Food Knowledge Representation Learning with Adversarial Substitution

Diya Li*, **Mohammed J. Zaki**
Computer Science Department, Rensselaer Polytechnic Institute
916lidiya@gmail.com, zaki@cs.rpi.edu

Abstract

Knowledge graph embedding (KGE) has been well-studied in general domains, but has not been examined for food computing. To fill this gap, we perform knowledge representation learning over a food knowledge graph (KG). We employ a pre-trained language model to encode entities and relations, thus emphasizing contextual information in food KGs. The model is trained on two tasks – predicting a masked entity from a given triple from the KG and predicting the plausibility of a triple. Analysis of food substitutions helps in dietary choices for enabling healthier eating behaviors. Previous work in food substitutions mainly focuses on semantic similarity while ignoring the context. It is also hard to evaluate the substitutions due to the lack of an adequate validation set, and further, the evaluation is subjective based on perceived purpose. To tackle this problem, we propose a collection of adversarial sample generation strategies for different food substitutions over our learnt KGE. We propose multiple strategies to generate high quality context-aware recipe and ingredient substitutions and also provide generalized ingredient substitutions to meet different user needs. The effectiveness and efficiency of the proposed knowledge graph learning method and the following attack strategies are verified by extensive evaluations on a large-scale food KG.

1 Introduction

Structured knowledge furnishes an in-depth understanding of the world. Knowledge graph embedding (KGE) maps entities and relations into vectors while retaining their semantics (Wang et al., 2017; Lin et al., 2018). KGE has been well-studied and applied in general KGs with common ontological knowledge (i.e., WordNet (Miller, 1995), DBpedia (Auer et al., 2007), and Freebase (Bollacker et al., 2008)). Only a few works have targeted domain-specific KGs (Mohamed et al., 2021; Bonner et al., 2021) and to the best of our knowledge, there is no work for KGE in the food domain. Even though previous work (Li and Zaki, 2020) trains recipe embeddings on a large-scale dataset, KG information is utilized only as side information to assist embedding learning and only recipes get represented, and other node types in the food KG, such as ingredients are ignored. To fill this gap, we aim to conduct knowledge representation learning over the entire food KG to get high-dimensional vectors of nodes and relations while capturing their semantic meanings.

As for encoding models in KGE, most deep learning-based methods like convolutional neural networks (CNN) (Dettmers et al., 2018), recurrent neural networks (RNN) (Guo et al., 2019) and graph neural networks (GNN) (Schlichtkrull et al., 2018; Shang et al., 2019) allow a single static embedding for each entity or relation to describe its global meaning in a given KG. However, their intrinsic contextual nature is ignored, i.e., entities and relations may appear in different graph contexts and exhibit different properties. Transformer-based models (Vaswani et al., 2017) have boosted contextualized text representation learning. Thus, to emphasize the contextual information in knowledge graphs, we employ Transformers to encode entities and relations. Specifically, we adopt BERT (Devlin et al., 2019) to encode the triples in the food KG as paths. The model is trained with two typical tasks in pretrained language models and knowledge graph embedding: to predict a masked entity from a given path, and to predict the plausibility of a triple in the KG.

Large-scale food data offers rich knowledge that can help many issues related to healthy eating behaviors. Among various food related research, the food substitution problem is gaining increasing attention owing to its applicability in tasks like food question answering (Yagcioglu et al., 2018; Chen

* Diya Li is the corresponding author.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 653–664
November 20–23, 2022.

et al., 2021) and personalized dietary recommendation (Min et al., 2019). In practice, there is a rising demand for people seeking food substitutions due to health concerns, ingredient shortage, or personal preferences (Epstein et al., 2010). For instance, there are numerous posts on *reddit* asking for food alternatives like "*substitutes for tomatoes in pizza*".

Previous work discovers suitable substitution options based on semantic similarity via explicit substitution rules and additional context (Akkoyunlu et al., 2017; Pan et al., 2020; Shirai et al., 2020). They require many handcrafted features and there is no formal evaluation. Efforts to apply machine learning methods to efficiently select substitutions have been limited due to the lack of public datasets with valid substitutions. Moreover, evaluating the quality of ingredient substitutions is difficult since the validity of an ingredient substitution may be influenced by personal preference and perceived purpose of the substitution.

Massive food KGs have become good sources for suggesting substitutions, since they provide unified and standardized concepts and their relationships in structured form, which is very valuable for food related studies. However, KGs often suffer from sparseness if one only uses structure information in observed triple facts (Shirai et al., 2020). We notice that the degree of nodes in Food KGs are mostly small (Qin et al., 2019; Haussmann et al., 2019), and therefore contextual information will be ignored if we model food substitution directly on the KG. Besides, we observe that the food substitutions should be distinct from context or be generalized according to different user query scenarios. For the first case, people often ask for ingredient substitutions with reference to a particular food or recipe. For example, "*applesauce*" can be a good substitute for "*sugar*" in "*carrot cake*", while "*honey*" is better for "*sugar*" in "*brown sugar meatloaf*". Thus, context is important in such scenarios. The second case refers to the huge number of queries on search engines asking for food substitutions for general purpose. For instance, "*what can be substituted for heavy cream*".

