

本科毕业论文文献综述

1 国内外发展现状

早期的问答系统例如 Baseball[4] 是针对特定的知识领域所设计的可扩展性比较差, 适用范围也比较小。随着互联网的发展, 研究人员可用的语料信息的增加。问答系统的发展得到了极大的提升, 先后提出了基于逻辑推理的方法 [11]、基于模版匹配的方法 [13]、基于机器学习的方法 [15] 和基于数据冗余的方法 [8]。这些模型主要利用信息检索或浅层语义理解技术去从大量候选集中寻找答案从而构建智能问答系统, 但是检索式问答技术存在一个缺陷, 就是答案中一定至少包含一个用户问句中含有的字或者词, 但是这在实际情况中往往是不成立的。在此阶段, 阻碍问答系统进一步发展的主要困难是高质量的数据集和自然语言处理技术。随着互联网的进一步发展, 高质量的语料得到不断积累。而统计机器学习方法的兴起, 自然语言处理技术各个子领域都取得了很大的进步, 阻碍问答系统最大的两个问题正在逐步解决。正因如此, 很多基于问答系统的产品也逐渐问世, 例如: 苹果的 Siri、谷歌的 Google Assistant 和微软的 Cortana。

2 研究方向

实现一个完整的问答系统主要需要三个部分, 分别是用户问句的语义理解、信息检索和表示以及对知识进行推理, 从而得到最终答案。

2.1 问句理解

这个环节实际上要解决的问题是如何将自然语言最准确地转化为计算机可以表示和理解的形式。主要有两种传统的方法来实现, 分别是语义解析的方法和基于信息检索的方法。

基于语义解析的方法中目前最主流的是组合范畴语法 CCG[7][17]。CCG 首先自然语言问句中的词汇被映射到语义表达式中的词汇, 然后按照特定的语法规则将词汇组合起来, 进而得到了最终的语义表达式。

基于信息检索的方法,首先使用分词、实体识别等 NLP 技术找到问句中所涉及到的实体和关键词,然后去知识资源库中去进行检索。

2.2 知识图谱的构建

知识图谱的构建,主要是将互联网上存在的大量非结构化的文字语料,结构化,并且存储到知识数据库中。在这个过程中,我们需要先对文本进行实体识别,然后对实体进行分类和消歧,接着抽取出关系和事件,这样就能构建一个由实体、关系和事件组成的知识数据库。目前代表性的工作有:TextRunner[16]、Wanderlust[1]、NELL[7]等。

2.3 知识推理

早期的知识推理方法大多对现有知识归纳学习出符号逻辑的推理规则,比如 PRA[9]. 这种方法随着推理规则的数量随着关系的数量指数增长,因此很难扩展到大规模知识图谱中去。而随着神经网络等发展,基于 Memory 和 Attention 机制等神经网络,在知识归纳推理方面取得了令人瞩目的结果。这也是我这次毕设的研究基础。

3 关键技术

3.1 Word2Vec

Word2Vec(Mikolov et al., 2013[10]) 是一种表达词汇的方法,将一个单词用一个向量来表示,关系比较近的词(例如: king 和 queen, Shanghai 和 Beijing)在向量空间相对应的有比较近的距离。如何确定词的关系,一个朴素的想法就是在相似上下文中出现的单词关系比较近。Word2Vec 就是基于这一想法的一种高效的词汇表达方法。特别的,Word2Vec 有两种算法,分别是 Continuous Bag-of-Words(CBOW) 和 Skip-Gram。CBOW 是根据环境词预测中心词,而 Skip-Gram 是根据中心词预测环境词。

3.2 GloVe

尽管 Word2Vec 已经被实验证明可以发掘出复杂的语义相似关系,但是是根据局部上下文来预测,往往忽略了整体的信息。与之对应的, Pennington et al.[12] 提出的 Global Vectors for Word Representation (GloVe) 使用全局的统计信息预测单词 i 出现在单词 j 的上下文中的概率。具体来讲, GloVe 使用最小二乘作为目标函数训练词词共生矩阵。GloVe 在很多词汇相似性任务中已经表现出优越的结果。

