1 知识图谱构建过程

知识表示、知识抽取、知识融合、知识存储、知识推理

2 推荐系统特点

精确性、多样性、可解释性

3 知识图谱和专家系统的区别

知识图谱与传统专家系统时代的知识工程有着显著的不同。与传统专家系统时代主要依靠专家手工获取知识不同，现代知识图谱的显著特点是规模巨大，无法单一依靠人工和专家构建。

4 知识图谱和知识工程的区别

知识工程	知识图谱
主要依赖人工构建知识库，数据量一般在数万或数十万左右。	使用机器学习、自然语言处理等方法自动化构建，数据量都在数十到数百亿的量级上。
包括一阶逻辑谓词、产生式规则、描述逻辑等多种不同的知识表示方式，主要为了推理的应用。	主要以RDF三元组和属性图和分布式表示的方式来表示知识。
注重逻辑推理，包括确定性和不确定性推理。	注重事实知识的检索，也能完成一定的推理任务，可赋能智能搜索、智能问答、推荐系统等应用。

5 知识表示

定义：客观事物的机器标示，即知识表示首先需要定义客观实体的机器指代或指称。

作用：

与本体的关系：一组本体约定和概念模型，即知识表示还需要定义用于描述客观事物的概念和类别体系。

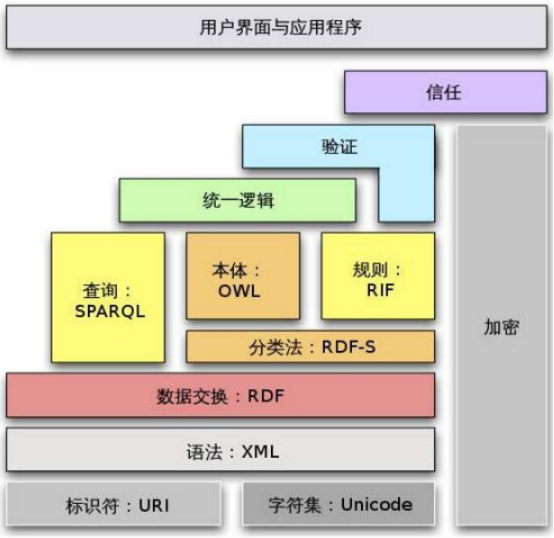
与推理的关系：支持智能推理的基础，即为机器推理的模型与方法奠定基础。

与数值计算的关系：用于高效计算的数据结构，即知识表示的数据结构要求可以由机器进行高效计算。

方便统一标识和人类认知：人可理解的机器语言，即知识表示还必须接近于人的认知，是人可理解的机器语言。

6 语义网络

- 语义网（Semantic Web）为知识表示提供了很好的应用前景
- 语义网的知识表示需要一套标准语言用于描述Web的各种信息
- W3C提供了新的标准语言：RDF，RDFS和OWL



W3C语义网标准栈

7 RDF

- 具体功能
  - 图表示：采用了基于三元组声明的图模型
  - 唯一标识：基于URI的可扩展词汇集
  - 信息交换：基于XML的序列化语法编码
  - 逻辑基础：形式化的语义和可证明的推论（描述逻辑）
  - 开放世界：允许任何人发表任何资源的声明

8 RDFS

- RDFS提供了最基本的对类和属性的描述元语
  - rdfs:type　　用于指定个体的类
  - rdfs:subClassOf　　用于指定类的父类
  - rdfs:subPropertyOf　　用于指定属性的父属性
  - rdfs:domain　　用于指定属性的定义域
  - rdfs:range　　用于指定属性的值域

9 OWL

- OWL1.0有三个子语言：OWL Lite，OWL DL和OWL Full

子语言	特　征	使用限制举例
OWL Lite	用于提供给那些只需要一个分类层次和简单的属性约束的用户	支持基数（cardinality），但允许基数为 0 或 1
OWL DL	在 OWL Lite 基础上包括了 OWL 语言的所有约束。该语言上的逻辑蕴涵是可判定的	当一个类可以是多个类的一个子类时，它被约束不能是另外一个类的实例
OWL Full	它允许在预定义的（RDF、OWL）词汇表上增加词汇，从而任何推理软件均不能支持 OWL Full 的所有 feature。OWL Full 语言上的逻辑蕴涵通常是不可判定的	一个类可以被同时表达为许多个体的一个集合以及这个集合中的一个个体。具有二阶逻辑特点