To tackle the above issues, we conduct textual adversarial attack on our learnt KGE model. We utilize a masked language model to generate high quality adversarial samples which finds substitutions that maximize the risk of making wrong assertions on KG triple plausibility prediction. We employ the generated adversarial samples as food substitutions. Furthermore, to meet the different food substitution purposes, we design a collection of attack strategies to generate three types of food substitutions: *context-aware recipe substitutions*, *context-aware ingredient substitutions* and *generalized ingredient substitutions*. In order to generate *context-aware recipe substitutions*, we first find the vulnerable tokens in recipes, defined as those that trigger an error in a target prediction model. Next, we apply a masked language model in a semantic-preserving way to generate substitutes, with flexibility to replace, add, or delete vulnerable tokens. The generation of *context-aware ingredient substitutions* is similar to recipe substitutions but only valid ingredients are selected as substitutions. The two types of substitutions are naturally aware of context since they are generated from a pre-trained language model, taking advantage of its superiority in contextualized information and rich linguistic knowledge. For the *generalized ingredient substitutions*, the adversarial attack is conducted among triples formed from all the ingredient's neighbors in the KG. A successful attack is achieved only when the adversarial sample fools most of its neighbors, preventing it to be contextualized to any specific neighbor.

The contribution of our work is twofold: First, we address the sparseness problem in food KG and enrich its representation through the retraining of a pre-trained language model on two tasks – masked entity and triple plausibility prediction. Second, we conduct the food substitution work over KGs to leverage the structured and large-scale knowledge. We propose a novel collection of attack strategies to create different types of food substitutions. *We are the first to deeply generate food substitutions in an adversarial attack manner, thus avoiding the problem of substitutions ground truth shortage.* Both automatic and human evaluations show the high quality of our food substitutions.

2 Related Work

2.1 Knowledge Graph Embeddings

The models that encode the interactions of entities and relations in knowledge graphs can be categorized into: linear/bilinear models, factorization models, and neural networks. Among the neural networks-based models, Convolutional Neural Networks (CNNs) are utilized for learning deep expressive features (Dettmers et al., 2018; Nguyen et al., 2018). Graph Neural Networks (GNNs) are intro-

duced for learning connectivity structure under an encoder-decoder framework (Schlichtkrull et al., 2018; Shang et al., 2019). Transformer-based models have boosted contextualized text representation learning. Wang et al. (2019) employed Transformers to encode edges and path sequences. Similarly, Yao et al. (2019) borrowed ideas from the BERT (Devlin et al., 2019) model as an encoder for entities and relations. Our proposed method for knowledge representation learning also utilizes transformers as the encoding model while two subtasks are considered for training. It is important to note that while there are many KGE works in the general domain, we are the first to propose effective KG embeddings for a large-scale food KG.

2.2 Food Substitution

Previous work on food substitutions is mainly based on semantic similarity with explicit substitution rules such as food taxonomy and food subclass information (Gaillard et al., 2015; Skjold et al., 2017), but it is not applicable for general use. Akkoyunlu et al. (2017) proposed a rule-based approach to extract food substitution if the two foods are consumed in a similar context. Pan et al. (2020) explored substitution of ingredients via simple embedding similarity while the quality of substitutes was not examined. Shirai et al. (2020) suggested substitutes based on user context, by leveraging explicit and implicit semantic information about ingredients from various sources. Without needing the effort for feature design and external rules, our work focuses on contextualized and generalized food substitutions. It can automatically suggest different ingredients according to the recipe context and also generalized ones.

2.3 Textual Adversarial Attack

An increasing amount of effort is being devoted to generating better textual adversarial examples with various attack methods. There are a lot of attack models to explore synonym substitution rules to enhance semantic meaning preservation (Jin et al., 2020; Li et al., 2020; Wang et al., 2021; Li et al., 2021; Garg and Ramakrishnan, 2020). Among them, Jin et al. (2020) replace tokens with their synonyms derived from counter-fitting word embeddings (Mrkšić et al., 2016). The mask-then-infill approaches are widely adopted to greedily replace tokens with the predictions from BERT (Li et al., 2020; Garg and Ramakrishnan, 2020; Li et al., 2021). Unlike the above works focusing on textual perturbation, we design a collection of attack strategies particularly for KG triples, with regards to entity property and substitution query purpose.

3 Methodology

In this section, we first encode a food KG into a pre-trained language model (BERT) to learn entity and relation representations. Then, we conduct attacks on BERT to generate different types of adversarial samples as food substitutions.

3.1 Contextualized KG Embedding

Given a KG $\mathcal{G}$ composed of head-relation-tail triples $\{(h, r, t)\}$. Each triple indicates a relation $r \in \mathcal{R}$ between two entities $h, t \in \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ are the entity and relation sets. The entities in food KG are recipes and ingredients. Here we formulate the triple (h, r, t) as a path $h \rightarrow r \rightarrow t$, e.g., *banana bread* $\rightarrow$ *consist_of* $\rightarrow$ *all purpose flour*.

The input to the model can be one triple or multiple triples of the form $h \rightarrow r \rightarrow t$. The first token of every input path is always a special classification token `[CLS]`. The head entity is represented as a tokens $x_1^h, \ldots, x_a^h$, and similarly for the relation and tail entities. The input tokens can therefore be represented as $X = \{x_1^h, \ldots, x_a^h, x_1^r, \ldots, x_b^r, x_1^t, \ldots, x_c^t\}$, where a, b, c are the lengths of head, relation, and tail entities. Additionally, the entities and relations are separated by a special token `[SEP]`.

