PCE task. PG-Transformer refers to the *Perception-guided Transformer*, which we compare with the Ensemble model introduced before.

Setting	Model	Accuracy	F1
3 Classes	PG-Transformer	**.7761**	.5294
	Ensemble	.7728	**.5628**
2 Classes	PG-Transformer	.8715	**.8196**
	Ensemble	**.8742**	.8112

5.7 Qualitative Analysis

Setup I: Qualitative analysis

For an in-depth examination of how each part contributes to the result, we investigated the classifications. When reduced to a binary outcome (correct prediction vs. wrong prediction), there are 8 possible combinations of how the three individual models can classify one sample sequence. To identify relevant samples for a deeper analysis, we exclude all samples that were classified correctly/incorrectly by all three models. This leads to a subset of samples where at least one of the models makes a different prediction than the others. In our test set, we identified 4 samples that all include the same image, but were assessed by different participants. In this stimulus, 54 percent of participants claimed that the caption mentions the central object, 37 percent claimed that it is unclear and 8 percent responded negatively. Table 15 shows in detail how the individual models classified the samples and with which confidence. **yes** represents a positive response to the question *Are the central objects in the image mentioned in the caption?*, **no** stands for a negative response.

As it can be seen, the transfomers appear to learn global trends in the data (which answer does an individual participant prefer and which evaluation is most common for a given stimulus). The LSTM appears to be able to differentiate between yes/no quite well, but struggles with the unclear option. If we take a closer look at the fixation sequences in Figures 26 and 27, we can clearly see that class *yes* and class *no* look considerably different in terms of sequence length and cov-

Table 15: In-depth analysis of the four samples for image ID 2412873.

Participant ID	Target	Model	yes	no	unclear
08K4	unclear	LSTM	.6356	.1654	.1990
		Transformer	.3404	0569	**.6026**
		Ensemble	.3292	.0063	**.6645**
EWCX	yes	**LSTM**	**.5739**	.2516	.1745
		Transformer	.3388	.0688	.5923
		Ensemble	**.5618**	.0113	.4269
MDVL	no	**LSTM**	.3613	**.4921**	.1466
		Transformer	.3279	.0713	.6008
		Ensemble	.6631	.0593	.2776
WNTO	yes	**LSTM**	**.6237**	.2088	.1675
		Transformer	.3353	.0577	.6070
		Ensemble	**.5526**	.0214	.4261

ered area, while the unclear option in Figure 28 lies somewhere in between. It appears that the information needed to predict the outcome *unclear* is located in the stimulus itself and cannot be drawn from the fixation sequence alone.

Setup II: Qualitative analysis

We again take a closer look at the classification results and compare the Perception-Guided Transformer approach to the vanilla Transformer from the first experiment. By doing so, we investigate how the classification results change when we make the fixation sequence data available to the Transformer. The samples are the same as above and the results can be seen in Table 16. Obviously, the classification in the *PGT* setting is more flexible than the vanilla transformer. Although the third sample (Participant MDVL, Target Class *no*) is not predicted correctly by any of the models, we can see that the *Perception-Guided* variant deviates from the rather static behavior of the vanilla transformer and, in comparison, puts more weight towards *yes*, which is the prediction target for this sample.

By injecting the transition matrix derived from the scan path, we can change the behavior of the transformer to be less restricted to global data statistics and enable it to account for different fixation sequences and more participant-specific

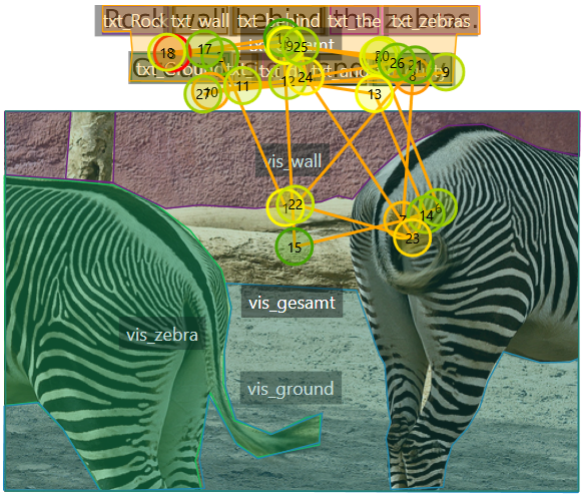

Figure 26: Fixation Sequence with negative response

Participant ID MDVL. Stimulus ID 2412873.

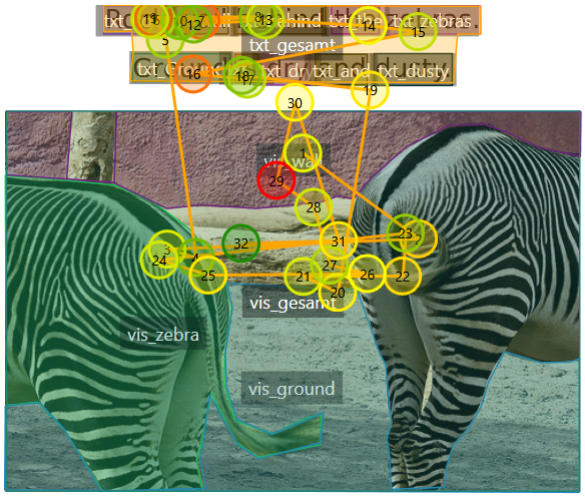

Figure 27: Fixation Sequence with positive response

Participant ID EWCX. Stimulus ID 2412873.

