7-2021

Calibrating Knowledge Graphs

Aishwarya Rao
ar2711@rit.edu

Calibrating Knowledge Graphs

by

Aishwarya Rao

A thesis submitted in partial fulfillment of the
requirements for the degree of
Master of Science
in Computing and Information Sciences

B. Thomas Golisano College of Computing and
Information Sciences
Rochester Institute of Technology

July 2021

MS IN COMPUTING AND INFORMATION SCIENCES
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

MS DEGREE THESIS

The MS degree thesis of Aishwarya Rao
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
MS degree in Computing and Information Sciences

Dr. Carlos R. Rivero, Thesis Advisor

Dr. Xumin Liu, Reader

Dr. Peizhao Hu, Observer

Date

Calibrating Knowledge Graphs

by

Aishwarya Rao

Submitted to the
B. Thomas Golisano College of Computing and Information Sciences
Department of Computer Science
in partial fulfillment of the requirements for the
Master of Science Degree
at the Rochester Institute of Technology

Abstract

A knowledge graph model represents a given knowledge graph as a number of vectors. These models are evaluated for several tasks, and one of them is link prediction, which consists of predicting whether new edges are plausible when the model is provided with a partial edge. Calibration is a postprocessing technique that aims to align the predictions of a model with respect to a ground truth. The idea is to make a model more reliable by reducing its confidence for incorrect predictions (overconfidence), and increasing the confidence for correct predictions that are closer to the negative threshold (underconfidence). Calibration for knowledge graph models have been previously studied for the task of triple classification, which is different than link prediction, and assuming closed-world, that is, knowledge that is missing from the graph at hand is incorrect. However, knowledge graphs operate under the open-world assumption such that it is unknown whether missing knowledge is correct or incorrect. In this thesis, we propose open-world calibration of knowledge graph models for link prediction. We rely on strategies to synthetically generate negatives that are expected to have different levels of semantic plausibility. Calibration thus consists of aligning the predictions of the model with these different semantic levels. Nonsensical negatives should be farther away from a positive than semantically plausible negatives. We analyze several scenarios in which calibration based on the sigmoid function can lead to incorrect results when considering distance-based models. We also propose the Jensen-Shannon distance to measure the divergence of the predictions before and after calibration. Our experiments exploit several pre-trained models of nine algorithms over seven datasets. Our results show that many of these pre-trained models are properly calibrated without intervention under the closed-world assumption, but it is not the case for the open-world assumption. Furthermore, Brier scores (the mean squared error before and after calibration) using the closed-world assumption are generally lower and the divergence is higher when using open-world calibration. From these results, we gather that open-world calibration is a harder task than closed-world calibration. Finally, analyzing different measurements related to link prediction accuracy, we propose a combined loss function for calibration that maintains the accuracy of the model.

Acknowledgments

The satisfaction that accompanies the successful completion of any task would be incomplete without acknowledging the people that helped bring it to fruition. I would first like to thank my advisor Dr. Carlos Rivero for giving me the opportunity to explore and venture into the field of knowledge graph research. It is only through his guidance, aid, and support that my thesis is what it is today. I would also like to express my gratitude to Dr. Xumin Liu and Dr Peizhao Hu for taking the time to serve in my committee and provide constructive feedback. I am grateful to the faculty and staff of the Computer Science department who have directly or indirectly helped me by guiding me towards developing the necessary skills I needed and providing indispensable resources and support.

Finally, I would like to thank my family for their unyielding support in my pursuit of a Master's degree.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Knowledge graphs represent data as entities (vertices) and the relationships (edges) between them [17]. Knowledge graphs usually store the data in the form of (subject, predicate, object) triples, where the subject and object are entities, and the predicate is the label of the relationship between them. One of the key assumptions in knowledge graphs is that triples not present in the graph can be either false (true negatives), or just missing relationships (false negatives). This is known as the open-world assumption, and is the main motivation behind knowledge graph models [32].

A knowledge graph model is a machine learning model that contains at least an embedding (vector) to encode every entity and predicate in the knowledge graph at hand [16]. These embeddings are learned to have high accuracy for a particular task. Having an input triple, the model outputs a score based on the learned embeddings. We focus on link prediction that consists of predicting new triples [5]. Link prediction is evaluated by taking each positive triple in the test split, and corrupting its subject with all possible entities such that the resulting triples are not present in the graph under evaluation, i.e., they are assumed to be negatives. The position (rank) of the positive triple is used to measure accuracy. Similarly, the object of the positive triple is replaced by other entities to generate negatives. A well-trained model is expected to rank positive triples higher than their negative counterparts. However, training these models is a challenging task because of the many factors involved, for instance, generating negatives under the open-world assumption is challenging, graphs are typically large and models need to be learned based on stochastic gradient descent, and the training step usually has a number of hyperparameters to be tuned [36].

Calibration is a post-processing machine learning technique that is used to align the predictions of a specific model closer to the actual distribution of the data, i.e., the ground truth [30]. Calibration is appealing because it transforms the scores of the model at hand closer to the expected values in the ground truth, which in turn makes the model more reliable [29]. A well-calibrated model predicts more extreme scores (high confidence) for the correct answers, mitigating the problem of overconfident and underconfident models. On one hand, overconfident models output scores that are very close to zero for positives and vice versa for negatives. On the other hand, underconfident models output scores that are very close to the threshold between positives and negatives when correctly predicting an input.

Recently, there have been some efforts to calibrate knowledge graph models [31, 35]. Calibrating knowledge graph models aims to increase the reliability of the models, but it comes with some challenges [35]. First, knowledge graphs do not generally have negative triples available, and it is thus required to generate negative triples synthetically [2]. Second, positive and negative triples are typically unbalanced since negatives tend to outnumber positives by several orders of magnitude [36]. Third, the expected semantic plausibility of negative triples depend on the strategy used to synthetically generate them, which implies that there are different levels of negatives [2]. Fourth, the open-world assumption makes it difficult to obtain reliable ground truth since certain negative triples may be missing from the knowledge graph at hand rather than being false.

Tabacof and Costabello [35] showed that knowledge graph models are typically uncalibrated and, then, exploited the local-world closed assumption strategy to generate negative triples. However, they focused on triple classification (deciding whether a triple is positive or negative) rather than link prediction. Furthermore, they used a rate to randomly select a subset of negatives per positive triple. However, since it is generally the case that negatives significantly outnumber positives, such a rate must be high to avoid biases during training, for instance, selecting only nonsensical negatives. Safavi et al. [31] focused on predicate rather than link prediction, i.e., predicting a predicate when both subject and object are fixed. Since the number of predicates is typically much smaller than entities, predicate prediction is expected to be an easier task than link prediction [31]. The authors explored closed-world calibration in which missing triples are considered negatives, and open-world calibration based on the open-world assumption. Their conclusion is that calibration becomes significantly harder when shifting to the open-world assumption. Furthermore, to evaluate open-world calibration, they relied on manual labeling of negatives by the crowd rather than synthetic generation strategies. Unfortunately, due to polysemy, information disparity and other issues [4], it is unclear how accurate such manual labeling is.

1.1 Proposal

In this thesis, we propose open-world calibration for knowledge graph models. To avoid manual labeling of triples, we rely on a number of strategies to synthetically generate negative counterparts of positive triples [2]. These strategies are expected to generate negatives with different levels of semantic plausibility, e.g., nonsensical or semantically plausible. These strategies exploit the underlying structure of the knowledge graph at hand. We take advantage of that to define open-world calibration as a regression problem in which the triple scores output by a model under evaluation must be adjusted to the different semantic levels between 0 and 1. Note that closed-world calibration is a binary classification: either positives or negatives. As a result, negatives that are expected to be semantically plausible should be closer to positives than negatives that are expected to be nonsensical, which should be considered "pure" negatives. The semantic possibilities are generated based on how the negatives are generated, which in turn depends on the structure of the graph itself. For triple classification, we also maintain the 0.5 threshold between the positives and negatives and the assigned levels are all below 0.5.

We discuss several observations in the context of calibrating knowledge graph models. We illustrate practical issues when applying the sigmoid function, which is used by calibration, to model scores computed based on distances and similarities. The sigmoid function assumes that input values lie in the range of $(-\infty, \infty)$; however, distance-based models like TransE [5] produce scores in the $(0, \infty)$ range. This can be detrimental for calibration purposes as all scores higher than one are approximated to one by the sigmoid function. Depending on the precision of the system, this could lead to loss of information. We propose min-max scaling of such scores to deal with these situations.

