《人工神经网络》大作业最终报告

周正平
计算机科学与技术系
清华大学
zhouzp15@mails.tsinghua.edu.cn

张钰晖
计算机科学与技术系
清华大学
yuhui-zh15@mails.tsinghua.edu.cn

Abstract

本次实验致力于提高KBQA（Knowledge Based Question Answering）系统中的问句关系抽取准确率。我们首先基于seq2seq模型计算问句与关系之间的相似度，其后使用记忆网络架构引入自然语言文本作为外部记忆模块，以搭建**自然语言问句**与**结构化知识库**之间的桥梁。此外，我们使用句子级别的注意力机制对多个实例进行加权，以降低远程监督引入的噪声。

模型最终在WEBQUESTIONS数据集上取得了45%的F1值，在此前浩繁的研究成果中居于前列，但仍有值得改进完善的地方。

1 引言

基于知识库的问答系统（Knowledge Based Question Answering，KBQA）是NLP及IR领域研究的热点话题，其重要的关键技术是实体链指（Entity Linking）与关系抽取（Relation Extraction）。

目前state-of-art的KBQA系统Aqqu和Stagg在标准数据集WebQuestions上统计发现，在回答错误的问题中，有将近半数的错误来自于关系抽取错误，因此如果能很好的解决关系抽取问题，KBQA系统的性能将取得明显提高。

关系抽取可以简单定义为从一句话里抽取三元组（s，r，o），其中s和o是实体（Entity），r是s和o之间的关系（Relation）。例如从问题"Who is the ceo of Apple"，正确抽取的三元组是$(Apple\ Inc., directed_by, Steve\ Jobs)$。这不仅需要从问句里识别出Apple Inc.实体，更需要正确匹配关系$directed_by$。

Aqqu和Stagg在关系抽取步骤中主要采用了统计机器学习的方法，我们将在此基础上加以改进，采用前沿NLP领域的技术，如记忆网络（Memory Network）、注意力机制（Attention）、支持语句（Support Sentences），来尝试提高关系抽取的准确率，从而提高KBQA系统的准确率。

模型输入为问题（Query）q和三元组（Triple）(s, r, o)，输出为一个实数，代表q和$(s$fi$r$fi$o)$的匹配程度。

我们使用一个实例对系统的设计思路进行阐明：

给定一个问句q，我们的系统试图查询知识图谱$\mathcal{K}$，以得到答案A：

$$q = \text{"Who is the ceo of apple?"}, A = \text{"Steve Jobs"}$$

实体链指 实体链指是关系抽取的前置环节。在这一部分，我们从问句中识别出所有可能的存在于知识图谱中的**实体**。在此例中，读者容易看出“apple”一词指代苹果公司（Apple Inc.）。然而，这个映射关系并非显而易见。“apple”一词还有可能对应于植物名、歌曲名等等。因而在这一步中，我们会得到一个包含所有可能实体的集合，并得出每个候选实体的置信概率。

关系匹配 关系匹配是本次实验的核心环节。这一部分衡量q与知识图谱中定义的**关系**的匹配程度。共分为如下2个主要模块：

1. **字面匹配**：这一部分的主要任务是衡量问句与知识库中的关系在字面级别的相似程度。我们考虑2种可能的策略，并以关系$r = director_of$为例进行说明：
   (a) **整体匹配**：这种策略将s与r作为一个整体与q进行匹配。在上例中，有
   $$Similarity(q, r) = Similarity(\text{“}Who\, is\, the\, ceo\, of\, \mathbf{apple}\text{?”}, \text{“}\mathbf{Apple\ Inc.}\, directed\, by\text{”})$$
   (b) **解耦匹配**：这种策略单独将r与q进行匹配。其中，q中涉及的实体名称使用特殊符号<e>进行替换。在上例中，有
   $$Similarity(q, r) = Similarity(\text{“}Who\, is\, the\, ceo\, of\, <\mathbf{e}>\text{?”}, \text{“}directed\, by\text{”})$$
2. **语义匹配**：这一部分以记忆网络的架构引入自然语言文本增强r中蕴含的语义信息。
   由于q使用自然语言表述，而r为结构化知识库中预定义的关系，二者间往往难以建立良好的字面匹配。例如，在上例中，q中的短语“ceo of”与$r = directed_by$在字面级别存在一定的差异。然而，如果我们在维基百科上看到如下语句：
   Jobs is the ceo of Apple Inc.
   我们便有理由认为r与q之间存在某种语义的关联。
   直观地，如果一个自然语句s中同时出现了q中的主语（$apple$）与谓词（$ceo\, of$），又同时出现了r连接的2个实体（$s = Apple\ Inc.$与$o = Steve\ Jobs$），则这句话便对q与r之间的相关性提供了证据。我们称之为支持语句（Support Sentence）。