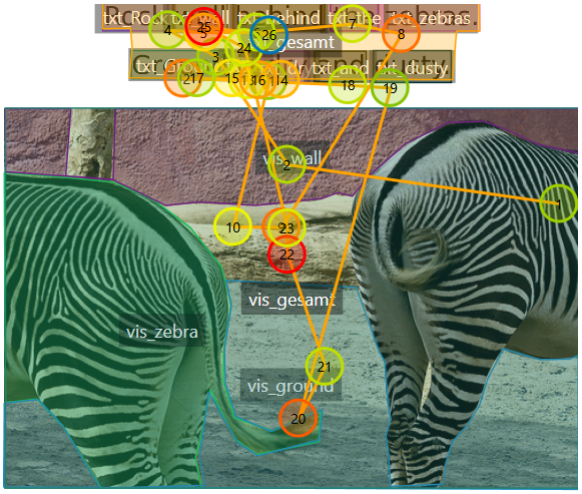

Figure 28: Fixation Sequence with unclear response

Participant ID 08K4. Stimulus ID 2412873.

Table 16: Comparison Transformer vs. Perception-Guided Transformer (PGT) on stimulus 2412873.

Participant	Target	Model	yes	no	unclear
08K4	unclear	**Transformer**	.3404	.0569	**.6026**
		PGT	.5008	.0665	.4327
EWCX	no	Transformer	.3388	.0688	.5923
		PGT	**.4709**	.1569	.3722
MDVL	yes	Transformer	.3279	.0713	.6008
		PGT	.4543	.1787	.3670
WNTO	no	Transformer	.3353	.0577	.6070
		PGT	**.5236**	.0483	.4281

choices. Again, the *unclear* option poses a difficulty, because the underlying sequence (Figure 28) is not as distinct from the *yes* option (Figure 27) as it is the case with the *no* option (Figure 26).

5.8 Conclusions

In order to answer *RQ 4 (What are the consequences of a static environment for embedding situational context?)* and *RQ 5 (Can situational context be expressed via non-relational data?)*, this chapter explored the application of the previously introduced tcKG modelling formalism in a scenario with a dynamic subject entity that acts in a static environment. For the scenarios from Chapter 4 where both the subject and the environment had dynamic characteristics, we proposed the RETRA approach for embedding these dynamics. However, the different nature of a static environment required adjustments to the situational contextualisation mechanisms.

The reason for this different nature mainly lies in the situational context. In previous scenarios, a given *subject - relation - object* triple could occur under different environmental aspects that could change over time. In a static scenario, this change does not happen, since all environmental circumstances are fixed from the beginning. This becomes apparent in the example of the *image - caption* stimuli of the PCE task. Independent of how long and often a participant looks at the stimulus, the perceived entities that constitute the stimulus never occur under different circumstances. This leads to a reduced informational value of the situational context in static environments. To alleviate this issue, this chapter presented alternative views on the tcKG formalism and especially situational context and how it can still contribute to dynamic entity embeddings in a static world:

Sequential Context in static environments does not differ from dynamic environments.

Situational Context in a static world is not time dependent. Therefore, we assume that all known situational context affects the subject entity simultaneously. In the closed world of the stimuli in the PCE task, this can be interpreted as *world knowledge*.

As a consequence, the interplay of sequential and situational context in a static environment has to be modelled differently than in Chapter 4. The varying situational contexts in the previous chapters could express rich semantics. By having only one situational context for a whole situation, its informational value de-

creases. To compensate this, non-KG modalities can serve as additional sources of situational context. In support of these observations, we conducted a series of experiments in which we model participants solving the PCE task.

In the first series of experiments, three models with different purposes for solving the PCE task were introduced:

Sequence Model: The sequence model implements an LSTM network to perform the sequential contextualisation of a participant while perceiving a stimulus. This purely symoblic scenario performs on a temporally ordered sequence of triples consisting of $(participant_{kt}, perceives, AOI_i)$ and, in the final step, predicts the evaluation of the stimulus in the incomplete triple:

$$(participant_{k|T|}, decides, ?)$$

Situational Context Model: The situational context model implements a Transformer Encoder that serves as the situational contextualisation function. There are two main differences when compared to the previous view on situational context:

1. Since context is not time dependent, situational contextualisation is performed only once and takes into account all available context from the stimulus. The result is one contextualized participant embedding.
2. Due to the static nature of a fixed stimulus, there is no additional context available in form of entities in a KG. As an alternative, the underlying images and captions are used directly. They are encoded into their vector spaces by pre-trained encoders and translated to the KG embedding space.

Similar to the LSTM model, the final contextualised participant embedding is used to solve the PCE task in form of completing the triple:

$$(participant_k, decides, ?)$$

Ensemble Model: Finally, an Ensemble model combines the two models by com-

bining the decision of both the sequentially and the situationally contextualised participant embedding. To ensure that the Ensemble leads to a combined view, the model parts are trained jointly on solving the PCE task with a combined decision.