Furthermore, we propose to use the Jensen–Shannon distance to analyze the divergence between uncalibrated and calibrated scores. We focus on calibration for link prediction in knowledge graphs and study the accuracy measured based on ranks of calibrated models, which, to the best of our knowledge, have not been studied yet.

We study closed-world vs. open-world assumption in our experiments. We use several pre-trained models over the seven de facto standard datasets used to evaluate link prediction: FB13, FB15K, FB15K-237, NELL-995, WN11, WN18 and WN18RR. The pre-trained models correspond to nine algorithms: Analogy [23], ComplEx [38], DistMult [41], HolE [28], RotatE [34], SimplE [20], TransD [18], TransE [5], and TransH [40]. Our results show that many of the original models are properly calibrated without intervention assuming closed-world, but they are not properly calibrated assuming open-world. Finally, we introduce a novel combined loss function for calibration while maintaining the accuracy of the original model. With this, we find that both closed-world and open-world calibration result in better calibrated models without hurting the accuracy. We observe that calibration models under the open-world assumption tend to have slightly higher divergence from the original scores than under the closed-world assumption.

The rest of the paper is organized as follows: Chapter 2 presents background information and related work; Chapter 3 discusses several observations in the context of knowledge graph model calibration; Chapter 4 presents our approach for open-world calibration; Chapter 5 discusses the experiments and results for the same and finally Chapter 6 summarizes the work and conclusion.

Chapter 2

Background

This chapter presents the link prediction task for knowledge graph models in Section 2.1, and calibration as a machine learning technique in Section 2.2. Once the foundations of both knowledge graph models and model calibration are set, Section 2.3 describes the work done so far for calibrating knowledge graph models.

2.1 Link Prediction for knowledge graphs

A knowledge graph stores data in the form of entities and relationships between them [17]. Entities are vertices in the graph and relationships are directed, labeled edges between vertices. These entities and relationships can be represented as (subject, predicate, object) triples, such that subject and object are entities, and predicate is the label of the relationship that starts in subject and ends in object (directed). An example of a knowledge graph that stores data related to movies is as follows: it contains entities representing actors, movies and countries. Relationships encode whether an actor acted in a movie, and a movie has a country of origin. For instance, "Carrie Fisher acted in Star Wars" is represented as (CarrieFisher, ActedIn, StarWars); "Star Wars has country of origin USA" is represented as (StarWars, Country, USA). There are standard knowledge graphs available and are commonly used in research. Freebase is a collaborative knowledge base from different sources and domains. We consider two versions extracted from Freebase - FB13 [33] and FB15k [6] based on the kind of relations and anomalies. WordNet contains English words and

Graph	Versions	Entities	Relations
Freebase	FB13 [33]	75,043	13
	FB15k237 [6]	14,541	237
Wordnet	WN18 [6]	40,943	18
	WN11 [11]	40,943	11
Never Ending Language Learning	NELL-995 [25]	75,492	200
Yet Another Great Ontology	YAGO3-10 [24]	123,182	37

Table 2.1: Summary of common knowledge graphs

their relations to each other. Similar to Freebase, there are multiple graphs derived from Wordnet [11] [6]. NELL [25] is extracted from unstructured web pages and YAGO [24] is built from WordNet, Wikipedia and GeoNames. The number of entities and relations in these graphs are described in Table 2.1

A knowledge graph model consists of a vector embedding for each entity and predicate available in the graph at hand [16]. These embeddings are learned using machine learning algorithms to solve a particular task, such as link prediction or triple classification [5]. In the case of link prediction, these models are generally trained to minimize a loss function such that the model predicts a low score for positive triples (triples that belong to the graph), and a high score for negative triples. The score of an input triple is computed based on the embeddings of its subject, predicate and object. Current approaches use different dissimilarity or distance measures as scoring functions [5, 18, 20, 22, 23, 28, 34, 40, 41].

TransE [5] is the first translation based knowledge graph model algorithm. Every entity and relationship is randomly assigned a vector embedding. TransE uses the max-margin loss function shown in equation 2.1 to increase the margin between the scores of positive and negative triples. In equation 2.1, $score_p$ and $score_n$ refer to the scores of positive and negative triples respectively and the margin is the target margin between the positives and negatives.

$$Loss(score_p, score_n) = max(0, score_p - score_n + margin) \tag{2.1}$$

The score of a triple is the L1 or L2 norm of the sum of the head and relationship's embedding and the tail's embedding [5] as shown in equation 2.2.

$$d = ||h + r - t|| \tag{2.2}$$

Other translation models [40] [18] [22] followed TransE and vary in their choice of distance function and embedding space.
Nickel et al. [27] and Cai et al. [9] present a more detailed summary of knowledge graph models.

2.1.1 Evaluation of knowledge graph models

Knowledge graph models are evaluated for link prediction using ranking-based metrics [7]. Since the goal of the model is to predict low scores for positive triples, it follows that the metric should measure the ability of the model to do so. The idea is to register the position (rank) of a positive triple with respect to its negative counterparts. The most common evaluation metric is the Mean Reciprocal Rank (MRR), which is the average of the inverse of the ranks of positive triples. However, MRR has been recently criticized, and the Mean Rank (MR), the mean of the ranks of positive triples, is recommended to assess link prediction [3,14,36]. MR can be adjusted for chance as follows [3]: $\overline{MR} = 1 - MR/\mathbb{E}[MR]$, where $\mathbb{E}[MR]$ is the expected MR of a random model assuming that all individual ranks are independent. $\overline{MR} \in [-1, 1]$ such that $\overline{MR} > 0$ entails better accuracy than a random model, $\overline{MR} = 0$ means that the model is indistinguishable from a random model, and $\overline{MR} < 0$ entails worse accuracy than a random model.

Based on the ranking metrics, the quality of a knowledge graph models can also be evaluated using the number of ties between scores of positive and negative triples. Since the models are trying to separate these triples, the lower this value is, the better the model.

2.1.2 Negative Generation Strategies

Knowledge graphs only contain positive triples. Furthermore, they usually operate under the open-world assumption, i.e., triples that are not present in the graph at hand may be either missing or negatives [13]. As a result, it is a challenge to generate the negative counterparts of a positive triple [2]. Typically, these negative counterparts are synthetically generated by corrupting a positive triple, i.e., changing its subject or object such that the new corrupted triple is not present in the graph [5]. There are multiple corruption strategies that are used to generate these negative triples [2]. These corruption strategies vary based on the set from which each corrupted entity is selected from. Assuming that we wish to corrupt a positive triple (s, p, o) as either (s', p, o) and (s, p, o') such that they are not present in the graph (otherwise, they are positives), corruption strategies are as follows:

- Global-Naïve [2]: s' is selected as entities that are never subjects in the entire graph. Similarly, o' entities are those that are never objects in the graph. We denote as N_G the set of negative triples generated by this strategy.
- Local-Closed World Assumption (LCWA) [13]: s' and o' are all the entities available in the graph at hand except those that appear as subjects of $(\_, p, o)$ triples, or objects of $(s, p, \_)$ triples. We denote this set of negatives as N_L.
- Type-constrained LCWA (TLCWA) [2]: s' entities are subjects of $(\_, p, \_)$ triples, except those that appear as subjects of $(\_, p, o)$ triples. Similarly, o' entities are objects of $(\_, p, \_)$ triples, except those that appear as objects of $(s, p, \_)$ triples. We denote these negatives as N_T.

N_L consists of all entities in the graph as long as the resulting triple is not positive. N_G and N_T are sets containing only certain entities depending on whether they are in the subject or object positions in the graph. Therefore, both N_G and N_T are subsets of N_L: $N_G \subseteq N_L$, $N_T \subseteq N_L$. The definitions of N_G and N_T are such that the former has entities that are never in subject/object positions, and the latter has entities only from subject/object positions. Therefore, it stands that $N_G \cap N_T = \emptyset$.