训练排序 训练采用Learning to Rank的思想，最终在上述2步中得分最高的候选结果作为答案输出。

2 相关工作

近年来，KBQA问题引起了信息检索领域（IR）及自然语言处理领域（NLP）的广泛关注。许多工作致力于对问句进行语义解析，其中关系抽取更是尤为关键。例如，（Berant et al., 2013）使用词语表对问句中的实体进行匹配，进而自底向上枚举所有可能的逻辑分析树，进而通过统计模型选出概率最大的逻辑解析；（Yih et al., 2015）对每个问句分阶段构造其逻辑子图，并在构造的过程中计算子图的概率；（Bast et al., 2015）则将问句分为3种可能的逻辑结构，并据此构造候选集合，使用tf-idf等传统NLP方法计算候选事实与问句之间的匹配程度。此外，还有部分工作着眼于手工对问句构造特征向量。（Yao et al., 2014）首先构造问句的语法分析树，而后对各个语法分析树的特征进行手工构造，并据此进行筛选；（Bast et al., 2015）则通过对词汇频次、关系数量等统计特征进行综合，对所得的特征向量使用SVM进行排序。无论是语义解析还是特征工程，上述方法均或多或少依赖于人工特征工程。基于深度学习的KBQA工作也因此大量涌现：（Bordes et al., 2014）使用词袋模型将问题与候选实体嵌入到低维空间，采用向量建模的方法计算其匹配程度；（Yih et al., 2015）使用CNN模型对问句中的实体与关系进行匹配。

另一些学者在KBQA的问答数据集之外，引入自然语言文本作为辅助语义信息。（Chen et al., 2017）创新性地仅使用维基百科中的自然文本，基于阅读理解方式构建问答系统；（Xu et al., 2016）则首先使用神经网络对问答数据集进行学习，而后引入维基百科中的文本对上一步的结果进行补充完善；（Berant et al., 2014）则基于统计模型，在语料库上学习得到一个问句生成模型。通过比较该模型根据候选事实生成的问句与原问句，可以确定候选事实的匹配程度。

尽管上述工作已经在KBQA领域取得了巨大的成功，我们仍在其中发现2点可改进之处：

1. **关系抽取**：参考此前部分state-of-the-art的KBQA系统的错误分析，可以看出，关系抽取乃是许多KBQA系统亟待解决的瓶颈问题：

KBQA系统	错误分析
AQQU（Bast et al., 2015）	Entity Linking：9.82% **Relation Extraction：41.96%** 其他（包括数据噪声等）：48.22%
STAGG（Yih et al., 2015）	Entity Linking：8% **Relation Extraction：35%** 其他（包括数据噪声等）：57%
QUESTIONANSWERINGOVERFB（Xu et al., 2016）	Entity Linking：15% **Relation Extraction：50%** 其他（包括数据噪声等）：35%

2. **自然文本**：通过文献整理，我们发现，引入了自然文本作为辅助训练数据的KBQA系统相对较少。（Li et al., 2017）首次引入支持语句的概念，然而在这份工作中，自然语言仅作为模型的第二阶段进行补充训练，而未进行端到端的集成。

记忆网络（Memory Network）近年来在自然语言处理领域得到了广泛的应用。（Sukhbaatar et al., 2015）首次引入端到端的记忆网络，文中指出，记忆网络可以作为外部记忆模块，以端到端的形式集成至系统中，相比传统的LSTM结构，可以存储更大的数据量。（Kumar et al., 2016）则提出动态记忆网络，在文中讨论了使用记忆网络将句子集合编码的多种方法；（Bordes et al., 2015）则将记忆网络架构应用于KBQA领域，通过在外部记忆模块中对FREEBASE进行存储，来实现对模型的远程监督。