- OWL2：是OWL的最新版本
  - OWL2定义了OWL的子语言，通过限制语法使用，使得这些子语言能更方便地实现和应用
  - OWL2有三个子语言
    - OWL2 QL：面向基于本体的查询
    - OWL2 EL：面向概念术语描述、本体推理
    - OWL2 RL：推理复杂度为多项式时间

OWL2 QL词汇总结

允许的核心词汇	对应的描述逻辑公理举例
rdfs:subClassOf	Mother ⊑ Person
rdfs:subPropertyOf	hasSon ⊑ hasChild
rdfs:domain	∃hasSon.⊤⊑ Person
rdfs:range	⊤⊑ ∀hasSon.Person
owl:inverseOf	hasChild≡hasParent
owl:disjointWith	Women ⊓ Man ⊑ ⊥

OWL2 EL词汇总结

允许的核心词汇	对应的描述逻辑公理举例
rdfs:subClassOf	Mother ⊑ Person
rdfs:subPropertyOf	hasSon ⊑ hasChild
owl:someValuesOf	∃hasSon.Children ⊑ Person Parent ⊑ ∃hasSon.Children
owl:intersectionOf	Star ⊓ Women ⊑ Scandal
owl:TransitiveProperty	Tran(hasAncestor)

10 分布式表示

优点：显著提升计算效率、有效减少数据稀疏、便于多源数据融合

11 抽取原理

- 从关系数据库中抽取知识
  - 抽取原理：
    - 表（Table）-类（Class）
    - 列（Column）-属性（Property）
    - 行（Row）-资源/实例（Resource/Instance）
    - 单元（Cell）-属性值（Property Value）
    - 外键（Foreign Key）-指代（Reference）
  - 根据上述规则可将关系数据库转化为一个知识图谱。

12 DM映射

1.5 面向结构化数据的知识抽取

- 面向结构化数据的知识抽取：从数据库这种结构化数据中抽取知识
- 直接映射示例

1. 首先通过SQL语句创建如图所示的两个数据库表

表示人员和地址的两个数据库表

People 表

PK		→ Address(ID)
ID	fname	Address(ID)
7	Bob	18
8	Sue	NULL

Address 表

PK		
ID	city	state
18	Cambridge	MA

创建如上所示的两个数据库表的SQL语句

```
CREATE TABLE "Addresses" (
"ID" INT, PRIMARY KEY("ID"),
"city" CHAR(10),
"state" CHAR(2)
)

CREATE TABLE "People" (
"ID" INT, PRIMARY KEY("ID"),
"fname" CHAR(10),
"addr" INT,
FOREIGN KEY("addr")REFERENCES "Addresses"("ID")
)

INSERT INTO "Addresses" ("ID", "city", "state")VALUES (18, 'Cambridge',
'MA')
INSERT INTO "People" ("ID", "fname", "addr")VALUES (7, 'Bob', 18)
INSERT INTO "People" ("ID", "fname", "addr")VALUES (8, 'Sue', NULL)
```

1.5 面向结构化数据的知识抽取

- 面向结构化数据的知识抽取：从数据库这种结构化数据中抽取知识
- 直接映射示例

2. 基于直接映射标准，上述两个表可以输出如下的RDF数据

主语：由前缀和表名(People)、主键列名(ID)、主键值(7)串联而成的国际化资源标识符(IRI)。
谓词：由前缀和表名、列名连接形成的IRI。
值：是从列值的词汇形式的RDF文字。
每个外键都会生成一个三元组，其谓词由引用表和引用的列名组成。这些三元组的宾语是被引用三元组的行标识符(如<Addresses / ID = 18>)。
直接映射不会为NULL值生成三元组。

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

主语

```
<People/ID=7> rdf:type <People> .
<People/ID=7> <People#ID> 7 .
<People/ID=7> <People#fname> "Bob" .
<People/ID=7> <People#addr> 18 .
<People/ID=7> <People#ref-addr> <Addresses/ID=18>
<People/ID=8> rdf:type <People> .
<People/ID=8> <People#ID> 8 .
<People/ID=8> <People#fname> "Sue" .

<Addresses/ID=18> rdf:type <Addresses> .
<Addresses/ID=18> <Addresses#ID> 18 .
<Addresses/ID=18> <Addresses#city> "Cambridge" .
<Addresses/ID=18> <Addresses#state> "MA" .
```