Note that different elements separated by `[SEP]` have different segment embeddings: the tokens head and tail entities share the same segment embedding $\mathbf{e}_A$, while the tokens in relation have another segment embedding $\mathbf{e}_B$. For token x_i^h in head entity, we construct its input representation as $\mathbf{E}_i^h = \mathbf{x}_i^h + \mathbf{p}_i^h + \mathbf{e}_A$, where $\mathbf{x}_i^h$ and $\mathbf{p}_i^h$ are the token and position embeddings. After constructing all input representations, we feed them into a stack of L Transformer encoders (Vaswani et al., 2017) to encode the path and obtain:

$$w\mathbf{T}_i^h = \text{Transformer}(\mathbf{E}_i^h)$$

The final hidden states $\mathbf{T}_i^h \in \mathbb{R}^H$ are taken as the desired representations for entities and relations within X, where H is the hidden state size. These representations are naturally contextualized, and automatically adaptive to the input.

Afterwards, the encoding model is retrained with two tasks: predicting a masked ingredient entity and predicting the plausibility of a triple.

Predicting a masked ingredient entity

During training, for each input path $X = \{x_1^h, \ldots, x_a^h, x_1^r, \ldots, x_b^r, x_1^t, \ldots, x_c^t\}$, we create the training instance by replacing the head entity or tail entity with a special token `[MASK]` if it is an ingredient. Then, the masked sequence is fed into the Transformer encoding blocks. The final hidden state corresponding to `[MASK]` is used to predict the target entity:

$$\mathbf{u}^t = \text{softmax}(W_2 \cdot \text{Feedforward}(\mathbf{T}^t))$$

where $W_2 \in \mathbb{R}^{V \times H}$ is a trainable parameter, V is the entity vocabulary size, $\mathbf{u}^t$ is the predicted distribution of $t = \{x_1^t, \cdots, x_c^t\}$ over all ingredients. Here we only do masked ingredient entity prediction because the vocabulary size of recipes is too large for training. We compute a cross-entropy loss over the one-hot label $\mathbf{y}^t$ and the prediction $\mathbf{u}^t$:

$$\mathcal{L}_1 = -\sum_i^V y_i^t \log(u_i^t)$$

Predicting the plausibility of a triple

Given triples that reveal rich graph structures, similar to knowledge graph embeddings (Ji et al., 2021), the second training task is to predict the plausibility of the triples. The final hidden state of $\mathbf{T}_{[CLS]}$ is used as the aggregate path representation for computing triple scores. The scoring function $f_r(h,t)$ for a triple $\tau = (h,r,t)$ is defined as:

$$s_\tau = f_r(h,t) = \text{sigmoid}(\mathbf{T}_{[CLS]} W^T)$$

where $W \in \mathbb{R}^{1 \times H}$ is a trainable parameter and $s_\tau \in [0,1]$ is the triple plausibility score. Given the positive triple set $\mathbb{D}^+$ and a negative triple set $\mathbb{D}^-$, we compute the cross-entropy loss with s_τ and triple labels:

$$\mathcal{L}_2 = -\sum_{\tau \in \mathbb{D}^+ \cup \mathbb{D}^-} (y_\tau \log(s_\tau) + (1-y_\tau)\log(1-s_\tau))$$

where $y_\tau \in \{0,1\}$ is the triple label. The negative triple set $\mathbb{D}^-$ is simply generated by replacing head entity h or tail entity t in a positive triple $(h,r,t) \in \mathbb{D}^+$ with a random entity, that is, via negative sampling.

3.2 Generating Food Substitutions

After training the knowledge graph embedding model, we conduct attacks to generate feasible adversarial samples as recipe, ingredient and generalized ingredient substitutions, respectively, with three different attack strategies.

3.2.1 Problem Formulation

We utilize an attack model to find vulnerable tokens in KG triples $\tau = (h,r,t)$ and replace them with generated substitutions that maximize the risk of making wrong assertions on a target model. Here we assume it is a KG triple plausibility classifier $f_r(h,t)$ since we have used it in our preceding KGE model.

An adversarial entity t' is supposed to modify the text in t to trigger an error in the target model $f_r(h,t)$. For simplicity, we assume the tail entity t (it can also be the head entity h and recipe entities are always in the head of triples) is formatted as $t = \{x_1, \ldots, x_i, \ldots, x_c\}$. At the same time, perturbations on t should be minimal, such that t' is close to t.

