

文章编号: 2095-2163(2019)03-0001-06

中图分类号: TP393.01

文献标志码: A

# 基于序列到序列模型的文本到信息框生成的研究

鲍军威<sup>1</sup>, 周明<sup>2</sup>, 赵铁军<sup>1</sup>

(1 哈尔滨工业大学 计算机科学与技术学院, 哈尔滨 150001; 2 微软亚洲研究院, 北京 100080)

**摘要:** 本文展示了一种序列到序列的模型(Seq2Seq)来基于文本生成信息框(Infobox),信息框指的是一组“属性-值”对。该模型以端到端的方式工作,可利用一个编码器将一个文本段落表示成一个隐向量序列,然后通过解码器来生成信息框。本文在 WIKIBIO 数据集上进行实验。研究提出的序列到序列模型取得了 58.2 的  $F_1$  值,该结果比流水线式的基准方法显著提升了 21.0 个百分点。实验结果表明,本模型具有以生成序列的方式来生成“属性-值”对的能力。引入注意力与拷贝机制可以提升模型的准确率。更重要的是,研究观察到该拷贝机制有能力从输入文本中拷贝稀有词来生成目标端信息框中的“值”。

**关键词:** 文本到信息框生成; 序列到序列模型; 注意力机制; 拷贝机制

## Text-to-infobox generation with sequence to sequence learning

BAO Junwei<sup>1</sup>, ZHOU Ming<sup>2</sup>, ZHAO Tiejun<sup>1</sup>

(1 School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China;

2 Microsoft Research Asia, Beijing 100080, China)

**[Abstract]** The paper proposes a sequence-to-sequence (Seq2Seq) model to generate infoboxes from the text, where an infobox is a set of attribute-value pairs. The model works in an end-to-end fashion, which represents a paragraph as a sequence of hidden vectors with an encoder and then generates an infobox with a decoder. The paper conducts experiments on the WIKIBIO dataset. The Seq2Seq model achieves an average  $F_1$  score of 58.2, which significantly outperforms a pipeline-based approach by 21.0 points. Experiment results show that the seq2seq model has the ability to generate a list of attribute-value pairs in a sequential way. Incorporating both the attention and copying mechanisms improves the accuracy of the model. Furthermore, the results demonstrate that the copying mechanism is capable of replicating rare words to generate values.

**[Key words]** text-to-infobox; sequence-to-sequence model; attention mechanism; copying mechanism

## 0 引言

信息框(Infobox)是一种结构化的知识(<https://en.wikipedia.org/wiki/Infobox>),其中包含一组关于相应文章主题的“属性-值”对。信息框可以应用在很多需要推理与推断的场景中,例如基于知识图谱的自动问答和语义解析等<sup>[1-4]</sup>。然而,基于文本通过人工来构建信息框将会耗费大量的资源和成本。而且,更新已经构建好的信息框也需付出大量时间。因此,自动地从文本中抽取信息框受到学界越来越多的关注与重视<sup>[5-7]</sup>。传统的信息框抽取方法通常包含一个复杂的流程,并主要依赖于专家知识和特征工程。同时,这些系统却常会遭受错误传导的问题。为此,探索端到端的方法来自动学习生成信息框是十分必要的。

最近,序列到序列(Seq2Seq)的模型在很多任务中得到了成功运用,例如机器翻译<sup>[8-9]</sup>,文本摘要<sup>[10]</sup>等。本文提出一种新的方法将 Seq2Seq 模型

扩展到文本到信息框的生成任务上。总地来说,该模型将输入文本通过编码器表征成一组实值向量,而后通过解码器基于这些实值向量来生成信息框。在本文中,研究将一组“属性-值”对转化成序列,由此使得 Seq2Seq 模型能够以序列的形式来生成信息框。注意力机制被用于计算各种各样的自然语言表达与“属性”的相似度。同时,研究中还采用拷贝机制从源文本中拷贝稀有词来生成信息框中的“值”。

本次研究在包含了 728 321 个“人物传记-信息框”对的 WikiBio 数据集<sup>[11]</sup>上进行了实验。Seq2Seq 模型取得了 58.2 的平均  $F_1$  值,该结果比流水线式的方法显著提升了 21.0 个百分点。经由大量的实验证明可知,Seq2Seq 模型有能力以生成序列的方式来生成信息框。注意力与拷贝机制的引入最终将大幅提升模型的准确率。而在研究后却又发现拷贝机制有能力从输入文本拷贝稀有词来生成信息框中的“值”。尤需一提的是,模型分析表明 Seq2Seq 模型在高频属性上的表现会更好。错误分析则表明

