Towards Knowledge-Based, Robust Question Answering

by

So Yeon Min

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

Author .....
Department of Electrical Engineering and Computer Science
May 17, 2020

Certified by .....
Peter Szolovits
Professor
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Towards Knowledge-Based, Robust Question Answering

by

So Yeon Min

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2020, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Current question answering systems face two major challenges; the ability to employ external knowledge and to robustly generalize to unseen expressions of questions need to be improved. In this thesis, I introduce two works that can together help advance question answering. First, I introduce TransINT, a novel and interpretable knowledge graph embedding method that isomorphically preserves the implication ordering among relations in the embedding space. Second, I present methods to train sequence-to-sequence semantic parsing models robust to unseen paraphrases. These two works could together serve as steps to create human-like question answering systems that can understand unseen paraphrases and link existing and external facts for logical inference.

Thesis Supervisor: Peter Szolovits
Title: Professor

Acknowledgments

I deeply thank my supervisor, Professor Peter Szolovits, who is both incredibly bright and extremely kindhearted. It was very fortunate of me to work with an exceptional person like him; I will never be able to forget the day Pete has accepted me into his group.

I also thank Dr. Preethi Raghavan, who has sponsored this research with funding from the MIT-IBM Watson AI Lab. Extremely smart and patient, she has actively collaborated with me and has taught me many invaluable skills.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Near the end of 2018, my supervisor Professor Peter Szolovits and I were envisioning a project to combine two relational ontologies in the medical domain - the MRREL table of the Unified Medical Language System (UMLS) [7] and the relations of the i2b2 relations challenge dataset [52]. While these two sources of knowledge both contain clinical multi-relational facts (such as (abciximab,`may_treat`, Myocardial Ischemia)), some relations in the UMLS MRREL table subsume one in the i2b2 dataset (e.g. `may_treat` of UMLS MRREL implies `TrIP` (Treatment improves medical problem) of the i2b2 relations challenge). Thus, I wanted to create a seamless combination of two knowledge graphs embedded to $\mathbb{R}^d$, such that subsumptions of relations in the embedding space are identically represented as in the inclusion ordering in the human mind. While I could not achieve the original goal I had envisioned, I did devise a method to represent implication rules in knowledge graphs isomorphically in their embeddings.

On the other hand, with our collaborator Dr. Preethi Raghavan, I had been working on making a family of models for semantic parsing, the task of transforming natural language utterances into uniquely and exactly identified expressions, generalizable to unseen paraphrases. For example, consider a physician using a question answering system to query patient notes; one may choose to phrase one's information need in a manner that the system has not observed before. A reliable system should be able to respond appropriately to a paraphrase of the question. Thus, we developed

a paraphrase-robust semantic parsing mechanism.

While these two projects are fairly orthogonal, they are both useful for the common goal of knowledge-based and robust question answering (QA). Current question answering systems face two major challenges; the ability to employ external knowledge and to robustly generalize to unseen expressions of questions need to be improved.

Questions asked by humans often presume common sense or logical knowledge about the world, which may not exist in the database to be queried. For example, humans subconsciously assume that "bananas" are "yellow" (common sense) and a father is a parent (logical implication), but current QA models seldom address such external knowledge, if not contained in the query database itself. Recently, works such as ERNIE [62] have attempted to merge knowledge graph embeddings with existing language models and have shown that such a method significantly improves performance on various downstream natural language tasks. Thus, my work in knowledge graph representation is a stepping stone to benefit knowledge-based question answering.

Another important challenge in QA is improving model robustness to paraphrases, which is the central problem of my project in semantic parsing. Semantic parsing is a key component of structured question answering, in that it translates human questions into executable queries that can later retrieve answers from a database. Because semantic parsing is closely related to question understanding, my semantic parsing project adequately addresses the paraphrase-robustness problem of current QA models. Thus, we hope that the two projects that I have pursued during my master's candidacy become fruitful steps towards building knowledge-based and robust question answering system.

In Chapter 2, I introduce TransINT, a novel and interpretable knowledge graph embedding method that isomorphically preserves the implication ordering among relations in the embedding space. In Chapter 3, I describe methods to train sequence-to-sequence semantic parsing models robust to unseen paraphrases. In Chapter 4, I conclude the thesis and suggest future work.

Chapter 2

TransINT: Embedding Implication Rules in Knowledge Graphs with Isomorphic Intersections of Linear Subspaces

Knowledge Graphs (KG), composed of entities and relations, provide a structured representation of knowledge. For easy access to statistical approaches on relational data, multiple methods to embed a KG into $f(\text{KG}) \in \mathbb{R}^d$ have been introduced. Logical rules are known to enhance knowledge graph embeddings by eliminating unwanted solutions. Implication ordering, such as $is\_father\_of \Rightarrow is\_parent\_of$, is one of the most common types of rules. We propose TransINT, a novel and interpretable KG embedding method that isomorphically preserves the implication ordering among relations in the embedding space. Given implication rules, TransINT maps sets of entities (tied by a relation) to continuous sets of vectors that are inclusion-ordered isomorphically to relation implications. With a novel parameter sharing scheme, TransINT enables automatic training on missing but implied facts without rule grounding. On two benchmark datasets, we outperform the best existing state-of-the-art rule integration embedding methods with significant margins in link prediction and triple

classification. The angles between the continuous sets embedded by TransINT provide an interpretable way to mine semantic relatedness and implication rules among relations.

2.1 Introduction

Learning distributed vector representations of multi-relational knowledge is an active area of research [8, 42, 31, 59, 9]. These methods map components of a KG (entities and relations) to elements of $\mathbb{R}^d$ and capture statistical patterns, regarding vectors close in distance as representing similar concepts. The use cases of such embeddings include detection of absent edges, discovery of nodes' properties, and clustering nodes by their connectivity with other nodes. Particularly, because many large-scale knowledge bases are far from complete [40, 7], multi-relational embeddings can be useful in automatic knowledge base completion — automatically inferring missing facts from existing ones.

However, current KG embedding approaches, whose only concerns are to simply learn embeddings compatible with facts in the given KG, come with limitations. Because KG's are largely incomplete, only requiring compatibility with existing facts can lead to inference of wrong facts when the learned embeddings are applied to downstream tasks. For example, if the relation *has_spouse* appears infrequently in the KG, the embedding may not learn that *has_spouse* only holds between entities of *person* types, predicting incorrect facts such as *(Iphone 7, has_spouse, Ipad Pro)*. Integration of rules that declare constraints on *has_spouse* would eliminate clearly undesirable solutions in a large solution space of the embeddings, which would result in fewer wrong predictions.

Thus, one focus of current research is to bring logical rules to KG embeddings [27, 56, 60]. While existing methods impose hard geometric constraints and embed asymmetric orderings of knowledge [41, 53, 54], many of them only embed hierarchy (unary *Is_a* relations), and cannot embed binary or n-ary relations in KG's. On the other hand, other methods that integrate binary and n-ary rules [27, 21, 45, 13]

do not come with empirical results that support their significant effects in reducing logically wrong predictions.

We propose TransINT, a new and extremely powerful KG embedding method that isomorphically preserves the implication ordering among relations in the embedding space. Given pre-defined implication rules, TransINT restricts entities tied by a relation to be embedded to vectors in a particular region of $\mathbb{R}^d$, included isomorphically by the order of relation implication. For example, we map any entities tied by *is_father_of* to vectors in a region that is included in the region for *is_parent_of*; thus, we can automatically know that if John is a father of Tom, he is also his parent even if such a fact is missing in the KG. Such embeddings are constructed by sharing and rank-ordering the basis of the linear subspaces in which the vectors are required to exist. The parameter sharing of our method refines the large solution space of KG embeddings, which are learned with stochastic gradient descent, starting from random initialization.

Mathematically, a relation can be viewed as a set of entities tied by a constraint [47]. We take such a view on KG's, since it gives consistency and interpretability to model behavior. We show that angles between embedded relation sets can identify semantic patterns and implication rules — an extension of the line of thought as in word/ image embedding methods such as [36] and [24] to relational embedding.

The main contributions of our work are: (1) A novel KG embedding such that implication rules in the original KG are guaranteed to unconditionally, not approximately, hold. (2) Our model suggests possibilities of learning semantic relatedness between groups of objects. (3) We significantly outperform state-of-the-art rule integration embedding methods, [27] and [21], on two benchmark datasets, FB122 and NELL Sport/Location.

2.2 TransINT

In this section, we describe the intuition and justification for our method. We first define relation as sets, and revisit TransH [59] as mapping relations to sets in $\mathbb{R}^d$.

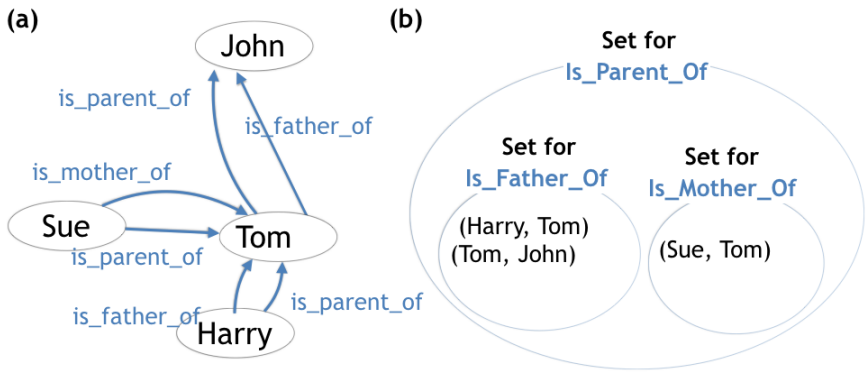

Figure 2-1: Two equivalent ways of expressing relations. (a): relations defined in a hypothetical KG. (b): relations defined in a set-theoretic perspective (Definition 1). Because *is_father_of* $\Rightarrow$ *is_parent_of*, the set for *is_father_of* is a subset of that for *is_parent_of* (Definition 2).

Finally, we propose TransINT. We put * next to definitions and theorems we propose/introduce. Otherwise, we use existing definitions and cite them.

2.2.1 Sets as Relations

We define relations as sets and implication as inclusion of sets, as in set-theoretic logic.

Definition *(Relation Set):* Let r_i be a binary relation and x, y entities. Then, a set $\mathbf{R_i}$ such that $r_i(x, y)$ if and only if $(x, y) \in \mathbf{R_i}$ always exists [47]. We call $\mathbf{R_i}$ the **relation set** of r_i.

For example, consider the relations in Figure 2-1a and their corresponding sets in Figure 2-1b; *Is_Father_Of(Tom, Harry)* is equivalent to *(Tom, Harry)* $\in \mathbf{R}_{\text{Is\_Father\_Of}}$.

Definition *(Logical Implication):* For two relations, r_1 implies r_2 (or $r_1 \Rightarrow r_2$) iff $\forall x, y$,

$$(x, y) \in \mathbf{R_1} \Rightarrow (x, y) \in \mathbf{R_2} \quad \text{or equivalently,} \quad \mathbf{R_1} \subset \mathbf{R_2}. \text{ [47]}$$

For example, *Is_Father_Of* $\Rightarrow$ *Is_Parent_Of.* (In Figure 2-1b, $\mathbf{R}_{\mathbf{Is\_Father\_Of}} \subset \mathbf{R}_{\mathbf{Is\_Parent\_Of}}$).

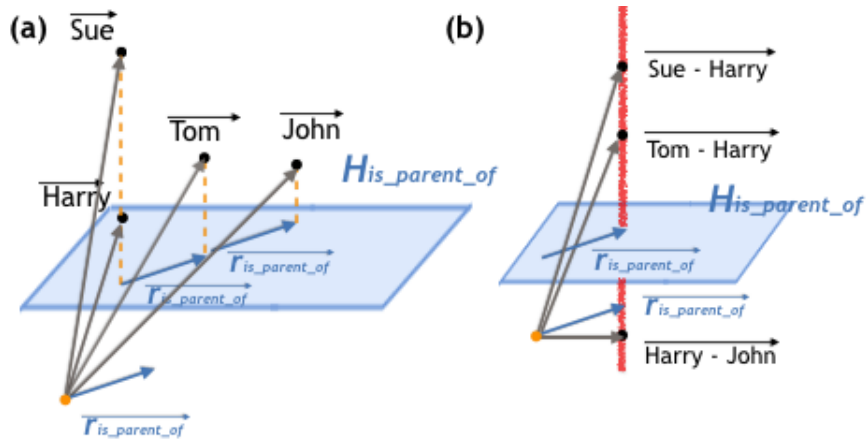

Figure 2-2: Two perspectives of viewing TransH in $\mathbb{R}^3$; order of operations can be flipped. (The orange dot is the origin, to emphasize that translated vectors are equivalent.) (a): projection first, then difference — first projecting $\vec{h}$ and $\vec{t}$ onto $H_{is\_parent\_of}$, and then requiring $\vec{h_\perp} + \vec{r_j} \approx \vec{t_\perp}$ (b): difference first, then projection — first subtracting $\vec{h}$ from $\vec{t}$, and then projecting the difference $(\overrightarrow{t-h})$ to $H_{is\_parent\_of}$ and requiring $(\overrightarrow{t-h})_\perp \approx r_j$. All $(\overrightarrow{t-h})_\perp$ belong to the red line, which is unique because it is when $\overrightarrow{r_{is\_parent\_of}}$ is translated to the origin.