本次实验综合考虑以上文献的几种方法，提出如下几个改进之处：

1. **关系抽取**：显示地将关系抽取与KBQA的其他环节进行解耦，并通过实验对比论证其合理性。
2. **自然文本**：在记忆网络的架构之下引入自然语言文本作为外部记忆模块，并使用注意力机制对多个语句进行加权降噪。
3. **远程监督**：上述自然文本，基于远程监督的思想（Mintz et al., 2009）进行选取。

3 方法

给定一个问句q，我们设计的问答系统试图查询知识图谱$\mathcal{K}$，以得到答案A。其实现基于记忆网络（Memory Network）架构，主要包括如下几个模块：

1. **候选集合**C：通过实体链指与关系枚举，将问题的搜索空间压缩至候选事实集合。
2. **输入模块**I：将输入的问句、候选的事实、自然文本语句等转化为向量表示。
3. **泛化模块**G：根据当前输入，更新外部记忆中的内容。
4. **输出模块**O：对记忆内容基于注意力机制进行加权，得到输出特征向量。
5. **响应模块**R：根据输出的特征向量，计算候选事实对问题的匹配程度。

图1展示了模型的示意图：

3.1 候选集合C

在这一步骤中，我们首先在知识图谱中确定**候选事实**集合

$$C(q) = \{c|c = (s, r, o), s \in E(q) \wedge r \in out(s)\}$$

其中：

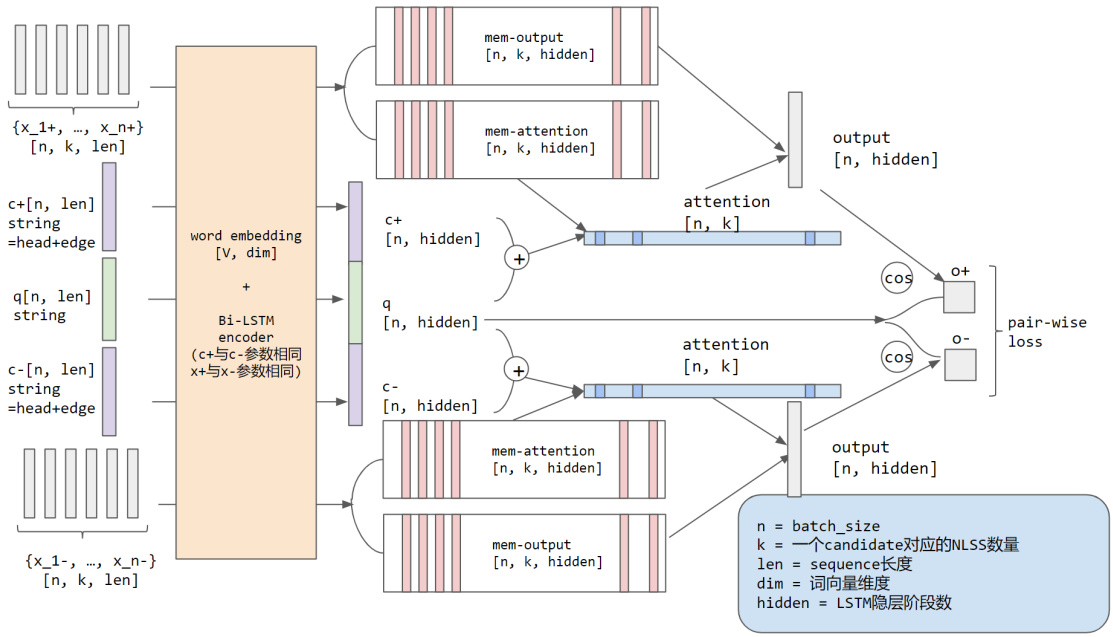

Figure 1: 关系匹配中使用的记忆网络与注意力模型

1. $s \in E(q)$表示q中可能出现的某一实体。在上例中，有$E(q) = \{Apple\ Inc., Apple\ Tree, ...\}$。
2. $r \in out(s)$表示实体s在知识图谱中的可能出边。例如对$s = Apple\ Inc.$，有$out(s) = \{directed_by, located_in, ...\}$。
3. o表示q的候选答案之一，由s与r在知识图谱$\mathcal{K}$中唯一确定。例如对$s = AppleInc., r = directed_by$，有$o = Steve\ Jobs$。