2.2 Calibration in Machine Learning

A machine learning model usually predicts a class label with a certain probability; these probabilities become the confidence values of the model at hand for such labels. While an ideal scenario is not always achievable, calibration allows shifting the predicted probabilities of a model to reflect the distribution of a ground truth [30]. For instance, if a ground truth contains 50% positives, a perfectly calibrated model has a .5 probability of predicting a positive. Calibration is important since it helps users understand how reliable a prediction made by the model is. Furthermore, a model that predicts incorrect answers with a high probability is also not reliable. For example, in a binary classification, a model that predicts incorrect answers with 85% probability or above is not desirable. A more reliable model would predict incorrect answers with lower probabilities. Calibration can also help in these cases [29].

Calibration is thus a post-processing technique that learns a transformation function that maps the probabilities predicted by a model to the expected probabilities of a ground truth. The transformation function acts as an additional layer atop the machine learning model that takes the scores (a.k.a. logits) of the model, and converts them into a probability between zero and one. This transformation function is learned over the validation split by reducing the loss between the model's and the expected probabilities of a ground truth. The original model before calibration is thus left untouched during this process.

Isotonic regression [29] is a non-parametric calibration method that uses a monotonically increasing function to transform the output scores z_i of a model into the probabilities that match a certain validation split p_i, which are considered the ground truth:

$$p_i = m(z_i) + B \tag{2.3}$$

where m is a monotonic function. Isotonic regression typically requires large amounts of data, and is also prone to suffer from overfitting [19]. Additionally, isotonic regression is a non-differentiable function, which makes it incompatible with modern infrastructure generally used for training machine learning models [35].

Platt scaling [30] is another calibration method that uses a logistic regression parametric function. The equation this method solves is as follows [15]:

$$p_i = \sigma(Az_i + B) \tag{2.4}$$

where A and B are calculated by optimizing a loss function over the validation split using the model scores.

Positive and negative triples are represented by their target labels 1 and 0 respectively. To prevent overfitting further, instead of using these labels, we use class sizes translated to the following probabilities:

$$p_+ = \frac{P+1}{P+2} \qquad p_- = \frac{1}{N+2} \tag{2.5}$$

where P is the number of positives, N is the number of negatives, p_+ refers to the probability of positives, and p_- refers to the probability of negatives [30]. As a result, p_+ is expected to have a high value closer to one (but not exactly one), and p_- is expected to have a low value closer to zero (but not exactly zero).

Platt scaling [30] using the Negative Log-Likelihood function as the loss to train. For a two-class problem, the loss function reduces to Binary Cross Entropy (BCE) [10] which is shown in the equation below:

$$BCE(p_i, f_i) = -w_i \times [p_i * log(\sigma(f_i)) + (1 - p_i) * log(1 - \sigma(f_i))] \tag{2.6}$$

In the above equation, p_i is the expected ground truth and f_i the prediction. It works by moving the negative scores closer to zero and the positives closer to one. We can also leverage weights with the BCE Loss as demonstrated by w_i.

The calibration of a given model can be evaluated in multiple ways. A reliability diagram is a plot of the frequency of positive samples against the predicted probabilities [15, 29]. For a perfectly calibrated model, this plot is the identity function as it predicts low probabilities for positives and high probabilities for negatives. Expected calibration error is the weighted mean difference between the accuracy and predicted probabilities per bin after dividing the predictions into equally spaced bins [26]. While reliability diagrams provide a good visual understanding of the calibration error, the expected calibration error provides a numerical measure of it.

The Brier score is another numerical measure equal to the mean squared difference between the predicted probabilities and the expected probabilities of the ground truth [8]. The score is computed as follows:

$$\frac{1}{n}\sum_{i=1}^{n}(p_i - f_i)^2$$

In the above equation, n is the number of samples, p_i is the expected probability of the ground truth, and f_i is the predicted probability by the model. Note that $f_i = \sigma(Az_i + B)$, where z_i is the score (prediction) of the original model. When computing the Brier score of the original model under evaluation, we assume that $A = 1$ and $B = 0$ [35].

2.3 Knowledge graph model calibration

Calibration can be used on knowledge graph models as well to improve their reliability. Calibration in such models works by taking the scores predicted by the model and transforming them into probability values. Note that different knowledge graph models have different score ranges [16]. Using closed-world calibration, one can assume a .5 default threshold between positives and negatives, regardless of the range of the scores for the model at hand. Without calibration, the threshold between positive and negative triples must be manually determined depending on the algorithm and model scores.

Using reliability diagrams, Tabacof and Costabello [35] showed that off-the-shelf knowledge graph models are generally not properly calibrated for the task of triple classification. Different than link prediction, triple classification is a binary classification task that places every triple into one of two classes - positives that belong to the graph, or negatives that do not. The models output probabilities of a triple being positive, and the threshold between the two classes is .5. Uncalibrated models predict lower probabilities than expected for negative triples, and higher probabilities for positive triples. Tabacof and Costabello [35] rely on Platt scaling and isotonic regression to calibrate knowledge graph models for triple classification. However, knowledge graphs do not usually contain negatives, only positives, and calibration requires samples from both classes (positives and negatives) to learn. One of the main contributions of Tabacof and Costabello [35] is their use of synthetic negative triples for the calibration process. Synthetic triples are generated by using the LCWA corruption strategy. The corruption rate decides the number of negatives to be generated per positive triple. The positive triples and their corresponding negatives are weighted according to two user defined variables:

1. corruption rate η which determines the number of negatives N to generate per positive triple

2. positive base rate $\alpha = P/(P + N)$ where P is the number of positives and N the number of negatives

The main drawback of Tabacof and Costabello [35] is that not all possible negatives are generated for every positive triple. Only a fixed number of negative triples are generated based on a η, and these negatives are selected randomly. Depending on the semantic plausibility of the selected negatives, the training of the calibration model may significantly differ. In one experiment [35], different α values were used to find the most suitable one. However, each run picks random negatives with different semantic levels which makes the results harder to compare fairly.

Safavi et al. [31] explored the difference in the effectiveness of calibration between open-world and closed-world assumptions for predicate prediction. On one hand, the closed-world assumption considers all missing triples are negatives rather than unknown, making the expected probabilities of the ground truth easy to generate: triples that are present in the graph at hand are considered positives, while triples that are not present are considered negatives. On the other hand, predicate prediction consists of, given a subject s and an object o, for every predicate p, the model under evaluation provides a score for (s, p, o). The accuracy is measured based on the percentage of results where the predicate with the highest score (ranked in the first position) exists in the test split. The authors showed that knowledge graph model calibration for the close-world assumption results in low expected calibration errors and high accuracy for predicate prediction.

In a different set of experiments, Safavi et al. [31] trained calibration models using the closed-world assumption, but the models were evaluated through crowdsourcing based on the open-world assumption. Crowdsourcing was accomplished based on a specific knowledge graph (Wikidata) such that participants needed to decide whether a given triple is factually correct. These results showed that open-world calibration resulted in high expected calibration errors and overconfident models. As Safavi et al. [31] aptly points out, this difference arises because the models were trained under the close-world assumption and, then, evaluated based on crowdsourcing and the open-world assumption. Furthermore, crowdsourcing in knowledge graphs is challenging due to polysemy, information disparity, extraction errors, and knowledge that is not explicitly stated [4]. As a result, calibration of knowledge graph models trained and tested under the open-world assumption for link prediction remains unexplored.

Chapter 3

Close-world Calibration

In this chapter, we discuss several observations in the context of close-world calibration of knowledge graph models for link prediction and how they could affect model performance. Section 3.1 discusses the merits of preprocessing knowledge graph scores for calibration, section 3.2 addresses the need to balance classes for knowledge graphs and 3.3 proposes a new metric to evaluate calibration.

3.1 Scaling functions

The Platt scaling technique for model calibration exploits a sigmoid function to transform scores into the [0, 1] range. Many machine learning models output scores (or logits) that are in the $(-\infty, \infty)$ range, which makes the sigmoid function a good choice. However, in knowledge graph models, scores are either distances or similarities [39]. For example, translation models exploit either L^1- or L^2-norms as their scoring functions [5,18,40]. Thus, the range of the predicted scores is $[0, \infty)$ for these models. Even though these boundaries are typically not discussed in similarity-based approaches, the same model typically outputs negative and positive scores [1], which suggests that the range is $(-\infty, \infty)$.