**作者简介:** 鲍军威(1988-),男,博士研究生,主要研究方向:自然语言处理、语义解析、自动问答等;周明(1964-),男,博士,教授,博士生导师,主要研究方向:自然语言处理;赵铁军(1962-),男,博士,教授,博士生导师,主要研究方向:自然语言处理、机器学习。

收稿日期: 2018-11-13

哈尔滨工业大学主办 ◆ 学术研究与应用

Seq2Seq 模型在那些需要逻辑推理的“属性”与“值”的生成方面具有一定局限。

## 1 任务与数据集

### 1.1 任务

给定文本  $x = \{x_1, \dots, x_T\}$ , 文本到信息框的生成任务是基于  $x$  来生成一个包括一组“属性-值”对的信息框  $I = \{ \langle A_1, V_1 \rangle, \dots, \langle A_N, V_N \rangle \}$ 。其中, “属性”  $A_i = \{y_1^{(i)}, \dots, y_M^{(i)}\}$  是包含  $M$  个词的序列, “值”  $V_i = \{z_1^{(i)}, \dots, z_N^{(i)}\}$  是包含  $N$  个词的序列。

### 1.2 数据集

WikiBio<sup>[11]</sup> 是来自于 Wikipedia 的一组包含“人物传记-信息框”对的数据集。人物传记是对应的 Wikipedia 文章的第一段, 同时该段落往往包含了多个句子。WikiBio 包含了 728 321 个实例, 且被划分为 3 个部分, 包括 582 659 条训练数据、72 831 条开发数据、以及 72 831 条测试数据。在本文中, 研究使用该数据集来基于人物传记生成信息框。在本文的设定中, 如果一个“属性-值”对的“值”中没有这个词出现在输入文本中, 那么研究将会过滤掉这个

“属性-值”对。如果一个“值”中包含超过 5 个词, 那么相应的“属性-值”对也会被移除。最终, 本文获得了 WikiBio 的一个子集, 集合中包含了 580 069 个实例作为训练数据, 72 502 个实例作为开发集数据, 72 517 个实例作为测试数据。在此基础上, 研究得出的过滤后的训练数据的统计信息见表 1。

表 1 训练数据上的统计信息

Tab. 1 Statistics on the proposed training set

类型	值
每个实例中文本的平均词长	97.10
每个“值”的平均词长	2.58
每个实例中“属性”的平均个数	5.09
数据集的“属性”的个数	3 366
数据集中频率大于 10 的属性的个数	1 158

## 2 方法

在本节中, 研究展示一个整合了注意力机制与拷贝机制的 Seq2Seq 模型来进行文本到信息框的生成。继而给出了本方法进行文本到信息框生成的设计展示如图 1 所示。对此可做探讨分述如下。

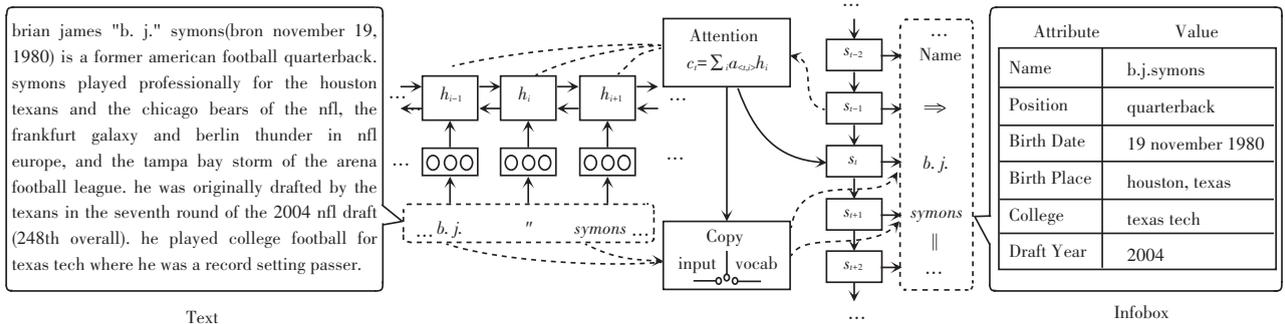


图 1 文本到信息框生成的框架总览

Fig. 1 Overview of the proposed text-to-infobox generation framework

### 2.1 编码器

为了表示输入文本的语义, 研究使用一个编码器来将变长的文本映射到一组定长的连续向量。而后又采用双向的基于门循环单元 (Gated Recurrent Unit, GRU) 的循环神经网络 (Recurrent Neural Network, RNN)<sup>[12]</sup> 将输入文本  $x = \{x_1, \dots, x_T\}$  编码成一个隐向量序列  $\mathbf{H} = \{h_1, \dots, h_T\}$ , 其中  $T$  是序列的长度。每一个在  $t$  时刻的隐向量  $h_t$  都是由双向隐向量  $\bar{h}_t$  和  $\overleftarrow{h}_t$  拼接而成。