2.2.2 Background: TransH

Given a fact triple (h, r_j, t) in a KG (i.e., *(Harry, is_father_of, Tom)*), TransH maps each entity to a vector, and each relation r_j to a relation-specific hyperplane H_j and a fixed vector $\vec{r_j}$ on H_j (Figure 2-2a). For each fact triple (h, r_j, t), TransH wants

$$\vec{h_\perp} + \vec{r_j} \approx \vec{t_\perp} \cdot \cdot \cdot \cdot \cdot \quad \text{(Eq. 1)}$$

where $\vec{h_\perp}, \vec{t_\perp}$ are projections of $\vec{h}$, $\vec{t}$ onto H_j (Figure 2-2a).

Revisiting TransH We interpret TransH in a novel perspective. An equivalent way to put Eq.1 is to change the order of subtraction and projection (Figure 2-2b):

$$\text{Projection of } (\overrightarrow{t-h}) \text{ onto } H_j \approx \vec{r_j}.$$

This means that all entity vectors $(\vec{h}, \vec{t})$ such that their difference $\overrightarrow{t-h}$ belongs to the red line are considered to be tied by relation r_j (Figure 2-2b); $\mathbf{R_j} \approx$ the red line, which is the set of all vectors whose projection onto H_j is the fixed vector $\vec{r_j}$. Thus, upon a deeper look, **TransH actually embeds a relation set in KG** (Figure 2-1b)

to a particular set in $\mathbb{R}^d$. We call such sets *relation space* for now; in other words, a *relation space* of some relation r_i is the space where each (h, r_i, t)'s $\overrightarrow{t-h}$ can exist. We formally visit it later in Section **3.1**. Thus, in TransH,

$$r_i(x,y) \equiv (x,y) \in \mathbf{R_i} \qquad \textbf{(relation in KG)}$$
$$\cong \overrightarrow{y-x} \in \textit{relation space of } r_i \quad \textbf{(relation in } \mathbb{R}^{\mathbf{d}}\textbf{)}$$

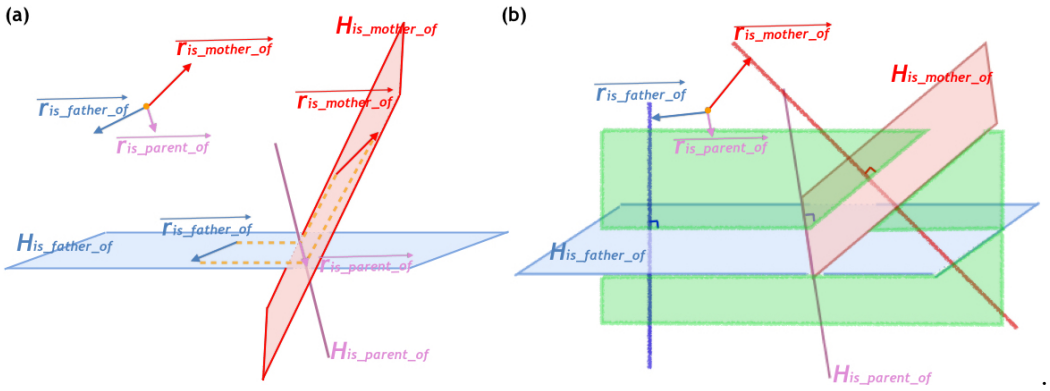

Figure 2-3: Two perspectives of viewing TransINT. (a): TransINT as TransH with additional constraints — by intersecting H's and projecting $\vec{r}$'s. The dotted orange lines are the projection constraint. (b): TransINT as mapping of sets (relations in KG's) into linear subspaces (viewing TransINT in the *relation space* (Figure 2-2b)). The blue line, red line, and the green plane are, respectively, *is_father_of*, *is_mother_of* and *is_parent_of*'s *relation spaces* — where $\overrightarrow{t-h}$'s of h, t tied by these relations can exist. The blue and the red line lie on the green plane — *is_parent_of*'s *relation space* includes the other two's.

2.2.3 TransINT

We propose TransINT, which, given pre-defined implication rules, guarantees isomorphic ordering of relations in the embedding space. Like TransH, TransINT embeds a relation r_j to a (subspace, vector) pair $(H_j, \vec{r_j})$. However, TransINT modifies the relation embeddings $(H_j, \vec{r_j})$ so that the *relation spaces* (i.e., red line of Figure 2-2b) are ordered by implication; we do so by intersecting the H_j's and projecting the $\vec{r_j}$'s (Figure 2-3a). We explain with familial relations as a running example.

Intersecting the H_j's TransINT assigns distinct hyperplanes $H_{is\_father\_of}$ and $H_{is\_mother\_of}$ to *is_father_of* and *is_mother_of*. However, because *is_parent_of* is implied by the aforementioned relations, we assign

$$H_{is\_parent\_of} = H_{is\_father\_of} \cap H_{is\_mother\_of}.$$

In $\mathbb{R}^3$, TrainsINT's $H_{is\_parent\_of}$ is not a hyperplane but a line (Figure 2-3a), unlike in TransH where all H_j's are hyperplanes; in some $\mathbb{R}^d$, Figure 2-3a's $H_{is\_parent\_of}$ will be a linear subspace whose basis has rank $d-2$.[1]

Projecting the $\vec{r_j}$'s TransINT constrains the $\vec{r_j}$'s with projections (Figure 2-3a's dotted orange lines). First, $\overrightarrow{r_{is\_father\_of}}$ and $\overrightarrow{r_{is\_mother\_of}}$ are required to have the same projection onto $H_{is\_parent\_of}$. Second, $\overrightarrow{r_{is\_parent\_of}}$ is that same projection onto $H_{is\_parent\_of}$.

We introduced the constraints on H_j's and $\vec{r_j}$ above because they result in ordering the *relation spaces* of r_j's isomorphically to their relation sets. Figure 2-3b graphically illustrates that *is_parent_of*'s *relation space* (green hyperplane) includes those of *is_father_of* (blue line) and *is_mother_of* (red line). More generally, the two constraints above guarantee that ($R_i \subset R_j$) iff (r_i's relation space $\subset$ r_j's relation space).

One thing not to be confused by is that H_j is not r_j's *relation space*. H_j is a "projecting hyperplane" assigned uniquely to each r_j, so that Eq. 1 can be imposed on each fact triple (h, r_j, t); on the other hand, the *relation space* of r_j is the set where all such $\overrightarrow{t-h}$ belong (Figure 2-3). The focus and interest of our work are to inclusion-order the *relation spaces* of r_j's, not H_j's..

[1]One weakness of TransINT is that in $\mathbb{R}^d$, the dimension of the basis of H_j can be as low as $d-h-1$, when there is a chain of h relations that imply r_j (i.e., $r_1 \Rightarrow r_2 \Rightarrow ... \Rightarrow r_h \Rightarrow r_j$). However, in existing benchmark datasets [7, 57, 8], the highest "h" is normally less than 10, while the embedding dimension d is chosen among {50, 100, 200} for TransINT as well as existing KG embedding methods [8, 59]. Thus, it is unlikley that $d-h-1$ becomes significantly smaller than d. Furthermore, TransINT shows great experimental results despite the dimensions of some H_j's being lower than $d-1$ (Section 2.5); thus, the reduction of dimensions for some H_j's should not be a serious problem.

In summary, TransINT requires that

For distinct relations r_i, r_j, require the following if and only if $r_i \Rightarrow r_j$:
Intersection Constraint: $H_j = H_i \cap H_j$.
Projection Constraint: Projection of $\vec{r_1}$ to H_j is $\vec{r_j}$.
where $\overrightarrow{H_i}, \overrightarrow{H_j}$ and $\vec{r_i}, \vec{r_j}$ are distinct.

In Section 2.3, we prove that these two constraints guarantee that an ordering isomorphic to implication holds in the embedding space:

$(r_i \Rightarrow r_j)$ iff (r_i's rel. space $\subset$ r_j's rel. space) or equivalently,

$(R_i \subset R_j)$ iff (r_i's rel. space $\subset$ r_j's rel. space).

2.3 TransINT's Isomorphic Guarantee

In this section, we formally state TransINT's isomorphic guarantee. We denote all $d \times d$ matrices with capital letters (e.g., A) and vectors with arrows on top (e.g., $\vec{b}$).

2.3.1 Projection and *relation space*

In $\mathbb{R}^d$, there is a bijection between each linear subspace H_i and a projection matrix P_i; $\forall \vec{x} \in \mathbb{R}^d, P_i x \in H_i$ [48]. A random point $\vec{a} \in \mathbb{R}^d$ is projected onto H_i iff multiplied by P_i; i.e., $P_i a = \vec{b} \in H_i$. In the rest of the paper, we denote P (or P_i) as the projection matrix onto a linear subspace H (or H_i). Now, we formally define a general concept that subsumes *relation space* (Figure 2-3b).

Definition* $(Sol(P, \vec{k}))$: Let H be a linear subspace and P its projection matrix. Then, given $\vec{k}$ on H, the set of vectors that become $\vec{k}$ when projected on to H, or the solution space of $P\vec{x} = \vec{k}$, is denoted as $\mathbf{Sol}(\mathbf{P}, \vec{k})$.

With this definition, *relation space* (Figure 2-3b) is $(Sol(P_i, \vec{r_i}))$, where P_i is the projection matrix of H_i (subspace for relation r_i); it is the set of points $\overrightarrow{t-h}$ such that $P_i(\overrightarrow{t-h}) = \vec{r_i}$.

2.3.2 Isomorphic Guarantees

Main Theorem 1 *(Isomorphism):* Let $\{(H_i, \vec{r_i})\}_n$ be the (subspace, vector) embeddings assigned to relations $\{\mathbf{R_i}\}_n$ by the *Intersection Constraint* and *the Projection Constraint*; P_i the projection matrix of H_i. Then, $(\{Sol(P_i, \vec{r_i})\}_n, \subset)$ is isomorphic to $(\{\mathbf{R_i}\}_n, \subset)$.

In actual optimization, TransINT requires something less strict than $P_i(\overrightarrow{t-h}) = \vec{r_i}$:

$$P_i(\overrightarrow{t-h}) - \vec{r_i} \approx \vec{0} \equiv ||P_i(\overrightarrow{t-h} - \vec{r_i})||_2 < \epsilon,$$

for some non-negative and small ϵ. This bounds $\overrightarrow{t-h} - \vec{r_i}$ to regions with thickness 2ϵ, centered around $Sol(P_i, \vec{r_i})$ (Figure 2-4). We prove that isomorphism still holds with this weaker requirement.

Definition** $(Sol_\epsilon(P, k))$: Given any P, the solution space of $||P\vec{x} - \vec{k}||_2 < \epsilon$ (where $\epsilon \geq 0$) is denoted as $\mathbf{Sol}_\epsilon(\mathbf{P}, \vec{k})$.

Main Theorem 2 *(Margin-aware Isomorphism):* $\forall \epsilon \geq 0$, $(\{Sol_\epsilon(P_i, \vec{r_i})\}_n, \subset)$ is isomorphic to $(\{\mathbf{R_i}\}_n, \subset)$.

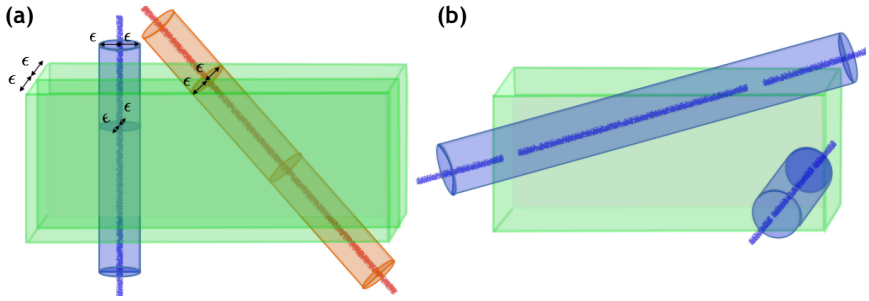

Figure 2-4: Figure 2-3(b)'s *relation spaces* when $P_i(\overrightarrow{t-h}) - \vec{r_i} \approx \vec{0} \equiv ||P_i(\overrightarrow{t-h} - \vec{r_i})||_2 < \epsilon$ is required. (a): Each *relation space* now becomes a region with thickness ϵ, centered around figure 2-3(b)'s *relation space*. (b): Relationship of the angle and area of overlap between two *relation spaces*. With respect to the green region, the nearly perpendicular cylinder overlaps much less with it than the other cylinder with much smaller angle.