在实验使用的知识图谱FREEBASE中，存储的事实总数高达29亿之多，因此使用候选集合可以对搜索空间进行压缩。此外，通过预筛选，可以排除大量的无关事实，从而起到降噪效果。

这一部分可分为如下几个环节：

实体链指 我们首先使用实体链指系统S-MART（Yang and Chang, 2015），得到候选实体集合$E(q)$。S-MART系统适用于较短的噪声文本，因而与数据集WEBQUESTIONS的实际情况相契合。

S-MART系统的实现策略简述如下：首先，对知识图谱中的每一实体e，该系统通过对多种数据来源进行统计，确定其在文本中所有可能的出现形式，得到一个列表；其次，对于输入问句q，它考虑所有出现于上述列表中的子串，并将其链接至相应实体e；最后，每个实体基于其出现频次，依统计模型得到其置信概率。

实验中，我们对每个问句q，保留其置信概率最高的10个实体作为$E(q)$。

关系枚举 对每一实体$s \in E(q)$，我们使用SPARQL（SPARQL Protocol and RDF Query Language）语言查询FREEBASE，得到其所有出边集合$out(s)$，以及其中每一出边r的终点o。

WEBQUESTIONS数据集中的大部分问题（约占66%）可在FREEBASE仅查询1条边就得到答案。另有约31%的问题需查询相邻的2条边。实际上，FREEBASE已经对部分相邻边进行了合并，可以使用SPARQL一步查询得到。因而对后一种情形，参考（Bordes et al., 2015）的处理方法，我们保留第2条边计入$out(s)$中。

实验中，考虑到许多问题的答案通常不唯一，我们对所有始点相同的边合并其终点。因此，每一候选事实c对应的结果o实际为一个实体集合。

F1值计算 对训练集中的每一候选事实c，计算其结果o与数据集提供的真实答案A_{gold}之间的F1值$F1 = 2 \cdot \frac{Precision(o,A_{gold}) \times Recall(o,A_{gold})}{Precision(o,A_{gold}) + Recall(o,A_{gold})}$作为其优度指标，用于在训练时计算Pairwise Loss。

对测试集，该指标仅用于调试与错误分析，在推断时不会向模型提供。

支持语句选取 对每一事实$c = (s, r, o)$，我们基于远程监督思想（Distant Supervision），从维基百科数据集中选取其对应的支持语句：具体地，考虑语句s，若s中同时出现了s与o的别名形式（通过FREEBASE提供的实体名称映射表进行软匹配），则将s加入事实c的支持语句集合$X(c)$。

3.2 输入模块I

对输入问句q、候选事实$c = (s, r, o)$、支持语句集合$X(c) = \{x_1, x_2, ..., x_n\}$，我们在这一步中采用Google Word2Vec预训练词向量对其中的单词进行向量化。

考虑q、c或x_i 中的单词序列$W = \{w_1, w_2, ..., w_m\}$。我们将其对应的词向量序列$\mathbf{W} = \{\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_m}\}$通过双向LSTM网络的最后一个隐状态，编码为单个实值向量：

$$Encoder(W) = BiLSTM([\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_m}])$$

从而，有

$$\mathbf{q} = Encoder(W_q) \tag{1}$$
$$\mathbf{c} = Encoder(W_c) \tag{2}$$
$$\mathbf{x_i} = Encoder(W_{x_i}) \tag{3}$$

其中，$\mathbf{q}, \mathbf{c}, \mathbf{x_i}$分别为$q, c, x_i$的向量化表示；$W_q, W_c, W_{x_i}$分别为$q, c, x_i$中的单词序列。

