PROBABILITY CALIBRATION FOR KNOWLEDGE GRAPH EMBEDDING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge graph embedding research has overlooked the problem of probability calibration. We show popular embedding models are indeed uncalibrated. That means probability estimates associated to predicted triples are unreliable. We present a novel method to calibrate a model when ground truth negatives are not available, which is the usual case in knowledge graphs. We propose to use Platt scaling and isotonic regression alongside our method. Experiments on three datasets with ground truth negatives show our contribution leads to well calibrated models when compared to the gold standard of using negatives. We get significantly better results than the uncalibrated models from all calibration methods. We show isotonic regression offers the best the performance overall, not without trade-offs. We also show that calibrated models reach state-of-the-art accuracy without the need to define relation-specific decision thresholds.

1 INTRODUCTION

Knowledge graph embedding models are neural architectures that learn vector representations (i.e. embeddings) of nodes and edges of a knowledge graph. Such knowledge graph embeddings have applications in knowledge graph completion, knowledge discovery, entity resolution, and link-based clustering, just to cite a few (Nickel et al., 2016).

Despite burgeoning research, the problem of calibrating such models has been overlooked, and existing knowledge graph embedding models do not offer any guarantee on the probability estimates they assign to predicted facts. Probability calibration is important whenever you need the predictions to make probabilistic sense, i.e., if the model predicts a fact is true with 80% confidence, it should to be correct 80% of the times. Prior art suggests to use a sigmoid layer to turn logits returned by models into probabilities (Nickel et al., 2016) (also called the expit transform), but we show that this provides poor calibration. Figure 1 shows reliability diagrams for off-the-shelf TransE and ComplEx. The identity function represents perfect calibration. Both models are miscalibrated: all TransE combinations in Figure 1a under-forecast the probabilities (i.e. probabilities are too small), whereas ComplEx under-forecasts or over-forecasts according to which loss is used (Figure1b).

Calibration is crucial in high-stakes scenarios such as drug-target discovery from biological networks, where end-users need trustworthy and interpretable decisions. Moreover, since probabilities are not calibrated, when classifying triples (i.e. facts) as true or false, users must define relation-specific thresholds, which can be awkward for graphs with a great number of relation types.

To the best of our knowledge, this is the first work to focus on calibration for knowledge embeddings. Our contribution is two-fold: First, we use Platt Scaling and isotonic regression to calibrate knowledge graph embedding models on datasets that include ground truth negatives. One peculiar feature of knowledge graphs is that they usually rely on the *open world assumption* (i.e. facts not present are not necessarily false, they are simply unknown). This makes calibration troublesome because of the lack of ground truth negatives. For this reason, our second and main contribution is a calibration heuristics that combines Platt-scaling or isotonic regression with synthetically generated negatives.

Experimental results show that we obtain better-calibrated models and that it is possible to calibrate knowledge graph embedding models even when ground truth negatives are not present. We

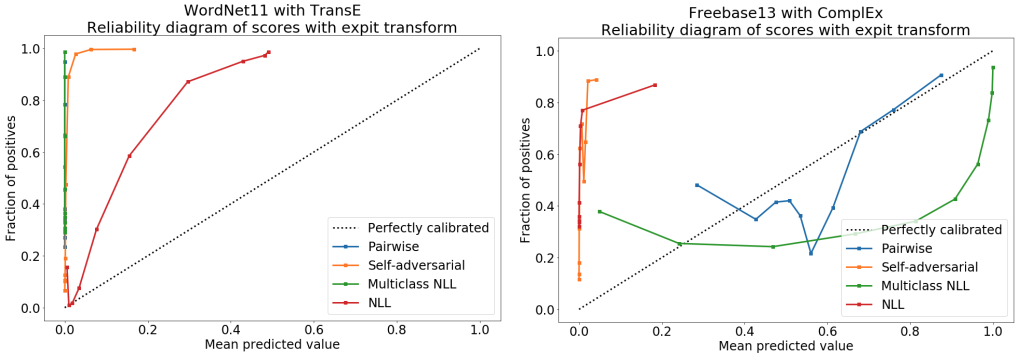

Figure 1: Reliability diagrams of uncalibrated models. Probabilities are generated by a logistic sigmoid layer. The larger the deviation from the diagonal, the more uncalibrated is the model. We present four different common loss functions used to train knowledge graph embedding models. (a) Uncalibrated TransE on WN11. (b) Uncalibrated ComplEx on FB13. Best viewed in colors.

also experiment with triple classification, and we show that calibrated models reach state-of-the-art accuracy without the need to define relation-specific decision thresholds.