There are lots of efforts being devoted to generating adversarial examples with various textual attack models on BERT (Jin et al., 2020; Li et al., 2020; Wang et al., 2021; Li et al., 2021; Garg and Ramakrishnan, 2020). The **mask-then-infill** perturbation approach (Li et al., 2020, 2021; Garg and Ramakrishnan, 2020) is widely-adopted. The approach usually chooses a masked language model as the attack model to find the vulnerable tokens in entities and replace them with adversarial sample. Specifically, we replace x_i in t with `[MASK]`, thus having $\hat{t} = \{x_1, \ldots, [\text{MASK}], \ldots, x_c\}$. We then select a token z to fill in, obtaining $t' = \{x_1, \ldots, z, \ldots, x_c\}$. Intuitively, the substitute token z is often constrained by three conditions:

i) z receives a high probability from the masked language model so it can smoothly fit into the original context; we regulate it by adding a condition $p_{MLM}(z|(h,r,\hat{t})) > k$.

ii) t' should be semantically similar to t, $\text{sim}(\mathbf{t'}, \mathbf{t}) > d$, where $\text{sim}(\mathbf{t'}, \mathbf{t})$ denotes the cosine similarity between representations of $\mathbf{t'}$ and $\mathbf{t}$.

iii) When placing t' in the retrained BERT model for KG triple plausibility classification, $f_r(h,t')$ yields low probability for the gold label y_τ which indicates that t' can trigger an error in the target model.

Under the attack theory, it might seem contradictory to treat t' as a food substitution, given that the triple (h,r,t') is less plausible in the KG. However, our assumption is the food KG is sparse (which it is in practice). The plausibility of the triple

formed from food substitution cannot be a standard to judge the quality of the substitution, since it can be a potential triple missed in the KG. Thus, a better gauge of the plausibility is based on the semantic similarity of the substitution or human evaluation, as done in our experiments.

3.2.2 Recipe Substitution Generation

Since recipes are usually short phrases, instead of mask-then-infill permutation, we consider more flexible actions to generate adversarial samples by *replacing*, *adding*, and *deleting* tokens. Given $t = \{x_1, \ldots, x_i, \ldots, x_c\}$, for the *replace* action, we have $\hat{t} = \{x_1, \ldots, x_{i-1}, \texttt{[MASK]}, x_{i+1}, \ldots, x_c\}$ by replacing x_i with `[MASK]`. For the *add* action, we have $\hat{t} = \{x_1, \ldots, x_{i-1}, \texttt{[MASK]}, x_i, \ldots, x_c\}$ by adding `[MASK]` before x_i. For the *delete* action, we have $\hat{t}= \{x_1, \ldots, x_{i-2}, \texttt{[MASK]}, x_{i+1}, \ldots, x_c\}$ by replacing $x_{i-1}x_i$ with `[MASK]`. For example, given a recipe entity "*blue cheese-stuffed potatoes with buffalo chicken tenders*", it can be formulated as "*blue cheese-stuffed potatoes with buffalo* `[MASK]` *tenders*", "*blue cheese-stuffed potatoes with buffalo* `[MASK]` *chicken tenders*", and "*blue cheese-stuffed potatoes with* `[MASK]` *tenders*" according to the *replace*, *add*, and *delete* actions.

For every $\hat{t}$ obtained from the above three actions, we estimate the action score by computing the decrease in probability of predicting the correct label y_τ. The action score I_i is defined as:

$$I_i = o_{y_\tau}((h, r, t)) - o_{y_\tau}((h, r, \hat{t}))$$

where $o_{y_\tau}(\cdot)$ denotes the logit output by the target model for correct label y_τ.

To conduct the attack on BERT, we sequentially apply this attack strategy over t until an adversarial example t' is found or a limit of permutation action M is reached. We filter the set of top K tokens (K is a pre-defined constant) predicted by the masked language model for the masked token according to condition ii). To represent $\mathbf{t}$ and $\mathbf{t}'$, previous work in textual adversarial attack often uses the universal sentence encoder (Cer et al., 2018). Here we adopt pretrained recipe embeddings (Li and Zaki, 2020) to calculate $\text{sim}(\mathbf{t}', \mathbf{t})$ because it is trained on recipe corpus, preserving stronger representational ability for recipe data.

3.2.3 Ingredient and Generalized Ingredient Substitution Generation

Different from recipes, most ingredients only consist of 1-3 words. The plausibility of generated ingredient substitutions is vital in our task. Therefore, we conduct entity-level perturbation on KG triples. We reuse the masked BERT model in Section 3.1 to detect vulnerable entities and suggest candidate ingredients. The attack process is similar to the attack on recipes. For instance, "*mozzarella cheese*" can be substituted with "*cream cheese*" in triple (*Philly cheese steak pizza, consist_of, mozzarella cheese*), where "*cream cheese*" is picked from the ingredient vocabulary. The ingredient generated in such a way can provide reasonable substitution for a particular recipe when recipe and ingredient make up the head and tail entities in a KG triple (h, r, t).

Moreover, we introduce a new attack strategy to produce more *generalized* ingredient substitutions since there are also many scenarios asking for ingredient substitution for general purpose without any context. Given an ingredient entity t, we retrieve its neighbors $\mathcal{N}^t$ in KG and form N triples $\{(h, r, t)|h \in \mathcal{N}^t\}$, note that a neighbor entity can also be a tail entity t in this triple set, we denote it as h for simplicity. Then, we obtain a candidate ingredient set $\mathcal{Z}$ via our pretrained masked BERT model. For every ingredient candidate z in $\mathcal{Z}$, we iteratively apply attack over $f_r(h, t)$ and record the attack success rate α until it reaches a threshold determined by βN (β is a pre-defined constant). Since the adversarial attack is conducted among all t's neighbor, a successful attack is achieved only when the adversarial sample t' fools most of its neighbors $\mathcal{N}^t$. Therefore, the t' is regulated by $\mathcal{N}^t$, preventing it to be contextualized to any specific neighbor.