However, depending on the algorithm to train a model and the training phase itself, the range of the scores predicted by such model varies, even significantly in certain cases [36]; for instance, scores may be in the $(0, 20]$ range. Furthermore, scores may be limited to a specific range in a postprocessing step, e.g., scores that are less than -20 or greater than 20 are transformed into exactly -20 or 20, respectively [1]. This is problematic in the context of link prediction as, depending on how ties are resolved, repeated scores may have an impact in the final accuracy values based on ranks [1].

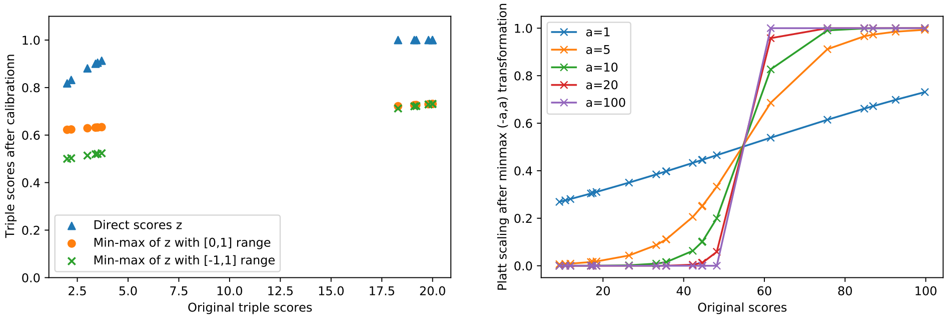
To tackle these issues, we focus on data scaling approaches to transform the scores of a given model for link prediction so that they lie within a common range. Our hypothesis is that calibration for link prediction may benefit from such data scaling. Out of the different approaches, min-max scaling is the most promising one since it maintains the margin between predictions learned by the original model. For a score z, min-max scaling transforms it into the $[0, 1]$ range as follows:

$$z' = \frac{z - min}{Max - min}$$

where min and Max are the minimum and maximum z values observed, respectively.

That min-max scaling maintains the learned margin is an important feature for link prediction since we should obtain the same ranks that measure accuracy after scaling. We consider two variants of min-max scaling - the regular presented above that results in range $[0, 1]$, and a modified version whose range is $[-a, a]$ as follows: $z'' = 2a\,z' - a$. While both variants maintain the learned margin between positive and negative triples as expected, the $[-a, a]$ variant also moves the range closer to the range the sigmoid function is expecting. The sigmoid curve is centered at zero such that all input values above zero results in a value above .5, and those below zero result in a value below .5. While distance-based scores transformed using the $[0, 1]$ range will result in values above .5, the $[-a, a]$ range allows us to use the .5 threshold to distinguish between positives and negatives.

To illustrate our discussion, Figure 3.1a plots the actual scores predicted by a distance-based model against the result of applying Platt scaling to the scores. Note that A and B are randomly selected and kept constant for all the comparisons. Additionally, Figure 3.1a plots the same scenario but using both variants of min-max scaling - the $[0, 1]$ and the $[-a, a]$ approach where $a = 1$. As it can be noted, direct scores and min-max scaling using the $[0, 1]$ range lead to scores above .5. One of the benefits of closed-world calibration is that the .5 threshold can be used to differentiate positives and negatives [35],



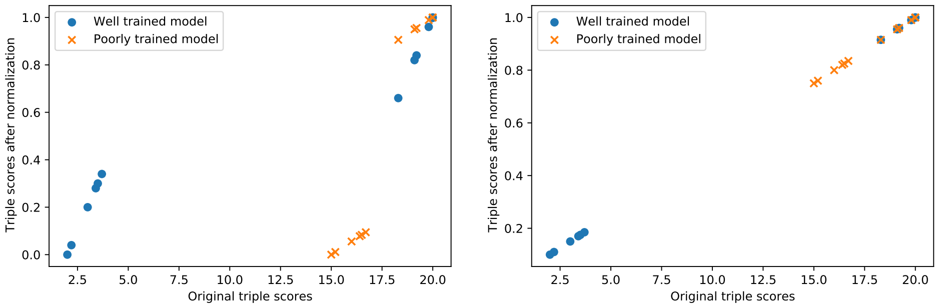
(a) Platt scaling

(b) Min-max with different ranges

Figure 3.1: Comparison of different data scaling approaches. Figure 3.1a plots Platt scaling (A and B are randomly selected) using the actual scores of the same model and two types of min-max scaling ([0, 1] and [-1, 1]). Figure 3.1b shows the difference when different ranges $[-a, a]$ are selected for the min-max scaling.

which is thus not achieved by direct scores and min-max scaling using the $[0, 1]$ range. However, this is achieved with min-max scaling using the $[-a, a]$ range, where some scores after scaling are below .5. We also consider the effect of different values of a on the scores in Figure 3.1b. We observe that low values like $a = 1$ do not work well with Platt scaling as the sigmoid function maps it to values in the range $[0.2, 0.8]$. Higher values of a like $10, 20, 100$ result in a large number of the scores tied at 0 or 1 after Platt scaling. Both of these scenarios can hurt the performance of the knowledge graph model.

The next step is to compute minimum and maximum values. One option is to use the local minimum and maximum of the model under evaluation. In the process of training knowledge graph models for link prediction, for the same embedding approach and knowledge graph, a validation split is typically used to select one or more models according to certain accuracy criteria [36]. These models comprise different hyperparameter values. As a result, we may need to calibrate several models of the same approach over the same knowledge graph. Furthermore, we would like to compare these new calibration models to select the best one during test. Since different equally good models can have different ranges, we propose to select the minimum and maximum limits individually for each model. This serves as an approximation of the actual



(a) Min-max single model

(b) Min-max across models

Figure 3.2: Selecting the limits for min-max scaling. Figure 3.2a presents min-max scaling using the minimum and maximum scores of each individual model. Figure 3.2b shows the same scaling but using the minimum and maximum scores across models. Both figures use $[0, 1]$ min-max scaling to demonstrate the differences when picking limits.

limits of the approach at hand which can be $\pm\infty$, while also maintaining the scores the model learned.

To illustrate our discussion, Figure 3.2a shows two models of the same embedding approach over the same knowledge graph. The first model predicts scores in the $(2, 20)$ range, and the second model in the $(15, 20)$ range. Note that the difference in ranges is realistic since the range of the scores depend on the hyperparameter values that were used to train the models. We assume that the model with the $(2, 20)$ range has been well trained since there is more separation among scores, while the other model was poorly trained as the scores are closer. In the figure, we use the local minimum and maximum per model, and we observe that min-max scaling makes the .5 threshold between positive and negative triples much clearer than the original scores in both cases. Figure 3.2b shows how minimum and maximum across models maintains the expected behavior after min-max scaling. In this figure, the minimum and maximum are considered to be 0 and 20, respectively.

3.2 Weights to balance classes

We observe that, in link prediction, negative counterparts typically outnumber the positive triples by a great factor. This imbalance is caused because negatives are generated for every positive triple by corrupting the subject and object in the triple at hand. Depending on the strategy to generate negative counterparts, the number of eligible entities that can be used to corrupt the subject and object can be very large [2]. While training a calibration model, if the majority of the samples in the validation split are negatives, the result is likely to be biased towards negatives, which implies that the A and B values tend to augment the scores. To avoid this, we explore the option of weighting the positive triples so the classes are balanced. For every batch containing one positive triple and n negative counterparts, the positive triples are resampled n times to create a balance. We also propose using the number of samples in each class as the basis of our weighting strategy. For X number of samples in a class, the weight used for that class is of the form $\frac{1}{\sqrt{X}}$

3.3 Evaluating calibration

Finally, we propose to study the effect of calibration on the predicted probability distribution of a model. Kullback–Leibler divergence is used to measure how different a probability distribution is from another probability distribution as follows [21]:

$$KL(P||Q) = \sum_{x \in X} P(x) log \left(\frac{P(x)}{Q(x)} \right)$$

where X is a set of measurements, and P and Q are the two probability distributions under study. In our case, X is the complete set of positive and negative triples. $P(x)$ is a function that takes each triple before calibration as input, and outputs the probability of the triple. Such a probability is computed by divided every score by the sum of all scores. $Q(x)$ is similar but relying on the scores after calibration.