2.4 Initialization and Training

The intersection and projection constraints can be imposed with parameter sharing. We first introduce some preliminary definitions.

Definition (Parent Relation):* For two relations r_1, r_2, if $r_1 \Rightarrow r_2$ and there is no r_3 such that $r_1 \Rightarrow r_3 \Rightarrow r_2$, then r_1 is *parent relation* of r_2; r_2 is child of r_1. For example, in Figure 1c of the submitted paper, *is_family_of* is *is_parent_of*'s *parent relation.*

Definition (Head Relation):* A relation r that has no *parent relation* is a *head relation.*

For example, in Figure 1b, *is_parent_family_of* is a *head relation.*

2.4.1 Parameter Sharing Initializaion

From initialization, we bind parameters so that they satisfy the two constraints. For each entity e_j, we assign a d-dimensional vector $\vec{e_j}$. To each $\mathbf{R}_i$, we assign $(H_i, \vec{r_i})$ (or $(A_i, \vec{r_i})$) with parameter sharing. We first construct the H's.

Intersection constraint Each subspace H can be uniquely defined by its orthogonal subspace. We define the orthogonal subspace of the H's top-down. To every *head relation* $\mathbf{R}_h$, assign a d-dimensional vector $\vec{a_h}$ as an orthogonal subspace for H_{R_h}, making H_{R_h} a hyperplane. Then, to each $\mathbf{R}_i$ that is not a *head relation*, additionally assign a new d-dimensional vector $\vec{a_i}$ linearly independent of the bases of all of its parents. Then, $\mathbf{R}_i$'s basis of the orthogonal subspace for H_{R_i} becomes $[\vec{a_h}, ..., \vec{a_p}, \vec{a_i}]$ where $\vec{a_h}, ..., \vec{a_p}$ are the vectors assigned to $\mathbf{R}_i$'s parent relations. Projection matrices can be uniquely constructed given the bases $[\vec{a_h}, ..., \vec{a_p}, \vec{a_i}]$ [48]. Now, we initialize the $\vec{r_i}$'s.

Projection Constraint To the head relation $\mathbf{R}_h$, pick any random $x_h \in \mathbb{R}^d$ and assign $\vec{r_h} = P_h x$. To each non-head $\mathbf{R}_i$ whose parent is $\mathbf{R}_p$, assign $\vec{r_i} = \vec{r_p} + (I - P_p)(P_i)x_i$ for some random x_i. This results in

$$P_p \vec{r_i} = P_p \vec{r_p} + P_p(I - P_p)(P_i)\vec{x_i} = \vec{r_p} + \vec{0} = \vec{r_p}$$

for any parent, child pair.

Parameters to be trained Such initialization leaves the following parameters given a KG with entities e_j's and relations r_i's: (1) a d-dimensional vector ($\vec{a_h}$) for the head relation, (2) a d-dimensional vector ($\vec{a_i}$) for each non-head relation, (3) a d-dimensional vector $\vec{x_i}$ for each head and non-head relation, (4) a d-dimensional vector $\vec{e_j}$ for each entity e_j. TransH and TransINT both assign two d-dimensional vectors for each relation and one d-dimensional vector for each entity; thus, TransINT has the same number of parameters as TransH.

2.4.2 Training

We construct negative examples (wrong fact triplets) and train with a margin-based loss, following the same protocols as in TransE and TransH.

Training Objective We adopt the same loss function as in TransH. For each fact triplet (h, r_i, t), we define the score function

$$f(h, r_i, t) = ||P_i(\overrightarrow{t-h}) - \vec{r_i}||_2$$

and train a margin-based loss L:

$$L = \sum_{(h,r_i,t)\in G} max(0, f(h, r_i, t)^2 + \gamma - f(h', r_i', t')^2).$$

where G is the set of all triples in the KG and (h', r_i', t') is a negative triple made from corrupting (h, r_i, t). We minimize this objective with stochastic gradient descent.

Automatic Grounding of Positive Triples Without any special treatment, our initialization guarantees that training for a particular (h, r_i, t) also automatically executes training with (h, r_p, t) for any $r_i \Rightarrow r_p$, at all times. For example, by traversing *(Tom, is_father_of, Harry)* in the KG, the model automatically also traverses *(Tom, is_parent_of, Harry), (Tom, is_family_of, Harry)*, even if they are missing in the

KG. This is because $P_p P_i = P_p$ (by Lemma 4 of Appendix A.1.1) with the given initialization and thus,

$$f(h, r_p, t) = ||P_p(\overrightarrow{t-h}) - \vec{r_p}||_2^{\ 2} = ||P_p(P_i((\overrightarrow{t-h}) - \vec{r_i}))||_2^{\ 2}$$
$$\leq ||(P_p + (I - P_p))P_i((\overrightarrow{t-h}) - \vec{r_i}))||_2^{\ 2} = ||(P_i((\overrightarrow{t-h}) - \vec{r_i}))||_2^{\ 2} = f(h, r_i, t)$$

In other words, training $f(h, r_i, t)$ towards less than ϵ automatically guarantees training $f(h, r_p, t)$ towards less than ϵ. This eliminates the need to manually create missing triples that are true by the implication rule.

2.5 Experiments

We evaluate TransINT on two standard benchmark datasets, Freebase 122 [8] and NELL sport/location [57], and compare against, respectively, KALE [27] and SimplE+ [21], state-of-the-art methods that integrate rules to KG embeddings, in the trans- and bilinear family. We perform link prediction and triple classification tasks on Freebase 122, and link prediction only on NELL sport/location (because SimplE+ only reported performance on link prediction). All codes for experiments were implemented in PyTorch [44].

2.5.1 Link Prediction on Freebase 122 and NELL Sport/Location

We compare link prediction results with KALE on Freebase 122 (FB122) and with SimplE+ on NELL Sport/Location. The task is to predict the gold entity given a fact triple with missing head or tail: if (h, r, t) is a fact triple in the test set, predict h given (r, t) or predict t given (h, r). We follow TransE, KALE, and SimplE+'s protocol. For each test triple (h, r, t), we rank the similarity score $f(e, r, t)$ when h is replaced with e for every entity e in the KG, and identify the rank of the gold head entity h; we do the same for the tail entity t. Aggregated over all test triples, we report for FB 122: (i) the mean reciprocal rank (**MRR**), (ii) the median of the ranks (**MED**), and (iii) the proportion of ranks no larger than n (**HITS@N**) which

are the same metrics reported by KALE. For NELL Sport/Location, we follow the protocol of SimplE+ and do not report MED. A lower MED, and a higher MRR and Hits HITS@N are better.

TransH, KALE, and SimplE+ adopt a "filtered" setting that addresses when entities that are correct, albeit not gold, are ranked before the gold entity. For example, if the gold entity is *(Tom, is_parent_of, John)* and we rank every entity e for being the head of *(?, is_parent_of, John)*, it is possible that *Sue*, *John*'s mother, gets ranked before *Tom*. To avoid this, the "filtered setting" ignores corrupted triplets that exist in the KG when counting the rank of the gold entity. (The setting without this is called the "raw setting").

TransINT's hyperparameters are: learning rate (η), margin (γ), embedding dimension (d), and learning rate decay (α), applied every 10 epochs to the learning rate. We find optimal configurations among the following candidates: $\eta \in \{0.003, 0.005, 0.01\}, \gamma \in \{1, 2, 5, 10\}, d \in \{50, 100\}, \alpha, \in \{1.0, 0.98, 0.95\}$; we grid-search over each possible (η, γ, d, α). We create 100 mini-batches of the training set (following the protocol of KALE) and train for a maximum of 1000 epochs with early stopping based on the best median rank. Furthermore, we try training with and without normalizing each of entity vectors, relation vectors, and relation subspace bases after every batch of training.

Experiment on Freebase 122

We compare our performance with that of KALE and previous methods (TransE, TransH, TransR) that were compared against it, using the same dataset (FB122). FB122 is a subset of FB15K [8] accompanied by 47 implication and transitive rules; it consists of 122 Freebase relations on "people", "location", and "sports" topics. Out of the 47 rules in FB122, 9 are transitive rules (e.g., `person/nationality(x,y)` $\wedge$ `country/official_language(y,z)` $\Rightarrow$ `person/languages(x,z)`) to be used for KALE. However, since TransINT only deals with implication rules, we do not take advantage of them, unlike KALE.

We also put us at some intentional disadvantages against KALE to assess TransINT's

	Raw					Filtered				
	MRR	MED	Hits N%			MRR	MED	Hits N%		
			3	5	10			3	5	10
TransE	0.262	10.0	33.6	42.5	50.0	0.480	2.0	58.9	64.2	70.2
TransH	0.249	12.0	31.9	40.7	48.6	0.460	3.0	53.7	59.1	66.0
TransR	0.261	15.0	28.9	37.4	45.9	0.523	2.0	59.9	65.2	71.8
KALE*	0.294	9.0	36.9	44.8	51.9	0.523	2.0	61.7	66.4	72.8
TransINT^G	**0.339**	**6.0**	**40.1**	**49.1**	**54.6**	**0.655**	**1.0**	**70.4**	**75.1**	**78.7**
TransINT^{NG}	0.323	8.0	38.3	46.6	53.8	0.620	1.0	70.1	74.1	78.3

Table 2.1: Results for link prediction on FB122.
*For KALE, we report the best performance by any of KALE-PRE, KALE-Joint, KALE-TRIP (3 variants of KALE proposed by [27]).

	Sport					Location				
	MRR		Hits N%			MRR		Hits N%		
	Filtered	Raw	1	3	10	Filtered	Raw	1	3	10
Logical Inference	-	-	28.8	-	-	-	-	27.0	-	-
SimplE	0.230	0.174	18.4	23.4	32.4	0.190	0.189	13.0	21.0	31.5
SimplE+	0.404	0.337	33.9	44.0	50.8	0.440	0.434	43.0	44.0	45.0
TransINT^G	**0.450**	0.361	**37.6**	**50.2**	**56.2**	**0.550**	**0.535**	**51.2**	**56.8**	**61.1**
TransINT^{NG}	0.431	**0.362**	36.7	48.7	52.1	0.536	0.534	51.1	53.3	59.0

Table 2.2: Results for link prediction on NELL sport/location.

robustness to absence of negative example grounding — the use of given rules to avoid false negatives in creating negative examples to be used in the margin-based loss L^2. In constructing negative examples for the margin-based loss L, KALE both uses rules (by grounding) and their own scoring scheme to avoid false negatives. While grounding with FB122 is not a burdensome task, it known to be very inefficient and difficult for extremely large datasets [15]. Thus, it is a great advantage for a KG model to perform well without grounding of training/test data. We evaluate TransINT on two settings, with and without rule grounding. We call them respectively TransINT^G (grounding), TransINT^{NG} (no grounding).

We report link prediction results in Table 2.1; since we use the same train, test and validation sets, we directly copy from [27] for baselines. While the *filtered* setting gives better performance (as expected), the trend is generally similar between *raw*

[2]The simplest method to construct negative examples is to replace the head or tail of an existing KG fact with another entity. For example, from *(Paris, is_city_of, France)*, a negative example *(Paris, is_city_of, England)* can be created; however, false negatives such as *(Paris, is_city_of, EU)* can be created as well. Rules such as *France is part of EU* can prevent such false negatives.

and *filtered*. TransINT outperforms all other models by large margins in all metrics, even without grounding; especially in the *filtered* setting, the **Hits@N** gap between TransINT^G and KALE is around 4∼6 times that between KALE and the best Trans Baseline (TransR).

Also, while TransINT^G performs higher than TransINT^{NG} in all settings/metrics, the gap between them is much smaller than that between TransINT^{NG} and KALE, showing that TransINT robustly brings state-of-the-art performance even without grounding. The results suggest two possibilities in a more general sense. First, the emphasis on true positives could be as important as or more important than avoiding false negatives. Even without manual grounding, TransINT^{NG} has automatic grounding of positive training instances enabled (Section 4.1.1.) due to model properties, and this could be one of its success factors. Second, hard constraints on parameter structures can yield a performance boost significantly larger than that by regularization or joint learning, which are softer constraints.