The key novelty of this setup is the situational contextualisation which is performed directly with data from the image and text modalities of the stimulus. The reasoning behind this is that due to the static nature of the stimulus, *external* sources (ResNet and BERT features) of information have to be used. The advantage of those representations is that they contain *World Knowledge* to a certain degree. This world knowledge fills the information gap caused by the static environment. Another specialty of static environments is that the sequential and the situational contextualisation do not longer happen simultaneously at every time step. Instead, situational contextualisation is performed *globally* (with respect to the current stimulus contents) and independent of the current t. The Ensemble models these differences of situational and sequential context by performing both operations separately, on two different representations of the same participant, but conditioned on the same task. This rather naive implementation already shows an increased performance in comparison to the approaches in isolation. In conclusion, this shows how both types of context complement each other even in a static environment.

Although the combination of sequential and situational context in the Ensemble model showed to perform well on the PCE task, having two distinct representations of the same participant is not an adequate solution for modelling the behavior of one entity. Since there is only one entity, this is a questionable solution from a formal point of view. Because of this, an alternative approach with only one participant representation was devised. This approach unifies the sequential and situational contextualisation function into one architecture:

Perception Guided Transformer: This model builds on top of the situational context transformer from the previous experiments. The difference, however, is that the contextualisation function now takes an additional Transition Matrix T as an argument. This matrix is derived from the fixation sequence

and represents the sequential information in a condensed format. The matrix provides the weights in a weighted self-attention over the stimulus contents. The result of this is one participant representation that captures both the situational and sequential context.

With this setup, we implemented a unified contextualisation function as weighed self-attention within a Transformer Encoder. Although the original fixation sequence cannot be restored from the Transition Matrix, there appears to be enough sequential information available for the model to outperform the pure LSTM or Transformer based approaches. It reaches a performance on the same level as the Ensemble model, with the advantage of relying on a single participant representation. Additionally, this approach is computationally cheaper than the Ensemble model, since the amount of trainable parameters remains the same as the Transformer only architecture.

In summary, this chapter provides new insights on how the different properties of static environments require a different approach to situational context. The consequences of a static environment are:

1. Situational and sequential context cannot be applied simultaneously, since situational context is not time dependent.

2. This requires new approaches for the combined sequential and situational contextualisation of dynamic entities.

3. Since the same situational context is now true during all interactions, it can be interpreted to function as source of world knowledge.

By establishing these properties and demonstrating how the different nature of situational context can be used in a KG embedding scenario, we answer RQ 4. Closely tied to the idea of situational context as world knowledge is RQ 5, which we answer by introducing the idea of translating feature vectors from different modalities to the KG Embedding space.

6 Conclusion

In this thesis, we introduced novel approaches for representing how entities act in their respective environments and how these interactions affect the participating entities. Based on the identified limitations of static KGs, we develop novel techniques to capture dynamic entities. Our proposed tcKG representation approach is able to formally model these dynamics. The complementary RETRA embedding approach proposes techniques to reflect these dynamics in a KG embedding setting. Furthermore, the Perception-guided Transformer demonstrates how the concepts of sequential and situational contextualisation translate to entities that act in a static world and how non-symbolic modalities can be exploited for representing context. To conclude this thesis, the subsequent section summarizes our findings and contributions and puts them into context of the initial motivation and the resulting research questions.

6.1 Summary

We argued that formal representations of real-world entities should be able to reflect *change* within entities. This change can be either caused by previous interactions of the focus entity, or by situational factors. Accordingly, we introduced the concepts of **sequential** and **situational** context under which an entity acts.

Sequential Context: Describes previous interactions that affect the current state of an entity.

Situational Context: Describes current environmental factors that affect an entity's behavior.

Additionally, we introduced the notion of **dynamic** and **static** entities and use these properties to describe scenarios on a high level. The real-world scenarios in which we model subject entity dynamics throughout this work fall in the following categories:

Dynamic Environment: The situational context for a given fact can vary over time.

Static Environment: The situational context for a given fact is always the same. This also implies that the object is a passive entity.

On the basis of these high-level concepts, we explored how dynamics in entity representations can be modelled in the Knowledge Graph formalism and how these dynamics can be reflected in Knowledge Graph Embeddings.

As a starting point, we established the limitations of static Knowledge Graphs with regard to representing entity dynamics. This corresponds with the first research question.

RQ 1: What are the limitations of static formalisms and their corresponding embedding approaches with regard to sequential and situational context?

To answer this question, we tested different methods for representing sequential and situational context within static formalisms on the example of a Location Recommendation Task. We expressed situational context through an n-ary hypergraph that extends the original binary *checks_in(user,location)* relation by adding the situational context as entities to the relation. For representing sequential context, we used a RNN that takes pre-trained entity Embeddings as inputs. By this, we identified the limitations summarized below:

- The main limitation of both approaches is that the representations remain static. *As such, the potentially lasting effects of situational and sequential context cannot be reflected adequately within an entity.*
- A hypergraph relation is not able to explicitly express the difference between the primary *subject - object* relation and the situational context that is affecting it.
- Although the RNN approach can model global sequential check-in patterns, *it does not reflect individual users and how their check-in history affects their current state.*

All these limitations are rooted in the underlying static KG formalism. To overcome these issues, additions to take into account situational and sequential dependencies for KG facts are required. This leads to RQ 2:

RQ 2: How can the KG formalism be extended to model sequential and situational context?