While the Kullback–Leibler divergence is asymmetric, the Jensen-Shannon distance is symmetric and uses the divergence between two distributions as follows [21]:

$$JS(P,Q) = \sqrt{\frac{KL(P||M) + KL(Q||M)}{2}}$$

where M is the element-wise mean between P and Q.

Chapter 4

Open-World Calibration

The open-world assumption in knowledge graphs establishes that it is unknown whether missing triples are positives or negatives [32]. In practice, many algorithms require the presence of negatives to train knowledge graph models [36]. To solve this issue, these algorithms exploit strategies to generate negatives by corrupting positive triples [2]. During training, these strategies assume that only the triples present in the training split are positives, ignoring the validation and test splits. During validation, they assume that the triples present in both training and validation splits are positives, but not the ones in the test split. Finally, during testing, the union of training, validation and test splits are considered positives. These assumptions may affect the accuracy of the model at hand [31]. We propose open-world calibration by combining different strategies to generate negatives as well as exploiting the expected semantic plausibility of these negatives. Section 4.1 discusses how we can observe different levels of plausibility through negative generation strategies.Section 4.2 utilizes these to generate target probabilities that can be used during calibration and section 4.3 discusses our adapted weighted strategy with these new probabilities.

4.1 Negative generation Strategies

We first focus on the expected semantic plausibility of the strategies to generate negative triples presented above [2]. The Global-Naïve strategy is expected to generate nonsensical negatives as the subjects (objects) selected for corruption are never subjects (objects) in the rest of the graph. In our movie example, this strategy outputs negatives such that the subjects of ActedIn predicates are countries, e.g., (USA, ActedIn, StarWars), which is nonsensical since a country cannot act in a movie. TLCWA generates negatives that are expected to be semantically plausible, and are more prone to be missing triples than other negatives, e.g., (ElleFanning, ActedIn, StarWars), which can be true as Elle Fanning has acted in several movies, but negative in this case. It is not possible to define the expected semantic plausibility of negatives generated by LCWA as the subjects and objects used for corruption are not directly related to the positive triple at hand.

We introduce a new strategy and expected semantic plausibility. The Local-Naïve strategy is a specialization of the Global-Naïve strategy such that, having (s, p, o), s' is selected as objects of predicate p that are never subjects of predicate p. Similarly, o' entities are those that are never objects related by p. We denote as N_C the set of negative triples generated by this strategy. Note that $N_C \cap N_G \neq 0$, therefore, it is expected that these negatives are also nonsensical; however, since they are local to a given predicate, they are more plausible than other entities absolutely unrelated to the predicate at hand. For instance, in our movie example, Local-Naïve outputs (StarTrek, ActedIn, StarWars), which is nonsensical since a movie cannot act in another movie.

4.2 Generating semantic probabilities

Instead of having a single probability $p_{-} = .0$ of belonging to the negative class, we adjust p_{-} to take different values based on these strategies and observations regarding the semantic plausibility of the negatives they generate. The goal of link prediction is to rank the positives above the negatives and break up ties. Keeping in mind triple classification, the semantic values for negative triples remain under the threshold 0.5;$p_{-} < .5$. Through this restriction, we keep the original positive-negative division intact while simultaneously adding semantic meaning to the negative triple predictions. This is further seen in the below equations where we use N_{-} as the total number of negatives.

To reduce overfitting, we take advantage of Equation 2.5 and N_{-}, the count of negative triples. Assuming that z_i is the score of a model for an input negative triple $t' = (s', p, o)$ or (s, p, o'), we adjust p_{-} as follows:

Relation	Positive	Global-Naive	LCWA	Local-Naive	TLCWA
religion	avicenna-shia islam	roman catholic church	denys rayner		antoine brutus menier
cause of death	gustav iii of sweden-regicide	roman catholic church	antoine brutus menier		denys rayner
place of death	alfred kinsey-bloomington indiana	roman catholic church	denys rayner	battling siki	antoine brutus menier
profession	el lissitzky-artist	roman catholic church	antoine brutus menier		denys rayner
location	maria winteler einstein-united states	roman catholic church	antoine brutus menier		nietzchka keene
gender	elizabeth montgomery-female	roman catholic church	rajiv gandhi		antoine brutus menier
nationality	yehudi menuhin-switzerland	roman catholic church	yongzheng emperor		antoine brutus menier
place of birth	mollie sugden-keighley	roman catholic church	denys rayner		antoine brutus menier
institution	william lyon mackenzie king - university of toronto	roman catholic church	antoine brutus menier		friedrich bessel
children	cornelius vanderbilt - george washington vanderbilt ii	roman catholic church	denys rayner	anthony asquith	antoine brutus menier
parents	alexander stewart earl of mar - alexander stewart 1st earl of buchan	roman catholic church	denys rayner	antoine brutus menier	anthony asquith
spouse	dolores del rio-cedric gibbons	roman catholic church	antoine brutus menier	archduke joseph of austria palatine of hungary	mary de bohun
ethnicity	william feller-croats	roman catholic church	antoine brutus menier		jefferson davis

Table 4.1: FB13 - Corrupting subject using different strategies

Relation	Positive	Global-Naive	LCWA	Local-Naive	TLCWA
religion	ethelbert of kent-christianity	denys rayner	cancer	antoine brutus menier	roman catholic church
cause of death	kenneth e hagin - cardiovascular disease	denys rayner	antoine brutus menier	anthony asquith	cancer
place of death	alfred kinsey - bloomington indiana	denys rayner	roman catholic church	antoine brutus menier	madison
profession	shintaro katsu - writer	denys rayner	antoine brutus menier	anthony asquith	mathematician
location	rich vogler - indiana	denys rayner	antoine brutus menier	anthony asquith	madison
gender	dora distria - female	denys rayner	roman catholic church	antoine brutus menier	male
nationality	frederick van nuys - united states	denys rayner	roman catholic church	antoine brutus menier	kingdom of england
place of birth	paolo mantegazza - monza	denys rayner	roman catholic church	antoine brutus menier	madison
institution	james jackson jr - university of cambridge	denys rayner	antoine brutus menier	anthony asquith	harvard university
children	cornelius vanderbilt - george washington vanderbilt ii	denys rayner	roman catholic church	antoine brutus menier	anthony asquith
parents	alexander stewart earl of mar - alexander stewart 1st earl of buchan	denys rayner	roman catholic church	anthony asquith	antoine brutus menier
spouse	dolores del rio - cedric gibbons	denys rayner	antoine brutus menier	george boole	mary de bohun
ethnicity	zora neale hurston - african american	denys rayner	antoine brutus menier	jefferson davis	united kingdom

Table 4.2: FB13 - Corrupting object using different strategies

Relation	**Positive**	**Global-Naive**	**LCWA**	**Local-Naive**	**TLCWA**
type of	gavialis 1 - reptile genus 1	anxiety disorder 1	solar array 1	penalise 1	spiritual bouquet 1
synset domain topic	convert 6-bowling 1	anxiety disorder 1	sympathy card 1	church of rome 1	spiritual bouquet 1
has instance	elate 1-beatify 1	anxiety disorder 1	genus xylomelum 1	spiritual bouquet 1	sympathy card 1
member holonym	genus orycteropus 1-family orycteropodidae 1	anxiety disorder 1	spiritual bouquet 1	church of rome 1	chamaecyparis lawsoniana 1
part of	lima bean 3-lima bean 2	anxiety disorder 1	spiritual bouquet 1	church of rome 1	solar array 1
has part	paddle wheel 1-paddle box 1	anxiety disorder 1	spiritual bouquet 1	photovoltaic cell 1	church of rome 1
member meronym	acris 1-cricket frog 1	anxiety disorder 1	spiritual bouquet 1	chamaecyparis lawsoniana 1	church of rome
similar to	ancient 2-old 1	anxiety disorder 1	spiritual bouquet 1	quality 1	freelance 1
subordinate instance of	henry rowe schoolcraft 1 - ethnologist 1	anxiety disorder 1	spiritual bouquet 1	literary critic 1	ciardi 1
domain region	music 1-pianissimo 1	anxiety disorder 1	sympathy card 1	spiritual bouquet 1	church of rome 1
domain topic	russia 2-borodino 1	anxiety disorder 1	spiritual bouquet 1	maisonette 2	australia 1

Table 4.3: WN11 - Corrupting subject using different strategies

1. $t' \in N_G \Rightarrow p_- = \frac{1}{N_- + 2}$
2. $t' \in N_C \setminus N_G \Rightarrow p_- = .125 + \frac{1}{N_- + 2}$
3. $t' \in N_T \Rightarrow p_- = .375 + \frac{1}{N_- + 2}$
4. $t' \in N_L \setminus (N_G \cup N_T \cup N_C) \Rightarrow p_- = .250 + \frac{1}{N_- + 2}$

Note that, since there is overlap between the sets of negatives, these adjustments must be accomplished by priority, e.g., (1) has higher priority than (2). The main idea behind these adjusted probabilities is to classify each negative according to its expected semantic plausibility based on the corruption strategy. The selected probabilities correspond to the quartiles of the negative probability space $[0, 0.5)$.

