CORE: Context-Aware Open Relation Extraction with Factorization Machines

Fabio Petroni
Sapienza University of Rome
Rome, Italy
petroni@dis.uniroma1.it

Luciano Del Corro
Max Planck Institute for Informatics
Saarbrücken, Germany
delcorro@mpi-inf.mpg.de

Rainer Gemulla
University of Mannheim
Mannheim, Germany
rgemulla@uni-mannheim.de

Abstract

We propose CORE, a novel matrix factorization model that leverages contextual information for open relation extraction. Our model is based on factorization machines and integrates facts from various sources, such as knowledge bases or open information extractors, as well as the context in which these facts have been observed. We argue that integrating contextual information—such as metadata about extraction sources, lexical context, or type information—significantly improves prediction performance. Open information extractors, for example, may produce extractions that are unspecific or ambiguous when taken out of context. Our experimental study on a large real-world dataset indicates that CORE has significantly better prediction performance than state-of-the-art approaches when contextual information is available.

1 Introduction

Open relation extraction (open RE) is the task of extracting new facts for a potentially unbounded set of relations from various sources such as knowledge bases or natural language text. The task is closely related to *targeted information extraction* (IE), which aims to populate a knowledge base (KB) with new facts for the KB's relations, such as wasBornIn(Sepp Herberger, Mannheim). Existing methods either reason within the KB itself (Franz et al., 2009; Nickel et al., 2011; Drumond et al., 2012) or leverage large text corpora to learn patterns that are indicative of KB relations (Mintz et al., 2009; Surdeanu et al., 2012; Min et al., 2013). In both cases, targeted IE methods are inherently limited to an (often small) set of predefined relations, i.e., they are not "open". The open RE task is also related to *open information extraction* (open IE) (Banko et al., 2007; Del Corro and Gemulla, 2013), which extracts large amounts of surface relations and their arguments from natural language text; e.g., "critizices"("Dante", "Catholic Church").[1] Although open IE is a domain-independent approach, the extracted surface relations are purely syntactic and often ambiguous or noisy. Moreover, open IE methods usually do not "predict" facts that have not been explicitly observed in the input data. Open RE combines the above tasks by predicting new facts for an open set of relations. The key challenge in open RE is to reason jointly over the *universal schema* consisting of KB relations and surface relations (Riedel et al., 2013).

A number of matrix or tensor factorization models have recently been proposed in the context of relation extraction (Nickel et al., 2012; Riedel et al., 2013; Huang et al., 2014; Chang et al., 2014). These models use the available data to learn latent semantic representations of entities (or entity pairs) and relations in a domain-independent way; the latent representations are subsequently used to predict new facts. Existing models often focus on either targeted IE or open RE. Targeted models are used for within-KB reasoning; they rely on the closed-world assumption and often do not scale with the number of relations. Open RE models use the open-world assumption, which is more suitable for the open RE task because the available data is often highly incomplete. In this paper, we propose CORE, a novel open RE factorization model that incorporates and exploits contextual information to improve prediction performance.

Consider for example the sentence "Tom Peloso joined Modest Mouse to record their fifth studio album". Open IE systems may extract the *surface fact* "join"(TP, MM) from this sentence. Note

[1] We mark (non-disambiguated) mentions of entities and relations in quotation marks throughout this paper.

that *surface relation* "join" is unspecific; in this case, it refers to becoming a member of a music band (as opposed to, say, an employee of a company). Most existing open RE systems use the extracted surface fact for further reasoning, but they ignore the context from which the fact was extracted. We argue in this paper that exploiting contextual information is beneficial for open RE. For our example, we may use standard NLP tools like a named entity recognizer to detect that TP is a person and MM an organization. These coarse-grained types give us hints about the domain and range of the "join" relation for the surface fact, although the actual meaning of "join" still remains opaque. Now imagine that the above sentence was extracted from a newspaper article published in the music section. This information can help to infer that "join" indeed refers to joining a band. Other contextual information, such as the words "record" and "album" that occur in the sentence, further strengthen this interpretation. A context-aware open RE system should leverage such information to accurately predict facts like "is band member of"(TP, MM) and "plays with"(TP, MM).

Note that the prediction of the fact "is band member of"(TP, MM) is facilitated if we make use of a KB that knows that TP is a musician and MM is a music band. If TP and/or MM are not present in the knowledge base, however, such a reasoning does not apply. In our work, we consider both linked *entities* (in-KB) and non-linked *entity mentions* (out of-KB). Since KB are often incomplete, this open approach to handle named entities allows us to extract facts for all entities, even if they do not appear in the KB.

In this paper, we propose CORE, a flexible open RE model that leverages contextual information. CORE is inspired by the combined factorization and entity model (FE) of Riedel et al. (2013). As FE, CORE associates latent semantic representations with entities, relations, and arguments. In contrast to FE, CORE uses factorization machines (Rendle, 2012) as its underlying framework, which allows us to incorporate context in a flexible way. CORE is able to leverage and integrate arbitrary contextual information associated with the input facts into its open RE factorization model. To support reasoning under the open-world assumption, we propose an efficient method for parameter estimation in factorization machines based on Bayesian personalized ranking (Rendle et al., 2009).