2 Related Work

A comprehensive survey of knowledge graph embedding models is out of the scope of this work. Works such as (Nickel et al., 2016) and (Cai et al., 2017) present a survey of recent work, although they do not include the most recent models such as ConvE (Dettmers et al., 2018) or RotatE (Sun et al., 2019). In this paper we limit our analysis to the three most established models: TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), and ComplEx (Trouillon et al., 2016). They do not address the problem of assessing the reliability of predictions, leave aside calibrating probabilities.

Besides well-established techniques such as Platt scaling (Platt et al., 1999) and isotonic regression (Zadrozny & Elkan, 2002), recent interest in neural architectures calibration show that modern neural architectures are poorly calibrated and that calibration can be improved with novel methods. For example, (Guo et al., 2017) successfully proposes to use temperature scaling for calibrating modern neural networks.

The Knowledge Vault pipeline in (Dong et al., 2014) extracts triples from unstructured knowledge and is equipped with Platt scaling calibration, but this is not applied to knowledge graph embedding models. To the best of our knowledge, the only work that adopts probability calibration to knowledge graph embedding models is Krompaß & Tresp (2015). The authors propose to use ensembles in order to improve the results of knowledge graph embedding tasks. For that, they propose to calibrate the models with Platt scaling, so they operate on the same scale. No further details on the calibration procedure are provided. Besides, there is no explanation on how to handle the lack of negatives.

3 Preliminaries

Knowledge Graph. Formally, a knowledge graph $\mathcal{G} = \{(s, p, o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a set of triples $t = (s, p, o)$, each including a subject $s \in \mathcal{E}$, a predicate $p \in \mathcal{R}$, and an object $o \in \mathcal{E}$. $\mathcal{E}$ and $\mathcal{R}$ are the sets of all entities and relation types of $\mathcal{G}$.

Triple Classification. Binary classification task where $\mathcal{G}$ (which includes only positive triples) is used as training set, and $\mathcal{T} = \{(s, p, o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a disjoint test set of *labeled* triples to classify. Note $\mathcal{T}$ includes positives and negatives. Since the learned models are not calibrated, multiple decision thresholds τ_i must be picked, where $0 < i < |\mathcal{R}|$, i.e. one for each relation type. This is done using a validation set (Bordes et al., 2013). Classification metrics apply (e.g. accuracy).

Link Prediction. Given a training set $\mathcal{G}$ that includes only positive triples, the goal is assigning a score $f(t) \in \mathbb{R}$ proportional to the likelihood that each *unlabeled* triple t included in a held-out set

	WN11	FB13	YAGO39K	FB15K-237	WN18RR
Training	112,581	316,232	354,996	272,115	86,835
Validation	5,218	11,816	18,682	17,535	3,034
Test	21,088	47,466	18,728	20,466	3,134
Entities	38,696	75,043	39,374	14,541	40,943
Relations	11	13	39	237	11

(a)

Model	**Scoring Function f_m**
TransE	$-\|\|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\|\|_n$
DistMult	$\langle \mathbf{e}_s, \mathbf{r}_p, \mathbf{e}_o \rangle$
ComplEx	$Re(\langle \mathbf{e}_s, \mathbf{r}_p, \overline{\mathbf{e}_o} \rangle)$

(b)

Table 1: (a) Triple classification datasets used in experiments (left); link prediction datasets used for positive base rate experiments (right); (b) Scoring functions of models used in experiments.

$\mathcal{S}$ is true. Note $\mathcal{S}$ does not have ground truth positives or negatives. This task is cast as a learning to rank problem, and uses metrics such as mean rank (MR), mean reciprocal rank (MRR) or Hits@N.