Tables 4.1, 4.2, and 4.3 contain the corrupted triples based on different strategies. While the strategies work with all graphs, the tables illustrate how they can vary depending on the knowledge graph being considered. In FB13, we observe that our expectations for the kinds of triples generated for the respective strategies are met. Table 4.1 contains positive triple Gustav III of Sweden whose cause of death was regicide. Corrupting this triple using Global Naive results in the Roman Catholic Church's cause of death being regicide - which is meaningless. Corrupting the same triple using TLCWA results in replacing person Gustav III of Sweden with another person Denys Rayner - which has a higher level of plausibility than the triple the Global strategy generated. Table 4.2 shows the triples generated when corrupting objects of a positive triple and we observe similar behavior for Global and TLCWA. Our assigned labels for these two classes are on opposite ends of negativity with Global being most negative and TLCWA being least negative. Through these tables, we observe that it is not possible to assign a level of negativity to LCWA and is most likely to contain missing triples. For instance, Denys Rayner was selected to corrupt the subject of Place of Death. If there was actually was an entry for Denys Rayner's place of death in the graph, it would have been part of TLCWA.

Table 4.3 contains the corrupted triples for WN11. We observe that the strategies are less reliable for WN11 as compared to FB13. While these negative generation strategies are supposed to be general-purpose, they are by-design extensible and can be improved with manual tweaking based on domain-knowledge about the graph in question.

4.3 Using weights

We adapt the weighting strategy from Chapter 3 as we are now dealing with multiple divisions in the negative class. For open-world calibration, the weights are calculated based on the number of triples in each of the defined classes.

$$Wt_G = \frac{1}{\sqrt{|N_G|}} \qquad Wt_C = \frac{1}{\sqrt{|N_C \setminus N_G|}}$$

$$Wt_T = \frac{1}{\sqrt{|N_T|}} \qquad Wt_L = \frac{1}{\sqrt{|N_G \cup N_T \cup N_C|}}$$

We hypothesise that these weights will better help the model calibrate as the number of triples generated by these strategies vary vastly. For instance, in the knowledge graph FB13 [33], the number of triples generated by LCWA is about two times that of triples generated by Naive and five times that of TLCWA [2].

Chapter 5

Experiments

We study closed-world and open-world calibration for link prediction. We selected a number of pre-trained models that are publicly-available [36]. These models were trained using the implementations of the algorithms provided by the OpenKE framework [16]. A variety of hyperparameter values were used and the best models were selected based on their accuracy over the validation splits using $\overline{MR}$. We focused on Analogy [23], ComplEx [38], DistMult [41], HolE [28], RotatE [34], SimplE [20], TransD [18], TransE [5] and TransH [40] models, which is a combination of distance-based and similarity-based models. The datasets used in our experiments are as follows: FB13 [33], FB15K [6], FB15K-237 [37], NELL-995 [25], WN11 [11], WN18 [5], and WN18RR [12]. These datasets are the de facto standard to evaluate link prediction [16, 36]. These datasets are publicly available and already divided into training, validation and test splits. However, [36] detected imbalanced among the splits, and proposed a new set of splits that are well balanced and guaranteed to preserve the topology of the original knowledge graph. We rely on these new splits. Note that, for a certain algorithm and dataset, several models with similar accuracy over the validation split may be available. We narrow down by keeping the worst model in terms of positive triples Brier score among the different variants. The elimination is based on the fact that calibration is unnecessary for already well-calibrated models. We consider only positive triples as negatives significantly outnumber the positives and tends to hide poor calibration results.

Through our experiments, we observe that normalizing the model scores

Table 5.1: Brier scores of positive triples before calibration - model scores are normalized with min-max

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	0.531	0.897	0.950	0.604	0.953	0.850	0.970
ComplEx	0.396	0.929	0.952	0.877	0.024	0.891	0.771
DistMult	0.009	0.037	0.567	0.397	0.403	0.045	0.726
HolE	0.759	0.973	0.933	0.780	0.902	0.767	0.733
RotatE	0.095	0.486	0.192	0.414	0.352	0.201	0.434
SimplE	0.007	0.969	0.311	0.951	0.411	0.532	0.899
TransD	0.683	0.778	0.656	0.533	0.831	0.699	0.698
TransE	0.637	0.823	0.856	0.820	0.765	0.534	0.685
TransH	0.733	0.804	0.808	0.770	0.749	0.555	0.871

with the min-max function provides more meaningful results before calibration than the sigmoid function. The sigmoid function leads to all the scores being equal to 1 and making the overall positive-negative Brier score ≈ 1 (as all negatives are mislabeled). All further experiments rely on min-max scaling. From these before calibration results, we also choose to use weights for all further experiments. This is because without weights, the positives do not have any effect on the training or the final metrics.

Table 5.1 contains the worst Brier scores of only positive triples before calibration for each model-graph type. Since we are using min-max normalization with limits chosen based on model, the high error observed indicates that there are some positives with significantly low scores. We choose to focus on the worst ones as these models are the ones that will benefit the most from calibration.

We perform calibration on the models whose Brier scores are listed in Table 5.1. The calibration models based on Platt scaling are trained using the validation split. We exploited LCWA to train calibration models in both settings: closed-world and open-world. Furthermore, we use the filtered generation of negatives [5], i.e., in the validation split, triples present in training and validation are not considered negatives, ignoring the test split; in the test split, triples present in all splits are not considered negatives. After training, we used the test split to evaluate the calibration models. In the context of

Table 5.2: AMR of models for closed-world calibration by minimizing BCE Loss

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	-0.530	0.974	0.924	-0.745	-0.890	-0.483	-0.772
ComplEx	-0.766	0.958	0.961	-0.874	0.358	-0.624	-0.718
DistMult	0.424	0.756	0.878	-0.893	-0.378	-0.378	-0.820
HolE	-0.589	0.985	0.934	-0.795	-0.880	-0.751	-0.809
RotatE	0.566	0.735	0.572	0.647	0.640	0.301	0.661
SimplE	0.440	0.982	0.954	-0.220	-0.287	-0.360	-0.802
TransD	0.894	0.987	0.985	0.964	0.973	0.906	0.971
TransE	0.912	0.994	0.981	-0.941	0.976	0.946	0.973
TransH	0.898	0.994	0.975	-0.957	0.960	0.948	0.932

closed-world and open-world calibration, we also evaluated the following: no transformation of the model scores and min-max scaling to the $[-a, a]$ range with $a = 5$. We use weighted training based on the approach discussed in Chapter 3 to balance positives and negatives. Platt scaling is performed by minimizing the Negative Log Likelihood Loss function.

Tables 5.2 and 5.3 show the Adjusted Mean Rank AMR for these models after calibration. We observe that most of the accuracy values remain the same before and after calibration (for both closed-world and open-world). This can be explained by the fact that Platt scaling is a linear transformation and does not directly affect the ranks. One stark difference we observe between ranks before and after calibration are that some of them are inverted. For instance, closed-world calibration for ComplEx - FB13 has the score -0.766 whereas for open-world, it has the better invert 0.766. This can be explained by the fact that A and B in the Platt scaling equation can be minimized to a positive or negative number. Since it is randomly initialized, the final score could tilt either way. For a negative value of A, the final ranks are reversed with all the positives ranking below the negatives.

To address this issue, we modify the loss function to minimize two parameters - the binary cross entropy loss to move the scores towards their expected values and the Margin Ranking Loss that makes sure that each positive triple is still ranked above its negative counterparts. By prioritizing the positive

Table 5.3: AMR of models for open-world calibration by minimizing BCE Loss

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	-0.530	0.974	0.924	-0.745	-0.890	-0.483	-0.772
ComplEx	0.766	0.958	0.961	-0.874	0.358	-0.624	0.718
DistMult	0.424	0.756	0.878	-0.893	0.378	-0.378	0.820
HolE	-0.589	0.985	0.934	-0.795	-0.880	-0.751	-0.809
RotatE	0.566	0.735	0.572	0.647	0.640	-0.301	0.661
SimplE	0.440	0.982	0.954	0.220	-0.287	-0.360	-0.802
TransD	0.894	0.987	0.985	0.964	0.973	0.906	0.971
TransE	0.912	0.994	0.981	0.941	0.976	0.946	0.973
TransH	0.898	0.994	-0.975	0.957	0.960	0.948	0.932

triples' position, this combined loss function also has the added benefit of acting as an additional weight for positives. Through this, we can calibrate the knowledge graph model without hurting its accuracy.

$$Loss = BCE(Platt(z_i), p_i) + MRL(Platt(z_{i+}), Platt(z_{i-})) \tag{5.1}$$

Equation 5.1 shows the combined loss function used. The first part of the equation is the same Binary Cross Entropy Loss that is performed on the Platt scaled scores from the model. The second part of the equation is the new Margin Ranking Loss that takes the positive triple score z_{i+} as the first argument and the negative triple scores z_{i-} as the second. Equation 5.2 shows how this loss is calculated where margin determines how far apart the two arguments should be. The default argument for margin is set to 0. x_1 in the equation represents the list that is expected to rank higher than the list x_2. In our case, x_1 would be the positive scores and x_2 the negative ones.

$$MRL(x_1, x_2) = max(0, -(x_1 - x_2) + margin) \tag{5.2}$$

Tables 5.4 and 5.5 contain the accuracy results after training with this combined loss. As we can see, most of the issues with inversion is fixed through our proposed combined loss.

The labels for open-world assumption were based on the negative generation strategies. To test the validity of the assumptions made, we compare the accuracy of open-world calibration with those when the labels are flipped such

Table 5.4: AMR of models for closed-world calibration by minimizing BCE and MRL Loss

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	0.436	0.974	0.924	0.745	0.890	0.483	0.772
ComplEx	0.761	0.958	0.961	0.874	0.358	0.624	0.718
DistMult	0.424	0.702	0.878	0.893	-0.378	0.343	0.820
HolE	0.589	0.985	0.934	0.795	0.880	-0.751	0.809
RotatE	0.566	0.735	-0.572	0.647	0.640	0.301	0.661
SimplE	0.440	0.982	0.954	0.214	0.287	-0.360	-0.802
TransD	0.894	0.987	0.985	0.964	0.973	0.906	0.971
TransE	0.912	0.994	0.981	0.941	0.976	0.946	0.973
TransH	0.898	0.994	0.975	0.957	0.960	0.948	0.932

that the least . In this case, the positive labels are kept as is and the flipped negative labels are as follows-

1. $t' \in N_G \Rightarrow p_- = 0.375 + \frac{1}{N_- + 2}$
2. $t' \in N_C \setminus N_G \Rightarrow p_- = .250 + \frac{1}{N_- + 2}$
3. $t' \in N_T \Rightarrow p_- = \frac{1}{N_- + 2}$
4. $t' \in N_L \setminus (N_G \cup N_T \cup N_C) \Rightarrow p_- = .125 + \frac{1}{N_- + 2}$

Table 5.6 shows the accuracy we obtain when the models are calibrated with these flipped labels. We observe that almost all of the models are significantly lower when compared to results from before calibration and open-world calibration. This difference in accuracy when the positives are kept the same indicates that the original probabilities assumed for open-world calibration align better with the graphs.

We also analyze the Brier scores after calibration with the combined loss function. From tables 5.7 and 5.8, we observe that open-world calibration is better at calibrating positive triples compared to closed-world. Weighted calibration has more of an advantage in closed-world where there are only two classes with one (negative) being about 100 times more than the other. In contrast, in open-world, we have 5 classes who sizes are slightly less biased.

Table 5.5: AMR of models for open-world calibration by minimizing BCE and MRL Loss

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	0.521	0.974	0.924	0.745	0.890	0.483	0.772
ComplEx	0.766	0.958	0.961	0.874	0.358	0.624	0.718
DistMult	0.432	0.740	0.878	0.893	0.376	0.378	0.819
HolE	0.589	0.985	0.934	0.795	0.880	0.751	0.809
RotatE	0.566	0.735	-0.572	0.647	0.640	0.301	0.661
SimplE	-0.444	0.982	0.954	-0.220	0.258	0.360	0.802
TransD	0.894	0.987	0.985	0.964	0.973	0.906	0.971
TransE	0.912	0.994	0.981	0.941	0.976	0.946	0.973
TransH	0.898	0.994	0.975	0.957	0.960	0.948	0.932

Table 5.6: AMR of models for open-world calibration where the labels for the negative triples are flipped

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	-0.010	-0.011	-0.052	-0.001	-0.018	-0.008	-0.013
ComplEx	-0.002	-0.058	-0.009	-0.003	-0.002	-0.003	0.073
DistMult	0.000	-0.002	-0.020	-0.030	-0.003	-0.000	0.001
HolE	-0.016	-0.170	-0.030	-0.056	-1.402	-0.003	0.000
RotatE	-0.002	-0.028	0.016	-0.003	-0.004	-0.001	-0.000
SimplE	0.001	-0.067	-0.044	-0.001	-0.001	-0.138	-0.014
TransD	-0.007	-0.173	-0.076	-0.098	-0.041	-0.017	-0.035
TransE	-0.034	-0.242	-0.140	-0.001	-0.149	-0.102	-0.170
TransH	-0.010	-0.055	-0.048	-0.049	-0.026	-0.030	-0.005

Table 5.7: Brier scores for only positive triples for closed-world calibration

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	0.995	0.993	0.993	0.964	0.894	0.949	0.992
ComplEx	0.978	0.994	0.992	0.923	0.993	0.000	0.998
DistMult	1.000	0.993	0.991	0.000	0.999	0.986	0.997
HolE	0.000	0.996	0.993	0.000	0.000	0.624	0.998
RotatE	0.979	0.000	1.000	0.798	0.888	0.785	0.995
SimplE	0.999	0.990	0.993	0.989	0.995	1.000	0.011
TransD	0.005	0.998	0.998	0.918	0.955	0.799	0.997
TransE	0.993	0.996	0.997	0.951	0.949	0.941	0.997
TransH	0.993	0.996	0.997	0.863	0.981	0.867	0.994

Table 5.8: Brier scores for only positive triples for open-world calibration

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	0.709	0.653	0.695	0.719	0.760	0.810	0.842
ComplEx	0.744	0.693	0.660	0.001	0.573	0.000	0.937
DistMult	0.704	0.639	0.644	0.725	0.576	0.639	0.935
HolE	0.804	0.774	0.726	0.734	0.863	0.804	0.951
RotatE	0.953	0.996	0.981	0.905	0.896	0.909	0.996
SimplE	0.702	0.700	0.695	0.693	0.579	0.646	0.000
TransD	0.938	0.912	0.893	0.735	0.895	0.817	0.982
TransE	0.899	0.621	0.834	0.781	0.884	0.740	0.971
TransH	0.872	0.722	0.811	0.757	0.856	0.827	0.007

However, we observe from Table 5.1, 5.7, and 5.8 that Brier score increases after calibration when we take into consideration only positive triples. Even with weights, it appears that the negatives overshadow the positive triples. This can also be observed by the fact that the Brier scores of the negative triples are all 0 for closed-world calibration and $\leq$0.5 for open-world calibration.

Tables 5.9 and 5.10 contain the overall Brier scores for closed-world calibration (training and testing) and open-world calibration (training and testing) respectively.