An an example of generalized substitution, given an ingredient entity "*couscous*", we first retrieve all its neighbors in the food KG, forming a triple set $\{(h, r, t)|h \in \mathcal{N}^t\}$. The masked language model suggests {"*quinoa*", "*sorghum*", "*millet*", $\cdots$} as the candidate substitution set. When conducting the adversarial attack, "*quinoa*" successfully attacks the target model $f_r(h, t')$ over βN times, thus we take "*quinoa*" as the generalized substitution of "*couscous*". Comparing to other candidates, triple (*pesto chicken wrap with sun dried tomatoes*, *consist_of*, *quinoa*) triggers an error in triple plausibility prediction, whereas triples (*pesto chicken wrap with sun dried tomatoes*, *consist_of*, *sorghum*) and (*pesto chicken wrap with sun dried tomatoes*, *consist_of*, *millet*) are predicted as true. Engaging more entity neighbors from the KG to

conduct attacks makes the final substitution more generic.

4 Experiments

4.1 Dataset and Experimental Setup

We use the FoodKG (Haussmann et al., 2019) knowledge graph as the main source for KGE and food substitutions due to its rich structured knowledge of recipes with ingredients. The FoodKG contains food-relevant instances including recipe and ingredient information extracted from Recipe1M (Marin et al., 2019). We extract 4 million triples from FoodKG and randomly divide them into training, validation, and test datasets according to the ratio of 8:1:1. The BERT-base model is used to the encode the KG and generate substitutions, which is implemented with Hugging Face transformers (`github.com/huggingface/transformers`). More experimental details are given in Appendix A.1. Our code is publicly available at `https://github.com/DiyaLI916/FoodKGE`.

4.2 Knowledge Graph Embedding Results

We compare our BERT-based KGE model with some typical KGE methods with regards to encoding models, including:

- ***Linear models***: TransE (Bordes et al., 2013) and TransR (Lin et al., 2015). TransE learns vector representations of h, t, and r following the translational principal $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. TransR further introduces separated spaces for entities and relations to tackle the problem of insufficiency of a single latent space for both entities and relations.

- ***CNN/GNN models***: ConvE (Dettmers et al., 2018) and R-GCN (Schlichtkrull et al., 2018). ConvE uses 2-D convolution over embeddings and multiple layers of nonlinear features to model the interactions between entities and relations. R-GCN encodes KGs with graph convolutional networks and addresses the multi-relational data characteristic of KG by reshaping head entity and relation into a 2-D matrix.

- ***Transformer-based models***: KG-BERT (Yao et al., 2019) and CoKE (Wang et al., 2019). KG-BERT borrows the idea from language model pre-training and takes the BERT model as an encoder for entities and relations. Similarly, CoKE employs a stack of transformer blocks to encode edges and path sequences. In contrast, our KGE model has a multi-task training setting.

Metrics

Following the evaluation protocol of KGE models described in the previous works like Bordes et al. (2013), the performance of the KG representations are typically evaluated by two tasks: triple plausibility classification and entity linking prediction. Triple classification aims to judge whether a given triple (h, r, t) is correct or not, thus accuracy is reported in this task. It is in the same form as our training task of predicting the plausibility of a triple with negative sampling. The link prediction task aims to predict the head entity h given (?, r, t) or the tail entity t given (h, r, ?), where ? means the missing entity. Here, we only do prediction of ingredient entity. It is in the same form as our training task of predicting masked ingredient entities. For entity linking, we report MRR (Mean Reciprocal Rank of all the ground truth triples) and Hits@10 (the proportion of correct entities ranked in top 10, for all the ground truth entities) as our evaluation metrics. We only report results under the filtered setting (Bordes et al., 2013) which removes all corrupted triples that appear in training, validation, and test set before getting the ranking lists.

Table 1: Knowledge graph embedding results on triple plausibility classification and link prediction tasks. Higher is better. All scores are statistically significant at $p < .01$ employing a two-sample t-test.

	Triple Plausibility Accuracy	Link Prediction MRR	Link Prediction Hits@10
TransE	0.730	0.318	0.441
TransR	0.758	0.322	0.469
ConvE	0.836	0.402	0.517
R-GCN	0.814	0.350	0.482
KG-BERT	0.893	0.417	0.521
CoKE	0.872	0.451	0.540
Our model	**0.916**	**0.460**	**0.549**

Results and Analysis

The results of the two tasks on FoodKG are shown in Table 1. The linear models (TransE/TransR) do not achieve high scores in triple classification and link prediction tasks. Even though TransR alleviates the problem of TransE in dealing with multiple relations, the improvement in TransR is slight because the relation types in FoodKG is very small. TransR projects head and tail entities into relation space by a projection matrix. However, for most

triples in FoodKG, head and tail entities are of different types. ConvE shows decent results, which suggests that CNN models can capture global interactions among the entity and relation embeddings by nonlinear feature learning through multiple layers. Though R-GCN emphasizes the graph structure and the multi-relational data characteristic of KG, R-GCN performs worse than ConvE due to the scarce relation types in FoodKG.