可以看出，输入的向量编码取决于其单词序列W。我们在本次实验中共考虑W的2种可能选取策略：

整体表示 这种策略的基本思想是将q中的主语映射到c中的头实体s，q的其余部分映射到c中的关系r：

1. W_q为q中所有单词构成的集合。
2. W_c由s与r的所有单词序列拼接得到。
3. W_{x_i}为x_i中所有单词构成的集合。

解耦表示 这种策略的基本思想是将q中的实体与关系对匹配程度的影响进行解耦：

1. 在q 的单词集合中，把c中头实体s对应的连续单词序列替换为特殊符号$< e >$，得到W_q。
2. W_c为r的单词序列构成的集合。
3. 在x_i的单词集合中，将c中头实体s、尾实体o对应的连续单词序列各自替换为特殊符号$< e >$，得到W_{x_i}。

3.3 泛化模块G

在记忆网络中，我们使用外部记忆模块（External Memory）M对支持语句集合$X = \{x_1, x_2, ..., x_n\}$进行存储。泛化模块$G$负责根据当前输入，更新记忆模块$M$ 中存储的记忆。

M中共包括2个外部记忆模块：

输出记忆（Output Memory，OM） 其中存储的记忆$\{m_{o1}, m_{o2}, ..., m_{on}\}$用于计算模型的输出。

注意力记忆（Attention Memory，AM） 其中存储的记忆$\{m_{a1}, m_{a2}, ..., m_{an}\}$用于计算支持语句集合上的选择性注意力（Selective Attention）。

G通过如下方式对M进行改写：

OM 对输出记忆，有

$$m_{oi} = \mathbf{x_i}_o$$

其中，$\mathbf{x_i}_o$为使用输出双向LSTM对句子x_i编码得到的句向量。

AM 对注意力记忆，有

$$m_{ai} = \mathbf{x_i}_a$$

其中，$\mathbf{x_i}_a$为使用注意力双向LSTM对句子x_i编码得到的句向量。

3.4 输出模块O

输出模块根据输入的向量，对记忆M中存储的支持语句进行加权处理，从而将记忆按照与输入问句q及候选事实c的相关程度进行组合，得到输出向量o。

输出向量o可以看做输出记忆OM中各个句向量的加权均值：

$$o = \sum_i a_i \mathbf{m_{oi}}$$

其中，a_i表示第i个输出记忆$\mathbf{m_{oi}}$的权值。

在本次实验中，我们采用3种方式定义a_i：

最大值（Maximum） 这种方式仅考虑相关程度最高的支持语句的贡献。此时输出向量等于输出记忆中与$\mathbf{q}+\mathbf{c}$内积最大的句向量：

$$o = \mathbf{argmax}_{\mathbf{m_{oi}}} \mathbf{m_{ai}} \cdot (\mathbf{q}+\mathbf{c}), \mathbf{m_{oi}} \in OM$$

其中$\mathbf{m_{ai}}$为注意力记忆AM模块的第i个句向量。这种方式丢失了支持语句中的大部分信息，主要用于后续对比。

平均值（Average） 这种方式假设各个支持语句对问句q与事实c有着相等的相关程度。此时输出向量等于输出记忆中各个句向量的算术平均值：

$$o = \sum_i \frac{1}{n} \mathbf{m_{oi}}$$

这种方式不考虑对噪声语句的筛选，同样用作后续对比。

注意力（Attention） 这种方式依据各个支持语句与问题q及候选事实c的相关程度，对输出记忆中的各个句向量进行加权求和：

$$a_i = \frac{\exp(\mathbf{m_{ai}} \cdot (\mathbf{q}+\mathbf{c}))}{\sum_k \exp(\mathbf{m_{ak}} \cdot (\mathbf{q}+\mathbf{c}))}$$

其中，$e_i = \mathbf{m_{ai}} \cdot (\mathbf{q}+\mathbf{c}))$采用内积形式衡量第$i$个支持语句与问题及候选事实的匹配程度。在此基础上使用Softmax对权值进行归一化，便得到了注意力a_i。

3.5 响应模块R

响应模块接受输出模块的输出，计算问题q与事实$c=(s,r,o)$的相似程度$f(q,c)$。考虑在输入模块I处使用的2种输入编码方式，我们在响应模块R中采用不同的方式计算相似度:

$$f_{RDF}(q,c)=\cos(\mathbf{q},(\mathbf{c}+\mathbf{o_c})) \tag{4}$$

$$f_{Relation}(q,c)=\log(score_s)\cdot\cos(\mathbf{q},(\mathbf{c}+\mathbf{o_c})) \tag{5}$$