To solve this problem, we expanded the triples in the binary KG formalism to allow situational context as additional context entities. Furthermore, to capture the sequential component, we added time information to the extended facts. From the resulting *tcKG* facts, ordered sequences of triples with the same subject entity can be created to represent the sequential and situational context. In order to use tcKG for tasks like Link Prediction, a suitable Embedding approach is required. Its development and application is covered by RQ 3.

RQ 3: How can static knowledge graph embeddings be transformed into contextualized representations?

We proposed the *RETRA* framework (Recurrent Transformers) in response to RQ 3. RETRA consists of recurrent transformer encoders that implement a situational contextualisation function and a sequential contextualisation function for entities. With RETRA, pre-trained entity embeddings from any static KGE approach can be transformed into dynamic representations. Additionally, entity embeddings can be trained in an end-to-end fashion with arbitrary scoring functions. RETRA implements two functions to represent situational and sequential context. They can be either used together or individually, depending on the use-case:

Situational Contextualisation: Implemented as multi-headed self attention across the *subject - relation - object* triple and the current context entities.

Sequential Contextualisation: Implemented as a recurrent step where the contextualised subject entity is used as input for the next step.

Depending on the desired perspective, it is possible to let the situational context act on the relation or the subject entity. Since the situational context is captured within the original triple, well established binary scoring functions can be used to rank facts with an arbitrary number of context entities.

We evaluated these approaches empirically on three different Link Prediction tasks. All tasks have in common that the underlying world has a *dynamic environment*. The first task used the same data as the Location Recommendation scenario from before. The focus of the conducted experiments and their task a laid out below.

Location Recommendation: Contextualising pre-trained entity embeddings with RETRA.

Driving Scene Classification: Investigating the effects of increasing the amount of sequential information.

Event Prediction: Investigating the effects of added situational context.

The experiments showed that the added context provides beneficial information with regard to solving the tasks. Additionally, we demonstrated how different scoring functions can be used within the RETRA framework.

The aforementioned scenarios on which the initial RETRA implementation was tested all have in common that they consist of a dynamic environment in which a dynamic subject entity acts. This leaves the open question of how the concept of situational context functions in a static environment:

RQ 4: What are the consequences of a static environment for embedding situational context?

In a static environment, the situational context for a given *subject - object* interaction never changes. This means that situational context can be captured on a global level, independently of the current t. This global validity makes situational context less circumstantial and moves into the direction of background knowledge.

This allows for interpreting situational context as *world knowledge* that is not explicitly modelled in the underlying graph, but still available to the subject that is acting in the static world.

For empirical testing, we devised a novel task which we named Perception-guided Crossmodal Entailment. In this task, a participant (subject entity) views multi modal stimuli (text and images) and evaluates if the text modality describes the image modality. We recorded the eye movements of the participant during stimulus exposure, of which we then use the sequence of fixations on different stimulus objects. This forms an ordered sequence of triples *perceives (participant, stimulus object)*. Note that the subject entity may revisit certain objects in the stimulus for multiple times until it comes to a final evaluation of the text - image combination. While it is apparent that sequential contextualisation of the subject entity can be achieved in the same way as within the RETRA framework, different strategies for situational contextualisation have to be developed. Similar to RETRA, we use the self-attention mechanism of a transformer encoder to implement situational contextualisation. However, since the context never changes, all contexts are used at the same time and affect the subject entity directly. This can be interpreted as providing the subject entity with knowledge about the stimulus contents. Since sequential contextualisation occurs at every t, while situational contextualisation occurs only once in a static environment, the challenge is to model how both affect the subject entity. In our empirical study on the PCE task, we evaluated the following strategies:

- Sequential and Situational contextualisation is performed with two separate models. Each model returns an individual contextualised subject entity representation. In the final step, both entity representations are merged into one single representation on which the final evaluation of the text - image combination is done.

- The situational context transformer is additionally fed with derived data from the fixation sequence. The fixation data is transformed to a transition matrix, which is fed as bias into the attention weights of the transformer. As a result, the model returns the sequentially and situationally contextualized subject

entity.

When evaluated on the final decision task, both approaches achieve a comparable performance. However, the unified approach has the computational advantage of requiring less trainable parameters in comparison to the approach with two separate models. From a formal perspective, this procedure also has the advantage of resulting in only one unified entity representation, which is a better reflection of the underlying scenario.

Since situational context in static environments is fixed for all t, it does not carry the same informational value as it does in dynamic environments. In order to address this issue, we explored different ways of expressing situational context. As our data contains not only the symbolic fixation sequence, but also the underlying data as images (and image regions) and corresponding captions, we attempt to integrate this data directly into our approaches and formulate the corresponding research question:

RQ 5: Can situational context be expressed via non-symbolic data?

The motivation behind this research question is that a) there is a reduced informational value of situational context in static environments and b) there is additional informational potential available in non-symbolic modalities in our collected data for the PCE task.

Since the data for the PCE task contains a caption and a corresponding image with annotated image regions, we explore options of using these modalities within the symbolic fixation sequence. We identify two options of exploiting non-symbolic modalities within a KG framework:

Entity Extraction: The entities and possibly their relations can be extracted from the sources (i.e. texts and images) and added as additional facts to a KG.

Feature Translation: The entities are directly encoded into their respective vector representations using modality specific encoder models (i.e ResNet and BERT). These vector representations can then be translated into the KG embedding space.