Experiment on NELL Sport/Location

We compare TransINT against SimplE+, a state-of-the-art method that outperforms ComplEx [51] and SimplE [32], on NELL (Sport/Location) for link prediction. NELL Sport/Location is a subset of NELL [40] accompanied by implication rules; a complete list of them is available in Appendix A.2. Since we use the same train. test and validation sets, we directly copy from [21] for baselines (Logical Inference, SimplE, SimplE+). The results are shown in Table 2.2. Again, TransINT^G and TransinT^{NG} significantly outperform other methods in all metrics. The general trends are similar to the results for FB 122; again, the performance gap between TransINT^G and TransINT^{NG} is much smaller than that between TransINT^{NG} and SimplE+.

2.5.2 Triple Classification on Freebase 122

The task is to classify whether an unobserved instance (h, r, t) is correct or not, where the test set consists of positive and negative instances. We use the same protocol and

TransE	**TransH**	**TransR**	**KALE***	**TransINT^G**	**TransINT^{NG}**
0.634	0.641	0.619	0.677	**0.781** (0.839/ 0.752)	**0.743** (0.709/ 0.761)

Table 2.3: Results for triple classification on FB122, in Mean Average Precision (MAP).

test set provided by KALE; for each test instance, we evaluate its similarity score $f(h, r, t)$ and classify it as "correct" if $f(h, r, t)$ is below a certain threshold (σ), a hyperparameter to be additionally tuned for this task. We report on mean average precision (MAP), the mean of classification precision over all distinct relations (r's) of the test instances. We use the same experiment settings and training details as in Link Prediction other than additionally finding optimal σ.

Triple classification results are shown in Table 2.3. Again, TransINT^G and TransINT^{NG} both significantly outperform all other baselines. We also separately analyze MAP for relations that are/are not affected by the implication rules (those that appear/do not appear in the rules), shown in parentheses of Table 2.3 with the order of (influenced relations/uninfluenced relations). We can see that both TransINT's have MAP higher than the overall MAP of KALE, even when the TransINT's have the penalty of being evaluated only on uninfluenced relations; this shows that TransINT generates better embeddings even for those not affected by rules. Furthermore, we comment on the role of negative example grounding; we can see that grounding does not help performance on unaffected relations (i.e., 0.752 vs 0.761), but greatly boosts performance on those affected by rules (0.839 vs 0.709). While TransINT does not necessitate negative example grounding, it does improve the quality of embeddings for those affected by rules.

2.6 Semantics Mining with Overlap Between Embedded Regions

Traditional embedding methods that map an object (i.e., words, images) to a singleton vector learn soft tendencies between embedded vectors with cosine similarity, or

		Relation	Angle	*imb*
Not Disjoint	Relatedness	`/people/person/nationality`	22.7	1.18
	Implication	`/people/person/place_lived/location`*	46.7	3.77
Disjoint		`/people/cause_of_death/people`	76.6	n/a
		`/sports/sports_team/colors`	83.5	n/a

Table 2.4: Examples of angles and *imb* between `/people/person/place_of_birth` and other relations

angular distance between two embddings. TransINT extends such a line of thought to semantic relatedness between groups of objects, with angles between *relation spaces*. In Figure 2-4b, one can observe that the closer the angle between two embedded regions, the larger the overlap in area. For entities h and t to be tied by both relations r_1, r_2, $\overrightarrow{t-h}$ has to belong to the intersection of their *relation spaces*. Thus, we hypothesize the following over any two relations r_1, r_2 that are not explicitly tied by the pre-determined rules:

Let V_1 be the set of $\overrightarrow{t-h}$'s in r_1's *relation space* (denoted as Rel_1) and V_2 that of r_2's.

(1) Angle between Rel_1 and Rel_2 represents semantic "disjointness" of r_1, r_2; the more disjoint two relations, the closer their angle is to 90°.

When the angle between Rel_1 and Rel_2 is small,

(2) if majority of V_1 belongs to the overlap of V_1 and V_2 but not vice versa, r_1 implies r_2.

(3) if majority of V_1 and V_2 both belong to their overlap, r_1 and r_2 are semantically related.

(2) and (3) consider the imbalance of membership in overlapped regions. Exact calculation of this involves specifying an appropriate ϵ (Figure 2-3). As a proxy for deciding whether an element of V_1 (denoted by v_1) belongs in the overlapped region, we can consider the distance between v_1 and its projection to Rel_2; the further away v_1 is from the overlap, the larger the projected distance. Call the mean of such distances from V_1 to Rel_2 as d_{12} and the reverse d_{21}. The imbalance in d_{12}, d_{21} can be quantified with $\frac{1}{2}(\frac{d_{12}}{d_{21}} + \frac{d_{21}}{d_{12}})$, which is minimized to 1 when $d_{21} = d_{12}$ and increases as d_{12}, d_{21} are more imbalanced; we call this factor *imb*.

For hypothesis (1), we verified that the vast majority of relation pairs have an-

gles near to 90°, with the mean and median respectively 83.0° and 85.4°; only 1% of all relation pairs had angles less than 50°. We observed that relation pairs with angle less than 20° were those that can be inferred by transitively applying the predetermined implication rules. Relation pairs with angles within the range of $[20°, 60°]$ had strong tendencies of semantic relatedness or implication; such a tendency drastically weakened past 70°. Table 2.4 shows the angle and *imb* of relations with respect to `/people/person/place_of_birth`, whose trend agrees with our hypotheses. Finally, we note that such an analysis could be possible with TransH as well, since their method too maps $\overrightarrow{t-h}$'s to lines (Figure 2-2b).

In all of link prediction, triple classification, and semantics mining, TransINT's theme of assigning optimal regions to bound entity sets is unified and consistent. These two qualities were missing in existing works such as TransE, KALE, and SimplE+.

2.7 Related Work

Our work is related to two strands of research. The first is Order Embeddings [53] and their extensions [54, 3], which are significantly limited in that only unary relations and their hierarchies can be modeled. While [41] also approximately embeds unary partial ordering, their focus is on achieving reasonably competent results with unsupervised learning of rules in low dimensions, while ours is achieving state-of-the-art in a supervised setting.

The second strand is those that enforce the satisfaction of common sense logical rules for binary and n-ary relations in the embedded KG. [56] explicitly constrains the resulting embedding to satisfy logical implications and type constraints via linear programming, but it only requires doing so during inference, not learning. On the other hand, [27, 45, 21] encourage embeddings to follow a set of logical rules during learning, but their approaches involve soft induction instead of hard constraints, resulting in rather insignificant improvements. Our work combines the advantages of both [56] and works that impose rules during learning. Finally, [13] models unary re-

lations only and [38] transitivity only, whose contributions are fundamentally different from ours.

2.8 Conclusion

We presented TransINT, a new KG embedding method such that relation sets are mapped to continuous sets in $\mathbb{R}^d$, inclusion-ordered isomorphically to implication rules. Our method is extremely powerful, outperforming existing state-of-the-art methods on benchmark datasets by significant margins. We further proposed an interpretable criterion for mining semantic similarity and implication rules among sets of entities with TransINT.

Chapter 3

Advancing Seq2seq Semantic Parsing with Joint Paraphrase Learning

We address the problem of model generalization for sequence to sequence (seq2seq) architectures. We propose going above and beyond data augmentation by jointly learning paraphrases along with the main task. We observe that this is particularly useful in correctly handling unseen sentential paraphrases in semantic parsing — mapping English utterances to logical forms (structured representations that uniquely and exactly capture natural language meanings (Figure 3-1)). The proposed approach significantly outperforms state-of-the-art seq2seq models for semantic parsing on diverse domains: on Overnight, by up to 3.2%, and on emrQA, by 7%).

3.1 Introduction

Natural language provides a vast number of alternative ways to state something or to ask a question. This poses a daunting challenge to natural language processing methods because there is no possible way to enumerate all these alternatives. As a result, many popular machine learning systems trained on benchmark datasets are surprisingly fragile to such previously unobserved variations of the training input at test time. An attempt to ameliorate this problem is to augment the original training data with paraphrases. Obtaining such paraphrases from people is time-

consuming and expensive, leaving most possible paraphrases out of an augmented corpus. Furthermore, regardless of the magnitude of data augmentation, there always exist unseen instances that can break the model. Thus, data augmentation alone is an insufficient and incomplete remedy for improving model brittleness.

We propose to go above and beyond data augmentation in handling model generalization for sequence-to-sequence (seq2seq) semantic parsing and improve model generalization to test sets that entirely consist of unseen paraphrases of the training set. Assuming that data augmentation already took place in the training set, we propose new models that actively employ the properties of paraphrase-augmented data as part of the training objective.

We incorporate multi-task paraphrase detection and generation learning to sequence models for semantic parsing. We show that our models compare over and above other popular generalization schemes, such as feature-based or fine-tuned word embeddings [35, 14] or paraphrase-based methods such as paraphrase embeddings [61]. The proposed models outperform state-of-the-art models [43, 30] when evaluated across a variety of settings on emrQA [43] and Overnight for semantic parsing in the clinical and the open domain.

The main contributions of our work are as follows: (1) We propose novel multi-task learning seq2seq semantic parsing that significantly improves model generalization to unseen paraphrases at test time, in both the clinical and the open domain. (2) We introduce new methods of splitting data into train/ test sets that more realistically evaluates model generalization to paraphrases. (3) We present the first competitive baseline for semantic parsing on the emrQA dataset.

3.2 Related Works

Dealing with unseen paraphrastic variants of the input has been a fundamental problem [39, 18]. Recently, multiple works have shown that models easily "break" when evaluated on adversarial examples, which are noisy variants of the training inputs [25, 29]. However, there is relatively little work that goes beyond augmentation and

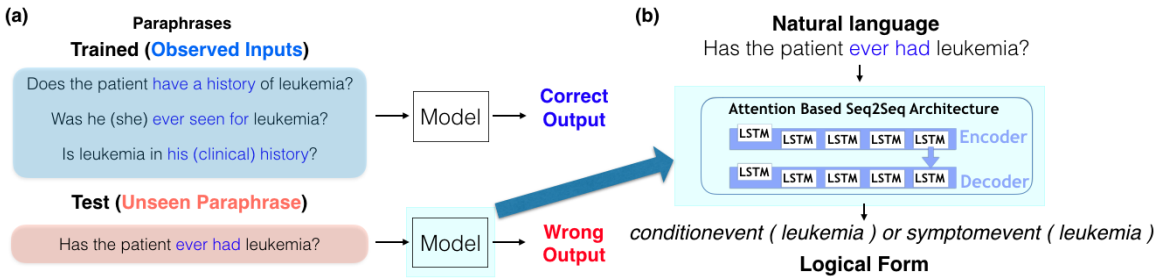

Figure 3-1: An overview of our work. (a) The objective is to train a seq2seq paraphrase model that is capable of accurately generalizing to unseen sentential paraphrases only observed at test time (red). Phrases highlighted in blue are synonymous when accompanied by a clinical condition, such as leukemia. (b) Example inputs and outputs for semantic parsing with the emrQA dataset.

employs a new training scheme — one that actively optimizes paraphrastic generalization along with learning the main NLP task at hand, in neural settings.

In non-neural settings, the idea that leveraging paraphrases facilitates modeling sentential semantics has been repeatedly verified across various NLP tasks. In semantic parsing, [5] deal with understanding the myriad paraphrastic variants in which knowledge base relations can be expressed in human language. They use a paraphrase of the original input utterance as an intermediary, which is used as an ancillary factor in ranking the likelihood of each candidate logical form.

In neural settings, the most widespread approach is to simply generate paraphrases for data augmentation, as used by [19] in question answering and [58] in semantic parsing. There are relatively few approaches that explicitly incorporate pairwise paraphrastic equivalence of inputs as part of the model. In semantic parsing, [16] applies CNN to learn paraphrase detection in a multi-task manner; [49] generate the simplest paraphrases for input utterances and uses them as intermediaries for mapping input to output.

In question answering, several multi-task learning works learn paraphrase detection along with the main task; [10] optimizes a multi-task objective (negative cosine similarity) that encourages embeddings of paraphrases to have small angular distance in every other iteration of training. Additionally, [17] uses an auxiliary multi-task learning objective for paraphrase detection in training multi-column convolutional

neural networks for structured question answering. Both of these works leverage the paraphrase clusters of the WIKIANSWERS [20] dataset. However, [17] found that their multi-task learning method gives almost no advantage. Moreover, both works did not analyze which domains or types of validation inputs benefited from paraphrase learning. Most importantly, these works are fundamentally and methodologically different from ours, in that they leveraged the paraphrases from WIKIANSWERS not as inputs to the main model, but only for learning paraphrase detection. On the other hand, our work uses paraphrase instances for both multi-task paraphrase learning and the main task, which is the driving factor behind the significant performance boost by our models.