Knowledge Graph Embeddings. Knowledge graph embedding models are neural architectures that encode concepts from a knowledge graph $\mathcal{G}$ (i.e. entities $\mathcal{E}$ and relation types $\mathcal{R}$) into low-dimensional, continuous vectors $\in \mathbb{R}^k$ (i.e, the embeddings). Embeddings are learned by training a neural architecture over $\mathcal{G}$. Although such architectures vary, the training phase always consists in minimizing a loss function $\mathcal{L}$ that includes a *scoring function* $f_m(t)$, i.e. a model-specific function that assigns a score to a triple $t = (s, p, o)$ (more precisely, the input of f_m are the embeddings of the subject $\mathbf{e}_s$, the predicate $\mathbf{r}_p$, and the object $\mathbf{e}_o$). The goal of the optimization procedure is learning optimal embeddings, such that the scoring function f_m assigns high scores to positive triples t^+ and low scores to triples unlikely to be true t^-. Existing models propose scoring functions that combine the embeddings $\mathbf{e}_s, \mathbf{r}_p, \mathbf{e}_o \in \mathbb{R}^k$ using different intuitions. Table 1b lists the scoring functions of the most common models. For example, the scoring function of TransE computes a similarity between the embedding of the subject $\mathbf{e}_s$ translated by the embedding of the predicate $\mathbf{e}_p$ and the embedding of the object $\mathbf{e}_o$, using the L_1 or L_2 norm $||\cdot||$. Such scoring function is then used on positive and negative triples $t^+ \in \mathcal{G}, t^- \in \mathcal{N}$ in the loss function. This is usually a pairwise margin-based loss (Bordes et al., 2013), negative log-likelihood, or multi-class log-likelihood (Lacroix et al., 2018). Since the training set usually includes positive statements, we generate synthetic negatives $t^- \in \mathcal{N}$ required for training. We do so by corrupting one side of the triple at a time (i.e. either the subject or the object), following the protocol proposed by (Bordes et al., 2013).

Calibration. Given a knowledge graph embedding model identified by its scoring function f_m, with $f_m(t) = \hat{p}$, where $\hat{p}$ is the estimated confidence level that a triple $t = (s, p, o)$ is true, we define f_m to be *calibrated* if $\hat{p}$ represents a true probability. For example, if $f_m(\cdot)$ predicts 100 triples all with confidence $\hat{p} = 0.7$, we expect exactly 70 to be actually true. Calibrating a model requires reliable metrics to detect miscalibration, and effective techniques to fix such distortion.

Reliability Diagram (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005). Also known as *calibration plot*, this diagram is a visual depiction of the calibration of a model (see Figure 1 for an example). It shows the expected sample accuracy as a function of the estimated confidence. A hypothetical perfectly calibrated model is represented by the diagonal line (i.e. the identity function). Divergence from such diagonal indicates calibration issues (Guo et al., 2017).

Brier Score (Brier, 1950). It is a popular metric used to measure how well a binary classifier is calibrated. It is defined as the mean squared error between n probability estimates $\hat{p}$ and the corresponding actual outcomes $y \in 0, 1$. The smaller the Brier score, the better calibrated is the model. Note that the Brier score $B \in [0, 1]$.

$$B = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{p}_i)^2 \tag{1}$$

Log Loss is another effective and popular metric to measure the reliability of the probabilities returned by a classifier. The logarithmic loss measures the relative uncertainty between the probability estimates produced by the model and the corresponding true labels.

$$L_{log} = -(y \cdot log(\hat{p}) + (1 - y) \cdot log(1 - \hat{p})) \tag{2}$$

Platt Scaling. Proposed by (Platt et al., 1999) for support vector machines, Platt scaling is a popular parametric calibration techniques for binary classifiers. The method consists in fitting a logistic regression model to the scores returned by a binary classifier, such that $\hat{q} = \sigma(a\hat{p} + b)$, where $\hat{p} \in \mathbb{R}$ is the uncalibrated score of the classifier, $a, b \in \mathbb{R}$ are trained scalar weights. and $\hat{q}$ is the calibrated probability returned as output. Such model can be trained be trained by optimizing the NLL loss with non-binary targets derived by the Bayes rule under an uninformative prior, resulting in an Maximum a Posteriori estimate.