其中，f_{RDF}为整体表示策略对应的相似度函数，$f_{Relation}$为使用解耦表示策略对应的相似度函数。

$\mathbf{q},\mathbf{c}$分别为问题q、事实c的向量表示；$score_s$为实体s对应的置信概率；$\mathbf{o_c}$为c的支持语句集合$X(c)=\{x_1,x_2,...,x_n\}$经输出模块O编码后得到的语义向量。

训练 训练采用Learning to Rank思想，利用正负例采样，训练模型对相关度较高的问题-事实对做出较大的响应。其损失函数定义如下:

$$loss=\max(0,1-f(q,c^+)+f(q,c^-))$$

其中，$(c^+,c^-)\in P$，集合P根据候选事实的F1值定义如下:

$$P=\{(c^+,c^-)|F1(c^+,q)>F1(c^+,q)\}$$

其中，候选事实$c=(s,r,o)$ 在问题q 上的F1值以如下方式计算:

$$F1(c,q)=2\cdot\frac{Precision(o,A)\times Recall(o,A)}{Precision(o,A)+Recall(o,A)}$$

其中，A为WEBQUESTIONS数据集为q提供的标准答案集合。

排序 将模型的输出进行排序，选取其中的最大值，得到最佳匹配事实

$$c_{best}=\mathbf{argmax}_c f(q,c)$$

其中，$c_{best}=(s_{best},r_{best},o_{best})$，则答案集合$o_{best}$即为问题$q$的最终回答。

4 实验

4.1 数据集

问答数据集 问答数据集采用开放QA领域主流数据集WEBQUESTIONS，其中包含5810个问题和来自Freebase数据库的答案。官方将其划分为3776（70%）个问题的训练集和2032（30%）个问题的测试集。该数据集由（Berant et al., 2013）首次引入，其收集方法为：首先在Google Suggest API上爬取问题数据，而后在Amazon Mechanical Turk平台采用Crowd Source方式获取人工标注的答案（通常为FREEBASE中的实体名称）。

知识图谱 实验采用FREEBASE作为知识知识图谱。FREEBASE中包含了约4400万实体，以及29亿与它们相关的事实。我们在实验中使用其中的所有三元组作为候选集合的来源。

自然语言文本 我们使用WIKIPEDIA中的英文数据（enwiki）提供高质量的自然语言文本。我们采用其最近的一次data dump[1]，并使用开源工具WikiExtractor[2]进行数据处理（包括去除HTML标签、超链接提取、分句等）

[1]https://dumps.wikimedia.org/enwiki/latest/

[2]https://github.com/attardi/wikiextractor

4.2 评价方案

考虑到WEBQUESTIONS数据集中的问题多以一个集合作为答案，本次实验参考（Bast et al., 2015）、（Yih et al., 2015）等工作，使用模型在测试集上的平均F1作为评价指标：

$$average\ F1 = \frac{1}{n}\sum_{i=1}^{n} F1(c_i, A_i)$$

其中，F1函数的含义已经在模型部分进行了充分的说明。c_i, A_i则分别对应于模型选取的候选事实，以及数据集给出的真实答案。

4.3 实验方法

参数设定 实验中，我们选用维度为300的Google Word2Vec预训练词向量对词嵌入进行初始化；模型中所有特征向量维度取作100，也即为双向LSTM编码器的隐层节点数；最后，我们采用Adam优化器进行正负例采样训练，初始学习率为0.001，设置批大小为16。为避免过拟合，我们采用early stop技术，对所有模型统一训练到第5个epoch。

模型实现 在本次实验中，我们基于Python Tensorflow，实现了如下4种模型：

1. Baseline-RDF：使用上述的整体匹配方式，不使用支持语句。
2. Baseline-Relation：使用上述的解耦匹配方式，不使用支持语句。
3. NLSS-RDF：使用上述的整体匹配方式，使用支持语句。
4. NLSS-Relation：使用上述的解耦匹配方式，使用支持语句。