3.3 递归神经网络(RNN)

递归神经网络是一种特别适合对时序信息建模的神经网络架构。在 t 时刻, RNN 输入词矩阵 w_t 和上一步的隐式状态向量 h_{t-1} 进行下面的运算后,产生这一步点隐式状态向量:

$$h_t = f(Wx_t + Uht - 1 + b) \quad (1)$$

上式中 W, U 和 b 是这个 RNN 的参数。使用 RNN 对语段建模,能够提取到当前词的前面的所有词的信息考虑在内。

3.4 GRU

尽管 RNN 能够考虑时序关联信息,但是由于使用 backpropagation 进行优化目标函数时,求导产生的链式相乘,会导致,随着递归向前,产生权重指数级爆炸或消失的问题,难以捕捉长期时间关联。对于权重指数暴炸问题,可以简单的采用超出阈值后截断的方法,就能产生很好的结果。而 Chung et al.[3]Gated Recurrent Units(GRU) 是一种激活单元,就是为了解决权重指数消失问题,对网络结构进行的优化。下面的一组公式表明了 GRU 如何根据 h_{t-1} 和 x_t 产生 h_t :

$$z_t = \sigma(W^{(z)}x_i + U^{(z)}h_{t-1}) \quad (2)$$

$$r_t = \sigma(W^{(r)}x_i + U^{(r)}h_{t-1}) \quad (3)$$

$$\tilde{h}_t = \tanh(r_t \odot U h_{t-1} + W x_t) \quad (4)$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1} \quad (5)$$

式 (2) 确定了 h_{t-1} 应该多大程度被带入下一阶段。式 (3) 表明 h_{t-1} 多大程度上影响 \tilde{h}_t 。式 (4) 表明新的记忆 \tilde{h}_t 是由输入 x_t 和 h_{t-1} 确定的。式 (5) 表明隐式状态 h_t 最终是由以前的隐式状态 h_{t-1} 和 \tilde{h}_t 新的记忆组成的。

3.5 LSTM

Long Short Term Memories(LSTM)[5] 是另一种与 GRU 相似的激活单元, 与 GRU 起到的作用相同。下面一组式子表明了 LSTM 的数学结构:

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \quad (6)$$

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \quad (7)$$

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \quad (8)$$

$$\tilde{c}_t = \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \quad (9)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (10)$$

$$h_t = o_t \odot \tanh(c_t) \quad (11)$$

式 (6) 使用输入词和之前的隐式状态来判断输入词是否值得保存, 用来产生新记忆。式 (7) 用来评估过去的记忆是否对于计算现在的记忆有用。式 (8) 确定了最终记忆应该多大程度上保存到隐式状态中。式 (9) 根据输入词 x_t 和 h_{t-1} 产生新的记忆。式 (10) 根据新记忆和 h_{t-1} 产生最终记忆。

3.6 Attention 机制

一句话中的不同的单词的重要性是不同的, 例如名词、动词在就会比定冠词介词表达更多的意思。Bahdanau et al[2] 注意到了使用单向量的 RNN 的最终状态没有办法很好的表征不同输入的部分有不同重要性这一特征。更进一步的, 输出的不同部分可能更看重输入的特定部分。例如, 在翻译中, 输出的前几个单词通常是根据输入的前几个单词, 而输出的后几个单词则更多的依赖于输入的后几个单词。

针对这一特点, **Attention** 机制在编码的每一步, 首先观察整个输入序列, 然后决定在这一阶段, 哪一部分比较重要。

Bahdanau et al. 提出的模型主要分成两个部分, 编码部分和解码部分。编码部分对输入序列编码为向量, 解码部分提取编码信息解码成语句, 输出语句有一系列单词组成 y_1, \dots, y_m 。

3.6.1 编码器

令 (h_1, \dots, h_n) 为每个输入句子的隐式向量, 这些向量可以使用 **GRU** 或者 **LSTM** 来产生, 包含了词在上下文中的表达信息。