3.2.1 Paraphrases

Paraphrases are sentences or phrases that convey the same meaning using different wording [6]. Methods to construct paraphrases are largely divided into syntactic variation and substitution [6]. *"Does the patient have a history of leukemia?"* and *"Is there leukemia in the patient's history?"* are syntactic paraphrases, with overlapping words reordered. Most paraphrases are not fully syntactic, and involve substitutions with synonymous phrases by matching general semantics to that of a domain sublanguage. E.g., *"have a history of"* is a general phrase and is not always synonymous to *"seen for"*, but the two are paraphrases of each other when accompanied by a condition in the clinical domain (Figure 3-1a).

Table 3.1 shows examples of annotated paraphrases that are of syntactic variant and synonymous substitution types. Some of emrQA and Overnight's paraphrases respectively assume knowledge of clinical ("considered for" ≡ "seen for, diagnosed with" when collocated with a |clinical problem|) and quantitative sublanguage ("at most two" ≡ "one or two").

Para. Types	**emrQA**
Syntactic Para's	what medication has the patient used for |problem| what medications have been previously used for the treatment of |problem|
Substitution Para's	is there any mention of |problem| in the patients record has been the patient ever been considered for |problem|
Para. Types	**Overnight**
Syntactic Para's	find an additional author to an efron article who is the other author for the article written by efron
Substitution Para's	article that at most two articles cite articles cited by two or more articles

Table 3.1: Examples of annotated paraphrases in emrQA and Overnight. Syntactic variation paraphrases and synonymous substitution paraphrases are respectively abbreviated as Syntactic Para's and Substitution Para's.

3.2.2 Problem Statement

Our setup assumes (1) a paraphrase-augmented dataset and (2) a baseline seq2seq model [50], which maps an input sequence to an output sequence. Our goal is to achieve additional improvement in model generalization, given this setup.

More formally, we are given a paraphrase-augmented dataset that consists of N input utterances $\{x^1, ..., x^N\}$ and corresponding output utterances $\{y^1, ..., y^N\}$. Input utterances $\{x^1, ..., x^N\}$ can be partitioned into K paraphrase groups $P_1, ..., P_K$, where each input utterance x^i belongs to exactly one paraphrase group and each P_i $(i \in 1, ..., K)$ has at least two elements. The constituents of each P_i are considered semantically equivalent. Each P_i is the set of all possible sentences that are paraphrases of each other that we are aware of.

On the other hand, there always exist unseen paraphrases that were not included during data augmentation. Let $T_1, ..., T_K$ each be a subset of those paraphrases not in $P_1, ..., P_K$, where all elements of T_i are paraphrases of any element of P_i.

We use $P_1, .., P_K$ as the training set and $T_1, ..., T_K$ as the test set. Our goal here is to improve model generalization to the unobserved instances in each T_i, by leveraging observed instances in P_i — to output a mapping from x^j $(j \in \{1, ..., N\})$ to its corresponding output y^j that performs well for (x^i, y^i)'s in the test set. In other words, given a seq2seq task with a training and a test set of input-output pairs and several unseen observations in the test set that are paraphrases of the training observations, we want to learn a model that can generalize accurately to unseen paraphrases.

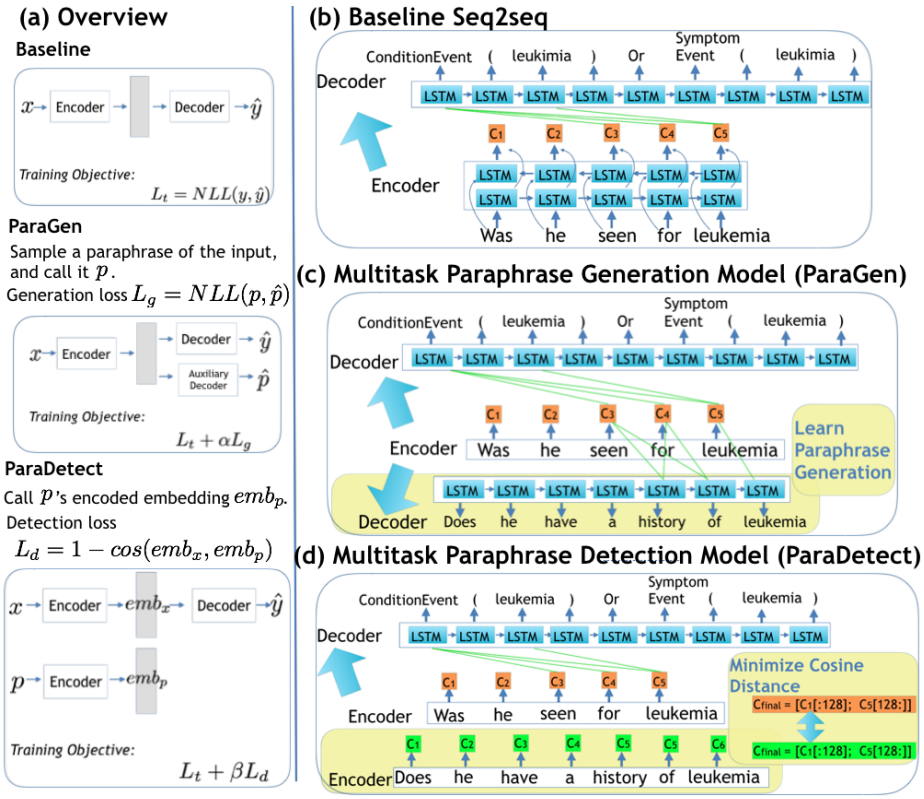

Figure 3-2: Proposed models. (a): Overview of all models; the encoder embeddings of inputs are depicted as gray boxes. (b): Simple Seq2seq (baseline) (c): Multitask Paraphrase Generation Model (d): Multitask Paraphrase Detection Model. Green lines represent attention weights, in (b), (c), (d). Detailed view of the multitask paraphrase generation and detection model is omitted for simplicity.

3.3 Methods: Seq2seq with Joint Paraphrase Learning

We incorporate auxiliary multi-task learning to the main seq2seq task — learning paraphrase generation (*ParaGen*), paraphrase detection (*ParaDetect*), and a combination of both tasks (*ParaGen + ParaDetect*). These methods work with any task whose inputs and outputs are sequences, on paraphrase-augmented data.

Our proposed models are alterations of the seq2seq [50] model to actively employ the natural properties that arise from data augmentation as part of the training objective. First, *ParaGen, ParaDetect, ParaGen + ParaDetect* sample a paraphrase of the given input and leverage it to reduce intra-class variance of paraphrases in

the representation space. The sampled paraphrase is a term inside each model's respective multi-task objective, which affects the encoded input embedding in the directions that reward paraphrastic homogeneity when back-propagated. However, this paraphrase sampling is only required during training; at test time, the multi-task portion of the model is discarded, and the input is passed through the seq2seq model only. This is a realistic test scenario that does not require paraphrase identification among test inputs; the expectation is that the multi-task training has optimized the backbone seq2seq model's parameters for generalization at test time.

We introduce notations, with semantic parsing on emrQA as a running example (Figure 3-1). x is an input utterance (e.g., "Does the patient have a history of leukemia") and p is a paraphrase of it sampled from the training set (e.g., "Is leukemia in his clinical history?"). y is the desired output sequence (e.g., "ConditionEvent (Leukemia) or SymptomEvent (Leukemia)" when the translation target is a logical form); mapping from x to y is the *main task*, and L_t is the negative-log-likelihood (NLL) loss for this main task. $\hat{y}, \hat{p}$ are output sequence and paraphrase generated by the models. Finally, we note that we regard an attention-based [33] seq2seq with a bidirectional LSTM encoder and a LSTM decoder, with the dropout probability set to 0.1 [46], as the backbone baseline model (Figure 3-3b).

3.3.1 ParaGen: Multitask Paraphrase Generation Model

Given an input utterance x, we sample from the training set one of x's paraphrases, p, and learn paraphrase generation from x to p along with the main task. More specifically, from a shared encoder that accepts x as an input, we keep two separately parameterized decoders that respectively produce $\hat{y}$ (main task decoder) and $\hat{p}$ (paraphrase generation decoder) as desired outputs (Figure 3-3c). The resulting objective is a weighted sum of L_g, the loss for paraphrase generation, and L_t (main task objective), defined below:

$$L_{total} = L_t + \alpha L_g \tag{3.1}$$

where L_g is the NLL loss between p and $\hat{p}$, and α is a hyperparameter for the weighted sum.

ParaGen was inspired by the *association model* strategy of learning alignment of word tokens between two paraphrases via log linear models [5]. However, the association model requires paraphrase-augmented data whose inputs are labeled with pairwise alignment, a condition not met in many situations. Neural attention is often regarded as an unsupervised proxy for token alignment [4, 28]. By unsupervised learning of alignment between x and p via neural attention (green lines in Figure 3-3c) as an auxiliary task, we attempt to influence the encoder parameters in a way such that it maps synonymous subphrases to correlated vector representations.

3.3.2 ParaDetect: Multitask Paraphrase Detection Model

In this model, we again sample a paraphrase p but learn paraphrase detection as the auxiliary task — to identify whether x and p are paraphrases by looking at their embeddings emb_x and emb_p. We keep the same model structure as the baseline, but we pass p into the same encoder used for the input utterance x, to generate emb_p, a fixed-length vector representation of p. Then, we force emb_x and emb_p, vector representations of the two paraphrases, to have high cosine similarity — a criterion popularly used for paraphrase detection methods with input vector similarity [34, 37, 23]. The resulting objective is a weighted sum of L_d, loss for paraphrase detection, and L_t, loss for the target task:

$$L_{total} = L_t + \beta L_d \tag{3.2}$$

where

$$L_d = 1 - cos(emb_x, emb_p) = 1 - \frac{emb_x \cdot emb_p}{||emb_x||\ ||emb_p||}$$

and β is a hyperparameter for the weighted sum.

What we intend to achieve is twofold. First, we want to impose homogeneity in angular distance to semantically equivalent inputs. For high dimensional data, cosine distance is considered as a reasonable approximation of the semantic similarity

among embeddings, as considered by seq2seq decoders and other models, more than Euclidean or other distance measures [12, 55]. Also, we want encoder parameters to develop agnosticity with respect to choice of expression given identical semantics; a different paraphrase p_i of x will be sampled at each iteration, and different expression among the p_i's will be encouraged to be ignored in cosine distance. Thus, after training, we expect angularly close inputs (paraphrases) to map to the same desired output.

3.3.3 Multitask Paraphrase Generation and Detection Model

We propose a combination of both models where we learn both paraphrase generation and detection as ancillary tasks. The resulting objective is a weighted sum of L_t, L_g, L_d:

$$L_{total} = L_t + \alpha(L_g + \beta L_d) \tag{3.3}$$

where α, β are hyperparameters for the weighted sums. We hope to gain both advantages of ParaGen and ParaDetect by summing their objectives.

3.4 Datasets and Novel Splitting Schemes

In this section, we explain the datasets and propose a novel train/test splitting scheme to accurately evaluate model generalization.

3.4.1 Datasets

The emrQA dataset consists of 1 million clinical-domain questions, their corresponding lf's and answer evidence in clinical notes. emrQA was created via a semi-automated process that uses annotations for various clinical NLP tasks and used them to slot fill natural language question templates and lf's. The question templates were created by normalizing medical entities in real questions collected from physicians from various health care institutions. Thus, the paraphrases in this dataset genuinely represent how physicians would phrase their information needs in different

ways.

Overnight is a semantic parsing dataset that provides questions and logical form pairs for multiple sub-domains, such as recipes and basketball [58]. This dataset was generated via crowd-sourcing on Amazon Mechanical Turk where multiple paraphrases were generated for every lf by crowd-source workers. They also generated sets of semantically similar questions (paraphrase groups); each paraphrase group was randomly split across the training and test set.