We conducted an experimental study on a real-world dataset using contextual information along the lines mentioned above. Our model is extensible, i.e., additional contextual information can be integrated when available. Even with limited amount of contextual information used in our experiments, our CORE model provided higher prediction performance than previous models. Our findings validate the usefulness of contextual information for the open RE task.

2 Related Work

There is a large body of related work on relation extraction; we restrict attention to methods that are most similar to our work.

Targeted IE. Targeted IE methods aim to extract from natural-language text new instances of a set of predefined relations, usually taken from a KB. Most existing methods make use of distant supervision, i.e., they start with a set of seed instances (pairs of entities) for the relations of interest, search for these seed instances in text, learn a relation extractor from the so-obtained training data, and optionally iterate (Mintz et al., 2009; Surdeanu et al., 2012; Min et al., 2013). Open RE models are more general then targeted IE methods in that they additionally reason about surface relations that do not correspond to KB relations. For this reason, Riedel et al. (2013) argued and experimentally validated that open RE models can outperform targeted IE methods.

Open IE. In contrast to targeted IE, the goal of open IE is to extract all (or most) relations expressed in natural-language text, whether or not these relations are defined in a KB (Banko et al., 2007; Fader et al., 2011; Del Corro and Gemulla, 2013). The facts obtained by open IE methods are often not disambiguated, i.e., the entities and/or the relation are not linked to a knowledge base; e.g., "criticizes"("Dante", "Catholic Church"). The goal of our work is to reason about extracted open-IE facts and their contextual information. Our method is oblivious to the actual open IE method being used.

Relation clustering. One way to reason about KB and surface relations is to cluster the relations: whenever two relations appear in the same cluster, they are treated as synonymous (Hasegawa et al., 2004; Shinyama and Sekine, 2006; Yao

et al., 2011; Takamatsu et al., 2011; Min et al., 2012; Akbik et al., 2012; de Lacalle and Lapata, 2013). For example, if "criticizes" and "hates" are clustered together, then we may predict "hates"("Dante", "Catholic Church") from the above fact (which is actually not true). The general problem with relation clustering is its "black and white" approach to relations: either two relations are the same or they are different. This assumption generally does not hold for the surface relations extracted by open IE systems (Riedel et al., 2013); examples of other types of relationships between relations include implication or mutual exclusion.

Tensor factorization. Matrix or tensor factorization approaches try to address the above problem: instead of clustering relations, they directly predict facts. Both matrix and tensor models learn and make use of semantic representations of relations and their arguments. The semantic representations ideally captures all the information present in the data; it does not, however, establish a direct relationship (such as synonymy) between different KB or surface relations.

Tensor factorization models conceptually model the input data as a subject×relation×object tensor, in which non-zero values correspond to input facts. The tensor is factored to construct a new tensor in which predicted facts take large non-zero values. Examples of such tensor factorization models are TripleRank (Franz et al., 2009), RESCAL (Nickel et al., 2011; Nickel et al., 2012), or PITF (Drumond et al., 2012). Tensor factorization models are generally well-suited to reason within a KB because they are able to predict relations between arbitrary pairs of subjects and objects. In the context of open RE, however, these methods suffer from limited scalability with the number of relations as well as from their large prediction space (Chang et al., 2014).

Matrix factorization. The key difference between matrix and tensor factorization models is that the former restrict the prediction space, i.e., these models generally cannot predict arbitrary facts. Similar to distant supervision approaches, matrix factorization models focus on predicting facts for which some direct evidence exists. In more detail, most methods restrict the prediction space to the set of facts for which the subject and the object share at least some relation in the input data. For this reason, matrix factorization models are not suited for in-KB reasoning; an individual pair of entities usually does not occur in more than one KB relation. In the open RE context, however, input relations are semantically related so that many subject-object pairs belong to multiple relations. The key advantage of matrix methods is (1) that this restriction allows them to use additional features—such as features for each subject-object pair—and (2) that they scale much better with the number of relations. Examples of such matrix factorization models include (Tresp et al., 2009; Jiang et al., 2012; Fan et al., 2014; Huang et al., 2014). Chang et al. (2014) have also shown that a combination of matrix and tensor factorization models can be fruitful. Closest to our work is the "universal schema" matrix factorization approach of Riedel et al. (2013), which combines a latent features model, a neighborhood model and an entity model but does not incorporate context. Our CORE model follows the universal schema idea, but uses a more general factorization model, which includes the information captured by the latent features and entity model (but not the neighborhood model), and incorporates contextual information.

Using contextual information. It is well known that contextual information can improve IE methods. Information such as bag-of-words, part-of-speech tags, entity types, or parse trees have been integrated into many existing systems (Mintz et al., 2009; Zhang et al., 2012; Takamatsu et al., 2011; Zhou et al., 2007; de Lacalle and Lapata, 2013; Akbik et al., 2012). Our work differs in that we integrate contextual information into an open RE system. To do so, we leverage factorization machines (Rendle et al., 2011; Rendle, 2012), which have been successfully applied to exploit contextual information in the context of recommender systems. We show how to model open RE data and context with factorization machines and provide a method for parameter estimation under the open-world assumption.

3 The CORE Model

Input data. We model the input data as a set of *observations* of the form (r, t, c), where r refer to a KB or surface relation, t refer to a subject-object pair of entities (or entity mentions) and c to contextual information. An observation obtained from the example of the introduction may be ("join", (TP, MM), { types:(person,org),

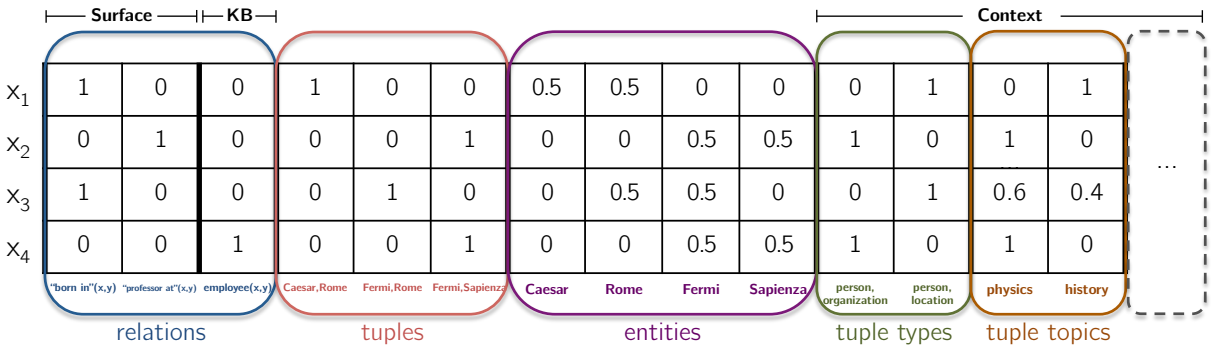

Figure 1: Example for representing a context-aware open RE problem with CORE

topic:music, word:record, word:album, ...}). Denote by R the set of all observed *relations*, by E the set of all observed *entities*, and by $T \subseteq E \times E$ the set of all observed entity pairs, which we refer to as *tuples*. A *fact* takes form $r(t)$ and is composed of a relation $r \in R$ and a tuple $t \in T$; e.g., "join"(TP, MM). Note that there may be multiple observations for a fact. Finally, denote by C the set of all *contextual variables*; each observation is associated with a set $c \subseteq C$ of context variables. In this paper we restrict attention to categorical context variables; our model can potentially handle continuous context as in (Rendle, 2012).

Problem definition. The open RE task is to produce a ranked list of tuples $T_r \subseteq T$ for each relation $r \in R$; the list is restricted to new tuples, i.e., tuples $t \in T$ for which $r(t)$ has not been observed in the input. The rank of each tuple reflects the model's prediction of the likelihood that the corresponding fact is indeed true. A good model thus ranks correct facts higher than incorrect ones.

Modeling facts. Denote by $V = R \cup T \cup E \cup C$ the set of all observed relations, tuples, entities, and contextual variables. For ease of exposition, we refer to the elements of V as *variables*. We model the input data in terms of a matrix in which each row corresponds to a fact (i.e., not an observation) and each column to a variable. We group columns according to the type of the variables; e.g, there are relation columns, tuple columns, entity columns, and a group of columns for each type of contextual information. The matrix is populated such that in each row the values of each column group sum up to unity, i.e., we normalize values within column groups. In particular, we set to 1 the values of the variable of the relation and the tuple of the corresponding fact. We set to 0.5 the variables corresponding to the two entities referred to by the fact. An example is shown in Fig. 1. Here the first row, for instance, corresponds to the fact "born in"(Caesar, Rome). Note that we model tuples and entities separately: the entity variables expose which arguments belong to the fact, the tuple variables expose their order.

Modeling context. As described above, we model the data in terms of a matrix in which rows corresponds to facts (instead of observations). The reasoning behind this approach is as follows. First, we may see a fact in multiple observations; our goal is to leverage all the available context. Second, facts but not observations are the target of our predictions. Finally, we are interested in predicting new facts, i.e., facts that we have not seen in the input data. For these facts, there is no corresponding observation so that we cannot directly obtain contextual information. To address these points, our model aggregates the context of relevant observations for each fact; this approach allows us to provide comprehensive contextual information for both observed and unobserved facts.

We group contextual information by the type of information: examples include metadata about the extraction sources (e.g., from an article on music), types of the entities of a tuple (e.g., (person, location)), or the bag-of-words in the sentence from which an extraction has been obtained. We aggregate the contextual information for each tuple $t \in T$; this tuple-level approach allows us to provide contextual information for unobserved facts. In more detail, we count in how many observations each contextual variable has been associated with the tuple, and then normalize the count values to 1 within each group of columns. The so-obtained values can be interpreted as the relative frequencies with which each contextual variable is associated with the tuple. The contextual information as-

sociated with each fact is given by the aggregated, normalized context of its tuple.

Fig. 1 shows context information arranged in two groups: tuple types and tuple topics. We capture information such as that the tuple (Caesar, Rome) has only been seen in articles on history or that tuple (Fermi, Rome) is mentioned in both physics and history articles (slightly more often in the former). Since context is associated with tuples, facts 2 and 4 on (Fermi, Sapienza) share contextual information. This form of context sharing (as well as entity sharing) allows us to propagate information about tuples across various relations.

Factorization model. CORE employs a matrix factorization model based on factorization machines and the open-world assumption to capture latent semantic information about the individual variables. In particular, we associate with each variable $v \in V$ a *bias term* $b_v \in \mathbb{R}$ and a *latent feature vector* $\boldsymbol{f}_v \in \mathbb{R}^d$, where the *dimensionality* d of the latent feature space is a hyperparameter of our model. Denote by X the set of rows in the input matrix, which we henceforth refer to as *training points*. For each training point $x \in X$, denote by x_v the value of variable $v \in V$ in the corresponding row of the matrix. Our model associates with training point $x \in X$ a *score* $s(x)$ computed as follows:

$$s(x) = \sum_{v \in V} x_v b_v + \sum_{v_1 \in V} \sum_{v_2 \in V \setminus \{v_1\}} x_{v_1} x_{v_2} \boldsymbol{f}_{v_1}^T \boldsymbol{f}_{v_2} \tag{1}$$

Here the bias terms models the contribution of each individual variable to the final score, whereas the latent feature vectors model the contribution of all pairwise interactions between variables. Note that only bias terms and feature vectors corresponding to non-zero entries in x affect the score and that x is often sparse. Since we can compute $s(x)$ in time linear to both the number of nonzero entries in x and the dimensionality d (Rendle, 2012), score computation is fast. As discussed below, we (roughly) estimate bias terms and feature vectors such that observed facts achieve high scores. We may thus think of each feature vector as a low-dimensional representation of the global information contained in the corresponding variable.

Prediction. Given estimates for bias terms and latent feature vectors, we rank unobserved facts as follows. Fix a relation $r \in R$ and a tuple $t \in T$ such that $r(t)$ has not been observed. As indicated above, our model overcomes the key problem that there is no observation, and thus no context, for $r(t)$ by context aggregation and sharing. In particular, we create an *test point* $\hat{x}$ for tuple $r(t)$ in a way similar to creating data points, i.e., we set the relation, tuple, and entity variables accordingly and add the aggregated, normalized context of t. Once test point $\hat{x}$ has been created, we can predict its score $s(\hat{x})$ using Eq. (1). We then rank each unobserved tuple by its so-obtained score, i.e., tuples with higher scores are ranked higher. The resulting ranking constitutes the list T_r of predicted facts for relation r.

Bayesian personalized ranking. The parameters of our model are given by $\Theta = \{\, b_v, \boldsymbol{f}_v \mid v \in V \,\}$. In approaches based on the closed-world assumption, Θ is estimated by minimizing the error between model predictions and target values (e.g., 1 for true facts, 0 for false facts). In our setting of open RE, all our observations are positive, i.e., we do not have negative training data. One way to handle the absence of negative training data is to associate a target value of 0 to all unobserved facts. This closed-world approach essentially assumes that all unobserved facts are false, which may not be a suitable assumption for the sparsely observed relations of open RE. Following Riedel et al. (2013), we adopt the open-world assumption instead, i.e., we treat each unobserved facts as unknown. Since factorization machines originally require explicit target values (e.g., feedback in recommender systems), we need to adapt parameter estimation to the open-world setting.

In more detail, we employ a variant of the Bayesian personalized ranking (BPR) optimization criterion (Rendle et al., 2009). We associate with each training point x a set of *negative samples* X_x^-. Each negative sample $x^- \in X_x^-$ is an unobserved fact with its associated context (constructed as described in the prediction section above). Generally, the negative samples x^- should be chosen such that they are "less likely" to be true than fact x. We maximize the following optimization criterion:

$$\frac{1}{|X|} \sum_{x \in X} \left(\sum_{x^- \in X_x^-} \frac{\ln \sigma(\delta(x, x^-))}{|X_x^-|} - \lambda \|\Theta_x\|^2 \right) \tag{2}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ denotes the logistic function, $\delta(x, x^-) = s(x) - s(x^-)$ denotes the difference of scores, and $\Theta_x = \{ b_v, \boldsymbol{f}_v \mid x_v \neq 0 \}$ the subset of the model parameters relevant for training point x. Here we use L2 regularization controlled by a single hyperparameter λ. In essence, the BPR criterion aims to maximize the average "difference" $\ln \sigma(\delta(x, x^-))$ between the score of fact x and each of its negative samples x^-, averaged over all facts. In other words, we aim to score x higher than each x^-. (Note that under the closed-world assumption, we would instead consider x^- as being false.) For a more in-depth discussion of BPR, see (Rendle et al., 2009).

Sampling negative evidence. To make BPR effective, the set of negative samples needs to be chosen carefully. A naive approach is to take the set of all unobserved facts between each relation $r \in R$ and each tuple $t \in T$ (or $E \times E$) as the set X_x^-. The reasoning is that, after all, we expect "random" unobserved facts to be less likely to be true than observed facts. This naive approach is problematic, however, because the set of negative samples is independent of x and thus not sufficiently informative (i.e., it contains many irrelevant samples).

To overcome this problem, the negative sample set needs to be related to x in some way. Since we ultimately use our model to rank tuples for each relation individually, we consider as negative evidence for x only unobserved facts from the same relation (Riedel et al., 2013). In more detail, we (conceptually) build a negative sample set X_r^- for each relation $r \in R$. We include into X_r^- all facts $r(t)$—again, along with their context—such that $t \in T$ is an observed tuple but $r(t)$ is an unobserved fact. Thus the subject-object pair t of entities is not observed with relation r in the input data (but with some other relation). The set of negative samples associated with each training point x is defined by the relation r of the fact contained in x, that is $X_x^- = X_r^-$. Note that we do not actually construct the negative sample sets; see below.

Parameter estimation. We maximize Eq. (2) using stochastic gradient ascent. This allows us to avoid constructing the sets X_x^-, which are often infeasibly large, and worked well in our experiments. In particular, in each stochastic gradient step, we randomly sample a training point $x \in X$, and subsequently randomly sample a negative sample $x^- \in X_x^-$. This sampling procedure can be implemented very efficiently. We then perform the following ascent step with learning rate η:

$$\Theta \leftarrow \Theta + \eta \nabla_\Theta \left(\ln \sigma(d(x, x^-)) - \lambda \|\Theta_x\|^2 \right)$$

One can show that the stochastic gradient used in the formula above is an unbiased estimate of the gradient of Eq. (2). To speed up parameter estimation, we use a parallel lock-free version of stochastic gradient ascent as in Recht et al. (2011). This allows our model to handle (reasonably) large datasets.

	size	info
facts	453.9k	14.7k Freebase, 174.1k surface linked, 184.5k surface partially-linked, 80.6k surface non-linked.
relations	4.7k	94 Freebase, 4.6k surface.
tuples	178.5k	69.5k linked, 71.5k partially-linked, 37.5k non-linked.
entities	114.2k	36.8k linked, 77.4k non-linked.

Table 1: Dataset statistics.

4 Experiments

We conducted an experimental study on real-world data to compare our CORE model with other state-of-the-art approaches.[2] Our experimental study closely follows the one of Riedel et al. (2013).

4.1 Experimental Setup

Dataset. We made use of the dataset of Riedel et al. (2013), but extended it with contextual information. The dataset consisted of 2.5M surface facts extracted from the New York Times corpus (Sandhaus, 2008), as well as 16k facts from Freebase. Surface facts have been obtained by using a named-entity recognizer, which additionally labeled each named entity mention with its coarse-grained type (i.e., person, organization, location, miscellaneous). For each pair of entities found within a sentence, the shortest dependency path between these pairs was taken as surface relation. The entity mentions in each surface fact were

[2] Source code, datasets, and supporting material are available at `http://dws.informatik.uni-mannheim.de/en/resources/software/core/`

Relation	#	PITF	NFE	CORE	CORE+m	CORE+t	CORE+w	CORE+mt	CORE+mtw
person/company	208	70 (0.47)	92 (0.81)	91 (0.83)	90 (0.84)	91 (0.87)	92 (0.87)	95 (0.93)	**96** (**0.94**)
person/place_of_birth	117	1 (0.0)	92 (*0.9*)	90 (0.88)	92 (*0.9*)	92 (*0.9*)	89 (0.87)	**93** (*0.9*)	92 (*0.9*)
location/containedby	102	7 (0.0)	63 (0.47)	62 (0.47)	63 (0.46)	61 (0.47)	61 (0.44)	62 (0.49)	**68** (**0.55**)
parent/child	88	9 (0.01)	64 (0.6)	64 (0.56)	64 (0.59)	64 (0.62)	64 (0.57)	67 (**0.67**)	**68** (0.63)
person/place_of_death	71	1 (0.0)	67 (0.93)	67 (0.92)	*69* (*0.94*)	67 (0.93)	67 (0.92)	*69* (*0.94*)	67 (0.92)
person/parents	67	20 (0.1)	51 (0.64)	52 (0.62)	51 (0.61)	49 (0.64)	47 (0.6)	*53* (**0.67**)	*53* (0.65)
author/works_written	65	24 (0.08)	45 (0.59)	49 (0.62)	51 (0.69)	50 (0.68)	50 (0.68)	51 (**0.7**)	**52** (0.67)
person/nationality	61	21 (0.08)	25 (0.19)	27 (0.17)	28 (0.2)	26 (0.2)	**29** (0.19)	27 (0.18)	27 (**0.21**)
neighbor./neighborhood_of	39	3 (0.0)	24 (0.44)	23 (0.45)	26 (0.5)	27 (0.47)	27 (0.49)	*30* (0.51)	*30* (**0.52**)
film/directed_by	15	7 (0.06)	7 (0.15)	11 (0.22)	9 (0.25)	10 (0.27)	**15** (**0.52**)	11 (0.28)	12 (0.31)
company/founders	11	0 (0.0)	*10* (*0.34*)	*10* (*0.34*)	*10* (0.26)	*10* (0.21)	*10* (0.22)	*10* (0.22)	*10* (0.24)
sports_team/league	11	1 (0.0)	7 (0.24)	*10* (0.23)	*10* (*0.3*)	7 (0.22)	*10* (0.27)	8 (0.29)	9 (*0.3*)
structure/architect	11	7 (0.63)	7 (0.63)	9 (0.7)	*11* (0.84)	*11* (0.73)	*11* (**0.9**)	*11* (0.8)	10 (0.77)
team/arena_stadium	9	2 (0.01)	6 (0.14)	6 (0.19)	6 (0.18)	6 (0.15)	6 (0.18)	*7* (**0.29**)	*7* (0.2)
team_owner/teams_owned	9	4 (0.05)	6 (0.17)	7 (0.18)	7 (0.33)	6 (0.27)	7 (0.19)	6 (0.22)	**8** (**0.34**)
film/produced_by	8	1 (0.03)	4 (0.06)	3 (0.13)	2 (0.12)	3 (0.03)	*6* (0.09)	3 (0.13)	*6* (**0.15**)
roadcast/area_served	5	0 (0.0)	4 (0.71)	4 (**0.73**)	4 (0.65)	4 (0.66)	4 (0.66)	*5* (0.64)	*5* (0.72)
person/religion	5	2 (0.0)	*3* (0.21)	2 (0.22)	1 (0.2)	*3* (0.22)	*3* (**0.25**)	2 (0.21)	*3* (0.21)
composer/compositions	3	*2* (0.1)	*2* (0.34)	*2* (0.35)	*2* (0.34)	*2* (0.35)	1 (0.33)	*2* (0.22)	*2* (**0.36**)
Average $MAP^{100}_{\#}$		0.09	0.46	0.47	0.49	0.47	0.49	0.49	**0.51**
Weighted Average $MAP^{100}_{\#}$		0.14	0.64	0.64	0.66	0.67	0.66	*0.70*	*0.70*

Table 2: True facts and $MAP^{100}_{\#}$ (in parentheses) in the top-100 evaluation-set tuples for Freebase relations. We consider as context the article metadata (m), the tuple types (t) and the bag-of-words (w). Best value per relation in bold (unique winner) or italic (multiple winners). Average weighs are # column values.

linked to Freebase using a simple string matching method. If no match was found, the entity mention was kept as is. There were around 2.2M tuples (distinct entity pairs) in this dataset, out of which 580k were fully linked to Freebase. For each of these tuples, the dataset additionally included all of the corresponding facts from Freebase. Using the metadata[3] of each New York Times article, we enriched each surface fact by the following contextual information: news desk (e.g., sports desk, foreign desk), descriptors (e.g., finances, elections), online section (e.g., sports, business), section (e.g., a, d), publication year, and bag-of-words of the sentence from which the surface fact has been extracted.

Training data. From the raw dataset described above, we filtered out all surface relations with less than 10 instances, and all tuples with less than two instances, as in Riedel et al. (2013). Tab. 1 summarizes statistics of the resulting dataset. Here we considered a fact or tuple as *linked* if both of its entities were linked to Freebase, as *partially-linked* if only one of its entities was linked, and as *non-linked* otherwise. In contrast to previous work (Riedel et al., 2013; Chang et al., 2014), we retain partially-linked and non-linked facts in our dataset.

Evaluation set. Open RE models produce predictions for all relations and all tuples. To keep the experimental study feasible and comparable to previous studies, we use the full training data but evaluate each model's predictions on only the subsample of 10k tuples ($\approx 6\%$ of all tuples) of Riedel et al. (2013). The subsample consisted of 20% linked, 40% partially-linked and 40% non-linked tuples. For each (surface) relation and method, we predicted the top-100 new facts (not in training) for the tuples in the subsample.

Considered methods. We compared various forms of our CORE model with PITF and the matrix factorization model NFE. Our study focused on these two factorization models because they outperformed other models (including non-factorization models) in previous studies (Riedel et al., 2013; Chang et al., 2014). All models were trained with the full training data described above.

PITF (Drumond et al., 2012). PITF is a recent tensor factorization method designed for within-KB reasoning. PITF is based on factorization machines so that we used our scalable CORE implementation for training the model.

NFE (Riedel et al., 2013). NFE is the full model proposed in the "universal schema" work of Riedel et al. (2013). It uses a linear combina-

[3]Further information can be found at `https://catalog.ldc.upenn.edu/LDC2008T19`.

Relation	#	PITF	NFE	CORE	CORE+m	CORE+t	CORE+w	CORE+mt	CORE+mtw
head	162	34 (0.18)	80 (0.66)	83 (0.66)	82 (0.63)	76 (0.57)	77 (0.57)	83 (0.69)	**88** (**0.73**)
scientist	144	44 (0.17)	76 (0.6)	74 (0.55)	73 (0.56)	74 (0.6)	73 (0.59)	*78* (0.66)	*78* (**0.69**)
base	133	10 (0.01)	85 (0.71)	86 (0.71)	86 (0.78)	88 (0.79)	85 (0.75)	83 (0.76)	**89** (**0.8**)
visit	118	4 (0.0)	73 (0.6)	75 (0.61)	76 (0.64)	80 (0.68)	74 (0.64)	75 (0.66)	**82** (**0.74**)
attend	92	11 (0.02)	65 (0.58)	64 (0.59)	65 (0.63)	62 (0.6)	66 (0.63)	62 (0.58)	**69** (**0.64**)
adviser	56	2 (0.0)	42 (0.56)	**47** (0.58)	44 (0.58)	43 (0.59)	45 (*0.63*)	43 (0.53)	44 (*0.63*)
criticize	40	5 (0.0)	31 (0.66)	33 (0.62)	33 (**0.7**)	33 (0.67)	33 (0.61)	35 (0.69)	**37** (0.69)
support	33	3 (0.0)	19 (0.27)	22 (*0.28*)	18 (0.21)	19 (*0.28*)	22 (0.27)	**23** (0.27)	21 (0.27)
praise	5	0 (0.0)	2 (0.0)	2 (0.01)	4 (*0.03*)	3 (0.01)	3 (0.02)	**5** (*0.03*)	2 (0.01)
vote	3	2 (0.01)	*3* (0.63)	*3* (0.63)	*3* (0.32)	*3* (0.49)	*3* (0.51)	*3* (0.59)	*3* (**0.64**)
Average $\text{MAP}_{\#}^{100}$		0.04	0.53	0.53	0.51	0.53	0.53	0.55	**0.59**
Weighted Average $\text{MAP}_{\#}^{100}$		0.08	0.62	0.61	0.63	0.63	0.61	0.65	**0.70**

Table 3: True facts and $\text{MAP}_{\#}^{100}$ (in parentheses) in the top-100 evaluation-set tuples for surface relations. We consider as context the article metadata (m), the tuple types (t) and the bag-of-words (w). Best value per relation in bold (unique winner) or italic (multiple winners). Average weighs are # column values.

tion of three component models: a neighborhood model (N), a matrix factorization model (F), and an entity model (E). The F and E models together are similar (but not equal) to our CORE model without context. The NFE model outperformed tensor models (Chang et al., 2014) as well as clustering methods and distantly supervised methods in the experimental study of Riedel et al. (2013) for open RE tasks. We use the original source code of Riedel et al. (2013) for training.

CORE. We include multiple variants of our model in the experimental study, each differing by the amount of context being used. We consider as context the article metadata (m), the tuple types (t) and the bag-of-words (w). Each tuple type is a pair of subject-object types of (e.g. (person, location)). The basic CORE model uses relations, tuples and entities as variables. We additionally consider the CORE+t, CORE+w, CORE+mt, and CORE+mtw models, where the suffix indicates which contextual information has been included. The total number of variables in the resulting models varied between 300k (CORE) to 350k (CORE+mtw). We used a modified version of *libfm* for training.[4] Our version adds support for BPR and parallelizes the training algorithm.

Methodology. To evaluate the prediction performance of each method, we followed Riedel et al. (2013). We considered a collection of 19 Freebase relations (Tab. 2) and 10 surface relations (Tab. 3) and restrict predictions to tuples in the evaluation set.

Evaluation metrics. For each relation and method, we computed the top-100 evaluation set predictions and labeled them manually. We used as evaluation metrics the mean average precision defined as:

$$\text{MAP}_{\#}^{100} = \frac{\sum_{k=1}^{100} I_k \cdot P@k}{\min\{100, \#\}} \tag{3}$$

where indicator I_k takes value 1 if the k-th prediction is true and 0 otherwise, and $\#$ denotes the number of true tuples for the relation in the top-100 predictions of all models. The denominator is included to account for the fact that the evaluation set may include less than 100 true facts. $\text{MAP}_{\#}^{100}$ reflects how many true facts are found by each method as well as their ranking. If all $\#$ facts are found and ranked top, then $\text{MAP}_{\#}^{100} = 1$. Note that our definition of $\text{MAP}_{\#}^{100}$ differs slightly from Riedel et al. (2013); our metric is more robust because it is based on completely labeled evaluation data. To compare the prediction performance of each system across multiple relations, we averaged $\text{MAP}_{\#}^{100}$ values, in both an unweighted and a weighted (by $\#$) fashion.

Parameters. For all systems, we used $d = 100$ latent factors, $\lambda = 0.01$ for all variables, a constant learning rate of $\eta = 0.05$, and ran 1000 epochs of stochastic gradient ascent. These choices correspond to the ones of Riedel et al. (2013); no further tuning was performed.

4.2 Results.

Prediction performance. The results of our experimental study are summarized in Tab. 2 (Freebase relations) and Tab. 3 (surface relations). As mentioned before, all reported numbers are with

[4] http://www.libfm.org

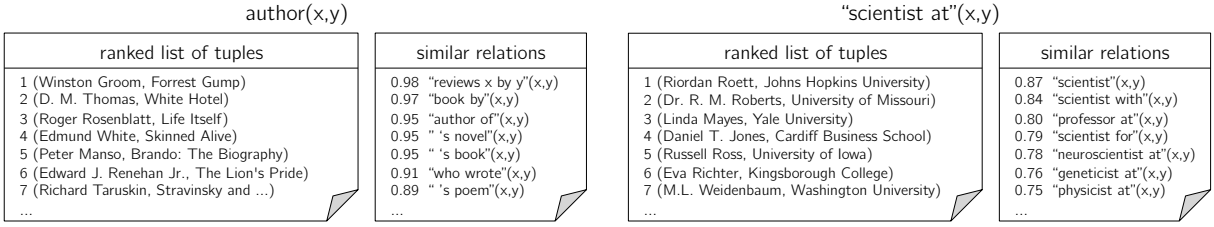

Figure 2: Some facts predicted by our model for the Freebase relation author(x,y) and the surface relation "scientist at"(x,y). Most similar relations also reported, using cosine similarity between the corresponding latent feature vectors as distance.

respect to our evaluation set. Each entry shows the number of true facts in the top-100 predictions and, in parentheses, the $\text{MAP}_{\#}^{100}$ value. The # column list the total number of true facts found by at least one method. The last two lines show the aggregated $\text{MAP}_{\#}^{100}$ scores.

We start our discussion with the results for Freebase relations (Tab. 2). First note that the PITF model generally did not perform well; as discussed before, tensor factorization models such as PITF suffer from a large prediction space and cannot incorporate tuple-level information. NFE and CORE, both matrix factorization models, performed better and were on par with each other. This indicates that our use of factorization machines does not affect performance in the absence of context; after all, both methods essentially make use of the same amount of information. The key advantage of our model over NFE is that we can incorporate contextual information. Our results indicate that using such information indeed improves prediction performance. The CORE+mtw model performed best overall; it increased the average $\text{MAP}_{\#}^{100}$ by four points (six points weighted) compared to the best context-unware model. Note that for some relations, including only subsets of the contextual information produced better results than using all contextual information (e.g., film/directed_by). We thus conjecture that extending our model by variable-specific regularization terms may be beneficial.

Tab. 3 summarizes our results for surface relations. In general, the relative performance of the models agreed with the one on Freebase relations. One difference is that using bag-of-word context significantly boosted prediction performance. One reason for this boost is that related surface relations often share semantically related words (e.g., "professor at" and "scientist at") and may occur in similar sentences (e.g., mentioning "university", "research", ...).

Anecdotal results. Fig. 2 shows the top test-set predictions of CORE+mtw for the author and "scientist at" relations. In both cases, we also list relations that have a similar semantic representation in our model (highest cosine similarity). Note that semantic similarity of relations is one aspect of our model; predictions incorporate other aspects such as context (i.e., two "similar" relations in different contexts are treated differently).

Training time. We used a machine with 16-cores Intel Xeon processor and 128GB of memory. Training CORE took roughly one hour, NFE roughly six hours (single core only), and training CORE+mtw took roughly 20 hours. Our implementation can handle reasonably large data, but an investigation of faster, more scalable training methods appears worthwhile.

5 Conclusion

We proposed CORE, a matrix factorization model for open RE that incorporates contextual information. Our model is based on factorization machines and the open-world assumption, integrates various forms of contextual information, and is extensible. Our experimental study suggests that exploiting context can significantly improve prediction performance.

References

[Akbik et al.2012] Alan Akbik, Larysa Visengeriyeva, Priska Herger, Holmer Hemsen, and Alexander Löser. 2012. Unsupervised discovery of relations and discriminative extraction patterns. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*.

[Banko et al.2007] Michele Banko, Michael J Cafarella, Stephen Soderl, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*.

[Chang et al.2014] Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[de Lacalle and Lapata2013] Oier Lopez de Lacalle and Mirella Lapata. 2013. Unsupervised relation extraction with general domain knowledge. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Del Corro and Gemulla2013] Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*.

[Drumond et al.2012] Lucas Drumond, Steffen Rendle, and Lars Schmidt-Thieme. 2012. Predicting rdf triples in incomplete knowledge bases with tensor factorization. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC)*.

[Fader et al.2011] Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Fan et al.2014] Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Thomas Fang Zheng, and Edward Y Chang. 2014. Distant supervision for relation extraction with matrix completion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Franz et al.2009] Thomas Franz, Antje Schultz, Sergej Sizov, and Steffen Staab. 2009. Triplerank: Ranking semantic web data by tensor decomposition. In *Proceedings of the 8th International Semantic Web Conference (ISWC)*.

[Hasegawa et al.2004] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL)*.

[Huang et al.2014] Yi Huang, Volker Tresp, Maximilian Nickel, Achim Rettinger, and Hans-Peter Kriegel. 2014. A scalable approach for statistical learning in semantic graphs. *Semantic Web*, 5(1):5–22.

[Jiang et al.2012] Xueyan Jiang, Volker Tresp, Yi Huang, and Maximilian Nickel. 2012. Link prediction in multi-relational graphs using additive models. In *Proceedings of the 2012 International Workshop on Semantic Technologies meet Recommender Systems & Big Data (SeRSy)*.

[Min et al.2012] Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. 2012. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

[Min et al.2013] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*.

[Mintz et al.2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*.

[Nickel et al.2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML)*.

[Nickel et al.2012] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st International Conference on World Wide Web (WWW)*.

[Recht et al.2011] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS)*.

[Rendle et al.2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*.

[Rendle et al.2011] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM conference on Research and development in Information Retrieval (SIGIR)*.

[Rendle2012] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57.

[Riedel et al.2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*.

[Sandhaus2008] E. Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12).

[Shinyama and Sekine2006] Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*.

[Surdeanu et al.2012] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

[Takamatsu et al.2011] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2011. Probabilistic matrix factorization leveraging contexts for unsupervised relation extraction. In *Advances in Knowledge Discovery and Data Mining*, pages 87–99. Springer.

[Tresp et al.2009] Volker Tresp, Yi Huang, Markus Bundschus, and Achim Rettinger. 2009. Materializing and querying learned knowledge. In *Proceedings of the 2009 International Workshop on Inductive Reasoning and Machine Learning for the Semantic Web (IRMLeS)*.

[Yao et al.2011] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Zhang et al.2012] Congle Zhang, Raphael Hoffmann, and Daniel S. Weld. 2012. Ontological smoothing for relation extraction with minimal supervision. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*.

[Zhou et al.2007] GuoDong Zhou, Min Zhang, DongHong Ji, and QiaoMing Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.