3.6.2 解码器

我们可以通过下面的递归式计算隐式状态 s_i

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \quad (12)$$

在这里 s_{i-1} 是前一个隐式状态, y_{i-1} 是上一个输出单词, 上下文向量 c_i 是在解码的第 i 步根据解码状态对原文全文信息等提取表示。

对于输入序列中的隐式向量, 计算下式得到一个权重值

$$e_{i,j} = a(s_{i-1}, h_j) \quad (13)$$

上式中 a 可以是任何定义域是 R 的函数, 计算会得到一系列值 $e_{i,1}, \dots, e_{i,n}$ 。使用 **softmax** 归一化

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^n \exp(e_{i,k})} \quad (14)$$

最后对输入隐式序列加权后生成最终的上下文状态向量。

$$c_i = \sum_{j=1}^n \alpha_{i,j} h_j \quad (15)$$

直观的来讲, 这个向量能够在第 i 步捕获原文的上下文信息。

3.7 Dynamic Memory Network

Kumar et al.[6] Dynamic Memory Network 由四个模块组成, 分别是: 输入模块、片段记忆模块、问题模块、回答模块。整体的架构如图表所示。

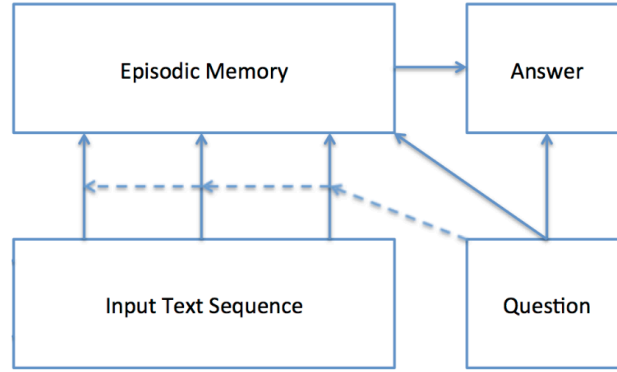


图 1 Dynamic Memory Network 模型架构

3.7.1 输入模块

输入模块接受一段文字, 首先将文字中的每个单词转换为对应的词向量表示, 然后将每个单词按顺序依次送入 GRU。在每句话的结尾处, 输出最终的隐式状态。片段记忆模块将接收这些隐式状态, 并进行总结推理。更加形式化的表示为, 对于一个单词序列 $T_I w_1, \dots, w_{T_I}$, 我们根据下式来更新状态。

$$h_t = GRU(L[w_t], h_{t-1}) \quad (16)$$

然后对于由 T_I 作为子序列组成的序列 s_1, \dots, s_{T_I} , 在每个子序列结束的时候, 我们将最终的隐式状态 $h_{s_1}, \dots, h_{s_{T_I}}$ 输出

3.7.2 问题模块

问题模块与输入模块相似, 接受问题作为输入, 将问题序列中的每个单词转换为对应的词向量的表示, 然后将每个词向量送入 GRU, 当所有问题单词都送入后, 输出 GRU 的最终表示。所以对于问题, 形式化的表示为, 对于一个包含单词 w_1, \dots, w_{T_Q}

的问题 T_Q , 我们使用下式更新隐式状态

$$h_t = GRU(L[w_t], h_{t-1}) \quad (17)$$

最后的输出为 h_{T_Q}

3.7.3 片段记忆模块

片段记忆模块对于输入模块在每句话结束时输出的隐式状态和问题模块输出的隐式模块进行总结推理, 产生一个最终记忆状态输出给回答模块, 用于产生回答。

片段记忆模块主要由嵌套的两层 GRU 组成, 内层 GRU 负责产生片段序列, 外层 GRU 使用问题向量初始化后, 根据片段序列产生最终记忆模块作为输入。

内层 GRU 每次通过遍历输入模块的输入序列产生一个片段, 在每个片段结束后, 内层 GRU 会把这个片段产生的最终状态送入外层 GRU, 下面的公式给出了内层 GRU 更新状态的方法: [14]

$$z_t^i = [t_t, m, q, c_t \odot q, c_t \odot m, |c_t - q|, |c_t - m|] \quad (18)$$

$$Z_t^i = W^{(2)} \tanh(W^{(1)} z_t^i + b^{(1)}) + b^{(2)} \quad (19)$$

$$g_t^i = \frac{\exp(Z_t^i)}{\sum_{k=1}^{M_i} \exp(Z_k^i)} \quad (20)$$

$$h_t^i = g_t^i GRU(c_t, h_{t-1}^i + (1 - g_t^i) h_{t-1}^i) \quad (21)$$

上面的式子中结合当前状态 c_t , 当前记忆状态 m 和记忆状态 q 来决定当前句子是否值得编码进入回答 g_t^i 中, 如果 $g_t^i \approx 0$, 当前句子将被忽略, 现在的状态即为之前的状态。这个片段的最终状态是当内层 GRU 遍历完所有句子之后产生的状态 $e^i = h_{T_i}$ 外层 GRU 将根据之前的记忆状态和本次片段状态来更新片段状态。

$$m^t = GRU(e^t, m^{t-1}) \quad (22)$$

外层 GRU 的最终记忆状态将被送入回答模块。

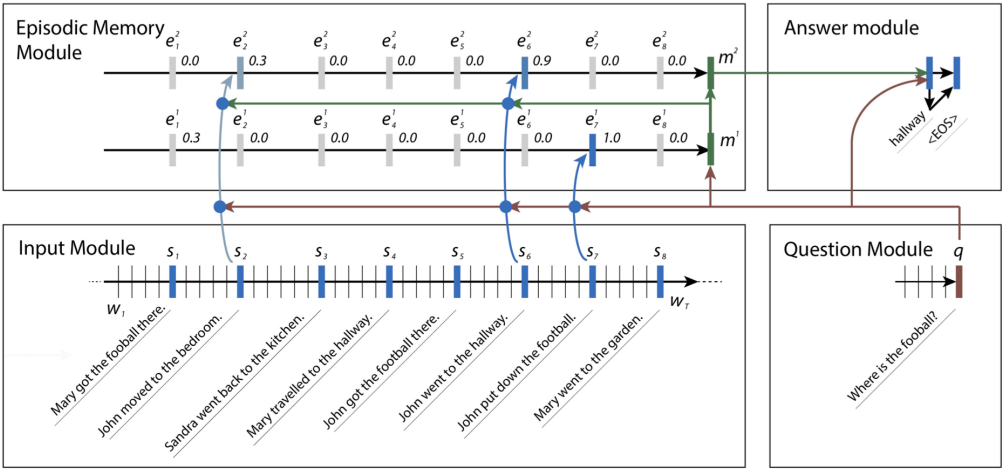


图 2 片段记忆模块架构

3.7.4 回答模块

回答模块通过 softmax 产生对于回答标签的概率分布,然后产生单个单词回答,也可以将输入状态向量 RNN 来产生多个单词的回答。

参考文献

- [1] Alan Akbik and Jürgen Broß. Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *WWW Workshop*, 2009.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [3] Junyoung Chung, Caglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Gated feed-back recurrent neural networks. In *ICML*, pages 2067–2075, 2015.
- [4] Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224. ACM, 1961.
- [5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [6] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, abs/1506.07285, 2015.
- [7] Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523. Association for Computational Linguistics, 2011.
- [8] Cody Kwok, Oren Etzioni, and Daniel S Weld. Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)*, 19(3):242–262, 2001.
- [9] Ni Lao, Tom Mitchell, and William W Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics, 2011.

- [10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [11] Dan I Moldovan and Vasile Rus. Logic form transformation of wordnet and its applicability to question answering. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 402–409. Association for Computational Linguistics, 2001.
- [12] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.
- [13] Martin M Soubotin and Sergei M Soubotin. Patterns of potential answer expressions as clues to the right answers. In *TREC*, 2001.
- [14] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. *arXiv*, 1603, 2016.
- [15] Hui Yang and Tat-Seng Chua. The integration of lexical knowledge and external resources for question answering. In *TREC*, 2002.
- [16] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 25–26. Association for Computational Linguistics, 2007.
- [17] Luke S Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*, 2012.