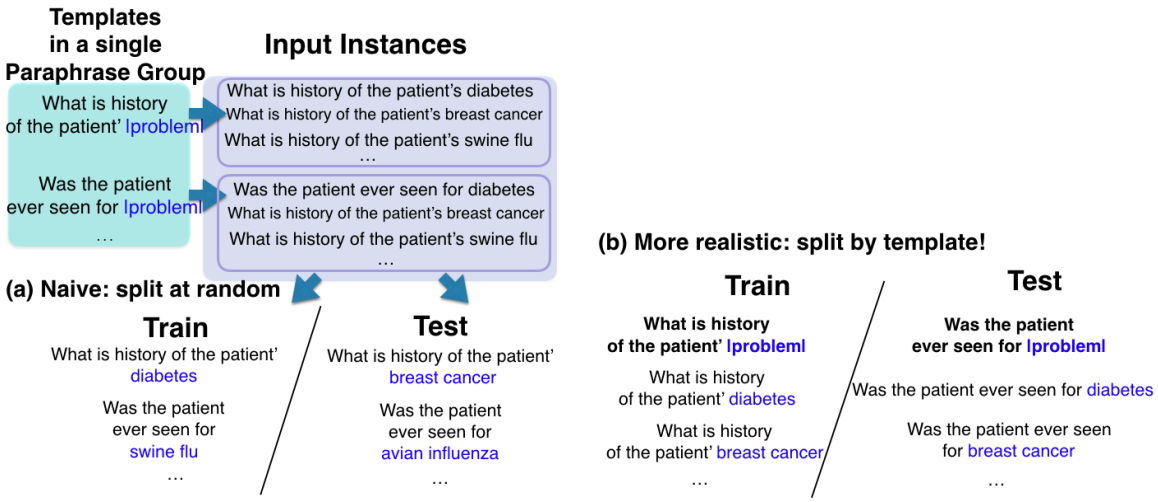

Figure 3-3: Illustration of the two splits. 2a is a naive split that is usually used in existing datasets; at deployment, this is too unrealistic because it contains too many recurring forms that were seen during training. 2b, on the other hand, disallows seen forms (during training) appearing in the test set; it is both more challenging and realistic than 2a.

3.4.2 Novel Train/Test Splitting Schemes

Not all paraphrases are created equal; some paraphrases are much less challenging than others in evaluating model performance. A common yet undesirable scenario in NLP datasets is that the form of input utterances can be repeated across training/test splits. For example, in Overnight's *recipe* domain, there are several questions in the form of "*how many* x *are there*" where x is some recipe-related entity, such as "*recipes*", "*ingredients*", "*meals*", etc. With such datasets, the test set often contains too many repeating forms of the training set. Such a train/test split is an unrealistic evaluation

of model generalization.

We propose a new, more realistic way to split paraphrase-augmented data, with the emrQA dataset as an example (Figure 3-2). emrQA consists of paraphrase groups of inputs. Within a single group, "templates" are filled with clinical entities to produce actual input instances (Figure 3-2 purple box). 2(a) shows a naive splitting scheme where the input instances are split at random. On the other hand, 2(b) is a more realistic scenario where *a form that was seen during training* ***never*** *appears at test time*. For example, all instances of "Has the pat. ever been exposed to |problem|?" belong to training and never when the model is evaluated. On the other hand, all instances of "Does this patient have a history of |problem|?" never appear during training yet do so at test time; thus, the model is tested whether it can infer the meaning of this form only from its paraphrased forms seen during training (such as "Has the pat. ever been exposed to |problem|?"). While this split is more challenging than the naive one, test instances are still semantically equivalent to some training instance, so the model is expected to catch this and generalize to unseen forms.

3.5 Experiments

We evaluate the proposed models on emrQA and Overnight, with the target task being semantic parsing. We split emrQA into train and test sets with both "naive" and "stricter" (Section 4) schemes, and create four distinct splits for each scheme for fair model evaluation; Overnight has officially released train/test sets (unlike emrQA) so we use the official splits (that are "naive") for comparison with previous work.

Our accuracy metric is "exact match", which only considers model outputs that are identical to the labeled ones as correct. We mention this because "denotation accuracy", which considers logical forms that return the label answer from the database as correct, has been used in several works on the Overnight dataset. We find this problematic, because it often considers model outputs of quantity-related questions right by chance; for example, models often wrongly interpret "less than or equal to x" as "$< x$", but this would be considered correct if the database does not contain

entries that are exactly x in time, amount, etc. Because many questions in Overnight are quantitative, we consider exact match accuracy to be a fairer metric.

3.5.1 Methods for Comparison

To adequately judge the effect of joint paraphrase learning, we use seq2seq methods that have been established as State-of-the-Art for each dataset as the backbone baseline; proposed joint paraphrase learning is added on top of these backbones.

Seq2Seq SOTA's No previous work exists on semantic parsing for emrQA; thus, we establish the first competitive baseline with the copy mechanism [26] added on the backbone seq2seq described in Section 5, for copying of medical entities (e.g., *"leukemia"*). For Overnight, we implemented [58]'s model as baseline and compared our models with the methods of [11], which is the only work on this dataset with exact match accuracy. We also note that, with our implementation of [58], we achieved a baseline higher than both of the baseline and proposed methods of [11].

Paraphrase-based Generalization Methods Our primary goal is to show that on paraphrase augmented data, active leveraging of it in the model gives additional benefits; thus comparisons with Seq2Seq SOTA's that don't leverage the presence of paraphrase clusters suffice to prove this. However, for these experiments, we also compare our models with existing paraphrase-based generalization methods that can be used under seq2seq settings, and show that our joint training outperforms them.

[61] introduced Gated Average Recurrent Networks (GRAN) — a GRU with an additional averaging gate — that learn paraphrastic sentence embeddings. The authors reported that pre-training with their method resulted in a performance boost in transfer learning on SemEval tasks. To compare our methods with pre-training via GRAN, we replace the encoder of our baseline seq2seq with a GRAN encoder pre-trained on our tasks' training set, with the GRAN encoder's parameters not frozen (allowed to be optimized).

We also compare with BERT [14] (shown to be powerful in many NLP tasks) fine-tuned on paraphrase detection, which we framed as a sentence pair binary classification task to paraphrase/non-paraphrase, applying the procedure in [14]. For fine-

tuning, we constructed the training set with all the paraphrase pairs in the original corpus and added the same number of non-paraphrase pairs, sampled randomly. ClinicalBERT [2] was used for emrQA and 12-layer base BERT (English Wikipedia) was used as the pre-trained base for Overnight. On both datasets, BERT was fine-tuned well enough to identify paraphrases with around 85% accuracy. For comparison, we took sentence embeddings from the fine-tuned BERT and replaced the encoder with it. We could not compare with end-to-end BERT models because, to our knowledge, no such prior work on semantic parsing exists.

Pre-trained Word Embeddings. Since pre-trained word embeddings are known to help generalization, the idea is to evaluate the contributions of the proposed paraphrase model over using standard methods to ensure generalization. We hypothesize two scenarios: (1) when pre-trained embeddings are available for a large-scale corpus beyond training data, and (2) when only corpus-trained embeddings are available. As large-scale embeddings, we use clinical word2Vec [35] trained on all i2b2 [52] datasets for emrQA, and officially released general English word2vec for Overnight.

3.5.2 Results

emrQA. For emrQA (Table 3.2), we can see that the proposed models outperform the baseline under both split schemes, but do so significantly worse under the "stricter" split; this shows that our models are capable of robustly generalizing to unseen syntactic variants, but does demonstrate that our stricter splitting criterion make the problem harder. We further compare our models with the different generalization methods mentioned (Table 3.4). ParaGen + ParaDetect is overwhelmingly dominant over other methods when large-scale corpus word embeddings are not available.

In emrQA, there were 338 test inputs with words that never appear during training (such as "considered" in 2nd example of emrQA's Substitution paraphrase in Table 3.1). These inputs largely determined model performance, with overall exact match accuracy being proportional to that on the exact match accuracy on these inputs. Especially, ParaGen could not capture the topic of the question (e.g., medical evaluation, treatment, etc.) when specific words were replaced with more general ones

Method	**emrQA *"naive"* split (random split)**	**emrQA *"stricter"* split (unseen paraphrases only in test set)**
Baseline: Seq2seq with copy	85.24%	54.65%
Paraphrase Generation (ParaGen)	85.87%	61.97
Paraphrase Detection (ParaDetect)	85.37%	62.04%
ParaGen + ParaDetect	**86.55%**	**63.75%**

Table 3.2: Exact match accuracy results on semantic parsing for emrQA, averaged across four splits.

Method / Domain	**Basketball**	**Blocks**	**Calendar**	**Publications**	**Recipes**	**Restaurants**	**Housing**	**SocialNetwork**
Baseline: Seq2seq with copy	82.8%	39.3%	**59.5%**	60.2%	75.0%	53.3%	47.1%	67.6%
Paraphrase Generation (ParaGen)	82.09%	40.9%	54.8%	59.6%	**75.5%**	**53.9%**	**49.2%**	**68.3%**
Paraphrase Detection (ParaDetect)	**83.8%**	**42.4%**	54.2%	60.9%	74.5%	51.5%	44.4%	**68.3%**
ParaGen + ParaDetect	82.6%	38.6%	56.5%	**63.4%**	70.4%	52.4%	45.5%	67.1%
Simple Seq2Seq (Damonte et al.)	69.6%	25.1%	43.5%	32.9%	58.3%	37.3%	29.6%	51.2%
Transfer Learning (Damonte et al.)	71.1%	25.1%	48.8%	40.4%	63.4%	39.2%	38.1%	54.5%

Table 3.3: Exact match accuracy results on semantic parsing on all domains of the Overnight dataset.

(e.g., "diagnosed for" → "considered for"). ParaDetect's errors usually occurred in mistakenly copying entities.

Overnight Across 7 out of 8 domains of Overnight, the best performing model (ParaGen) outperformed baseline by up to 3.2% (*Publications*) with a 1.6% boost on average (Table 3.3). We further compare our models with different generalization methods (Table 3.4). Word2vec was not effective, as in [49], and pre-training with BERT and GRAN were less effective than Para(Gen+Detect).

Method	**emrQA**	**Overnight (*Publication*)**
Baseline: Seq2seq with Copy*	54.65%	60.2 %
Baseline + Corpus Word2Vec	27.66%	57.1 %
Baseline + Large-Scale Word2Vec	**67.57%**	44.1%
BERT	52.48%	26.1%
GRAN	58.25%	58.6%
Paraphrase Generation (ParaGen)	61.97%	59.6%
ParaGen + Corpus Word2Vec	51.14%	60.25%
ParaGen + Large-scale Word2Vec	64.86%	39.8%
Paraphrase Detection (ParaDetect)	62.04%	60.9%
ParaDetect + Corpus Word2Vec	46.92%	57.8%
ParaDetect + Large-scale Word2Vec	63.02%	56.5%
Para(Gen+Dectect)	**63.75%**	**63.4%**
Para(Gen+Dectect) + Corpus Word2Vec	53.04%	60.2%
Para(Gen+Dectect) + Large-scale Word2Vec	66.67%	51.55%

Table 3.4: Exact match accuracy results on semantic parsing for emrQA ("stricter" split scheme, averaged over 4 splits) and Overnight (*publication* domain only).

3.6 Discussion: Cosine Distance Analysis (emrQA)

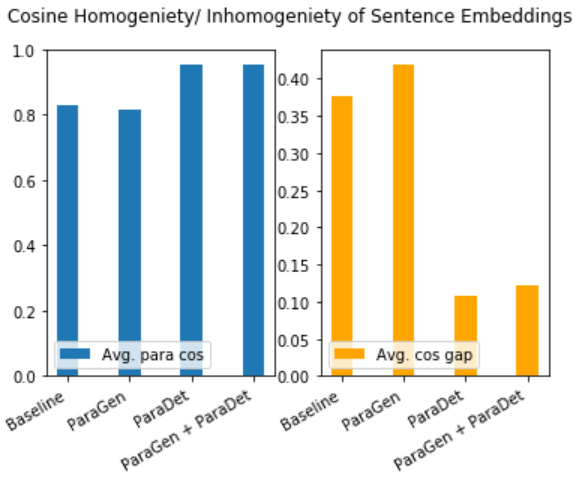

Figure 3-4: Results on cosine similarity between test question pairs in emrQA. Blue: homogeneity of paraphrases; orange: nonhomogeneity of non-paraphrases.

To understand the contribution of the proposed paraphrase models, we study the cosine similarity between embeddings of sentence pairs, a general metric in textual similarity and paraphrase detection [1, 22] in Figure 3-4. We do this by calculating the similarity between the last hidden state of the encoder for a pair of input utterances in the test set, for the four splits of emrQA's "stricter" splitting scheme. We calculate two metrics: (1) the average cosine similarity between pairs of paraphrase utterances (*Avg. para cos*, blue bar in Figure 3-4) and (2) the average difference between cosine similarity of paraphrase pairs and that of non-paraphrase pairs (*Avg. cos gap*, orange bar in Figure 3-4). They respectively quantify (1) how *homogenously* paraphrase utterances are embedded as vectors, and (2) how *nonhomogeneously* non-paraphrase utterances are embedded; *high* numbers in both quantities are ideal, if our models behave as intended in the methods section. We observed that ParaDetect achieves noticeably the highest *Avg. para cos*, and ParaGen the highest *Avg. cos gap*; ParaGen + ParaDetect shows something in between the two but closer to ParaDetect. These cosine statistics of embeddings seem to be indicative of model performance. ParaGen + ParaDetect, which embeds both paraphrases homogenously and non-paraphrases nonhomogenously, performs the best in terms of exact match accuracy; the other two

models also achieve higher performance than baseline, with much higher *Avg. para cos* and *Avg. cos gap* than baseline.