For the two transformer-based models, KG-BERT is particularly trained on the triple classification task, thus achieving a higher score in triple plausibility prediction. The CoKE model formulates multi-hop paths in the KG into sequences consisting of entities and relations. The model is trained to predict masked entities and relations and improves the multi-hop reasoning ability in KG, resulting in higher scores in link prediction task. Our model outperforms all the competitive baselines in these two evaluation tasks, and the improvements are statistically significant ($p < 0.01$). This demonstrates the superiority of our two-stage training strategy which explicitly captures the contextual information to help the triple fact assertion and is also powerful in single-hop reasoning.

4.3 Adversarial Attack Results on BERT

We compare our method with recent state-of-the-art adversarial attack methods against pre-trained language models as follows:

- ***BERT-Attack*** (Li et al., 2020): This model proposes a typical mask-then-infill approach which greedily replaces tokens with the predictions from BERT.
- ***BAE*** (Garg and Ramakrishnan, 2020): Similar to BERT-Attack, while BAE allows adding a token via perturbation.
- ***CLARE*** (Li et al., 2021): This model proposes three contextualized perturbations – Replace, Insert and Merge – that allow for generating different lengths of adversarial samples.

Metrics

We follow previous work on textual adversarial attack (Jin et al., 2020; Li et al., 2020), and adopt three metrics to automatically evaluate the attacking results: i) the attack success rate, representing the percentage of adversarial examples that can successfully attack the target model, ii) the perturbation rate, denoting the percentage of modified tokens, and iii) the textual similarity, computed as the cosine similarity between the representations of original entity and the alternative, as described in Section 3.2.

Table 2: Adversarial example generation performance in attack success rate (Attack), perturbation rate (Perturb), and textual similarity (Similarity). Best results are marked in bold. For Attack and Similarity, higher is better; for Perturb lower is better. All scores are statistically significant at $p < .01$ employing a two-sample t-test.

	Recipe Substitution		
	Attack	**Perturb ↓**	**Similarity**
BERT-attack	77.5	69.5	0.74
BAE	78.3	69.0	0.75
CLARE	80.6	**67.3**	0.82
Our model	**80.9**	67.7	**0.83**
	Ingredient Substitution		
	Attack	**Perturb ↓**	**Similarity**
BERT-attack	75.1	93.1	0.79
BAE	74.8	**90.7**	0.81
CLARE	81.3	94.3	0.82
Our model	**84.4**	100	**0.85**
	Generalized Ingredient Substitution		
	Attack	**Perturb ↓**	**Similarity**
Our model	**67.8**	100	**0.86**

Results and Analysis

We perform adversarial attacks on our KGE model and summarize the results in Table 2. Across models, our attack strategies are almost always more effective than the three baseline attack methods, achieving the highest average attack success rate and textual semantic similarities. Though the perturbation rate is widely-used to evaluate textual attack methods, where a lower perturbation rate is better; our goal is to generate high quality food substitutions, thus perturbation rate is not as important in our task. We do entity-level replacement for ingredient substitutions, therefore the perturbation rate is 100% in our cases.

We observe that BERT-attack and BAE models have close performance. BERT-attack only replaces tokens. BAE allows adding a token while it inserts only near the replaced token, thus limiting its attacking capability. CLARE uses three different perturbations (Replace, Insert and Merge), each allowing efficient attacking against any position of the input, and can produce outputs of varied lengths. Our model's attack strategy is similar to CLARE for recipe substitution, with a different action scoring function. It is reasonable that CLARE performs close to our model.

For ingredient substitution, the three baselines

Table 3: Human evaluation performance. Scores are based on a 5-point scale.

	Recipe Substitution		
	Original	CLARE	Ours
Appropriateness	4.37	4.18	4.22
Grammar	4.76	4.30	4.36
Semantic	-	3.51	3.65
	Ingredient Substitution		
	Original	CLARE	Ours
Appropriateness	4.67	4.52	4.60
Semantic	-	4.50	4.55
	Generalized Ingredient Substitution		
	Shirai et al. (2020)		Ours
Semantic	4.53		4.46

focus on token-level perturbation since they are proposed for textual adversarial attack. In contrast, we aim to generate different kinds of food substitutions over KG. Our model directly does entity-level perturbation for ingredient substitution, and outperforms all the baselines by a big margin. Besides, we also create an additional strategy to do generalized ingredient substitution by employing ingredient's neighbors in the KG to regulate its contextualization property. The new attack strategy achieves a high score of 0.86 in textual similarity.

Human Evaluation

It is important to note that our main focus is not purely on successful attacks, but rather on the quality of generated samples. Therefore, to further examine the quality of the food substitutions and compare with previous adversarial attack work CLARE (Li et al., 2021) and food substitution work (Shirai et al., 2020), we conduct a human evaluation study on 150 food substitutions. Specifically, we randomly selected 50 recipe substitutions and 50 ingredient substitutions which our model and CLARE successfully attack on the test dataset, and 100 generalized ingredient substitutions which our model successfully attacks (note that previous attack models cannot produce generalized ingredient substitutions). We recruited 10 annotators to evaluate the three types of food substitutions. For *recipe substitutions*, the recipe along with its ingredients are presented to the evaluators, who are requested to give scores on a 5-point scale (1-bad, 2-poor, 3-fair, 4-good, 5-excellent) in terms of three aspects: i) Appropriateness: recipe substitution appropriateness with regards to its ingredients; ii) Grammar: grammatical correctness of the substitute; and iii) Semantic: semantic similarity between the original recipe and its substitute as there is no ground truth for recipe substitution. The human evaluation for *ingredient substitutions* has a similar setting, but we do not assess the grammatical aspect because we do entity-level substitutions with new ingredients picked directly from the vocabulary. Shirai et al. (2020) has created a ground truth dataset for *generalized ingredient substitutions*. Thus, we evaluate the semantic similarity between the ground truth ingredient substitution and the substitutes provided in Shirai et al. (2020)'s work and the *generalized substitute* generated from our adversarial model.