13 条件随机场（CRF）

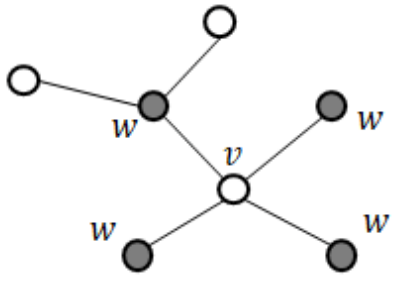
3 面向非结构化数据的知识抽取

- 基于统计模型的实体识别方法
- 模型训练：基于条件随机场(CRF)的实体识别方法

设$X=(X_1,\cdots,X_n)$是输入观测序列，$Y=(Y_1,\cdots,Y_n)$是输出标签序列$P(Y|X)$是在给定观测序列X条件下Y的条件分布，若

$$P(Y_v|X,Y_w,w\neq v)=P(Y_v|X,Y_w,w\sim v)$$

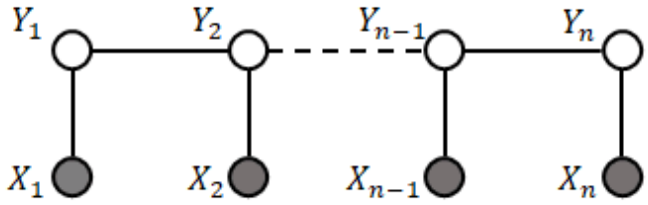
若上式对任意节点v都成立，则称条件分布$P(Y|X)$为条件随机场。其中$w\neq v$表示w是除v以外的所有节点，$w\sim v$表示在无向图$G=(V,E)$中w是与v相邻的所有节点。



3 面向非结构化数据的知识抽取

- **基于统计模型的实体识别方法**
- **模型训练：基于条件随机场(CRF)的实体识别方法**

对于文本序列，输入观测序列$X = (X_1, \cdots, X_n)$和输出标签序列$Y = (Y_1, \cdots, Y_n)$具有相同的结构，如图。



因此可以简化为线性条件随机场，即

$$P(Y_i|X, Y_1, \cdots, Y_{i-1}, Y_{i+1}, \cdots, Y_n) = P(Y_i|X, Y_{i-1}, Y_{i+1})$$

上式表示概率$P(Y_i)$取决于输入观测序列X，以及与$P(Y_i)$相邻的输出标签Y_{i-1}, Y_{i+1}

14 深度学习实体识别

3 面向非结构化数据的知识抽取

- **基于深度学习的实体识别方法**
- **优点：**
  - 具有强非线性映射能力，将原始数据映射成更高层次、更抽象表达。
  - 深度学习节省了人工设计特征的代价，传统基于特征的方法需要大量的工程技巧，而深度学习能从原始输入中自动学习特征
- **主要步骤：**
  - **输入数据的分布表示：**

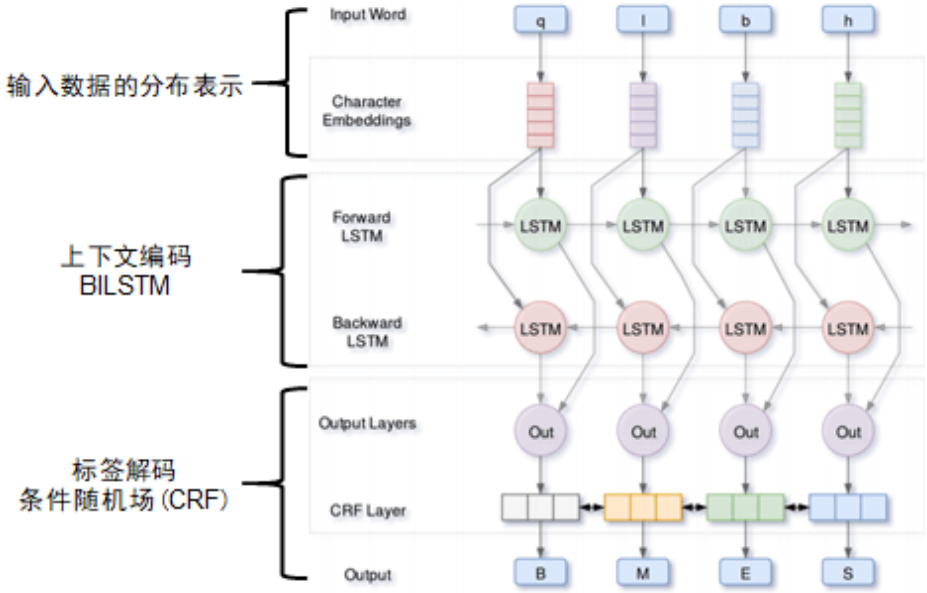
    利用深度学习网络，将输入文本无监督地表示为低维稠密的实值向量，实值向量的每一维表示文本潜在的语义特征。
  - **上下文编码：**