We also notice that, in general, the Brier scores after open-world calibration

Table 5.9: Best-performing, closed-world, weighted Brier scores for closed-world calibration

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	0.003	0.006	0.006	0.004	0.006	0.004	0.003
ComplEx	0.003	0.006	0.006	0.005	0.003	0.994	0.003
DistMult	0.003	0.006	0.006	0.961	0.003	0.003	0.003
HolE	0.971	0.006	0.006	0.981	0.991	0.046	0.003
RotatE	0.003	0.983	0.006	0.014	0.006	0.016	0.003
SimplE	0.003	0.006	0.006	0.004	0.003	0.003	0.799
TransD	0.854	0.006	0.006	0.005	0.004	0.010	0.003
TransE	0.003	0.006	0.006	0.004	0.004	0.004	0.003
TransH	0.003	0.006	0.006	0.008	0.004	0.006	0.003

Table 5.10: Best-performing, open-world, weighted Brier scores for open-world calibration

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	0.024	0.013	0.017	0.021	0.028	0.034	0.024
ComplEx	0.025	0.016	0.016	0.522	0.010	0.611	0.033
DistMult	0.024	0.012	0.015	0.022	0.010	0.019	0.033
HolE	0.029	0.025	0.020	0.023	0.047	0.033	0.035
RotatE	0.047	0.068	0.058	0.052	0.054	0.051	0.041
SimplE	0.024	0.016	0.017	0.019	0.010	0.019	0.727
TransD	0.045	0.053	0.048	0.026	0.058	0.041	0.040
TransE	0.040	0.016	0.035	0.031	0.062	0.030	0.038
TransH	0.036	0.022	0.031	0.032	0.047	0.041	0.602

Table 5.11: Jensen-Shannon distance between scores before and after closed-world calibration

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	0.000	0.002	0.001	0.006	0.006	0.010	0.000
ComplEx	0.000	0.004	0.003	0.002	0.000	0.004	0.001
DistMult	0.001	0.000	0.000	0.001	0.000	0.000	0.000
HolE	0.000	0.001	0.002	0.001	0.006	0.002	0.001
RotatE	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SimplE	0.000	0.002	0.005	0.000	0.000	0.000	0.001
TransD	0.031	0.029	0.012	0.050	0.003	0.005	0.004
TransE	0.036	0.052	0.074	0.025	0.001	0.003	0.004
TransH	0.039	0.054	0.065	0.027	0.077	0.032	0.125

Table 5.12: Jensen-Shannon distance between scores before and after open-world calibration

	FB13	FB15k	FB15k237	WN11	WN18	WN18RR	NELL995
Analogy	0.000	0.002	0.002	0.006	0.006	0.010	0.000
ComplEx	0.000	0.004	0.003	0.002	0.000	0.004	0.001
DistMult	0.000	0.000	0.000	0.001	0.000	0.000	0.000
HolE	0.000	0.002	0.002	0.002	0.006	0.002	0.001
RotatE	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SimplE	0.000	0.002	0.005	0.000	0.000	0.000	0.001
TransD	0.031	0.034	0.010	0.041	0.003	0.005	0.004
TransE	0.047	0.076	0.092	0.026	0.001	0.003	0.003
TransH	0.044	0.068	0.078	0.035	0.078	0.033	0.124

are higher when compared to close-world calibration. Open-world calibration is a comparatively harder task with several classes rather than binary, and the same number of trainable parameters (A and B).

Table 5.11 contains the Jensen-Shannon distance between the model's scores before and after closed-world calibration and Table 5.12 contains the same for open-world calibration. We find that the divergence between the scores are not very different for the two approaches. We also observe that the divergence is very low in both kinds of calibration. This can be explained by the fact that the listed divergence is calculated for models trained with the combined loss-approach which limits the divergence to preserve ranks. In contrast, the divergence of scores trained with BCE Loss alone is slightly higher than those listed in tables 5.11 and 5.12. Through these experiments, we obtain calibrated models while maintaining the original accuracy of the models for link prediction.

Chapter 6

Conclusion

In this thesis, we have studied the effect of calibration using both the closed-world and open-world assumptions on knowledge graph models. We identified several issues with knowledge graph model calibration and focused on addressing them. Knowledge graphs do not generally have negative triples available so we rely on synthetically generated negative triples for calibration. These generated negatives tend to outnumber the positives by a large margin so we propose a weighting strategy to balance the two classes. While the weights help improve the scores for positive triples to an extent, it is still in need of improvement. We observe that negatives tend to bias both training and results even when weights are used.

We note that knowledge graph models use different algorithms for training which in turn results in different score ranges (0 to ∞ for distance metrics) and are not directly comparable to neural networks. We introduced min-max scaling as a way to adapt knowledge graph model scores to be suitable for calibration techniques that work with neural networks. Using the $[-a.a]$ min-max scaling, we observe that the scores are spread out more and also avoids unnecessary ties during the calibration process.

Finally, we introduced a method of working with the open-world assumption without the need for manual labelling. We introduced a novel method of automatically labeling negative triples for the open-world assumption based on the semantic plausibility of different negative generation strategies. We study the effectiveness of these auto-generated labels on different graphs and find that these assumptions hold to an extend. However, these strategies can always be improved based on domain knowledge about the graph in question and are meant to be extensible.

Through our results, we observe that closed-world calibration is an easier task compared to open-world calibration. However, the latter is better at calibrating positive triples. We also proposed a novel combined loss function that ensures that the original model's training and accuracy is kept intact after calibration. We observed that both calibration techniques maintain the original accuracy of the model with this combined loss function.

We expect that open-world calibration will help pave the way for extension of different custom ground truth scoring techniques.

Bibliography

[1] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *CoRR*, abs/2006.13365, 2020.

[2] Iti Bansal, Sudhanshu Tiwari, and Carlos R. Rivero. The impact of negative triple generation strategies and anomalies on knowledge graph completion. In *CIKM*, page 45–54, 2020.

[3] Max Berrendorf, Evgeniy Faerman, Laurent Vermue, and Volker Tresp. Interpretable and fair comparison of link prediction or entity alignment methods with adjusted mean rank. *CoRR*, abs/2002.06914, 2020.

[4] Antoine Bordes and Evgeniy Gabrilovich. Constructing and mining web-scale knowledge graphs: KDD 2014 tutorial. In *SIGKDD*, page 1967, 2014.

[5] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 1–9, 2013.

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*, pages 1–9, 2013.

[7] Antoine Bordes, Jason Weston, Ronan Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. volume 1, 08 2011.

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

[9] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

[10] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[11] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[12] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D knowledge graph embeddings. In *AAAI*, pages 1811–1818, 2018.

[13] 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*, pages 601–610, 2014.

[14] Norbert Fuhr. Some common mistakes in IR evaluation, and how they can be avoided. *SIGIR Forum*, 51(3):32–41, 2017.

[15] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017.

[16] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 139–144, 2018.

[17] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga

Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *CoRR*, abs/2003.02320, 2020.

[18] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 687–696, 2015.

[19] Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Smooth isotonic regression: A new method to calibrate predictive models. *AMIA Summits on Translational Science Proceedings*, 2011:16, 2011.

[20] Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. *arXiv preprint arXiv:1802.04868*, 2018.

[21] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[22] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[23] Hanxiao Liu, Yuexin Wu, and Yiming Yang. Analogical inference for multi-relational embeddings. In *International conference on machine learning*, pages 2168–2178. PMLR, 2017.

[24] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *7th biennial conference on innovative data systems research.* CIDR Conference, 2014.

[25] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhanava Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018.

[26] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.1, 2015.

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

[28] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[29] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, page 625–632, 2005.

[30] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10, 06 2000.

[31] Tara Safavi, Danai Koutra, and Edgar Meij. Evaluating the calibration of knowledge graph embeddings for trustworthy link prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8308–8321, 2020.

[32] Baoxu Shi and Tim Weninger. Open-world knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[33] 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*, pages 926–934. Citeseer, 2013.

[34] 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.

[35] Pedro Tabacof and Luca Costabello. Probability calibration for knowledge graph embedding models, 2020.

[36] Sudhanshu Tiwari, Iti Bansal, and Carlos R. Rivero. Revisiting the evaluation protocol of knowledge graph completion methods for link prediction. In *TheWebConf*, pages 1–12, 2021.

[37] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *ACL Workshops*, pages 57–66, 2015.

[38] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080. PMLR, 2016.

[39] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743, 2017.

[40] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[41] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.