We compute the Fleiss's kappa coefficient to measure the agreement among evaluators, and the agreement score is 0.61, indicating moderate agreement. As shown in Table 3, for recipe substitution, the appropriateness and grammar scores of the adversarial samples are close to the original ones, indicating the high quality of these substitutions. The appropriateness score for ingredient substitution is very close to the original ingredients (4.67 vs. 4.73). This implies that the generated ingredient samples can be good substitutes with regards to their corresponding recipes. Our generated recipe and ingredient substitutions also achieve higher scores across all the three aspects when compared to CLARE. The semantic score of our generalized ingredient substitutions is close to Shirai et al. (2020)'s work which leverages various semantic sources and rules (4.46 vs. 4.53). In contrast with Shirai et al. (2020)'s work, our model automatically suggests generalized ingredient substitutions without the need for human-crafted features and rules.

Qualitative Analysis

In order to have a deep understanding of the adversarial samples, we conduct qualitative analysis over the three types of food substitutions. We observe the following:

- ***Recipe substitution:*** i) We have three perturbation actions during recipe substitution generation process. We calculate the action scores of these three and do perturbation according to the action with the highest score. In our final results, the **replace** action occurs most, accounting for 74.5% of the entire recipe substitutions. The noun token in recipes has a higher chance to be detected as a vulnerable token. The **delete** action often results in merging two noun tokens into one and the **add** action tend to insert tokens into noun phrase bi-grams. Table 4 lists some examples of

Table 4: Recipe substitution examples produced by our attack model. The token marked in red and blue are the vulnerable and generated ones, respectively.

Recipe	Action	Recipe Substitution
the sweetest blueberry muffins	replace	the sweetest cranberry muffins
spicy shrimp in coconut milk	delete	spicy shrimp in milk
banana cream muffins	add	tropical banana cream muffins
monterey jack chicken: bursting with flavor	replace	gouda jack chicken: bursting with flavor

Table 5: Ingredient substitution examples.

KG Triple	Ingredient Substitution
(chicken salad roll-ups appetizer, *consist_of*, poppy seed dressing)	sesame seed dressing
(beetroot yogurt, *consist_of*, beet)	carrot
(authentic Russian borscht, *consist_of*, beet)	turnip

Table 6: Generalized ingredient substitution examples.

Ingredient	Generalized Substitutions
milk	soy milk
kale	broccoli
grapefruit	lime
currant	cranberry
nutmeg	cinnamon
walnut	almond
green onion	garlic
arugula	lettuce

the three actions. For example, the token "*blueberry*" in recipe "*the sweetest blueberry muffins*" listed in Table 4 is replaced by "*cranberry*". ii) Semantic and grammatical errors often occur in recipe substitutions with long text. For instance, the token "*monterey*" in "*monterey jack chicken: bursting with flavor*" is replaced by "*gouda*" in Table 4. "*Monterey jack*" refers to the A*merican cheese Monterey Jack*, while "*gouda jack*" does not make sense in this substitution.

- ***Ingredient substitution:*** i) Rare ingredients with low frequency in the ingredient vocabulary (occurring less than 50 times in all triples) tend to be detected as vulnerable and are replaced by more common ones. As demonstrated in Table 5, "*poppy seed dressing*" is substituted by "*sesame seed dressing*" in "*chicken salad roll-ups appetizer*". This can be useful in practice, since people often ask for a substitute when an ingredient is not at hand. ii) Most ingredients are suggested different substitutions in different recipes. As shown in Table 5, "*beet*" is substituted by "*carrot*" in dessert "*beetroot yogurt*", whereas "*turnip*" is suggested to replace "*beet*" in main dish "*authentic Russian borscht*".

- ***Generalized ingredient substitution:*** We report some generalized ingredient substitutions that have successfully attacked the KGE model over 100 times. The results are listed in Appendix, Table 6. The substitutions are in line with human common sense. For example, "*milk*" may be substituted by "*soy milk*" in general over several recipes. Likewise, "*almond*" can be a substitute for "*walnut*". Thus, our generalized substitution approach can serve as a reasonable reference in applications where users seek ingredient substitutions for general purposes.

5 Conclusion and Future Work

In this work, we proposed a novel framework to learn food KG embeddings via a pre-trained language model and generate high quality food substitutions by conducting attacks in the language model. Specifically, we addressed the sparseness problem in food KG and enriched its contextualized representation via the retraining of BERT model on two tasks. We then employed a masked language model to iteratively generate feasible food substitutions via adversarial attacks on KGE. We further invented a collection of attack strategies to generate three types of food substitutions to meet different user needs: namely, contextualized recipe and ingredient substitutions for substitution queries with a given context, and generalized ingredient substitutions for general substitution purpose. For future work, we aim to take the health or nutrition information into consideration during adversarial sample generation, thus guiding healthier dietary choices for people.