基线 由于时间有限，我们将对KBQA系统的完整移植留作此后的工作。因此除上述4种模型之外，我们考虑关系抽取领域的基线CDSSM模型，并对其在WEBQUESTIONS上的性能与模型进行对比。

4.4 实验结果

平均F1值 在WEBQUESTIONS数据集上，上述4种模型与基线CDSSM取得的平均F1值如下：

方法	F1（%）
CDSSM	31.3
Baseline-RDF	38.1
Baseline-Relation	45.2
NLSS-RDF	37.0
NLSS-Relation	45.0

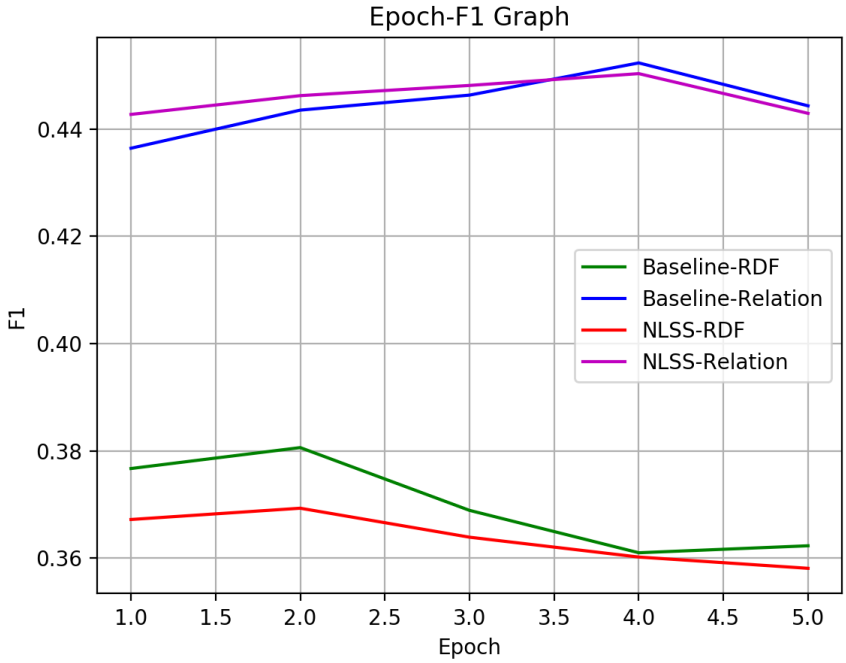

Figure 2: F1-epoch曲线图

F1-epoch曲线图

4.5 数据分析

在这一部分，我们对实验数据进行详尽的分析，并对不同模型取得的性能进行对比。

经过对结果的充分分析，我们发现以下因素对模型的性能具有显著的影响：

关系抽取与KBQA其他环节的解耦程度 从表格中可以看出，无论模型是否使用支持语句，模型在将关系抽取与KBQA的其他因素（如实体链指）进行解耦时均取得了显著的性能提升：Baseline-Relation的性能相比Baseline-RDF高出7.1%，NLSS-Relation相比NLSS-RDF提高了8%。

此外，从上面的F1-epoch曲线图中可以看出，使用解耦匹配方法时，模型的平均F1均在约第4个epoch时取到，且即使笔者尝试对其继续训练，也不会有显著的下降；然而在使用整体匹配方法时，这一峰值往往在第1-2个epoch达到，随后便随着训练时间的推移不断下滑，出现过拟合现象。导致这一现象的原因可能是解耦方法对问句中的实体子串进行了替换，从而起到了降噪效果，并增强了泛化能力。

支持语句的质量 支持语句的作用是为了强化“关系”的概念。我们在测试集上将问题支持语句及对应的注意力输出，发现数据集中存在极多完全无关的语句；在少数相关的语句中，部分语句又存在截断后实体在前半句话中没有体现的问题。当每个问题支持语句数目选择为20时，尚有部分合理的支持语句；语句数目为50时噪声语句占了绝大多数。

问题的质量 我们对模型预测结果与正确结果一一比对分析发现，一部分错误来源于问题中关键词无法匹配，这一部分问题可以采用KNRM方法加以辅助。另一部分错误来源于问题中关系晦涩，这一部分问题可以采用支持语句加以辅助。当存在相当一部分问题本身即不可回答（或是在Freebase中无法查出正确答案，或是问题答案本身错误，或存在可替换的极其相似的解）。

5 结论

表格显示出了我们实验的结果，通过以上表格我们可以得出以下结论。

神经网络的作用 即使是没有进行任何的处理，使用最简单的Baseline-RDF模型，其F1值也相比STAGG中的传统机器学习模型提高了大约7%，由此可以看出神经网络能够通过训练更加充分学习理解到句子的含义，进行关系抽取。

实体替换的作用 通过将问题中的实体替换为特殊标志，从而将实体链指与关系抽取两个步骤进行解耦，其F1值相比不进行实体替换提高了大约7%，由此可以看出通过简单的实体替换对上一步骤进行解耦，可以有效提高关系抽取的效果。

支持语句的作用 支持语句可以对关系进行补充与说明，增加了注意力机制的神经网络应可以更加充分的获取关系信息，但由于支持语句质量问题，引入支持语句并没有取得预想的提升，仅获得了一个可以媲美的结果。

最终，我们的模型Baseline-Relation和NLSS-Relation在WebQuestion数据集上达到了45%的F1值，已经好于绝大部分系统。

但是遗憾的是，面对这样一个大规模、强耦合、高复杂度的系统，尽管我们不断花费大量时间与精力处理数据、修改模型，但由于支持语句质量问题（噪声太高），加入了注意力机制的记忆网络并没有取得预想的提升，在今后的工作中，会尝试对支持语句进行筛选提高质量，从而继续提高模型准确率。

参考文献

[1] Bast H, Haussmann E. More accurate question answering on freebase[C]//Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 2015: 1431-1440.

[2] Li H, Xiong C, Callan J. Natural Language Supported Relation Matching for Question Answering with Knowledge Graphs[J].

[3] Xu K, Feng Y, Reddy S, et al. Enhancing freebase question answering using textual evidence[J]. CoRR abs/1603.00957, 2016.

[4] Yih S W, Chang M W, He X, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base[J]. 2015.

[5] Xu K, Reddy S, Feng Y, et al. Question answering on freebase via relation extraction and textual evidence[J]. arXiv preprint arXiv:1603.00957, 2016.

[6] Yu M, Yin W, Hasan K S, et al. Improved Neural Relation Detection for Knowledge Base Question Answering[J]. arXiv preprint arXiv:1704.06194, 2017.

[7] Sukhbaatar S, Weston J, Fergus R. End-to-end memory networks[C]//Advances in neural information processing systems. 2015: 2440-2448.

[8] Bordes A, Usunier N, Chopra S, et al. Large-scale simple question answering with memory networks[J]. arXiv preprint arXiv:1506.02075, 2015.

[9] Anything A M. Dynamic Memory Networks for Natural Language Processing[J]. Kumar et al. arXiv Pre-Print, 2015.

[10] Severyn A, Moschitti A. Modeling relational information in question-answer pairs with convolutional neural networks[J]. arXiv preprint arXiv:1604.01178, 2016.

[11] Yih S W, Chang M W, He X, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base[J]. 2015.

[12] Yu M, Yin W, Hasan K S, et al. Improved Neural Relation Detection for Knowledge Base Question Answering[J]. arXiv preprint arXiv:1704.06194, 2017.

[13] Chen D, Fisch A, Weston J, et al. Reading Wikipedia to Answer Open-Domain Questions[J]. arXiv preprint arXiv:1704.00051, 2017.

[14] Berant J, Chou A, Frostig R, et al. Semantic Parsing on Freebase from Question-Answer Pairs[C]//EMNLP. 2013, 2(5): 6.

[15] Yin W, Schütze H, Xiang B, et al. Abcnn: Attention-based convolutional neural network for modeling sentence pairs[J]. arXiv preprint arXiv:1512.05193, 2015.

[16] Golub D, He X. Character-level question answering with attention[J]. arXiv preprint arXiv:1604.00727, 2016.

[17] Berant J, Liang P. Semantic Parsing via Paraphrasing[C]//ACL (1). 2014: 1415-1425.