计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(8):164−169 [doi: 10.15888/j.cnki.csa.006474]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于深度学习的网站权威性预测①

杨海华[1,2], 冯仰德[1], 王　珏[1], 聂宁明[1], 刘　芳[1], 张博尧[1]

[1](中国科学院 计算机网络信息中心, 北京 100190)
[2](中国科学院大学, 北京 100049)
通讯作者: 冯仰德, E-mail: ydfeng@sccas.cn

摘　要: 网站权威性一般是由外部链接来衡量, 高质量的外部链接越多, 网站的权威性就越高; 常用的评价网站权威性的算法有 PageRank 等, 然而该类算法对网站权威性的影响是有选择性的, 使得这种方法具有一定的弊端. 本文利用深度学习的方法, 通过将搜索词和网址映射为向量, 计算两个向量之间的相似度来评判在某个搜索词下不同网址的权威性, 把计算结果相似度高对应的网站称为在该搜索词下权威性高的网站, 从而从另一种角度去衡量网站的权威性. 通过对比使用 Word2vec 和 LSTM 两种不同的模型实验, 在对公开的数据集上的实验结果表明使用这两种模型是有效的, 其中 LSTM 模型比 Word2vec 模型的效果要好.

关键词: 网站权威性; Word2vec; LSTM; 自然语言处理

引用格式: 杨海华,冯仰德,王珏,聂宁明,刘芳,张博尧.基于深度学习的网站权威性预测.计算机系统应用,2018,27(8):164–169. http://www.c-s-a.org.cn/1003-3254/6474.html

Website Authority Prediction Based on Deep Learning

YANG Hai-Hua[1,2], FENG Yang-De[1], WANG Jue[1], NIE Ning-Ming[1], LIU Fang[1], ZHANG Bo-Yao[1]

[1](Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Website authority is generally measured by external links. The more high-quality external links are, the more authoritative the website or web page itself is. Evaluation website authoritative algorithm has PageRank and so on. However, the impact of such algorithms on the authority of the website is selective, making this method has some drawbacks. This study uses the method of deep learning, by mapping search terms and URLs into vectors, and then calculates the similarity between two vectors to judge the authority of different websites under a certain search term. The website with high similarity of calculation results is referred to as an authoritative site under the search term, so we can use another view to measure the authority of website. By comparing two different model experiments using Word2vec and LSTM, the experimental results on open datasets show that it is effective to use both models, and LSTM model is better than Word2vec model.

Key words: website authority; Word2vec; LSTM; Natural Language Processing (NLP)

网站权威性一般是由外部链接数来衡量, 常用的算法有 PageRank[1]等算法. 一般而言高质量的外部链接越多, 网站或者网页本身的权威性就越高. 这类的方法的缺点在于外部链接对网站权威性的影响是有选择性的, 也就是说来自相关内容网站的链接, 对提高权威性帮助最大, 不相关内容的链接帮助很小. 比如在某个

① 基金项目: 国家重点研发计划 (2017YFB0203704)
Foundation item: National Key R & D Program of China (2017YFB0203704)
收稿时间: 2017-12-18; 修改时间: 2018-01-04; 采用时间: 2018-01-11; csa 在线出版时间: 2018-07-28

音乐网站首页上加一个链接到某个美食网站, 那么对美食网站的权威性几乎没什么帮助; 而且甚至在某些场景下这种方式的权威性可以被操作, 即可以有很多链接是人为设置的. 本文通过使用将搜索词和被召回的网站的 URL 映射为向量的方法, 计算两个向量之间的相似度来评价搜索词和网站之间的相似度, 相似度越高, 那么该网站的权威性越高, 所以如何精确的表达搜索词和网站的向量是最为关键的问题. 本文利用深度学习[2,3]的技术, 分别使用 Word2vec[4]模型与 LSTM (Long Short-Term Memory) 递归神经网络模型[5]来获取搜索词得向量表示; 通过随机初始化网址的向量, 然后在模型的训练中实时的更新词向量和网址向量, 从而得到最终的词汇表和网址向量表; 其中 Word2vec 模型和 LSTM 模型获取搜索词的向量方式有些不同, 具体见下一章节.