3.7 Conclusion

We presented a new general seq2seq semantic parsing framework where the main task is trained together with a paraphrase-learning objective to enhance model generalization. We also introduced new splitting schemes that reflect realistic evaluation for practical use. Our proposed approaches outperform the state-of-the-art across three datasets across both the open and clinical domain.

Chapter 4

Conclusion

We presented TransINT and a new framework to robustly train seq2seq semantic parsing models. The two works could together serve as steps to create human-like question answering systems that can understand unseen paraphrases and link existing and external facts for logical inference. While not touched upon in this thesis, one could attempt to extend TransINT to medical knowledge graphs, as I had originally envisioned. Furthermore, one could combine the two projects, by employing the logical knowledge of TransINT for paraphrase-robust semantic parsing. These will be interesting and meaningful future works to pursue.

Appendix A

Appendix for Chapter 2

A.1 Proof For TransINT's Isomorphic Guarantee

Here, we provide the proofs for Main Theorems 1 and 2. We also explain some concepts necessary in explaining the proofs. We put * next to definitions and theorems we propose/ introduce. Otherwise, we use existing definitions and cite them.

A.1.1 Linear Subspace and Projection

We explain in detail elements of $\mathbb{R}^d$ that were intuitively discussed. In this and later sections, we mark all lemmas and definitions that we newly introduce with *; those not marked with * are accompanied by reference for proof. We denote all $d \times d$ matrices with capital letters (ex) A) and vectors with arrows on top (ex) $\vec{b}$).

Linear Subspace and Rank

The linear subspace given by $A(x - \vec{b}) = 0$ (A is $d \times d$ matrix and $b \in \mathbb{R}^d$) is the set of $x \in \mathbb{R}^d$ that are solutions to the equation; its rank is the number of constraints $A(x - \vec{b}) = 0$ imposes. For example, in $\mathbb{R}^3$, a hyperplane is a set of $\vec{x} = [\ x_1, x_2, x_3] \ \in \mathbb{R}^3$ such that $ax_1 + bx_2 + cx_3 - d = 0$ for some scalars a, b, c, d; because vectors are bound by one equation (or its "A" only really contains one effective equation), a hyperplane's rank is 1 (equivalently $rank(A) = 1$). On the other hand,

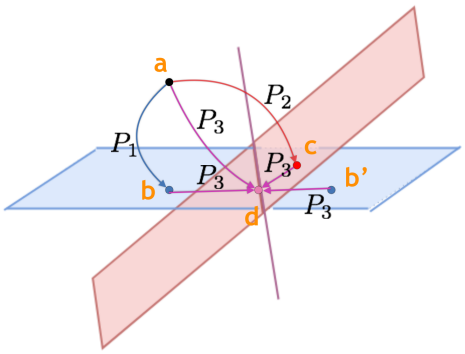

Figure A-1: Projection matrices of linear subspaces that inclusion-ordered.

a line in $\mathbb{R}^3$ imposes to 2 constraints, and its rank is 2 (equivalently $rank(A) = 2$).

Consider two linear subspaces H_1, H_2, each given by $A_1(\vec{x} - \vec{b_1}) = 0, A_2(\vec{x} - \vec{b_2}) = 0$. Then,

$$(H_1 \subset H_2) \Leftrightarrow (A_1(\vec{x} - \vec{b_1}) = 0 \Rightarrow A_2(\vec{x} - \vec{b_2}) = 0)$$

by definition. In the rest of the paper, denote H_i as the linear subspace given by some $A_i(\vec{x} - \vec{b_i}) = 0$.

Properties of Projection

Invariance For all $\vec{x}$ on H, projecting $\vec{x}$ onto H is still $\vec{x}$; the converse is also true.

Lemma 1 $P\vec{x} = \vec{x} \Leftrightarrow \vec{x} \in H$ [Linalg].

Orthogonality Projection decomposes any vector $\vec{x}$ to two orthogonal components - $P\vec{x}$ and $(I - P)\vec{x}$. Thus, for any projection matrix P, $I - P$ is also a projection matrix that is orthogonal to P (i.e. $P(I - P) = 0$) [Linalg].

Lemma 2 Let P be a projection matrix. Then $I - P$ is also a projection matrix such that $P(I - P) = 0$ [Linalg].

The following lemma also follows.

Lemma 3 $||P\vec{x}|| \leq ||P\vec{x} + (I - P)\vec{x}|| = ||\vec{x}||$ [Linalg].

Projection onto an included space If one subspace H_1 includes H_2, the order of projecting a point onto them does not matter. For example, in Figure 3, a random

point $\vec{a}$ in R^3 can be first projected onto H_1 at $\vec{b}$, and then onto H_3 at $\vec{d}$. On the other hand, it can be first projected onto H_3 at $\vec{d}$, and then onto H_1 at still $\vec{d}$. Thus, the order of applying projections onto spaces that includes one another does not matter.

If we generalize, we obtain the following two lemmas (Figure 5):

Lemma 4* Every two subspaces $H_1 \subset H_2$ if and only if $P_1P_2 = P_2P_1 = P_1$.

proof) By Lemma 1, if $H_1 \subset H_2$, then $P_2\vec{x} = \vec{x} \quad \forall \vec{x} \in H_1$. On the other hand, if $H_1 \not\subset H_2$, then there is some $\vec{x} \in H_1, \vec{x} \notin H_2$ such that $P_2\vec{x} \neq \vec{x}$. Thus,

$$\begin{aligned} H_1 \subset H_2 &\Leftrightarrow \forall \vec{x} \in H_1, \quad P_2\vec{x} = \vec{x} \\ &\Leftrightarrow \forall \vec{y}, \quad P_2(P_1\vec{y}) = P_1\vec{y} \Leftrightarrow P_2P_1 = P_1. \end{aligned}$$

Because projection matrices are symmetric [Linalg],

$$P_2P_1 = P_1 = {P_1}^T = {P_1}^T{P_2}^T = P_1P_2.$$

Lemma 5* For two subspaces H_1, H_2 and vector $\vec{k} \in H_2$,

$$H_1 \subset H_2 \Leftrightarrow Sol(P_2, \vec{k}) \subset Sol(P_1, P_1\vec{k}).$$

proof) $Sol(P_2, \vec{k}) \subset Sol(P_1, P_1\vec{k})$ is equivlaent to $\forall \vec{x} \in \mathbb{R}^d, P_2\vec{x} = \vec{k} \Rightarrow P_1\vec{x} = P_1\vec{k}$.

By Lemma 4, if $H_1 \subset H_2 \Leftrightarrow P_1P_2 = P1$. Since $\vec{k} \in P_2$, $P_2\vec{x} = \vec{k} \Leftrightarrow P_2(x - \vec{k}) = \vec{0} \Leftrightarrow P_1(P_2\vec{x} - \vec{k}) = \vec{0} \Leftrightarrow P_1P_2\vec{x} = P_1\vec{k} \Leftrightarrow P_1\vec{x} = P_1\vec{k}$.

Partial ordering If two subspaces strictly include one another, projection is uniquely defined from lower rank subspace to higher rank subspace, but not the other way around. For example, in Figure 3, a point $\vec{a}$ in R^3 (rank 0) is always projected onto H_1 (rank 1) at point $\vec{b}$. Similarly, point $\vec{b}$ on H_1 (rank 1) is always projected onto similarly, onto H_3 (order 2) at point d. However, "inverse projection" from H_3 to H_1 is not defined, because not only $\vec{b}$ but other points on H_1 (such as $\vec{b'}$) project to H_3 at

point $\vec{d}$; these points belong to $Sol(P_3, \vec{d})$. In other words, $Sol(P_1, \vec{b}) \subset Sol(P_3, \vec{d})$. This is the key intuition for isomorphism , which we prove in the next chapter.

A.1.2 Proof for Isomorphism

Now, we prove that TransINT's two constraints (section 2.3) guarantee isomorphic ordering in the embedding space.

Two posets are isomorphic if their sizes are the same and there exists an order-preseving mapping between them. Thus, any two posets $(\{A_i\}_n, \subset)$, $(\{B_i\}_n, \subset)$ are isomorphic if $|\{A_i\}_n| = |\{B_i\}_n|$ and

$$\forall i, j \quad A_i \subset A_j \Leftrightarrow B_i \subset B_j$$

Main Theorem 1 *(Isomorphism):* Let $\{(H_i, \vec{r_i})\}_n$ be the (subspace, vector) embeddings assigned to relations $\{\mathbf{R_i}\}_n$ by the *Intersection Constraint* and *the Projection Constraint*; P_i the projection matrix of H_i. Then, $(\{Sol(P_i, \vec{r_i})\}_n, \subset)$ is isomorphic to $(\{\mathbf{R_i}\}_n, \subset)$.

proof) Since each $Sol(P_i, \vec{r_i})$ is distinct and each $\mathbf{R_i}$ is assigned exactly one $Sol(P_i, \vec{r_i})$, $|\{Sol(P_i, \vec{r_i})\}_n| = |\{I_i\}_n|$.1

Now, let's show

$$\forall i, j, \quad R_i \subset R_j \Leftrightarrow Sol(P_i, \vec{r_i}) \subset Sol(P_j, \vec{r_j}).$$

Because the $\forall i, j$, intersection and projection constraints are true iff $R_i \subset R_j$, enough to show that the two constraints hold iff $Sol(P_i, \vec{r_i}) \subset Sol(P_j, \vec{r_j}$.

First, let's show $\mathbf{R_i} \subset \mathbf{R_i} \Rightarrow Sol(P_i, \vec{r_i}) \subset Sol(P_j, \vec{r_j})$. From the *Intersection Constraint*, $\mathbf{R_i} \subset \mathbf{R_i} \Rightarrow H_j \subset H_i$. By Lemma 5, $Sol(P_i, \vec{r_i}) \subset Sol(P_j, P_j\vec{r_i})$. From the *Projection Constraint*, $\vec{r_j} = P_j\vec{r_i}$. Thus, $Sol(P_i, \vec{r_i}) \subset Sol(P_j, P_j\vec{r_i}) = Sol(P_j, \vec{r_j})$. $\cdots\cdots$ 2

Now, let's show the converse; enough to show that if $Sol(P_i, \vec{r_i}) \subset Sol(P_j, \vec{r_j})$,

then the intersection and projection constraints hold true.

$$Sol(P_i, \vec{r_i}) \subset Sol(P_j, \vec{r_j})$$
$$\Leftrightarrow \forall \vec{x}, \quad P_i \vec{x} = \vec{r_i} \Rightarrow P_j \vec{x} = \vec{r_j})$$

If $P_i \vec{x} = \vec{r_i}$,

$$\forall \vec{x}, \quad P_j P_i \vec{x} = P_j \vec{r_i}$$
$$\forall \vec{x}, \quad P_j \vec{x} = \vec{r_j}$$

both have to be true. For any $\vec{x} \in H_i$, or equivalently, if $\vec{x} = P_i \vec{y}$ for some $\vec{y}$, then the second equation becomes $\forall \vec{y}, \quad P_j P_i \vec{y} = \vec{r_j}$, which can be only compatible with the first equation if $\vec{r_j} = P_j \vec{r_i}$, since any vector's projection onto a subspace is unique. (Projection Constraint)

Now that we know $\vec{r_j} = P_j \vec{r_i}$, by Lemma 5, $H_i \subset H_j$ (intersection constraint).$\cdots$ 3 From 1, 2, 3, the two posets are isomorphic.

In actual implementation and training, TransINT requires something less strict than $P_i(\overrightarrow{t-h}) = \vec{r_i}$:

$$P_i(\overrightarrow{t-h}) - \vec{r_i} \approx \vec{0} \equiv ||P_i(\overrightarrow{t-h} - \vec{r_i})||_2 < \epsilon,$$

for some non-negative and small ϵ. This bounds $\overrightarrow{t-h} - \vec{r_i}$ to regions with thickness 2ϵ, centered around $Sol(P_i, \vec{r_i})$ (Figure 4). We prove that isomorphism still holds with this weaker requirement.

Definition* $(Sol_\epsilon(P, k))$: Given a projection matrix P, we call the solution space of $||P\vec{x} - \vec{k}||_2 < \epsilon$ as $\mathbf{Sol}_\epsilon(\mathbf{P}, \vec{k})$.