Isotonic Regression (Zadrozny & Elkan, 2002). This popular non-parametric calibration techniques consists in fitting a non-decreasing piecewise constant function to the output of an uncalibrated classifier. As for Platt scaling, the goal is learning a function $\hat{q} = g(\hat{p})$, such that $\hat{q}$ is a calibrated probability. Isotonic regression learns g by minimizing the square loss $\sum_{i=1}^{n}(\hat{q}_i - y_i)^2$ under the constraint that g must be piecewise constant (Guo et al., 2017).

4 Calibrating Knowledge Graph Embedding Models Predictions

We propose two scenario-dependent calibration techniques: we first address the case with ground truth negatives $t^- \in \mathcal{N}$. The second deals with the absence of ground truth negatives.

Calibration with Ground Truth Negatives. We propose to use off-the-shelf Platt scaling and isotonic regression, techniques proved to be effective in literature. It is worth reiterating that to calibrate a model negative triples $\mathcal{N}$ are required from a held-out dataset (which could be the validation set). Such negatives are usually available in triple classification datasets (FB13, WN11, YAGO39K) [1].

Calibration with Synthetic Negatives. Our main contribution is for the case where no ground truth negatives are provided at all, which is in fact the usual scenario for *link prediction* tasks.

We propose to adopt Platt scaling or isotonic regression and to synthetically generate corrupted triples as negatives, while using sample weights to guarantee that the frequencies adhere to the base rate of the population (which is problem-dependent and must be user-specified). It is worth noting that it is not possible to calibrate a model without implicit or explicit base rate. If it is not implicit on the dataset (the ratio of positives to totals), it must be explicitly provided.

We generate synthetic negatives $\mathcal{N}$ following the standard protocol proposed by (Bordes et al., 2013): for every positive triple $t = (s, p, o)$, we corrupt one side of the triple at a time (i.e. either the subject s or the object o) by replacing it with other entities in $\mathcal{E}$. The number of corruptions generated per positive is defined by the user-defined corruption rate $\eta \in \mathbb{N}$. Since the number of negatives $N = |\mathcal{N}|$ can be much greater than the number of positive triples $P = |\mathcal{G}|$, when dealing with calibration with synthetically generated corruptions, we weigh the positive and negative triples to make the calibrated model match the population base rate $\alpha = P/(P + N) \in [0, 1]$, otherwise the base rate would depend on the arbitrary choice of η.

Given a positive base rate α, we propose the following weighting scheme:

$$\begin{aligned} \omega_+ &= \eta \quad \text{for positive triples } \mathcal{G} \\ \omega_- &= \frac{1}{\alpha} - 1 \quad \text{for negative triples } \mathcal{N} \end{aligned} \tag{3}$$

where $\omega_+ \in \mathbb{R}$ is the weight associated to the positive triples and $\omega_- \in \mathbb{R}$ to the negatives. The ω_+ weight removes the imbalance determined by having a higher number of corruptions than positive triples in each batch. The ω_- weight guarantees that the given positive base rate α is respected.

The above can be verified as follows. For the unweighted problem, the positive base rate is simply the ratio of positive examples to the total number of examples:

$$\alpha = \frac{P}{P + N} \tag{4}$$

If we add uniform weights to each class, we have:

$$\alpha = \frac{\omega_+ P}{\omega_- N + \omega_+ P} \tag{5}$$

[1] Negatives required to train the model are either provided or synthetically generated as described by (Bordes et al., 2013)

By defining $\omega_+ = \eta$, i.e. adopting the ratio of negatives to positives (corruption rate), we then have:

$$\alpha = \frac{P\frac{N}{P}}{N\omega_- + P\frac{N}{P}} = \frac{N}{\omega_- N + PN} = \frac{1}{\omega_- + 1} \tag{6}$$

Thus, the negative weights is:

$$\omega_- = \frac{1}{\alpha} - 1 \tag{7}$$

5 Results

We compute the calibration quality of our heuristics, showing that we achieve calibrated predictions even when ground truth negative triples are not available. We then show the impact of calibrated predictions on the task of triple classification.

Datasets. We run experiments on triple classification datasets that include ground truth negatives (Table 1). We train on the training set, calibrate on the validation set, and evaluate on the test set.