1 方法及模型介绍

本文在模型训练过程中一共使用了两张向量表, 一张用户搜索词集合的词汇表, 另外一张是所有网站 URL 集合的网址表, 两张表开始时都是随机初始化的. 在 Word2vec 模型的实验中, 最终的词汇表是由训练 Word2vec 模型得来的, 网址表是在整体的模型训练中生成的, 在计算相似度时, 使用的搜索词向量是通过查由 Word2vec 模型训练得到的词汇表获取的, 网址向量是在模型收敛时得到的网址表中获取的, 然后计算搜索词和网站之间的相似度. 例如用于计算相似度的搜索词为“我是中国人”, 将“我是中国人”通过切词得到“我”、“是”、“中国人”三个词, 然后通过查由 Word2vec 训练得到的词汇表获取相应的词的向量, 最后将查到的 3 个词的向量相加作为整个搜索词的向量; 在 LSTM 模型试验中, 最终的词汇表和网站表都是在利用整体网络端到端的训练过程中生成的, 在计算相似度时, 使用的搜索词向量是由两个步骤得到的: ① 搜索词切词后得到的词通过查词汇表得到每个词向量, 然后作为 LSTM 的输入; ② 取 LSTM 模型最后一个时刻的隐层输出得到最终整个搜索词的向量表达; 网址向量是在模型收敛时得到的网址表中获取的, 然后计算搜索词和网站之间的相似度. 例如用于计算相似度的搜索词为“我是中国人”, 将“我是中国人”通过切词得到“我”、“是”、“中国人”三个词, 然后通过查由整个模型训练得到的词汇表获取相应的三个词的向量作为 LSTM 的输入, 最后取 LSTM 模型最后一个时刻隐层的输出作为整个搜索词的向量.

1.2 Word2vec 模型训练词向量

Word2vec 模型是 Mikolov 等人在 NNLM[6]以及 Log-Bilinear 模型[7]基础上开发的工具, 分为连续 Bag Of Words (CBOW) 和连续 Skip-gram 两种模型. CBOW 模型利用上下文中的若干词去预测当前词; 而 Skip-gram 模型恰好相反, 利用当前词预测上下文的若干词.

两种模型的训练过程类似, 对于 CBOW 模型, 输入层是词的上下文中的若干词向量, 为了充分利用上下文信息预测中间词, 并且不损失句子长度信息的情况下, 将该词的上下文中的若干个词向量以累加的方式得到中间层的向量. 输出层则是以训练语料库中的词作叶子节点, 以各词出现次数为权值构造的一棵 Huffman 树. 通过随机梯度上升算法[4]对投影层向量可能表示的词进行预测, 使得 $p(W|con(W))$[8]值最大化, W 表示当前词, $con(W)$ 表示该词上下文中的若干词. 当神经网络训练完成时, 即可求出所有词的词向量. Word2vec 输出的词向量可以用于解决自然语言处理 (NLP) 方面的问题, 如情感分析[9]、图片描述[10]等. Lilleberg 等人[11]在做文本分类相关工作方面时, 为了更好的得到语义特征, 使用了 Word2vec 训练词向量, 并且得到不错的效果.

1.3 LSTM 网络模型

LSTM 网络是 RNN 的扩展, 它成功的解决了原始循环神经网络的缺陷, 成为当前最流行的 RNN. LSTM 的神经网络基本模块具有不同的结构, 这与传统的 RNN 不同, 传统的 RNN 模型的隐藏层只有一个状态, 即 h, 它对于短期的输入非常敏感. LSTM 基本模块中增加了一个新的单元状态 C, 如图 1 所示.

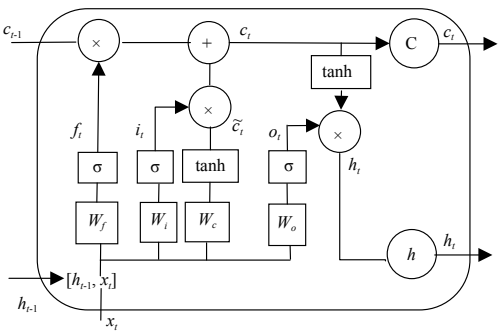

图 1 LSTM 网络模块示意图

LSTM 具有遗忘门 (forget gate)、输入门 (input gate)、和输出门 (output gates) 等三种门结构, 用以保持和更新细胞状态. 以下是三种门的具体作用方式:

(1) 遗忘门: 它决定了上一时刻的单元状态 c_{t-1} 有多少保留到当前时刻 c_t.

$$f_t = \sigma\left(w_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

其中, w_f 是遗忘门的权重矩阵, x_t 是当前时刻网络的输入值, h_{t-1} 是上一时刻 LSTM 的输出值, $[h_{t-1}, x_t]$表示把两个向量连接成一个更长的向量, b_f 是遗忘门的偏置项, σ 是 Sigmoid 函数.