***Main Theorem 2** (Margin-aware Isomorphism):* For all non-negative scalars ϵ, $(\{Sol_\epsilon(P_i, \vec{r_i})\}_n, \subset)$ is isomorphic to $(\{\mathbf{R_i}\}_n, \subset)$.

proof) Enough to show that $(\{Sol_\epsilon(P_i, \vec{r_i})\}_n, \subset)$ and $(\{Sol(P_i, \vec{r_i})\}_n, \subset)$ are isomorphic for all ϵ.

First, let's show

$$Sol(P_i, \vec{r_i}) \subset Sol(P_j, \vec{r_j}) \Rightarrow Sol_\epsilon(P_i, \vec{r_i}) \subset Sol_\epsilon(P_j, \vec{r_j}).$$

By Main Theorem 1 and Lemma 4,

$$Sol(P_i, \vec{r_i}) \subset Sol(P_j, \vec{r_j}) \Leftrightarrow \vec{r_j} = P_j\vec{r_i}, P_j = P_jP_i.$$

Thus, for all vector $\vec{b}$,

$$\begin{aligned}
&P_i(x - \vec{r_i}) = \vec{b} \\
&\Leftrightarrow P_jP_i(\vec{x} - \vec{r_i}) = P_j\vec{b} \\
&\Leftrightarrow P_j(\vec{x} - \vec{r_i}) = P_j\vec{b} \text{(Lemma 4)} \\
&\Leftrightarrow P_j(\vec{x} - \vec{r_j}) = P_j\vec{b} (P_j\vec{r_j} = \vec{r_j} = P_j\vec{r_i})
\end{aligned}$$

Thus, if $||P_i(\vec{x} - \vec{r_i})|| < \epsilon$, then $||P_j(\vec{x} - \vec{r_j})|| = ||P_j(P_i(\vec{x} - \vec{r_i}))|| < ||P_j(P_i(\vec{x} - \vec{r_i})) + (I - P)(P_i(\vec{x} - \vec{r_i}))|| = ||P_i(\vec{x} - \vec{r_i})|| < \epsilon. \cdots 1$

Now, let's show the converse. Assume $||P_i(\vec{x} - \vec{r_i})|| < \epsilon$ for some i. Then,

$$\begin{aligned}
||P_j(\vec{x} - \vec{r_j})|| &= ||P_j(\vec{x} - \vec{r_i}) + P_j(\vec{r_i} - \vec{r_j})|| \\
&= ||P_j(P_i(\vec{x} - \vec{r_i}) + (I - P_i)(\vec{x} - \vec{r_i})) + P_j(\vec{r_i} - \vec{r_j})|| \\
&= ||P_jP_i(\vec{x} - \vec{r_i}) + P_j(I - P_i)(\vec{x} - \vec{r_i}) + P_j(\vec{r_i} - \vec{r_j})|| \\
&\leq ||P_jP_i(\vec{x} - \vec{r_i})|| + ||P_j(I - P_i)(\vec{x} - \vec{r_i})|| + ||P_j(\vec{r_i} - \vec{r_j})||.
\end{aligned}$$

$||P_i(\vec{x} - \vec{r_i})|| < \epsilon$ bounds $||P_jP_i(\vec{x} - \vec{r_i})||$ to at most epsilon. However, because P, $(I - P)$ are orthogonal(Lemma 3) it tells nothing of $||(I - P_i)(\vec{x} - \vec{r_i})|| < \epsilon$, and the second term is unbounded.(Figure 5) The third term $||P_j(\vec{r_i} - \vec{r_j})||$ is unbounded as well, since $\vec{r_j}$ can be anything.

Thus, for $||P_i(\vec{x} - \vec{r_i})|| < \epsilon$ to bound $||P_j(\vec{x} - \vec{r_j})||$ at all for all $\vec{x}$,

$$P_j(I - P_i) = 0, P_j(\vec{r_i} - \vec{r_j}) = 0$$

need to hold. By Lemma 4 and 5,

$$P_j = P_j P_i \Leftrightarrow H_j \subset H_i$$
$$\Leftrightarrow Sol(P_i, \vec{r_i}) \subset Sol(P_j, P_j\vec{r_i}) = Sol(P_j, \vec{r_j}) \cdot \cdot 2$$

$|\{Sol_\epsilon(P_i, \vec{r_i})\}_n| = |\{Sol(P_i, \vec{r_i})\}_n|$ holds obviously; each $Sol(P_i, \vec{r_i})$ has a distinct $Sol_\epsilon(P_i, \vec{r_i})$ and each $Sol_\epsilon(P_i, \vec{r_i})$ also has a distinct "center" $(Sol(P_i, \vec{r_i})) \cdot \cdot 3$

From 1, 2, 3, the two sets are isomorphic.

A.2 Explanation on NELL Sport/ Location (section 5)

Here are the rules contained in NELL Sport/ Location, copied from [56] and [21].

Table A.1: Relations and Rules in Sport and Location datasets.

	Relations	**Rules**
Sport	AthleteLedSportsTeam	$(x, AtheleLedSportsTeam, y) \rightarrow (x, AthletePlaysForTeam, y)$
	AthletePlaysForTeam	$(x, AthletePlaysForTeam, y) \rightarrow (x, PersonBelongsToOrganization, y)$
	CoachesTeam	$(x, CoachesTeam, y) \rightarrow (x, PersonBelongsToOrganization, y)$
	OrganizationHiredPerson	$(x, OrganizationHiredPerson, y) \rightarrow (y, PersonBelongsToOrganization, x)$
	PersonBelongsToOrganization	$(x, PersonBelongsToOrganization, y) \rightarrow (y, OrganizationHiredPerson, x)$
Location	CapitalCityOfCountry	
	CityLocatedInCountry	$(x, CapitalCityOfCountry, y) \rightarrow (x, CityLocatedInCountry, y)$
	CityLocatedInState	$(x, StateHasCapital, y) \rightarrow (y, CityLocatedInState, x)$
	StateHasCapital	
	StateLocatedInCountry	

Appendix B

Appendix for Chapter 3

B.0.1 Fine-tuning BERT for Paraphrase Detection

We chose learning rate among $\{2e-5, 3e-5, 5e-5\}$, and trained for 5 epochs, stopping early at the highest validation accuracy.

B.0.2 Hyperparameter Selection

Hyperparameters consist of learning rate and α, β from Section 5. They were grid-searched iteratively; first, learning rate for the baseline model was grid-searched, and then α, β for each of the proposed models were grid-searched, with the learning rate fixed to what was found for the baseline. Finally, each of the proposed models' learning rates were grid-searched, with α, β fixed. emrQA's hyperparmeters were selected among $\alpha \in \{1, 0.1, 0.01\}, \beta \in \{1.25, 1, 0.75, 0.5\}$, learning rate $\in \{5e-4, 1e-3, 1.5e-3\}$; Overnight's hyperparameters among $\alpha \in \{1, 0.1, 0.01\}, \beta \in \{1.25, 1, 0.75, 0.5\}$, learning rate $\in \{1e-4, 3e-4, 5e-4\}$; Finally, CzEng 1.6's were among $\alpha \in \{1, 0.1, 0.01\}, \beta \in \{1.25, 1, 0.75, 0.5\}$, learning rate $\in \{1e-4, 3e-4, 5e-4, 7.5e-4\}$.

We also note that for each of emrQA, Overnight, and CzEng 1.6, models were trained up to 20, 50, and 100 epochs with early stopping at the epoch that returns best validation accuracy.

Bibliography

[1] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California, 2016.

[2] Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. Publicly available clinical BERT embeddings. *CoRR*, abs/1904.03323, 2019.

[3] Ben Athiwaratkun and Andrew Gordon Wilson. Hierarchical density order embeddings, 2018.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[5] Jonathan Berant and Percy Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[6] Rahul Bhagat and Eduard H. Hovy. What is a paraphrase? *Computational Linguistics*, 39:463–472, 2013.

[7] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32 Database issue:D267–70, 2004.

[8] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 2787–2795, USA, 2013. Curran Associates Inc.

[9] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.

[10] Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. In *Proceedings of the 2014th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECMLPKDD'14, pages 165–180, Berlin, Heidelberg, 2014. Springer-Verlag.

[11] Marco Damonte, Rahul Goel, and Tagyoung Chung. Practical semantic parsing for spoken language understanding. *CoRR*, abs/1903.04521, 2019.

[12] Cedric De Boom, Steven Van Canneyt, Steven Bohez, Thomas Demeester, and Bart Dhoedt. Learning semantic similarity for very short texts. In *2015 ieee international conference on data mining workshop (icdmw)*, pages 1229–1234. IEEE, 2015.

[13] Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. Lifted rule injection for relation embeddings. In *EMNLP*, 2016.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[15] Boyang Ding, Quan Wang, Bin Wang, and Li Guo. Improving knowledge graph embedding using simple constraints. In *ACL*, 2018.

[16] Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[17] Li Dong, Furu Wei, Ming Zhou, and Ke Xu. Question answering over Freebase with multi-column convolutional neural networks. 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 260–269, Beijing, China, July 2015. Association for Computational Linguistics.

[18] Allyson Ettinger, Rao Sudha, Daumé Hal, and M. Bender Emily. Towards linguistically generalizable nlp systems: A workshop and shared task. *ArXiv*, abs/1711.01505, 2017.

[19] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[20] Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *ACL*, 2013.

[21] Bahare Fatemi, Siamak Ravanbakhsh, and David Poole. Improved knowledge graph embedding using background taxonomic information. In *AAAI*, 2018.

[22] Samuel Fern and Mark Stevenson. A semantic similarity approach to paraphrase detection.

[23] Samuel Fernando and Mark Stevenson. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*, pages 45–52, 2008.

[24] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[25] Ian.J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[26] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics.

[27] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Jointly embedding knowledge graphs and logical rules. In *EMNLP*, 2016.

[28] Dichao Hu. An introductory survey on attention mechanisms in nlp problems. In Yaxin Bi, Rahul Bhatia, and Supriya Kapoor, editors, *Intelligent Systems and Applications*, pages 432–448, Cham, 2020. Springer International Publishing.

[29] Mohit Iyyer, , John Wieting, Kevin Gimpel, and Luke S. Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *NAACL-HLT*, 2018.

[30] Robin Jia and Percy Liang. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany, August 2016. Association for Computational Linguistics.

[31] Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4284–4295. Curran Associates, Inc., 2018.

[32] Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4284–4295. Curran Associates, Inc., 2018.

[33] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[34] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press, 2006.

[35] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[36] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[37] Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. Evaluating neural word representations in tensor-based compositional settings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–719, Doha, Qatar, October 2014. Association for Computational Linguistics.

[38] Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. Adversarial sets for regularising neural link predictors. *CoRR*, abs/1707.07596, 2017.

[39] Jeff Mitchell, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Extrapolation in nlp. *arXiv preprint arXiv:1805.06648*, 2018.

[40] Tom M Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, et al. Never-ending learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[41] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NIPS*, 2017.

[42] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, 2011.

[43] Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. emrQA: A large corpus for question answering on electronic medical records. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2357–2368, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[45] Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[47] Robert Roth Stoll. *Set theory and logic*. Courier Corporation, 1979.

[48] Gilbert Strang. *Linear algebra and its applications*. Thomson, Brooks/Cole, Belmont, CA, 2006.

[49] Yu Su and Xifeng Yan. Cross-domain semantic parsing via paraphrasing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1235–1246, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[50] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[51] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2071–2080. JMLR.org, 2016.

[52] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011.

[53] Ivan Vendrov, Jamie Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. *CoRR*, abs/1511.06361, 2015.

[54] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic embedding of knowledge graphs with box lattice measures. *arXiv preprint arXiv:1805.06627*, 2018.

[55] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: l 2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049. ACM, 2017.

[56] Quan Wang, Bin Wang, and Li Guo. Knowledge base completion using embeddings and rules. In *IJCAI*, 2015.

[57] Quan Wang, Bin Wang, and Li Guo. Knowledge base completion using embeddings and rules. In *IJCAI*, 2015.

[58] Yushi Wang, Jonathan Berant, and Percy Liang. Building a semantic parser overnight. 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 1332–1342, Beijing, China, July 2015. Association for Computational Linguistics.

[59] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[60] Zhuoyu Wei, Jun Zhao, Kang Liu, Zhenyu Qi, Zhengya Sun, and Guanhua Tian. Large-scale knowledge base completion: Inferring via grounding network sampling over selected instances. In *CIKM*, 2015.

[61] John Wieting and Kevin Gimpel. Revisiting recurrent networks for paraphrastic sentence embeddings. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[62] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics.