1 引言

1.1 现有研究

Resource Description Framework (RDF)：从思维到自然语言到图谱，是人们对智慧的整理过程。互联网内容的大规模、异质多元、组织结构松散的特点，给人们有效获取信息和知识提出了挑战。现在的大型知识图谱以其强大的语义处理能力和开放组织能力，为互联网时代的知识组织和智能应用奠定了基础。但知识图谱能否直接帮助用户管理自己的知识呢?

1.2 本研究

1.2.1 思想和想做的事

跨领域（包括跨语言）的重要性：
从现有研究来看，我关注到了知识图谱应用的一个问题：共用还是私用?
共用数据需要达到的效果侧重于：link prediction（增强搜索功能），kg 主要在做整理知识的工作。比起用知识专家，让 AI 用写好的知识进行传播效率更高。
私用数据则侧重于：知识抽取、表示、融合、推理，kg 主要在做挖掘资源的工作。
但比起知识专家，AI 专家的作用有多少呢? 即便 AI 一直在给予知识专家辅助，但知识专家永远可以指责 AI 专家给的知识没有用。除非 AI 脱离了人类智慧，否则这一指责永远不会失败。
人类对事物及其关系的认知是个黑匣子，AI 是对这个黑匣子的量化学习。向一个专家学习，AI 或许只能成为效能为 10% 的智能专家，但向 100 个专家学习，最大可达到 1000% 的效能——“跨领域学习”是让 AI 应用强势的关键。

强调路径而非 entity 的重要性：
另一方面，所谓思维，就是找到解决问题最近的路。AI 胜在可以同时考虑各种已知因素、从已知路径中找到最近的，人的思维在于懂得创造，如果可能的话，他们能创造出新的路——捷径。而在实际问题中，由于状况千变万化，大部分时候都需要这样一些“创新”。AI 能创造“捷径”吗?
现在，AI 与人类的关系看起来就好像父母与孩子，当孩子懵懂无知时，AI 提供的知识就可以让人受益匪浅，是最不偏心、偏颇的知识提供者；当孩子成为了专家可以独当一面时，AI 所提供的知识开始显得力不从心。
但 AI 所提供的重点不是知识本身，而是路径，AI 是最好的引领者。将来 AI 和人类的关系，可以是 DNA 和人类的关系。它能将信息分割成合适的单元，对这种单元的新组合（未知信息），也有早就写好的计算法则应对，不断接受信息的同时，因为反应迅速而一直站在指导人类的地位。这

对人类专家也是有好处的，顾虑周全的 AI 的可以让人类专家更“专”更“快”。
我想提出的概念，不是知识图谱，而是知识路径。因为人的任何一步，都受到了细思极恐般的无限的限制：生命、记忆、思维能力、技能、环境……总的来说，就是自己的位置和各种资源总量的限制。这种给予本体的推理对于资源预测和挖掘即位重要。
“先适应”和“最适化”的区别在于后者是计算现有条件下的最优选择，即现在的结果是最优；而“先适应”是指使选择做出后、下一步的结果最优。(但二者都属于自适应算法 self-adaptive 的研究)

强调 end-to-end 的重要性：
让数据能自我判断的空间更大，简化和优化人工操作。

强调 auto-structured 的难点：
人类脑中的世界是 continous 的，但语言是 discrete。sturcture data 本身是把语言更加 structure 化、强 structure 化，对专家而言仍然会被理解、且能节省时间，但对普通人而言，会使内容本身更难以被理解。
理解，一部分是对文字意思但理解，更多的则是对其组成和相互作用的理解。前者称为语义，后者称为知识。机器可以很好的理解明确定义的语义，但很难理解没有明确定义的知识。
一开始是不存在概念的，当区别出现后，就自然会聚类出概念以分类。

1.3 目标

“学以致用”，对于机器更是如此，因此，我认为以下 AI 应用研究很具有未来性：end-to-end 跨领域（包括跨语言）的知识图谱构建、和各资源限制下的先适应知识路径提供（基于本体推理 ontology reasoning）。
研究的指导思想是：让自己时刻都是被需要的。
我的应用目标是，能自动挖掘资源的 kg：vita.
1. 它是全方位的活着的人类记忆（思维，用与 kg 配套的立体空间 space 表示）。有这个目标的原因是，我相信总有一天，人类被虚假和针对性营销包围，会渴望找到和立体化自己的记忆的。那时的 AI 就不再只是工具和应用，而是结果和需求本身了。
2. 它也是抽象但足够被人类理解的语言（kg）。人对 entity 的理解就是对物的理解，但人希望了解的其实是事，所以所谓“事”该如何表示？就好像 DNA 一样，有了单词，如何连线呢？连成线的 entity，就成为了“事”。物是客观证据，事是人为想象，人和机器的理解（认知和表现力）会因所依据的 kg 不同而改变。即便不能比现有的自然语言更丰富多彩，但人类可以把语言这门艺术简化、或者说抽象化，形成一种更容易达成共识的机器语言——kg 语言。这种语言会是编程语言和自然语言的友善过渡，也是世界通用的充满智慧的语言。而掌握这种语言的 nlp 专家，将担任未来知识的导航者、甚至是先知，因为未来的知识必将更加庞大而精细，超出任何人的时间

限制，而 nlp 专家必须保证能领导人们拥有担任某项工作足够的知识。

1.3.1 方法

试图做的研究：
多源抽取融合：用词向量相加和 CNN 两种方法分别考虑文本的拓展表示和词序信息，比如 DKRL 可以坐 kbc 知识融合，再比如利用跨语言优势高效学习。
复杂关系表示：但不同的知识要有意义，所需的 kg 结构并不相同，比如树状或网状，所以融合手段是有限的，且最好要有机抽取融合。比如 distant supervision, open information extraction, CNN
path-infer：路径可以用来表达语义吗？相加、位乘、RNN 哪种算法可以提高 path 的可靠性？目前是相加最好。比如 path-constraint random walk, path ranking algorithm，path-based TransE (PTransE) 利用 path 预测关系

试图做的应用：
制作可视化工具或算法，让 embedding 更直观
用 kg 和 embedding 直接理解文章
用 kg 做私人活动监视器，让人们了解自己的信息摄入状况
用 kg 尝试做优于人类精算师的保险计算

2 embedding model with more rationality and interpretability

2.1 learning of existing methods

独热表示 one-hot 之后，现在主要研究的是分布式表示 distributed representation，难点在于增强实体和关系之间的联系。
为便于理解和比较不同模型的想法，以下尽量采用了统一的表示方法。其中，map 所表示的目前都是使用 linear map 进行计算的，而 non-linear map 的部分都用特定函数直接表示出来了。即我所写的 map 部分其实也可用 matrix 表示，之所以不写成 matrix，是因为考虑到将来的研究中，现在的 linear map 也可能拓展到 non-linear map, 所以，我只特别表示出了 non-linear map 的部

分，而其它部分则仍用 map 表示：

$$
\begin{aligned}
&entities \in \mathbb{R}^d \\
&knowledge = (entities, links, triples) \\
&triple = (head, link, tail) \\
&triple^- = (head, link^-, tail) \bigcap (head^-, link, tail) \bigcap (head, link, tail^-)
\end{aligned}
$$

2.1.1 距离模型

structured embedding (SE)：重要缺陷：head，tail 的 map 用两个不同的矩阵，协同性较差。

$$
\begin{aligned}
&map_1, map_2 \in \mathbb{R}^{d \times d} \\
link &= \operatorname{argmin} distance \\
&= \operatorname{argmin} |map_1 \times head - map_2 \times tail|_{L_1}
\end{aligned}
$$

2.1.2 单层神经网络模型

single layer model (SLM)：只增强了微弱的联系，且 $\tanh$ 的计算复杂度高。

$$
\begin{aligned}
&link^T \in \mathbb{R}^k \\
&map_1, map_2 \in \mathbb{R}^{d \times k} \\
&link = \operatorname{argmin} link^T \times \tanh(map_1 \times head + map_2 \times tail)
\end{aligned}
$$

2.1.3 语义匹配能量模型

semantic matching energy (SME): 也是为了增强联系，计算更复杂了。

$$
\begin{aligned}
&\otimes : A_{ij} \otimes B_{ij} = \{a_{ij} \times b_{ij}\}, \text{Hadamard product} \\
link &= \operatorname{argmin} (map_1 \times head + map_2 \times link + bias_1)^T \\
&\quad (map_3 \times tail + map_4 \times link + bias_2) \\
link &= \operatorname{argmin} (map_1 \times head \otimes map_2 \times link + bias_1)^T \\
&\quad (map_3 \times tail \otimes map_4 \times link + bias_2)
\end{aligned}
$$

2.1.4 双线性隐变量模型

Latent factor model (LFM): 协同性较好，计算复杂度低。

$$map \in \mathbb{R}^{d \times d}$$
$$link = \operatorname{argmin} head^T \times map \times tail$$

Dismult：计算大大减小，效果反而显著提升。

$$map \in D^{d \times d}, \text{diagonal matrix}$$

2.1.5 张量神经网络模型

Neural tensor network (NTN)：这里的实体向量是该实体中所有单词向量的平均值。这样可以充分重复利用单词向量，降低稀疏性问题，但计算复杂度非常高，难以扩大规模。

$$map_t \in \mathbb{R}^{d \times d \times k}, \text{third} - \text{order tensor}$$
$$map_1, map_2 \in P^{d \times k}, \text{projection matrix}$$
$$link = map^T \times \tanh(head \times map_t \times tail + map_1 \times head + map_2 \times tail + bias)$$

2.1.6 矩阵分解模型

RESACL：不仅优化存在 link 的位置，也优化不存在 link 的位置。

$$\exists link : head \times map \times tail = 1$$
$$\nexists link : head \times map \times tail = 0$$
$$\text{i. e.}$$
$$link = \operatorname{argmin} head \times map \times tail - 1$$
$$link^- = \operatorname{argmin} head \times map \times tail$$

2.1.7 翻译模型

TransE：根据平移不变现象，捕捉单词之间相同的隐含语意关系，相当于用相对的 link 去解释/翻译单词。随机替换 $triple$ 的人一元素以创建 $triple^-$，继续考虑了错误 link 的情况。参数较少，计

算复杂度低，在大规模稀疏 kg 上效果尤其惊人。因此成为以后大部分 kge 的基础模型。

$$
\begin{aligned}
&head + link = tail \\
&\text{e. g.} \\
&C(king) - C(queen) \approx C(man) - C(woman) \\
&\text{i. e.} \\
&link = \operatorname{argmin} |head + link - tail|_{L_1 \text{or} L_2} \\
&\text{for further optimization,} \\
&triple^- = (head, link^-, tail) \bigcap (head^-, link, tail) \bigcap (head, link, tail^-) \\
&Link = \sum_{triple} \sum_{triple^-} \max(0, link - link^- + triple - triple^-)
\end{aligned}
$$

Holographic embeddings (Hole)：让 head 和 tail 循环相关 circular correlation (an operator that combines the expressive power of the tensor product with the efficiency and simplicity of TranE.) 此操作的优点：1）不可交换性，许多 kg 中的 link 都是不可交换的；2）相关性，循环相关的得到的向量每一维都限制了 head 和 tail 的某种相似性，例如第一维相当于 head 和 tail 的内积，这利于更清楚地分清比较相似的关系；3）计算效率高，可用快速傅里叶变换加速计算。

$$
\begin{aligned}
&\star : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d, \text{circular correlation.} \\
&[head \star tail]_k = \sum_{i=0}^{d-1} head_i \times tail_{(i+k) \mod d} \\
&\text{for further optimization,} \\
&head \star tail = \text{Fourier}^{-1} \left(\overline{\text{Fourier}}(head) \odot \text{Fourier}(tail) \right) \\
&link = \operatorname{argmin} \sigma(link^T (head \star tail))
\end{aligned}
$$

TransH：处理 1-N，N-1，N-N 的复杂关系。如果 map 后存在无限个 link 超平面，则选取与 map 近似正交的 link 超平面。

$$
\begin{aligned}
&map : \text{normal vector of head and tail} \\
&link = \operatorname{argmin} |(head - map^T \times head \times map) + link - (tail - map^T \times tail \times map)|_{L_1 \text{or} L_2}
\end{aligned}
$$

TransR：使实体和关系可以 map 到不同的 space，再在不同 space 中建立 link。

$$
\begin{aligned}
&map \in \mathbb{R}^{d \times k} \\
&link = \operatorname{argmin} |head \times map + link - tail \times map|_{L_1 \text{or} L_2}
\end{aligned}
$$

CTransR：对 head 和 tail 的差进行聚类，从而将 link 细分成子关系 $link_{child}$，每个子关系分别学

习向量表示。

$$link = \text{argmin}\,|head \times map + link_{child} - tail \times map|_{L_1 \text{or} L_2}$$

TransD：认为让 map 仅和 link 有关系是不合理的，还是应该和 entity 有关；且 TransR 引入不同 space 导致参数急剧增加、计算复杂度大大提高。因此 TransD 设计了两个 map，让 map 与 entity 和 link 都相关，同时减小计算复杂度。

$$\begin{aligned}
&head_{project}, tail_{project} \in \mathbb{R}^d \\
&link_{project} \in \mathbb{R}^k \\
&map_{head} = link_{project} \times head_{project} + I^{d \times k} \\
&map_{tail} = link_{project} \times tail_{project} + I^{d \times k} \\
&link = \text{argmin}\,|head \times map_{head} + link - tail \times map_{tail}|_{L_1 \text{or} L_2}
\end{aligned}$$

TranSparse：认为 map 和 link 的异质性、link 之间的不平衡性是困难所在。因此定义稀疏度 sparsityθ，并考虑到 head 和 tail 的稀疏度不同。

$$\begin{aligned}
&\theta_{\min} \in [0,1], \text{hyperparameter of sparsity} \\
&\theta = 1 - (1 - \theta_{\min}) N_{link} / N_{maxlink} \\
&link = \text{argmin}\,|head \times map_{head}(\theta_{head}) + link - tail \times map_{tail}(\theta_{tail})|_{L_1 \text{or} L_2}
\end{aligned}$$

TransA：认为只用 L_1 或 L_2 距离不够灵活，且 entities，map 和 link 都每一维都被等同考虑了。因此提出用马氏距离 Mahalanobis distance，它是表示点与一个分布之间的距离，并让每一维学习不同的权重。

$$\begin{aligned}
&map: \text{weight matrix related to link} \\
&link = (head + link - tail)^T \times map(head + link - tail)
\end{aligned}$$

TransG：不用子关系 $link_{child}$ 来细分 link，而将在每个情况下的 link 的不同都用一个高斯分布 Normal distribution 来表示。如此就可以按照分布区分出同一个 link 的不同作用（与细分成子 link 不同），从而减少错误预测。

$$\begin{aligned}
&link = tail - head \\
&link \sim \sum_{m=1}^{M} \pi_{link,m} N(\mu_{link,m}, I)
\end{aligned}$$

KG2E：认为不论是 entity 还是 link 都是不确定的。因此用高斯分布的均值表示 entity 或 link 的 center，用高斯分布的协方差表示该 entity 或 link 的不确定度。并用 KL 距离（不对称相似度）和

期望概率（对称相似度）两种方法来判定 entity 和 link 的概率相似程度。

$$
\begin{aligned}
&link = head - tail \\
&probability_{entity} \sim N(\mu_{head} - \mu_{tail}, \sum_{head} + \sum_{tail}) \\
&probability_{link} \sim N(\mu_{link}, \sum_{link}) \\
&\forall l \in E \bigcup R, c_{\min} I \leq \sum_{l} \leq c_{\max} I, c_{\min} > 0. \text{to prevent overfitting} \\
link &= \operatorname{argmin} \int_{x \in \mathbb{R}^{k_e}} N(x; \mu_{link}, \sum_{link}) \log \frac{N(x; \mu_{entity}, \sum_{entity})}{N(x; \mu_{link}, \sum_{link})} dx \\
&= \frac{1}{2}(tr(\sum_{link}^{-1} \sum_{entity}) \\
&\quad + (\mu_{entity} - \mu_{link})^T \sum_{link}^{-1} (\mu_{entity} - \mu_{link}) \\
&\quad - \log \frac{\det(\sum_{entity})}{\det(\sum_{link})} + k_e) \\
link &= \operatorname{argmin} \int_{x \in \mathbb{R}^{k_e}} N(x; \mu_{link}, \sum_{link}) N(x; \mu_{entity}, \sum_{entity}) dx \\
&= \frac{1}{2}((\mu_{entity} - \mu_{link})^T (\sum_{entity} + \sum_{link})^{-1} (\mu_{entity} - \mu_{link}) \\
&\quad + \log(\det(\sum_{entity} + \sum_{link})) + k_e \log 2\pi)
\end{aligned}
$$