### 2.2 解码器

研究开发了一个解码器, 故而可基于编码器的

隐向量组  $\mathbf{H}$  来生成信息框  $I$ 。首先,  $I$  被转化成一个序列  $\{A_0, \rightarrow, V_0, \parallel, \dots, \parallel, A_N, \rightarrow, V_N\}$ , 其中字符“ $\rightarrow$ ”被插入到每个“属性”与“值”的中间, 字符“ $\parallel$ ”被插入到 2 个“属性-值”对的中间。通过这种转化, 解码器的目标转换为生成一个序列  $y = \{y_1, \dots, y_T\}$ 。就整体而言, 解码器被设计成 RNN, 其中每个时间步  $t$  都会生成一个词  $y_t$ , 直到生成一个特殊的 EOS 字符。这个过程中, 每个时间步都被形式化成以下过程:

$$p(y_t | y_{<t}, x) = p(y_t | y_{<t}, \mathbf{H}) = f(y_{t-1}, s_t, c_t). \quad (1)$$

其中,  $s_t = GRU(y_{t-1}, s_{t-1}, c_{t-1})$  是解码器的隐向量;  $c_t$  是通过注意力机制运行得到的上下文向量;  $f(\cdot)$  是参数化的函数,能够计算在输出候选空间上的概率分布。传统的方法中,输出候选空间只由一个相对较小的目标词典构成,且词典中只有一个 UNK 字符来表示所有可能的未登录词(out-of-vocabulary, OOV)。在本次工作中,研究采用了拷贝机制来处理 OOV 的问题。

### 2.3 注意力机制

这里,在分析基础上将观察到输入端的不同的词对生成特定的“属性-值”对的贡献是不同的。因此,解码器在生成每个词时就能够有选择性地使用输入文本中不同的信息。综合以上论述,本次研究就采用注意力机制<sup>[13]</sup>,在解码的每个时刻对输入文本中的每个词都赋予一个概率/权重值,该机制已成功地应用在许多任务中,诸如机器翻译<sup>[14]</sup>和阅读理解<sup>[15]</sup>。解码第  $t$  时刻的词时,对于输入文本中第  $i$  个词的注意力权重可定义为:

$$\alpha_{\langle t, i \rangle} = \frac{\exp [a(s_t, c_{t-1}, h_i)]}{\sum_{i=1}^T \exp [a(s_t, c_{t-1}, h_i)]}. \quad (2)$$

其中,函数  $a(\cdot)$  是一个多层感知器 (Multi-Layer Perceptron, MLP)。此后,注意力权重将用来计算上下文向量  $c_t = \sum_{i=1}^N \alpha_{\langle t, i \rangle} h_i$ 。

### 2.4 拷贝机制

研究观测到的另外一个现象是信息框中的“值”通常是 OOV 的词,并且往往被包含于输入文本中,例如命名实体和数字等。为了使模型具有从输入文本中拷贝 OOV 词的能力,研究不仅在目标词表  $v$  上计算了概率分布  $p_i^v(\cdot)$ ,也在输入文本  $x$  上计算了概率分布  $p_i^s(\cdot)$ ,来共同决定解码过程中每个时刻应该输出什么词。具体而言,研究定义  $p_i^s(\cdot) = \alpha_{\langle t, i \rangle}$ ,其中  $\alpha_{\langle t, i \rangle}$  是注意力权重。此外,还定义  $g_t(\cdot) = f_g(y_{t-1}, s_t, c_t)$  为一个值域为  $[0, 1]$  的开关,该开关可用来权衡是从目标词典中生成词、还是从输入文本中拷贝词。 $f_g(\cdot)$  则定义为带有  $\sigma$  激活函数的 MLP,该函数输出一个  $|u| = |v \cup x|$  维的向量。最终,生成一个词  $|\tilde{y}|$  的概率可以通过如下公式获得:

$$p(y_t | y_{\langle t \rangle}, x) = g_t(\tilde{y}) \odot p_i^s(\tilde{y}) + (1 - g_t(\tilde{y})) \odot p_i^v(\tilde{y}). \quad (3)$$

如果  $\tilde{y} \notin x$ , 则  $p_i^s(\tilde{y}) = 0$ ; 如果  $\tilde{y} \notin v$ , 则  $p_i^v(\tilde{y}) = 0$ 。

## 3 实验

### 3.1 实验设置

#### 3.1.1 数据集

文中在 2.2 节描述的过滤后的 WikiBio 数据集上进行实验。在本次研究的实验设置中,一个“属性”可当作是一个单独的词。如此一来,文中仅使用开发集中的前 1 000 个数据进行开发。

#### 3.1.2 评测指标

研究中采用准确率 ( $P$ )、召回率 ( $R$ )、以及  $F_1$  值并基于金标准的一组“属性-值”对来推得评价预测的一组“属性-值”对。最终在整个测试数据集上的  $P, R, F_1$  值是所有实例上相应得分的平均值。

#### 3.1.3 基准线方法

受已有研究<sup>[5]</sup>的启发,设计实现了一个流水线式的方法作为基准方法。该方法包括了一组分类器和一组抽取器。每个“属性”对应到一个分类器,该分类器用来确定一段文本是否可以生成该“属性”。每个抽取器将用来为某个“属性”进行相应“值”的抽取。被抽取出来的值,例如时间,都进行了正则化。这些分类器是基于 Classias (<http://www.chokkan.org/software/classias/usage.html.en>) 工具中的采用 L2 正则化的 L1 损失函数的 SVM 模型实现的。抽取器采用 CRF++ (<https://taku910.github.io/crfpp>) 模型。研究中,通过使用词汇化的特征,而非诸如词性(part of speech, POS)等句法特征,来训练抽取器。这样做的目的是使其与 Seq2Seq 模型运用相同的信息以增加可比性。S2S 是一个基于 GRU-RNN 结构的不带有注意力机制与拷贝机制的 Seq2Seq 模型。研究同时也实现了 Seq2Seq 模型的其他变种。这里,将在实验结果部分对这些模型变种做出阐释论述如下。

### 3.2 实验结果

生成“属性-值”对的实验结果见表 2。表 2 中, atn 表示注意力机制, copy 表示拷贝机制, A 表示属性, V 表示值。实验结果表明 2 个带有注意力与拷贝机制的 S2S 模型要明显好于基于流水线的方法(从 37.2 到 58.2, 从 37.2 到 55.8)。引入了注意力机制可以带来 17.6 个点的提升(从 22.5 到 40.1)。拷贝机制使得模型的结果提升了 18.1 个百分点(从 40.1 到 58.2)。更重要的是,将“值”加入到目标词

表并且使用拷贝开关来权衡可以带来 2.4 个点的提升(从 55.8 到 58.2)。

研究得到的分别在“属性”与“值”上单独评估的结果见表 3。表 3 中, Attributes 表示研究只对生成的“属性”进行评测, values 表示只对生成的值进行评测, 这些值对应的属性都是预测正确的。atn 表示注意力机制, copy 表示拷贝机制, A 表示属性, V 表示值。除了基本的 S2S 模型外, 其它带有注意力机制或者拷贝机制的模型在“属性”与“值”上的结果都要好于基于流水线的方法。经过分析可知, 注意力机制在“属性”与“值”的生成上都带来提升, 分别是 69.9 到 73.3 以及从 30.9 到 52.7。实验结果表明注意力机制对提升模型的准确率是很有帮助的。拷贝机制在“属性”生成上带来的提升比较有限(从 73.3 到 74.2), 在“值”的生成上带来的提升比较显著(从 52.7 到 76.4)。这是由于“属性”与“值”不同, 而且常常不会出现在文本表述中。通过观测数据, 研究发现不带有拷贝机制的 Seq2Seq 模型只能生成特殊符号 UNK 来表示 OOV 的词, 而带有拷贝机制的模型可以从文本中拷贝内容来生成稀有词。此外, 目标词表中只包含“属性”的 S2S+atn 模型在“属性”生成上取得了最好的结果。

表 2 “属性-值”对生成的实验结果

Tab. 2 Experiment results on attribute-value pair generation

模型	目标词表	$P$	$R$	$F_1$
Pipeline	-	40.9	36.6	37.2
S2S	A + V	24.2	22.5	22.5
S2S+atn	A + V	43.9	39.2	40.1
S2S+atn+copy	A + V	64.2	56.5	58.2
S2S+atn+copy	A	62.1	53.8	55.8

表 3 “属性”与“值”生成的各自的实验结果

Tab. 3 Experiment results on attributes and values, respectively

模型	目标词表	属性			值
		$P$	$R$	$F_1$	$F_1$
Pipeline	-	77.2	69.9	70.5	50.4
S2S	A + V	73.8	71.5	69.9	30.9
S2S+atn	A + V	81.1	71.7	73.3	52.7
S2S+atn+copy	A + V	81.9	72.3	74.2	76.4
S2S+atn	A	81.6	74.2	75.2	-
S2S+atn+copy	A	82.6	71.9	74.3	73.4

### 3.3 实验分析

#### 3.3.1 模型分析

研究进行实验来探索“属性”的频率对信息框

生成的影响。相应地, 研究基于训练数据集上“属性”的频率将测试数据集分成 7 份。图 2 给出了每个部分的运行结果。图 2 中, 横坐标表示“属性”的不同频率范围。该结果表明“属性”的频率越高, 模型取得的结果越好。

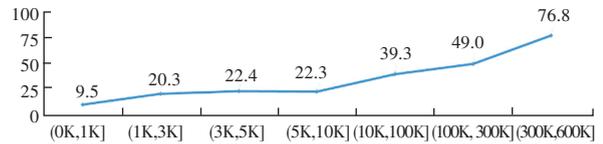
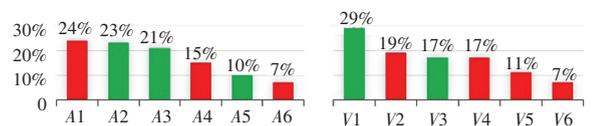


图 2 在 S2S+atn+copy 模型上采用 A+V 为目标词表的实验设定上不同部分的平均  $F_1$  值

Fig. 2 Average  $F_1$  scores on different portions of results on S2S+atn+copy with A+V as target vocabulary

#### 3.3.2 错误分析

错误分析的结果即如图 3 所示。为了分析“属性”生成中产生的错误, 研究随机选取了 100 个实例, 其中每个实例有至少一个预测错误的“属性”。由此则发现 54% 的实例的错误是可以接受的, 包括 A2 类(不恰当的金标准“属性”)、A3(不充足的金标准“属性”)、A5(预测了同义的“属性”)。“属性”生成的主要错误是 A1(需要逻辑推理)、A4(预测错的“属性”的频率太低)、A6(其它类型)。此外, 随机选取其它 100 个实例, 而且每个实例中都存在某个“属性”预测正确而“值”预测错误的“属性-值”对。这些实例中 46% 是可以接受的, 包括 V1(预测了同义的“值”)、V3(错误标注了的金标准的“值”)。主要的错误包括 V2(预测了相似的值)、V4(需要逻辑推理)、V5(预测部分“值”)、V6(其它)。



(a) “属性”错误分布 (b) “值”错误分布  
(a) Attribute error distribution (b) Value error distribution

图 3 不同错误类型的分布

Fig. 3 Distributions on different error types

## 4 相关工作

本工作主要涉及 2 条研究主线, 对此可做探讨分述如下。

#### 4.1 Wikipedia 上文本到结构化知识的预测

已有的一些知名的产品都是通过流水线式的方式, 并基于 Wikipedia 进行信息抽取而得到的, 例如 DBpedia<sup>[16]</sup>、YAGO<sup>[17]</sup>、YAGO2<sup>[18]</sup>、KYLIN<sup>[5]</sup>、IBminer<sup>[6]</sup>、

iPopulator<sup>[19]</sup>、以及 WOE<sup>[20]</sup>。Nguyen 等人<sup>[21]</sup>提出使用句法信息与语义信息,就可从 Wikipedia 中抽取关系。Zhang 等人<sup>[7]</sup>使用连接命名实体(重定向)来将一篇文章总结成信息框。在本文中,研究采用序列到序列(Seq2Seq)的神经网络模型来端到端地进行文本到信息框的生成。

#### 4.2 序列到序列(Seq2Seq)的学习

Seq2Seq 模型在很多任务上均可见到效果可观的应用尝试,例如机器翻译、对话回复生成和文本摘要<sup>[10,12,22-23]</sup>。在本文中,设计展示了一个 Seq2Seq 模型来进行文本到信息框的生成。文中的 Seq2Seq 模型采用了带有注意力机制与拷贝机制的“编码器-解码器”框架。不同的注意力机制<sup>[13-14, 24]</sup>也相继推出,用于将输入序列与输出序列进行对齐。受到这些工作的启发,本文研究采用一种注意力机制来连接自然语言与信息框。拷贝机制的提出受益于近期的关于指向网络(pointing network)与拷贝(copying)方法<sup>[10,25-29]</sup>的成功。

### 5 结束语

本文提出了一个序列到序列(Seq2Seq)的模型来进行文本到信息框的生成。实验结果表明 Seq2Seq 模型有能力以生成序列的方式来生成一个结构化的信息框。研究观察到引入了注意力与拷贝机制可以提升模型的准确率,同时拷贝机制有能力从原文本中拷贝稀有词来生成目标信息框中的“值”。

#### 参考文献

[1] MOONEY R J. Learning for semantic parsing[C] //Computational Linguistics and Intelligent Text Processing (CICLing). Mexico City:dblp,2007;311-324.

[2] BERANT J, CHOU A, FROSTIG R, et al. Semantic parsing on freebase from question-answer pairs[C] //Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics,2013;1533-1544.

[3] YIH W T, CHANG Mingwei, HE Xiaodong, et al. Semantic parsing via staged query graph generation: Question answering with knowledge base [C] // Proceedings of the 53<sup>rd</sup> Annual Meeting of the Association for Computational Linguistics and the 7<sup>th</sup> International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Beijing, China: Association for Computational Linguistics, 2015;1321-1331.

[4] BORDES A, USUNIER N, CHOPRA S, et al. Large-scale simple question answering with memory networks [J]. arXiv preprint arXiv:1506.02075,2015 .

[5] WU Fei, WELD D S. Autonomously semantifying wikipedia [C]

//Proceedings of the sixteenth ACM conference on Information and Knowledge Management. New York, USA: ACM, 2007;41-50.

[6] MOUSAVI H, GAO Shi, ZANIOLO C. Ibmminer: A text mining tool for constructing and populating infobox databases and knowledge bases [J]. Proceedings of the VLDB Endowment, 2013,6(12):1330-1333.

[7] ZHANG Kezun, XIAO Yanghua, TONG Hanghang, et al. The links have it: Infobox generation by summarization over linked entities [J]. arXiv preprint arXiv:1406.6449,2014.

[8] KALCHBRENNER N, BLUNSOM P. Recurrent continuous translation models [C] //Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL, 2013;1700-1709.

[9] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks [C] // Advances in Neural Information Processing Systems. Montreal, Canada: Nips, 2014; 3104-3112.

[10] NALLAPATI R, ZHOU Bowen, SANTOS C, et al. Abstractive text summarization using sequence-to-sequence RNNs and beyond [J]. arXiv preprint arXiv:1602.06023,2016.

[11] LEBRET R, GRANGIER D, AULI M. Neural text generation from structured data with application to the biography domain [C] // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas: Association for Computational Linguistics, 2016;1203-1213.

[12] CHO K, VAN MERRIENBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [C] //Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014;1724-1734.

[13] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate [J]. arXiv preprint arXiv:1409.0473,2014 .

[14] LUONG M H, PHAM H, Manning C D. Effective approaches to attention-based neural machine translation [J]. arXiv preprint arXiv:1508.04025,2015.

[15] KADLEC R, SCHMID M, BAJGAR O, et al. Text understanding with the attention sum reader network [J]. arXiv preprint arXiv:1603.01547,2016.

[16] AUER S, LEHMANN J. What have Innsbruck and leipzig in common? Extracting semantics from wiki content [C] // The Semantic Web: Research and Applications. Innsbruck Austria: Springer, 2007;503-517.

[17] SUCHANEK F M, KASNECI G, WEIKUM G. Yago: A core of semantic knowledge [C] // Proceedings of the 16<sup>th</sup> International Conference on World Wide Web. Banff, Alberta, Canada: dblp, 2007; 697-706.

[18] HOFFART J, SUCHANEK F M, BERBERICH K, et al. Yago2: A spatially and temporally enhanced knowledge base from wikipedia [J]. Artificial Intelligence, 2013,194;28-61.

[19] LANGE D, BÖHM C, NAUMANN F. Extracting structured information from wikipedia articles to populate infoboxes [C] // Proceedings of the 19<sup>th</sup> ACM International Conference on Information and Knowledge Management. Toronto, Ontario, Canada: ACM, 2010;1661-1664.