- **WN11** (Socher et al., 2013). A subset of Wordnet (Miller, 1995), it includes a large number of hyponym and hypernym relations thus including hierarchical structures.
- **FB13** (Socher et al., 2013). A subset of Freebase (Bollacker et al., 2008), it includes facts on famous people (place of birth and/or death, profession, nationality, etc).
- **YAGO39K** (Lv et al., 2018). This recently released dataset has been carved out of YAGO3 (Mahdisoltani et al., 2013), and includes a mixture of facts about famous people, events, places, and sports teams.

We also use two standard link prediction benchmark datasets, **WN18RR** (Dettmers et al., 2018) (a subset of Wordnet) and **FB15K-237** (Toutanova et al., 2015) (a subset of Freebase). Their test sets do not include ground truth negatives.

Implementation Details. The knowledge graph embedding models are implemented with the AmpliGraph library (Costabello et al., 2019) version 1.1, using TensorFlow 1.13 (Abadi et al., 2016) and Python 3.6 on the backend[2]. All experiments were run under Ubuntu 16.04 on an Intel Xeon Gold 6142, 64 GB, equipped with a Tesla V100 16GB.

Hyperparameter Tuning. For each dataset in Table 1a, we train a TransE, DistMult, and a ComplEx knowledge graph embedding model. We rely on typical hyperparameter values: we train the embeddings with dimensionality $k = 100$, Adam optimizer, initial learning rate $\alpha_0 =$ 1e-4, negatives per positive ratio $\eta = 20$, $epochs = 1000$. We train all models on four different loss functions: Self-adversarial (Sun et al., 2019), pairwise (Bordes et al., 2013), NLL, and Multiclass-NLL (Lacroix et al., 2018). Different losses are used in different experiments.

5.1 Calibration Results

Calibration Success. Table 2 reports Brier scores and log losses for all our calibration methods, grouped by the type of negative triples they deal with (ground truth or synthetic). All calibration methods show better-calibrated results than the uncalibrated case, by a considerable margin and for all datasets. In particular, to put the results of the synthetic strategy in perspective, if we suppose to predict the positive base rate as a baseline, for each of the cases in Table 2 (the three datasets share the same positive base rate $\alpha = 0.5$), we would get Brier score $B = 0.25$ and log loss $L_{log} = 0.69$, results that are always worse than our methods. There is considerable variance of results between models given a dataset, which also happens when varying losses given a particular combination of model and dataset (Table 3). TransE provides the best results for WN11 and FB13, while DistMult works best for YAGO39K. We later propose that this variance comes from the quality of the embeddings themselves, that is, better embeddings allow for better calibration.

In Figure 2, we also evaluate just the frequencies themselves, ignoring sharpness (i.e. whether probabilities are close to 0 or 1), using reliability diagrams for a single model-loss combination, for all

[2] We will open source our probability calibration in the camera-ready, should the paper be accepted.