Acknowledgements

This work is supported by IBM Research AI through the AI Horizons Network (under the RPI-IBM HEALS Project).

References

Sema Akkoyunlu, Cristina Manfredotti, Antoine Cornuéjols, Nicolas Darcel, and Fabien Delaere. 2017. Investigating substitutability of food items in consumption data. In *Second International Workshop on Health Recommender Systems (co-located with ACM RecSys)*, volume 5.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735. Springer.

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

Stephen Bonner, Ian P Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, and William L Hamilton. 2021. Understanding the performance of knowledge graph embeddings in drug discovery. *arXiv preprint arXiv:2105.10488*.

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

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Yu Chen, Ananya Subburathinam, Ching-Hua Chen, and Mohammed J Zaki. 2021. Personalized food recommendation as constrained question answering over a large-scale food knowledge graph. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 544–552.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 4171–4186.

Leonard H Epstein, Sarah J Salvy, Katelyn A Carr, Kelly K Dearing, and Warren K Bickel. 2010. Food reinforcement, delay discounting and obesity. *Physiology & Behavior*, 100(5):438–445.

Emmanuelle Gaillard, Jean Lieber, and Emmanuel Nauer. 2015. Improving ingredient substitution using formal concept analysis and adaptation of ingredient quantities with mixed linear optimization. In *Computer Cooking Contest Workshop*.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *International Conference on Machine Learning*, pages 2505–2514. PMLR.

Steven Haussmann, Oshani Seneviratne, Yu Chen, Yarden Ne'eman, James Codella, Ching-Hua Chen, Deborah L McGuinness, and Mohammed J Zaki. 2019. Foodkg: a semantics-driven knowledge graph for food recommendation. In *International Semantic Web Conference*, pages 146–162. Springer.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069.

Diya Li and Mohammed J Zaki. 2020. Reciptor: An effective pretrained model for recipe representation learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1719–1727.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Yankai Lin, Xu Han, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2018. Knowledge representation learning: A quantitative review. *arXiv preprint arXiv:1812.10901*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 43(1):187–203.

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

Weiqing Min, Shuqiang Jiang, and Ramesh Jain. 2019. Food recommendation: Framework, existing solutions, and challenges. *IEEE Transactions on Multimedia*, 22(10):2659–2671.

Sameh K Mohamed, Aayah Nounu, and Vít Nováček. 2021. Biological applications of knowledge graph embedding models. *Briefings in Bioinformatics*, 22(2):1679–1693.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2*, pages 327–333, New Orleans, Louisiana.

Yuran Pan, Qiangwen Xu, and Yanjun Li. 2020. Food recipe alternation and generation with natural language processing techniques. In *2020 IEEE 36th International Conference on Data Engineering Workshops*, pages 94–97. IEEE.

Li Qin, Zhigang Hao, and Liang Zhao. 2019. Food safety knowledge graph and question answering system. In *Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City*, pages 559–564.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3060–3067.

Sola S Shirai, Oshani Seneviratne, Minor E Gordon, Ching-Hua Chen, and Deborah L McGuinness. 2020. Identifying ingredient substitutions using a knowledge graph of food. *Frontiers in Artificial Intelligence*, 3.

Kari Skjold, Marthe Øynes, Kerstin Bach, and Agnar Aamodt. 2017. Intellimeal-enhancing creativity by reusing domain knowledge in the adaptation process. In *International Conference on Case-Based Reasoning Workshops*, pages 277–284.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Dong Wang, Ning Ding, Piji Li, and Hai-Tao Zheng. 2021. Cline: Contrastive learning with semantic negative examples for natural language understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 2332–2342.

Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, and Hua Wu. 2019. Coke: Contextualized knowledge graph embedding. *arXiv:1911.02168*.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

A Appendix

A.1 Experimental Setup

We use the following configuration for KG encoding: the number of Transformer layers: 12, number of self-attention heads: 12, and hidden size: 256. We choose BERT-base model instead of BERT-large because it achieves better results in triple plausibility classification, and the former is less sensitive to hyper-parameter choices. We employ dropout on all layers, with a 0.1 dropout rate.

Table 7: Parameter settings in BERT attack.

Parameter	Value
Recipe and Ingredient Substitution	
k	1e-2
d	0.6
M	30
K	20
Generalized Ingredient Substitution	
k	1e-2
d	0.75
K	10
β	0.2

We retrain the BERT model with batch size of 64 for at most 20 epochs, and use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 5e-5. The best hyper-parameter setting is determined by the validation set. For triple plausibility classification training, we sample one negative triple for every positive triple, which ensures class balance in binary classification. The parameter choices of the adversarial attacks on BERT are listed in Table 7. k is the learning rate and d is the dropout rate. For recipe and ingredient substitution generation, M is the maximum permutation actions to try for each attack and K is the filtered top K tokens predicted by the masked language model. β is the threshold rate to determine a successful attack in generalized ingredient substitution generation.