2.2 my idea and contribution

2.2.1 my idea

enrich words representation: sequential and spatial information, word root and it's character length

本设计对 relation 的概念进行了调整，具体为：

本设计中的图谱仅由 entity 和运算法则组成，而不另外规定 relation，以提高协同性。本设计将 relation 看作由 entity 之间的运算而形成的，且分为两种，一种是 entities 之间的渐变关系，对应 map 运算；一种是 entities 之间的社会联系，对应 link 运算（就是普通 kg 中的 relation），用 space 的共键关系来表示。

由此，增强“实体与关系之间的联系”问题，就可以更清楚地分为探索“实体与实体之间的 map 计算”和“map 与 link 之间的计算”这两个更单纯的子问题。

在具体计算方面，本设计考虑了以下几点：

entity：人需用名字来记住和代表一段记忆，包括受到环境限制的认知和表现

实体与实体之间的 map 的计算关键是：认知、表现和环境限制

map 与 link 之间的计算关键是：map 和 link 同处在一个环境限制中

那么问题就来了：要证明 map 和 link 的环境限制相同的话，得将二者表示出来，然后发现其中的关系。但如果想直接认为二者的环境限制相同（而不加以证明）的话，就应该将二者直接嵌合在一个模型中表示出来，进行 embedding 的学习。如此学习到的 embedding 不仅可以应

用于通俗文章，也可以灵活应对各种话题、场景、专业性等环境限制的文章。本文想做的事是后者。

距离就算一样，方向也会不一样。所以 map 和 link 不能只看数量关系，也必须看 spatial 关系。spatial 关系一半用 map 中的 hyper space 来表示，另一半则用 link 中的 probability 表示。不同的词语所面临的 environmental restriction 不同会导致用词的概率分布不同。
将 topic model 按照 environmental restriction model 来应用。我认为二者在方法上是相通的，只是在解释性方面，我希望自己能把它的概念描述得更准确。不同的是 topic 强调的是可能会出现的词语的概率，而我想考虑的 environmental restriction 强调的是可能出现的词语的概率，和可能不出现的词语的概率。

2.2.2 my contribution

1. the hypothesis
阐述 IF 图，并说明应用于各行各业时可以简化或更细化的操作。

2. the notation

$$knowledge = (entities, +, \times)$$
$$university = (dstudent + dteacher) = (denglish + djapanese) \times map$$

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \sim p(x|y)p(y)$$

3. the link concept
从过去到现在、从现有到将来的过渡关系。

$$(head_1, link_1, tail_1)$$
$$(head_2, link_2, tail_2)$$
$$head_1 \times map_1 \times map \times map_2 = tail_1$$
$$map = \frac{tail_1}{head_1 \times map_1 \times map_2}$$
$$link_1 = \tanh(map)$$
$$head_2 \times map_2 \times map \times map_1 = tail_2$$
$$map = \frac{tail_2}{head_2 \times map_2 \times map_1}$$
$$link_2 = \tanh(map)$$
$$link = (1 - \alpha) \times link_1 + \alpha \times link_2$$

$head_i, tail_i, map_i, link_i$ have consistency

map in a single dataset has consistency

$\exists link$:

$head_1 \times map_1 \times tail_1 = 1$

$head_1 \times map_1 \times map \times tail_1 = 1$

$head_1 \times map_1 \times map \times map_2 \times tail_1 = 1$

$\nexists link$:

$[1, 512][512, 512][512, 512][512, 512][512, 1]$