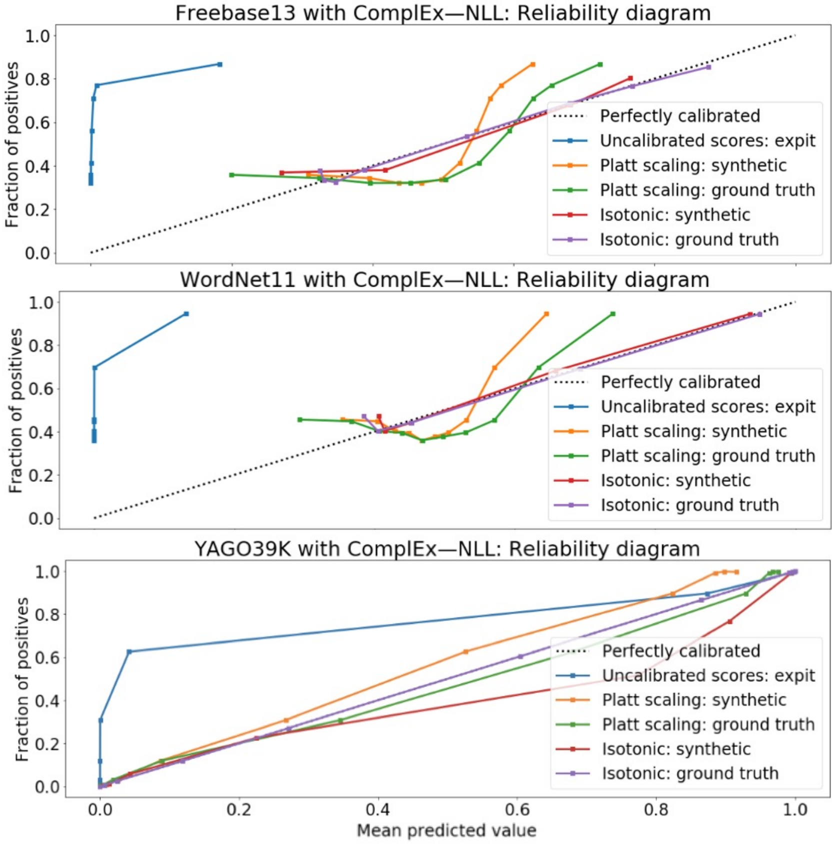

Figure 2: Calibration plots for the best calibrated model-loss combinations. Isotonic regression delivers the best results, getting very close to the perfectly calibrated line, both when used with the ground truth method or our proposed synthetic method. Best viewed in colors.

datasets (ComplEx+NLL). Calibration plots show a remarkable difference between the uncalibrated baseline (s-shaped blue line on the left-hand side) and all calibrated models (curves closer to the identity function are better). A visual comparison of uncalibrated curves in Figure 1 with those in Figure 2 also gives a sense of the effectiveness of calibration.

Ground Truth vs Synthetic. As expected, the ground truth method generally performs better than the synthetic calibration, since it has more data in both quantity (twice as much) and quality (two classes instead of one). Even so, the synthetic method is much closer to the ground truth than to the uncalibrated scores, as highlighted by the calibration plots in Figure 2. For WN11, it is actually as good as the calibration with the ground truth. This shows that our proposed method works as intended and could be used in situations where we do not have access to the ground truth, as is the case for most knowledge graph datasets.

Isotonic vs Platt. Isotonic regression performs better than Platt scaling in general, but in practice Isotonic regression has the disadvantage of not being a convex or differentiable algorithm Zadrozny & Elkan (2002). This is particularly problematic for the synthetic calibration, as it requires the generation of the synthetic corruptions, which can only be made to scale via a mini-batch based optimization procedure. Platt scaling, given that it is a convex and differentiable loss, can be made part of a computational graph and optimized with mini-batches, thus it can rely on the modern computational infrastructure designed to train deep neural networks.

Influence of Loss Function. We experiment with different losses, to assess how calibration affects each of them (Table 3). We choose to work with WN11 with TransE which is reported as a strong baseline in (Hamaguchi et al., 2017). Experiments show the choice of the loss has a big impact, greater than the choice of calibration method or embedding model. We believe such variability is

		Brier Score					**Log Loss**				
			Ground Truth		Synthetic			Ground Truth		Synthetic	
		Uncalib	Platt	Iso	Platt	Iso	Uncalib	Platt	Iso	Platt	Iso
WN11	TransE	.443	.089	**.087**	.092	.088	1.959	.302	**.295**	.311	.296
	DistMult	.488	.213	.208	.214	.208	5.625	.618	.604	.618	.601
	ComplEx	.490	.240	.227	.240	.228	6.061	.674	.651	.674	.650
FB13	TransE	.446	**.124**	**.124**	.148	.141	1.534	**.390**	.391	.459	.442
	DistMult	.473	.178	.170	.185	.192	2.177	.533	.518	.549	.567
	ComplEx	.481	.177	.170	.182	.189	2.393	.534	.516	.544	.565
YAGO 39K	TransE	.363	.095	.093	.106	.110	1.062	.319	.309	.370	.376
	DistMult	.284	.081	**.079**	.093	.089	1.043	.279	**.266**	.311	.308
	ComplEx	.264	.089	.084	.097	.095	1.199	.305	.278	.323	.313

Table 2: Calibration test results on different datasets. The lower score, the better. Best results in bold for each combination of dataset and metric. We present the results using the self-adversarial loss (Sun et al., 2019) on all datasets.

	Brier Score				**Log Loss**				**MRR**
	Ground Truth		Synthetic		Ground Truth		Synthetic		**(filtered)**
$\mathcal{L}$	Platt	Iso	Platt	Iso	Platt	Iso	Platt	Iso	
Pairwise	.202	.198	.209	.200	.591	.585	.606	.589	.06
NLL	.093	.088	.094	.088	.342	.299	.344	.301	.13
Multiclass-NLL	.204	.189	.204	.189	.599	.550	.599	.551	.11
Self-adversarial	.089	**.087**	.092	.088	.302	**.295**	.311	.296	**.16**

Table 3: Calibration test results using different losses $\mathcal{L}$ on WN11. Model used was TransE, as it provides a strong baseline on WN11 (Hamaguchi et al., 2017). For the calibration losses, the lower score, the better. The MRR is a ranking metric typically used to evaluate knowledge graph embeddings. We apply MRR only to the positive test triples. For MRR, the higher the better. The MRR results show a clear correlation between embedding quality and calibration performance. Best results in bold for each metric.

determined by the quality of the embeddings: the better the embeddings, the sharper the calibration. To support such claim, we report the mean reciprocal rank (MRR), which, for each true test triple, computes the (inverse) rank of the triple against synthetic corruptions, then averages the inverse rank (Table 3). We notice a clear correlation between the calibration results and the MRR. For all calibration methods, the best MRR (self-adversarial) leads to the best calibration results, while for all calibration methods but two exceptions, the worst MRR (obtained with pairwise loss) leads to the worst calibration results.

Positive Base Rate. We apply our synthetic calibration method to two link prediction benchmark datasets, FB15K-237 and WN18RR. As they only provide positive examples, we apply our method with varying base rates α_i, linearly spaced from 0.05 to 0.95. We evaluate results relying on the closed-world assumption, i.e. triples not present in training, validation or test sets are considered negative. For each α_i we calibrate the model using the synthetic method with both isotonic regression and Platt scaling. We sample negatives from the negative set under the implied negative rate, and calculate a baseline which is simply having all probability predictions equal to α_i. Figure 3 shows that isotonic regression and Platt scaling perform similarly and always considerably below the baseline. As expected from the previous results, the uncalibrated scores perform poorly, only reaching acceptable levels around some particular base rates.

Triple Classification and Decision Threshold. To overcome the need to learn $|\mathcal{R}|$ decision thresholds τ_i from the validation set, we propose to rely on calibrated probabilities, and use the natural threshold of $\tau = 0.5$. Table 4 shows how calibration affects the triple classification task, comparing with the literature standard of per-relation thresholds (last column). For simplicity, note we use the same self-adversarial loss in Table 2 and Table 4. We learn thresholds τ_i on validation sets, resulting in 11, 7, and 33 thresholds for WN11, FB13 and YAGO39K respectively.

Using a single $\tau = 0.5$ and calibration provides competitive results compared to multiple learned thresholds (note uncalibrated results with $\tau = 0.5$ are poor, as expected). It is worth mentioning that

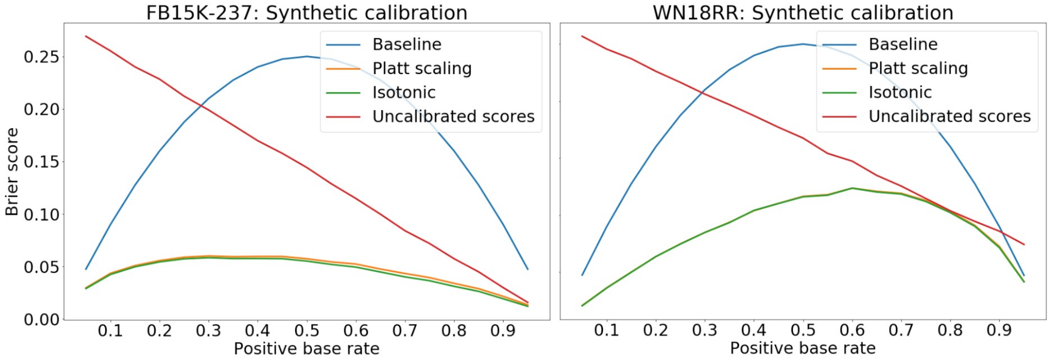

Figure 3: Synthetic calibration on FB15K-237 and WN18RR, with varying positive base rates. The baseline stands for using the positive base rate as the probability prediction. The results are evaluated under a closed-world assumption using the same positive base rate as was used to calibrate the models. Best viewed in colors.

