浙江大學

本科生毕业论文
开题报告

题　　目：基于 Dynamic Memory Network 的问答系统的优化与应用

姓　　名：杨凯

学　　号：3130000495

指导教师：蔡亮

专　　业：2013 级　软件工程

学　　院：计算机科学与技术

一、题目：基于 Dynamic Memory Network 的问答系统的优化与应用

二、指导教师对开题报告、外文翻译和文献综述的具体要求：

- 要求查阅相关的文献 10 篇以上（外文不少于 5 篇）。
- 翻译的外文文献必须是研究性的论文，并且与论文主题直接相关.
- 文献综述应包括国内外现状、研究方向、存在问题、参考依据等方面情况。
- 译文和文献综述字数要求各 3000 字以上，开题报告字数为 3500 字以上。
- 开题报告的内容应包括：

    1. 主要研究内容、目的和意义;
    2. 有关的国内外研究状况；
    3. 课题难点和拟解决的关键问题；
    4. 拟取的研究方法及其可行性、预期达到的目标;

指导教师（签名）：

年　　月　　日

毕业论文开题报告、外文翻译和文献综述考核

导师对开题报告、外文翻译和文献综述评语及成绩评定：

成绩比例	开题报告占 (20%)	中期报告占 (10%)	外文翻译占 (10%)
分值			

导师签字 _____

年　月　日

答辩小组对开题报告、外文翻译和文献综述评语及成绩评定：

成绩比例	开题报告占 (20%)	文献综述占 (10%)	外文翻译占 (10%)
分值			

答辩小组负责人（签名）_____

年　月　日

目　录

1　课题背景 1
2　目标和内容 2
3　可行性分析 2
4　研究方案和关键技术考虑 3
5　预期研究结果 8
6　进度计划 8
参考文献 10

本科毕业论文开题报告

1 课题背景

人类语言具有特殊的结构，能够方便的传达意义，自然语言处理 (NLP) 就是研究人类语言的一个学科。而问答系统 (Question Answering) 的 NLP 问题之一。NLP 中的很多问题都可以转换为 QA 问题 (例如：机器翻译、情感分析、实体标记等)。随着智能语音助手（例如：苹果的 Siri、谷歌的 Google Assistant 和微软的 Cortana）的流行，QA 问题更是被广泛的予以关注。而 QA 问题本身是一项很复杂的 NLP 任务。想要对问题做出回答，需要具有分析一段文字，进行归纳总结的能力。

传统的解决 QA 的方法通常依赖于固定结构的知识数据库，而这些数据库通常是由特定领域的专家人工搭建的。当进行问题回答时，首先解析问题，然后将问题转换成特定结构的查询语句，在数据库中搜寻，在将查询结果组合成回答呈现给用户。

随着深度学习领域的迅速发展，以神经网络为基础的方法已经在文字和图像分类识别领域取得了重大的进展。最近，开始逐渐有人将神经网络应用于比较复杂的任务，例如逻辑推理上。这些通过神经网络产生的隐式的语言表达不需要依赖于特定的语言结构，也不需要对语言做特定对解析等预处理行为。这些模型在需要很少的预处理过程的前提下，达到或者超过了传统的模型，取得了很好的结果。这些模型依赖于记忆 (Memory) 模块和注意力 (Attention) 模块，起到了能够从上下文总结信息、进行推理等作用。

Dynamic Memory Network 就是这样一个使用了记忆和注意力模块的模型。这个模型在能够取得当前领先的结果。模型由四个模块组成：输入模块、片段记忆模块、问题模块、回答模块。

本毕业论文将给予 Dynamic Memory Network, 首先实现这一模型，然后再对这一模型进行进一步扩展优化，最后将我们优化后的模型应用于技术文档数据集中。

2 目标和内容

本次毕业论文主要任务是基于 Dynamic Memory Network 模型的优化及其应用，具体来讲，分为以下几个子目标：

- 实现基准模型，作为实验对照
- 实现基础 Dynamic Memory Network
- 对 Dynamic Memory Networt 进行进一步优化（输入表达可以进一步尝试使用 GloVe 和基于字的词表达，记忆模块尝试使用 GRU、LSTM 和 bidirectional LSTM 优化)
- 在单词回答的技术上支持多词回答
- 将优化后的模型应用于技术文档数据集

3 可行性分析

从下面几个角度分析，本次毕业论文是可行的：

- 实验模型：Kumar et al.[3] 对这一基础模型的架构已经有了比较清晰的描述。使用主流的机器学习框架 Keras 和 TensorFlow 能够相应的减少一些所需要的基础工作量。
- 数据集：使用 Facebook bAbI 数据集。数据集是公开可用的，已经被问答系统模型广泛使用和验证，有较为全面的基准可以用于实验的对比。
- 实验机器：
  - Macbook Pro 2016，16GB 内存，2.60Hz Intel Core i7：代码编写，初步程序验证。
  - Precision Tower 7810 Workstation,32GB 内存,Intel Xeon i7, Nvidia GeForce GTX 970 显卡：程序验证，高阶参数调整。
  - Google Cloud Platform：32GB 内存，Nvidia Tesla K80 显卡：训练数据。

4 研究方案和关键技术考虑

4.1 研究方案

本次实验采用对照实验，一个基准组和两个对照组。基准组是将词向量直接作为输入进入最简单的神经网络架构中，来实现的。一个实验组是实现 Dynmaic Memory Network 的基础模型的实验组，另外一个实验组是对基础 Dynamic Memory Network 优化后的实验组。本次实验使用 Facebook bABi 数据集。bABi 数据集分成 20 个子集，每个子集具有不同的结构，图 1.1 为数据集中的一些例子 [7]。每个子集有 1000 个训练样本和 1000 个测试样本。我们使用训练样本对我们的三个模型进行训练，使用测试样本进行测试。一个模型只有能够正确回答 95% 的测试集问题，则称之为能够通过这个子任务集。

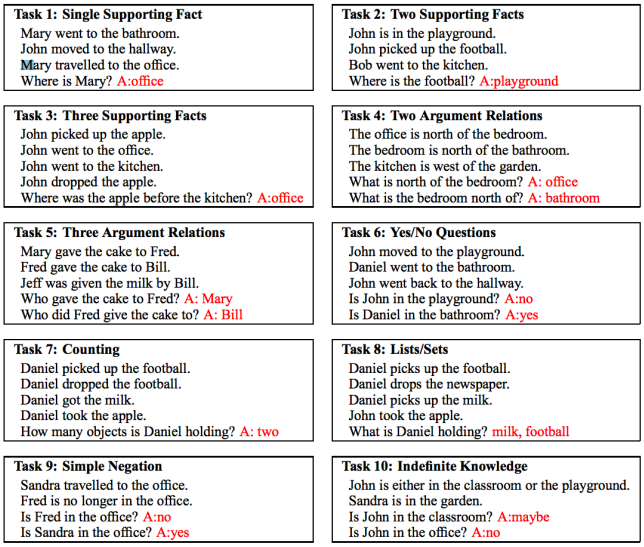

图 1 bABi 数据样例

4.2 Word2Vec

Word2Vec(Mikolov et al., 2013[4]) 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。

4.3 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.[6] 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。

4.4 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 [5]:

$$h_t = f(Wx_t + Uht - 1 + b) \tag{1}$$

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。

4.5 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.[1]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：

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

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

$$\widetilde{h}_t = tanh(r_t \odot Uh_{t-1} + Wx_t) \tag{4}$$

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

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 $\widetilde{h}_t$。式 (4) 表明新的记忆 $\widetilde{h}_t$ 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 $\widetilde{h}_t$ 新的记忆组成的。

4.6 LSTM

Long Short Term Memories(LSTM)[2] 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:

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

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

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

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

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t \tag{10}$$

$$h_t = o_t \odot tanh(c_t) \tag{11}$$

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。

4.7 Dynamic Memory Network

Kumar et al.[3] 提出的 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。

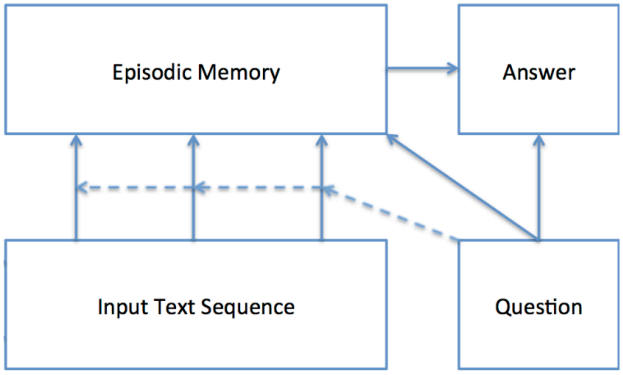

图 2 Dynamic Memory Network 模型架构

4.7.1 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 $w_1, ...w_{T_I}$，我们根据下式来更新状态。

$$h_t = GRU(L[w_t], h_{t-1}) \tag{12}$$

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

4.7.2 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 $w_1, ..., w_{T_Q}$

的问题 T_Q，我们使用下式更新隐式状态

$$h_t = GRU(L[w_t], h_{t-1}) \tag{13}$$

最后的输出为 h_{T_Q}

4.7.3 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：[8]

$$z_t^i = [t_t, m, q, c_t \odot q, c_t \odot m, |c_t - q|, |c_t - m|] \tag{14}$$

$$Z_t^i = W^{(2)} tanh(W^{(1)} z_t^i + b^{(1)}) + b^{(2)} \tag{15}$$

$$g_t^i = \frac{exp(Z_t^i)}{\sum_{k=1}^{M_i} exp(Z_k^i)} \tag{16}$$

$$h_t^i = g_t^i GRU(c_t, h_{t-1}^i + (1 - g_t^i)) h_{t-1}^i \tag{17}$$

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 遍历晚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。

$$m^t = GRU(e^t, m^{t-1}) \tag{18}$$

外层 GRU 的最终记忆状态将被送入回答模块。

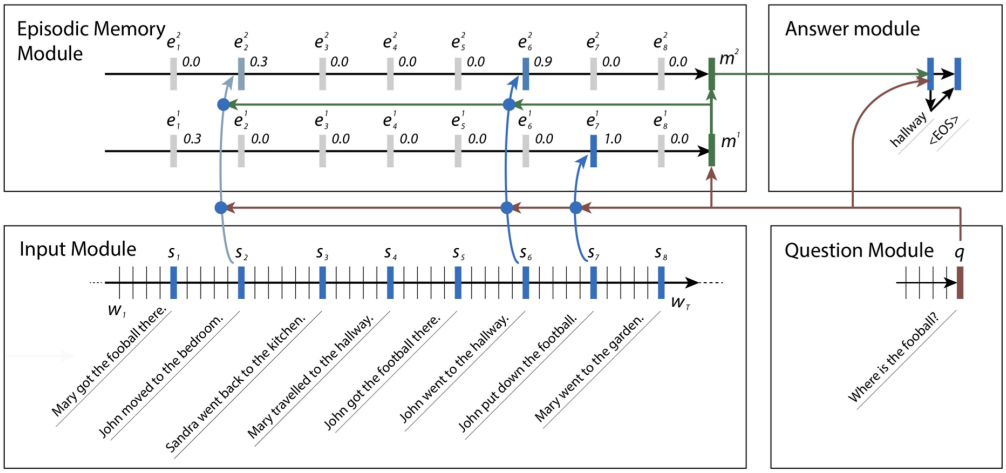

图 3　片段记忆模块架构

4.7.4　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。

5　预期研究结果

本次毕设预期结果是实现基础 Dynamic Memory Network，对起进行优化，提高回答准确率，在单词回答的技术上支持多词回答，并且最终应用到技术文档中。

6　进度计划

- 3 月 31 日 ~4 月 10 日: 编写基准模型和基础 Dynamic Memory Network 模型
- 4 月 11 日 ~4 月 17 日: 训练、验证和测试基准模型和基础 Dynamic Memory Network 模型
- 4 月 18 日 ~4 月 25 日: 编写代码，尝试对 Dynamic Memory Network 进行优化
- 4 月 26 日 ~4 月 29 日: 编写中期报告
- 4 月 30 日 ~5 月 16 日: 尝试拓展 Dynamic Memory Network，使模型具备多单词回答

- 5 月 17 日 ∼5 月 29 日: 编写论文
- 5 月 30 日 ∼ 答辩前: 准备答辩

参 考 文 献

[1] Junyoung Chung, Caglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *ICML*, pages 2067–2075, 2015.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[3] 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.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[5] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[6] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[7] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698, 2015.

[8] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. *arXiv*, 1603, 2016.