    利用深度学习网络记住文本上下文的局部或全局信息，形成中间语义，为后面输出标签序列的推测提供依据
  - **标签解码：**

    利用上下文编码得到的信息，推测最有可能的输出标签序列

3 面向非结构化数据的知识抽取

- 基于深度学习的实体识别方法
- 基于BiLSTM+CRF的实体识别模型框架：



3 面向非结构化数据的知识抽取

- 基于深度学习的实体识别方法

- 输入数据的分布表示

  词的分布表示模型Word2vec(Google,2013)，包括CBOW和Skip-gram

- CBOW(Continuous Bag-Of-Words Model)模型基本原理：

  将一个词 [如word(t)] 所在上下文的词 [如w(t-2)—w(t+2)，win_size=2] 的 one-hot 向量作为输入，预测 word(t) 的向量表示

  e.g. “The quick brown fox jumps over the lazy dog.”

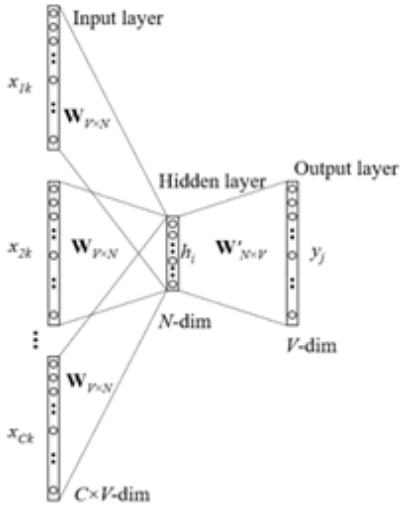
  输入[quick brown jumps over]预测fox

Word	One-hot vector
Quick	[0,1, 0, 0, 0, 0, 0, 0, 0, 0]
Brown	[0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
Jumps	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
Over	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]

3 面向非结构化数据的知识抽取

- 基于深度学习的实体识别方法
- 输入数据的分布表示

  词的分布表示模型Word2vec(Google,2013)，包括CBOW和Skip-gram
- CBOW(Continuous Bag-Of-Words Model)模型基本原理：

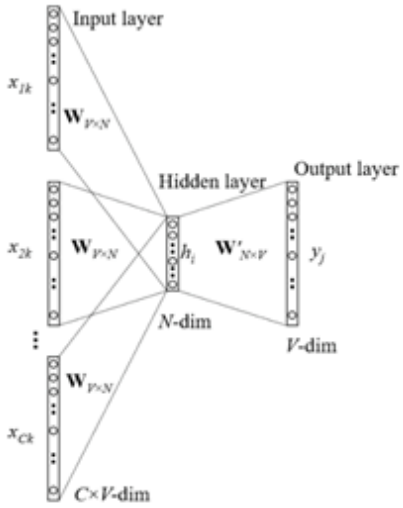


- ◆输入层：假定要预测的单词是fox，那么输入层为quick、brown、jumps、over，采用one-hot编码，分别为[0,1,0,0,0,0,0,0,0,0]、[0,0,1,0,0,0,0,0,0,0]、[0,0,0,0,1,0,0,0,0,0]、[0,0,0,0,0,1,0,0,0,0]
- ◆所有onehot 分别乘以共享的输入权重矩阵W. {V*N矩阵，N为自己设定的数，初始化权重矩阵W}，所得的向量相加求平均作为隐层向量, size为1*N.

3 面向非结构化数据的知识抽取

- 基于深度学习的实体识别方法
- 输入数据的分布表示

  词的分布表示模型Word2vec(Google,2013)，包括CBOW和Skip-gram
- CBOW(Continuous Bag-Of-Words Model)模型基本原理：



- ◆乘以输出权重矩阵W' 得到向量{1*V}，激活函数处理得到V-dim概率分布
- ◆概率最大的index 对应的单词为预测出的target word，与true label的onehot 向量做比较，误差越小越好（根据误差更新权重矩阵）
- ◆定义loss function（一般为交叉熵代价函数），采用梯度下降算法更新W和W'。
- ◆训练完毕后，输入层的每个单词与矩阵W相乘得到的向量的就是所需的词向量，即任何一个单词的onehot 乘以矩阵W都将得到自己的词向量。

15 Cypher

知识存储与检索PPT整个打印

16 知识图谱异构

1 知识图谱异构

- 定义：指知识图谱的语言层（language level）异构或模型层（model level或ontology level）异构。
  - 语言层异构：知识图谱语法（syntax）和表述（expressivity）方式上的异构。理论上可以分为四类：
    - 语法异构：采用不同描述语言。
    - 逻辑异构：逻辑表示不匹配。比如不相交可以表示为disjoint A B或A subclass-of （NOT B）。
    - 元语异构：元语的语义有差异，即同一个原语（如OWL和RDF Schema 中的<rdfs:domain>）定义不同。
    - 表达能力异构：不同语言表达能力差异

1 知识图谱异构

- 模型层异构：当融合的不同的本体描述相交或相关领域时出现的不匹配情况。理论上可以分为概念化不匹配和解释不匹配。
  - 概念化不匹配：概念范围或模型覆盖的不匹配。
    - 概念范围不匹配：同名概念在不同领域表示含义有差异。
    - 模型覆盖范围的不匹配：由于描述本体是不可避免地存在主观性，在模型的广度、粒度和对本体建模的观点上会出现的不匹配。
  - 解释不匹配：模型风格或建模术语的不匹配。
    - 范例不匹配：同一概念可以使用不同范例表述。
    - 建模术语的不匹配：对同义术语，同形异义或编码格式缺少转换规则或统一格式。

17 发现本体间映射算法

- 发现本体间映射的算法：
  - 基于术语
  - 基于结构
  - 基于实例

18 Dice系数

3.1.1 基于字符串的映射方法

- Dice系数：因为字符串可看作集合，因此可以使用衡量两个集合相似性的Dice距离度量字符串相似性。Dice系数定义：

$$sim_{Dice}(s,t) = \frac{2|S \cap T|}{|S| + |T|}$$

为计算字符串相似性，使用bigram（二元分词，按从头到尾的顺序把字符串每两个字符组成一个词语）形式计算相似度。

3.1.1 基于字符串的映射方法

◆以Lvensshtain和Levenshtein为例：

1.转换成bigram形式

Lvensshtain -> S: {Lv, ve, en, ns, ss, sh, ht, ta, ai, in}

Levenshtein -> T: {Le, ev, ve, en, ns, sh, ht, te, ei, in}

2.计算S、T集合的交集长度和集合大小

S∩T = {ve, en, ns, sh, ht, in}

|S∩T| = 6

|S|=10, T = 10

3.根据公式计算dice系数：2*6/(10+10)=0.6

19 Jaccard系数

3.1.1 基于字符串的映射方法

◆Jaccard系数：适合处理短文本的相似度，定义如下：

$$sim_{Jaccard}(s,t)=\frac{|S\cap T|}{|S\cup T|}$$

3.1.1 基于字符串的映射方法

◆以Lvensshtain和Levenshtein为例：

1. 转换成bigram形式

Lvensshtain -> S: {Lv, ve, en, ns, ss, sh, ht, ta, ai, in}

Levenshtein -> T: {Le, ev, ve, en, ns, sh, ht, te, ei, in}

2. 计算S、T集合的交集长度和集合大小

S∩T = {ve, en, ns, sh, ht, in} -> |S∩T| = 6

S∪T = {Lv, ss, ta, ai, Le, ev, te, ei, ve, en, ns, sh, ht, in}

|S∪T| = 14

3. 根据公式计算Jaccard系数：6/14=0.429

20 知识推理定义

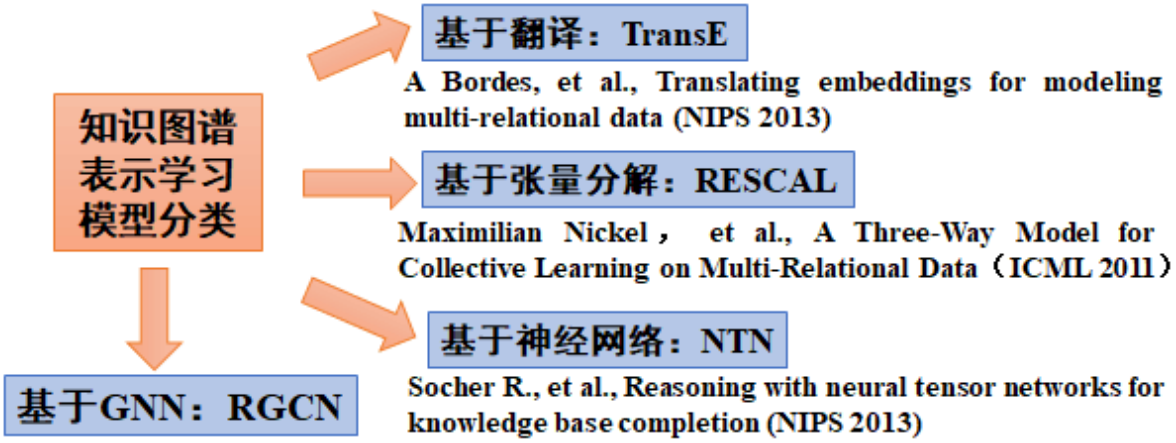
利用知识图谱中现有的显性知识预测图谱中尚未存储的隐形知识，逐步将知识图谱补充完整。