		Ground Truth		Synthetic		Uncalib.	Uncalib. (Per-Relation τ)	
		Platt	Iso	Platt	Iso	($\tau = 0.5$)	Reproduced	Literature
	TransE	88.8	**88.9**	88.8	**88.9**	50.7	88.2	
WN11	DistMult	66.5	67.2	66.1	67.1	50.8	67.2	**88.9***
	ComplEx	60.6	62.4	59.8	62.4	50.8	59.6	
	TransE	82.4	82.3	79.3	80.3	50.0	82.0	
FB13	DistMult	72.5	73.2	72.3	70.2	50.1	80.8	**89.1***
	ComplEx	73.8	74.2	74.2	72.4	50.1	83.6	
	TransE	87.1	87.8	86.6	84.9	50.2	88.8	
YAGO39K	DistMult	88.9	89.3	87.7	88.5	56.7	90.2	**93.8**†
	ComplEx	87.3	88.2	86.6	87.2	61.1	89.4	

Table 4: Effect of calibration on triple classification accuracy. For all calibration methods there is one single threshold, $\tau = 0.5$. For the per-relation τ, we have 11, 7, and 33 thresholds for WN11, FB13 and YAGO39K respectively, all learned from a validation set. Best results in bold for each dataset. Note that isotonic regression reaches state-of-the-art results for WN11. Results of $*$ from (Zhang et al., 2018); $\star$ from (Ji et al., 2016); $\dagger$ from (Lv et al., 2018).

we are at par with state-of-the-art results for WN11. Isotonic regression is again the best method, but there is more variance in the model choice. Our proposed calibration method with synthetic negatives performs well overall, even though calibration is performed only using half of the validation set (negatives examples are replaced by synthetic negatives).

6 Conclusion

We propose a method to calibrate knowledge graph embedding models. Our contribution addresses datasets with and without ground truth negatives. We Experiments on triple classification datasets and apply Platt scaling and isotonic regression with and without synthetic negatives controlled by our heuristics. All calibration methods perform significantly better than uncalibrated scores. We show that isotonic regression is the best choice in terms of calibration performance, with the trade-off of being more computationally expensive. Additional experiments on triple classification shows that calibration allows to use a single decision threshold, reaching state-of-the-art results without the need to learn per-relation thresholds.

Future work will evaluate additional calibration algorithms, such as beta calibration (Kull et al., 2017) or Bayesian binning (Naeini et al., 2015). We will also experiment on ensembling of knowledge graph embedding models, inspired by(Krompaß & Tresp, 2015). The rationale is that different models operate on different scales, but calibrating brings them all to the same probability scale, so their output can be easily combined.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250. AcM, 2008.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pp. 2787–2795, 2013.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques and applications. *arXiv preprint arXiv:1709.07604*, 2017.

Luca Costabello, Sumit Pai, Chan Le Van, Rory McGrath, Nicholas McCarthy, and Pedro Tabacof. AmpliGraph: a Library for Representation Learning on Knowledge Graphs, March 2019. URL `https://doi.org/10.5281/zenodo.2595043`.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Procs of AAAI*, 2018. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366`.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 601–610. ACM, 2014.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017.

Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge transfer for out-of-knowledge-base entities: a graph neural network approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1802–1808. AAAI Press, 2017.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Denis Krompaß and Volker Tresp. Ensemble solutions for link-prediction in knowledge graphs. In *2nd Workshop*, 2015.

Meelis Kull, Telmo M Silva Filho, Peter Flach, et al. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052–5080, 2017.

Timothee Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pp. 2869–2878, 2018.

Xin Lv, Lei Hou, Juanzi Li, and Zhiyuan Liu. Differentiating concepts and instances for knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1971–1979, 2018.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. Yago3: A knowledge base from multilingual wikipedias. 2013.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Procs of the IEEE*, 104(1):11–33, 2016.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 625–632. ACM, 2005.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, 2015.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *procs of ICML*, pp. 2071–2080, 2016.

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Procs of ICLR*, 2015.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699. ACM, 2002.

Zhao Zhang, Fuzhen Zhuang, Meng Qu, Fen Lin, and Qing He. Knowledge graph embedding with hierarchical relation structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3198–3207, 2018.