The second option, *Feature Translation*, is more suitable for our approach, because using pre-trained vector representations allows for leveraging the rich semantics they carry. Since pre-trained representations are conditioned on large-scale datasets, they contain implicit world knowledge. Word vectors carry information about the meaning of words in relation to their context. For example, the words *church* and *cathedral* have different surface forms, but are closely related on a meaning level. Isolated symbolic representations only enable us to state that they are different entities, while pre-trained vector representations also allow for expressing the degree of their semantic (dis)similarity. Analogous to this are image feature representations. They are conditioned on a classification task and therefore contain information about class membership. This also makes it possible to make statements about similarity between entities with respect to their class.

These useful properties speak in favor of using pre-trained vector representations instead of relying of symbolic representations (which would also rely on the availability of an extraction technique). The remaining challenge is to translate the vector representations from their modality specific Embedding space to the KG Embedding space, where the participant representations are located. Fortunately, it is possible to condition a linear transformation from one Embedding space to another. This is done in an end-to-end fashion, where the KG embedding space and the translation layer are jointly conditioned of the PCE task.

Contextualising the subject entity with non-symbolic data leads to an improved performance when compared to purely symbolic approaches. Therefore, our experiments provide support for using non-symbolic data as situational context in static environments.

6.2 Future Work

In this section, we discuss future research directions that expand on the concepts and ideas introduced in this work.

One property of datasets that we have not further discussed in this work are their **time frames**. By this, we refer to the time resolution or the frequency of interactions between entities. For the *Driving Scene* classification data, the cur-

rent state of interactions is recorded with a resolution of milliseconds, while in the *Event Prediction* scenario, even weeks may pass between two interactions. Reflecting this difference in temporal resolution might be beneficial if it were to be modelled within the RETRA framework. With an increasing duration between two interactions, the sequentially contextualised entity could slowly revert back to its *neutral* state. This would decrease the effect of the previous interaction based on the time that has passed. The rate of reverting back to a neutral state could be either defined individually for all entities, or be based on the entity type. For entities like countries, this rate would be lower than for individual persons. The goal would be to implement different variants and empirically test their viability by determining how they contribute to improving the results on link prediction tasks.

Another future research direction could focus on the **interplay** between situational and sequential context and different ways of its implementation. In the original formulation, we define situational context to affect a relation r and sequential context to affect a subject entity e^s. In the PCE task, we let both context types directly affect the subject entity e^s. This leads to the question whether it is necessary to balance the effects of the context sources, or - in other words - if they affect an entity equally. Applying both contextualisations to a subject entity directly enables the effect of the previous situational context to be carried on to the next time step. The question is whether this has negative side-effects on sequential contextualisation. An expanded study could investigate this and - if negative side effects were to be found - propose ways of alleviating them.

Also expanding on the ideas introduced with the RETRA approach, one proposed direction for further work is the modelling of **full inter-entity dynamics** for scenarios with dynamic environments. The approaches presented in this work focus on contextualising entities individually based on the sequence of their interactions with the environment. While this approach adequately represents how an entity is influenced by its previous interactions and the current situational context, it only models these effects for the subject entity that is currently in focus. Other entities that might interact with the current subject entity are also potentially influenced by environmental factors. Consider for example two persons $person_A$ and $person_B$. While the current approaches can model how the presence of $person_B$

could act as situational context on the focus subject $person_A$, they do not consider that $person_B$ might also be under the influence of its individual sequential and situational context. The current state of $person_B$ could potentially be meaningful with regard to how it affects $person_A$ in the given situation. Modelling these complex interactions would lead to a more detailed representation of how the entities affect each other.

Building on the idea of using non-symbolic entity representations for usage in KG scenarios, we also see potential in applying more recent models as feature encoders. The new generation of **large multi-modal language models** like GPT-4 have capabilities that go way beyond BERT. In terms of representations, it can be assumed that the underlying feature vectors of such large models carry even richer semantics than the older generation of models.

As Chapter 5 dealt with modelling human behavior in perception, a different direction for future studies could be to further explore the psychological component of modelling human entities. In Sections 4 and 5, we referred to the concept of semantic and episodic memory in psychology and how it relates to situational and sequential context. Future work could delve deeper into this relation by including state of the art LLMs for representing the world knowledge component. This could lead to an even more detailed representation of human perception processes on the basis of the KG formalism.

References

[1] ADOMAVICIUS, G., AND TUZHILIN, A. Context-aware recommender systems. In *Proceedings of the ACM Conference on Recommender Systems, RecSys* (2008).

[2] AKBARI, H., YUAN, L., QIAN, R., CHUANG, W.-H., CHANG, S.-F., CUI, Y., AND GONG, B. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text, 2021.

[3] BADDELEY, A., EYSENCK, M., AND ANDERSON, M. *Memory*. Taylor & Francis, 2015.

[4] BAHDANAU, D., CHO, K., AND BENGIO, Y. Neural machine translation by jointly learning to align and translate. *CoRR abs/1409.0473* (2014).

[5] BAO, J., ZHENG, Y., WILKIE, D., AND MOKBEL, M. F. Recommendations in location-based social networks: a survey. *GeoInformatica 19* (2015).

[6] BARAL, R., IYENGAR, S. S., LI, T., AND BALAKRISHNAN, N. Close: Contextualized location sequence recommender. In *Proceedings of the 12th ACM Conf. on Recommender Systems* (2018), RecSys '18.