建议把知识推理整个PPT打印

21 知识图谱表示学习模型分类

1.2 KG Embedding 典型模型

■知识图谱表示学习方法的分类



- 基于翻译: TransH,TransR,TransD,TranSparse,TransG,RotatE …
- 基于张量分解：DisMult, HolE, ComplEx …
- 基于神经网络: ConvE, ConvKB …
- 基于图神经网络：KGAT, KBAT …

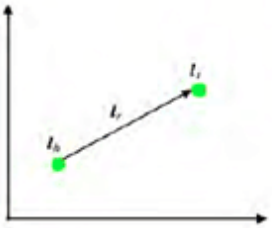
22 TransE

1.2 KG Embedding 典型模型

■TransE核心思想

- 对于每个三元组（h，r，t），将关系看成从头实体到尾实体的翻译，从几何上也看成是平移。$l_h + l_r \approx l_t$.
- 打分函数： $f_r(h,t) = |l_h + l_r - l_t|_{L_1/L_2}$

  距离函数　衡量翻译的好坏



- 模型训练过程中，TransE采用最大间隔方法的hinge loss。TransE 的 Loss function:

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r',t')\in S^-} \max(0, \gamma + f_r(h,t) - f_{r'}(h',t'))$$

KB中的正样本　负采样得到的负样本

$$\max(0,x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

1.2 KG Embedding 典型模型

■TransE算法实现过程

Algorithm 1 Learning TransE

input Training set $S = \{(h, \ell, t)\}$, entities and rel. sets E and L, margin γ, embeddings dim. k.

initialize $\ell \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$

  $\ell \leftarrow \ell / \|\ell\|$ for each $\ell \in L$

  $\mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$

loop

  $\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$

  $S_{batch} \leftarrow \text{sample}(S, b)$ // sample a minibatch of size b

  $T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets

  for $(h, \ell, t) \in S_{batch}$ **do**

    $(h', \ell, t') \leftarrow \text{sample}(S'_{(h,\ell,t)})$ // sample a corrupted triplet

    $T_{batch} \leftarrow T_{batch} \cup \{((h, \ell, t), (h', \ell, t'))\}$

  end for

  Update embeddings w.r.t. $\sum_{((h,\ell,t),(h',\ell,t')) \in T_{batch}} \nabla [\gamma + d(h + \ell, t) - d(h' + \ell, t')]_+$

end loop

1.2 KG Embedding 典型模型

■TransE算法实现过程

Algorithm 1 Learning TransE　　实体集合　　边际参数　　向量维度

input Training set $S = \{(h, \ell, t)\}$, entities and rel. sets E and L, margin γ, embeddings dim. k.

initialize $\ell \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each $\ell \in L$　　关系集合

  $\ell \leftarrow \ell / \|\ell\|$ for each $\ell \in L$

  $\mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ for each entity $e \in E$

loop

  $\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|$ for each entity $e \in E$

  $S_{batch} \leftarrow \text{sample}(S, b)$ // sample a minibatch of size b

  $T_{batch} \leftarrow \emptyset$ // initialize the set of pairs of triplets

  for $(h, \ell, t) \in S_{batch}$ **do**

    $(h', \ell, t') \leftarrow \text{sample}(S'_{(h,\ell,t)})$ // sample a corrupted triplet

    $T_{batch} \leftarrow T_{batch} \cup \{((h, \ell, t), (h', \ell, t'))\}$

  end for

  Update embeddings w.r.t. $\sum_{((h,\ell,t),(h',\ell,t')) \in T_{batch}} \nabla [\gamma + d(h + \ell, t)$

end loop

初始化 l, e

从知识库中采取minibatch得到样本

负采样，得到三元组对，构成训练样本

最小化目标函数，计算梯度，参数更新