[7] BARAL, R., IYENGAR, S. S., LI, T., AND ZHU, X. Hicaps: Hierarchical contextual poi sequence recommender. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2018), SIGSPATIAL '18.

[8] BENGIO, Y., COURVILLE, A., AND VINCENT, P. Representation learning: A review and new perspectives, 2014.

[9] BITZER, S., PARK, H., BLANKENBURG, F., AND KIEBEL, S. J. Perceptual decision making: drift-diffusion model is equivalent to a bayesian model. *Frontiers in Human Neuroscience 8* (2014).

[10] BORDES, A., USUNIER, N., GARCIA-DURÁN, A., WESTON, J., AND YAKHNENKO, O. Translating embeddings for modeling multi-relational

data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (2013).

[11] BOSCHEE, E., LAUTENSCHLAGER, J., O'BRIEN, S., SHELLMAN, S., STARZ, J., AND WARD, M. Icews coded event data, 2015.

[12] BROWN, T., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J. D., DHARIWAL, P., NEELAKANTAN, A., SHYAM, P., SASTRY, G., ASKELL, A., AGARWAL, S., HERBERT-VOSS, A., KRUEGER, G., HENIGHAN, T., CHILD, R., RAMESH, A., ZIEGLER, D., WU, J., WINTER, C., HESSE, C., CHEN, M., SIGLER, E., LITWIN, M., GRAY, S., CHESS, B., CLARK, J., BERNER, C., MCCANDLISH, S., RADFORD, A., SUTSKEVER, I., AND AMODEI, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (2020).

[13] BUCHER, H.-J., AND SCHUMACHER, P. The relevance of attention for selecting news content. an eye-tracking study on attention patterns in the reception of print and online media. *Communications. The European Journal of Communication Research 31* (2006).

[14] CAI, R., YUAN, J., XU, B., AND HAO, Z. Sadga: Structure-aware dual graph aggregation network for text-to-sql. *Advances in Neural Information Processing Systems 34* (2021).

[15] CHEN, Y.-C., LI, L., YU, L., KHOLY, A. E., AHMED, F., GAN, Z., CHENG, Y., AND LIU, J. Uniter: Universal image-text representation learning. In *ECCV* (2020).

[16] CHENG, C., YANG, H., LYU, M. R., AND KING, I. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI* (2013).

[17] CHO, E., MYERS, S. A., AND LESKOVEC, J. Friendship and mobility: user movement in location-based social networks. In *KDD* (2011).

[18] DASGUPTA, S. S., RAY, S. N., AND TALUKDAR, P. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of*

the 2018 Conference on Empirical Methods in Natural Language Processing (2018).

[19] DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K., AND FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009).

[20] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics* (2019).

[21] FATEMI, B., TASLAKIAN, P., VAZQUEZ, D., AND POOLE, D. Knowledge hypergraphs: Prediction beyond binary relations. In *IJCAI* (2020).

[22] FATEMI, B., TASLAKIAN, P., VAZQUEZ, D., AND POOLE, D. Knowledge hypergraph embedding meets relational algebra. *Journal of Machine Learning Research 24*, 105 (2023).

[23] FENG, S., LI, X., ZENG, Y., CONG, G., CHEE, Y. M., AND YUAN, Q. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI* (2015).

[24] GARCÍA-DURÁN, A., DUMANCIC, S., AND NIEPERT, M. Learning sequence encoders for temporal knowledge graph completion. In *Conference on Empirical Methods in Natural Language Processing* (2018).

[25] GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. *Deep Learning*. MIT Press, 2016.

[26] GOYAL, A., AND BENGIO, Y. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A 478* (2020).

[27] HABER, R., AND HERSHENSON, M. *The Psychology of Visual Perception*. Holt, Rinehart and Winston, 1980.

[28] HALILAJ, L., DINDORKAR, I., LUETTIN, J., AND ROTHERMEL, S. A knowledge graph-based approach for situation comprehension in driving scenarios. In *Eighteenth Extended Semantic Web Conference - In-Use Track* (2021).

[29] HAN, X., CAO, S., LV, X., LIN, Y., LIU, Z., SUN, M., AND LI, J. OpenKE: An open toolkit for knowledge embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Brussels, Belgium, Nov. 2018).

[30] HAN, Z., CHEN, P., MA, Y., AND TRESP, V. Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion. *CoRR abs/2011.03984* (2020).

[31] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[32] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural computation 9*, 8 (1997).

[33] HOGAN, A., BLOMQVIST, E., COCHEZ, M., D'AMATO, C., DE MELO, G., GUTIERREZ, C., GAYO, J. E. L., KIRRANE, S., NEUMAIER, S., POLLERES, A., NAVIGLI, R., NGOMO, A.-C. N., RASHID, S. M., RULA, A., SCHMELZEISEN, L., SEQUEDA, J., STAAB, S., AND ZIMMERMANN, A. *Knowledge Graphs*. Springer, 2022.

[34] JI, S., PAN, S., CAMBRIA, E., MARTTINEN, P., AND YU, P. S. A survey on knowledge graphs: Representation, acquisition and applications. *IEEE Transactions on Neural Networks and Learning Systems 33* (2020).

[35] KAZEMI, S. M., AND POOLE, D. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems* (2018).

[36] KRAJBICH, I., ARMEL, C., AND RANGEL, A. Visual fixations and the computation and comparison of value in simple choice. *Nature neuroscience 13 10* (2010).

[37] KRAJBICH, I., LU, D., CAMERER, C., AND RANGEL, A. The attentional drift-diffusion model extends to simple purchasing decisions. *Frontiers in psychology 3* (2012).

[38] KRISHNA, R., ZHU, Y., GROTH, O., JOHNSON, J., HATA, K., KRAVITZ, J., CHEN, S., KALANTIDIS, Y., LI, L.-J., SHAMMA, D. A., BERNSTEIN, M., AND FEI-FEI, L. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016.

[39] LEBLAY, J., AND CHEKOL, M. W. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018* (2018).

[40] LEFÈVRE, S., VASQUEZ, D., AND LAUGIER, C. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal 1* (2014).

[41] LENZ, D., DIEHL, F., TRUONG-LE, M., AND KNOLL, A. C. Deep neural networks for markovian interactive scene prediction in highway scenarios. In *IEEE Intelligent Vehicles Symposium, IV 2017, Los Angeles, CA, USA, June 11-14, 2017* (2017).

[42] LI, L. H., YATSKAR, M., YIN, D., HSIEH, C.-J., AND CHANG, K.-W. Visualbert: A simple and performant baseline for vision and language, 2019.

[43] LI, R., YANG, S., ROSS, D. A., AND KANAZAWA, A. Ai choreographer: Music conditioned 3d dance generation with aist++. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021).

[44] LI, X., JIANG, M., HONG, H., AND LIAO, L. A time-aware personalized point-of-interest recommendation via high-order tensor factorization. *ACM Transactions on Information Systems (TOIS) 35* (2017).

[45] LIN, T.-Y., MAIRE, M., BELONGIE, S., BOURDEV, L., GIRSHICK, R., HAYS, J., PERONA, P., RAMANAN, D., ZITNICK, C. L., AND DOLLÁR, P. Microsoft coco: Common objects in context, 2015.

[46] LIU, Y., HUA, W., XIN, K., AND ZHOU, X. Context-aware temporal knowledge graph embedding. In *International Conference on Web Information Systems Engineering* (2019).

[47] LOSHCHILOV, I., AND HUTTER, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* (2019).

[48] LU, J., BATRA, D., PARIKH, D., AND LEE, S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems* (2019).

[49] LUETTIN, J., ROTHERMEL, S., AND ANDREW, M. Future of in-vehicle recommendation systems @ bosch. *Proceedings of the 13th ACM Conf. on Recommender Sys.* (2019).

[50] MIKOLOV, T., CHEN, K., CORRADO, G. S., AND DEAN, J. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations* (2013).

[51] NAUMOV, M., MUDIGERE, D., SHI, H. M., HUANG, J., SUNDARAMAN, N., PARK, J., WANG, X., GUPTA, U., WU, C., 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., AND SMELYANSKIY, M. Deep learning recommendation model for personalization and recommendation systems. *CoRR abs/1906.00091* (2019).

[52] NICKEL, M., ROSASCO, L., AND POGGIO, T. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (2016).

[53] NOULAS, A., SCELLATO, S., LATHIA, N., AND MASCOLO, C. Mining user mobility features for next place prediction in location-based services. *2012 IEEE 12th International Conference on Data Mining* (2012).

[54] PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., DESMAISON, A., KÖPF, A., YANG, E., DEVITO, Z., RAISON, M., TEJANI, A., CHILAMKURTHY, S., STEINER, B., FANG, L., BAI, J., AND CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library, 2019.

[55] PFEIFFER, J., PFEIFFER, T., MEISSNER, M., AND WEISS, E. Eye-tracking-based classification of information search behavior using machine learning: Evidence from experiments in physical shops and virtual reality shopping environments. *Information Systems Research 31*, 3 (2020).

[56] PLUMMER, B. A., WANG, L., CERVANTES, C. M., CAICEDO, J. C., HOCKENMAIER, J., AND LAZEBNIK, S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *2015 IEEE International Conference on Computer Vision (ICCV)* (2015).

[57] RETTINGER, A., BOGDANOVA, V., AND NIEMANN, P. Towards learning cross-modal perception-trace models, 2019.

[58] RETTINGER, A., WERMSER, H., HUANG, Y., AND TRESP, V. Context-aware tensor decomposition for relation prediction in social networks. *Social Network Analysis and Mining 2*, 4 (2012).

[59] SU, W., ZHU, X., CAO, Y., LI, B., LU, L., WEI, F., AND DAI, J. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations* (2020).

[60] SUN, C., MYERS, A., VONDRICK, C., MURPHY, K. P., AND SCHMID, C. Videobert: A joint model for video and language representation learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019).

[61] SUN, Z., YANG, J., ZHANG, J., BOZZON, A., HUANG, L.-K., AND XU, C. Recurrent knowledge graph embedding for effective recommendation. *Proceedings of the 12th ACM Conference on Recommender Systems* (2018).

[62] TAN, H., AND BANSAL, M. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (2019).

[63] TAVARES, G., PERONA, P., AND RANGEL, A. The attentional drift diffusion model of simple perceptual decision-making. *Frontiers in Neuroscience 11* (2017).

[64] THOPPILAN, R., FREITAS, D. D., HALL, J., SHAZEER, N. M., KULSHRESHTHA, A., CHENG, H.-T., JIN, A., BOS, T., BAKER, L., DU, Y., LI, Y., LEE, H., ZHENG, H. S., GHAFOURI, A., MENEGALI, M., HUANG, Y., KRIKUN, M., LEPIKHIN, D., QIN, J., CHEN, D., XU, Y., CHEN, Z., ROBERTS, A., BOSMA, M., ZHOU, Y., CHANG, C.-C., KRIVOKON, I. A., RUSCH, W. J., PICKETT, M., MEIER-HELLSTERN, K. S., MORRIS, M. R., DOSHI, T., SANTOS, R. D., DUKE, T., SØRAKER, J. H., ZEVENBERGEN, B., PRABHAKARAN, V., DÍAZ, M., HUTCHINSON, B., OLSON, K., MOLINA, A., HOFFMAN-JOHN, E., LEE, J., AROYO, L., RAJAKUMAR, R., BUTRYNA, A., LAMM, M., KUZMINA, V. O., FENTON, J., COHEN, A., BERNSTEIN, R., KURZWEIL, R., AGUERA-ARCAS, B., CUI, C., CROAK, M. R., HSIN CHI, E. H., AND LE, Q. Lamda: Language models for dialog applications. *ArXiv abs/2201.08239* (2022).

[65] TRIVEDI, R., DAI, H., WANG, Y., AND SONG, L. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017).

[66] TROUILLON, T., WELBL, J., RIEDEL, S., GAUSSIER, E., AND BOUCHARD, G. Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning* (2016).

[67] TULVING, E. Episodic and semantic memory. *Organization of memory 1* (1972).

[68] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. Attention is all you need. In *Advances in Neural Information Processing Systems* (2017).

[69] WANG, H., ZHAO, M., XIE, X., LI, W., AND GUO, M. Knowledge graph convolutional networks for recommender systems. *The World Wide Web Conference* (2019).

[70] WANG, P., YANG, A., MEN, R., LIN, J., BAI, S., LI, Z., MA, J., ZHOU, C., ZHOU, J., AND YANG, H. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning* (2022).

[71] WANG, Q., MAO, Z., WANG, B., AND GUO, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering 29*, 12 (2017).

[72] WANG, Z., YU, J., YU, A. W., DAI, Z., TSVETKOV, Y., AND CAO, Y. SimVLM: Simple visual language model pretraining with weak supervision, 2021.

[73] WERNER, S., RETTINGER, A., HALILAJ, L., AND LÜTTIN, J. Embedding taxonomical, situational or sequential knowledge graph context for recommendation tasks. In *International Conference on Semantic Systems* (2021).

[74] WERNER, S., RETTINGER, A., HALILAJ, L., AND LÜTTIN, J. RETRA: Recurrent transformers for learning temporally contextualized knowledge graph embeddings. In *Eighteenth Extended Semantic Web Conference - Research Track* (2021).

[75] WOLF, T., DEBUT, L., SANH, V., CHAUMOND, J., DELANGUE, C., MOI, A., CISTAC, P., RAULT, T., LOUF, R., FUNTOWICZ, M., DAVISON, J., SHLEIFER, S., VON PLATEN, P., MA, C., JERNITE, Y., PLU, J., XU,

C., SCAO, T. L., GUGGER, S., DRAME, M., LHOEST, Q., AND RUSH, A. M. Huggingface's transformers: State-of-the-art natural language processing, 2020.

[76] XIE, M., YIN, H., WANG, H., XU, F., CHEN, W., AND 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).

[77] XIE, N., LAI, F., DORAN, D., AND KADAV, A. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706* (2019).

[78] XU, C., NAYYERI, M., ALKHOURY, F., YAZDI, H. S., AND LEHMANN, J. Tero: A time-aware knowledge graph embedding via temporal rotation. In *International Conference on Computational Linguistics* (2020).

[79] XU, P., ZHU, X., AND CLIFTON, D. A. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45* (2022).

[80] YANG, B., YIH, W., HE, X., GAO, J., AND DENG, L. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015).

[81] YANG, D., QU, B., YANG, J., AND CUDRE-MAUROUX, P. Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach. In *The World Wide Web Conference* (2019).

[82] YANG, D., QU, B., YANG, J., AND CUDRE-MAUROUX, P. Revisiting user mobility and social relationships in LBSNs: A hypergraph embedding approach, 2019.

[83] YENICELIK, D., SCHMIDT, F., AND KILCHER, Y. How does BERT capture semantics? a closer look at polysemous words. In *Proceedings of the Third*

BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (2020).

[84] YIN, H., WANG, W., WANG, H., CHEN, L., AND ZHOU, X. Spatial-aware hierarchical collaborative deep learning for poi recommendation. *IEEE Transactions on Knowledge and Data Engineering 29* (2017).

[85] YING, J. J.-C., LU, E. H.-C., KUO, W.-N., AND TSENG, V. S. Urban point-of-interest recommendation by mining user check-in behaviors. In *UrbComp '12* (2012).

[86] ZHANG, S., TAY, Y., YAO, L., AND LIU, Q. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems* (2019).

[87] ZHAO, S., KING, I., AND LYU, M. R. A survey of point-of-interest recommendation in location-based social networks. *CoRR abs/1607.00647* (2016).