(2) 输入门: 它决定当前时刻网络的输入 x_t 有多少保存到单元 c_t.

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

其中, w_i 是输入门的权重矩阵, b_i 是输入门的偏置项.

(3) 输出门: 它是控制单元状态 c_t 有多少输出到 LSTM 的当前输出值 h_t.

$$\tilde{c}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{4}$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \circ \tanh(c_t) \tag{6}$$

以上式子中符号∘表示按元素乘. 式 (3) 中$\tilde{c}_t$用于描述当前输入的单元状态, 它是根据上一次的输出和本次输入来计算的; 式 (4) 计算的是当前时刻的状态单元 c_t; 式 (5) 是把当前输入的单元状态$\tilde{c}_t$和 c_{t-1} 组合在一起, 形成新的状态单元 c_t; 式 (6) 的输出最终是由输出门和单元状态共同确定的.

LSTM 网络模型已被成功地应用于图片/视频描述[12–14]、文本/情感分类[15–18]、机器翻译[19]、智能问答[20,21]等自然语言处理任务中. 由于 LSTM 网络通过记忆单元去学习从细胞状态中忘记信息、去更新细胞状态的信息, 具有学习文本序列中远距离依赖的特性, 所以很自然地想到使用 LSTM 网络模型学习本文需要的搜索词的向量表达.

2 模型具体设计

2.1 使用 Word2Vec 模型的实验设计

使用 Google 开源的 Word2vec 模型, 训练语料库为数据集中的搜索词, 结合实际使用经验设置词向量的维度大小为 256; 由于用户输入的查询词一般在 10 个字以内, 所以设置模型中对应滑动窗口大小为 2; 这里使用 CBOW 方式训练 Word2vec 模型, 得到该实验最终的词向量表; 网址表的向量初始为随机生成, 维度大小为 256, 初始权重范围为: [–1.0, 1.0], 之后随着模型的训练实时更新; 使用 Tensorflow 框架搭建训练模型, 设置其初始的学习速率为 0.1, 使用 tensorflow 提供的指数衰减函数 tf.train.exponential_decay 动态的更新学习率; 为了防止模型在训练过程中过拟合, 采用了 dropout 方式, 其相关系数设置为 0.5, 并结合使用 early_stopping 的方式: 即记录到目前为止最好的 accuracy, 当连续 2 次 Epoch 没达到最佳 accuracy 时, 则停止训练. Batchsize 的大小设置为 256; 模型训练的最大迭代次数设为 100. 具体的训练步骤如下:

1) 将所有搜索词切词. 将切词后的搜索词通过调用 Word2vec 模型训练, 得到最终的词向量表;

2) 对于新来的一个搜索词, 通过切词, 然后通过查询由 Word2vec 生成的词向量表, 将切词后对应的词的向量相加作为该搜索词的向量表达, 记为 $Query_{EM}$.例如: 搜索词为“我是中国人”, 切词后分别为“我”、“是”、“中国人”三个词, 然后通过查询词向量表获取三个相应的词向量, 然后将这三个词向量相加得到关于“我是中国人”这个搜索词的向量表达;

3) 通过使用 tf.nn.embedding_lookup 函数查随机初始化的网址表得到 URL1、URL2 的向量表达, 分别记为 URL_{BEM}、URL_{WEM}, 并将其值设置为可训练的;

4) 将 Sigmoid 函数作用于搜索词的向量与点击率高的网址向量相加的结果上, 其结果记为 Q_{BSCORE}; 同理将 Sigmoid 函数作用于搜索词的向量与点击率低的网址向量相加的结果上, 其结果记为 Q_{WSCORE}; Q_{BSCORE}、Q_{WSCORE} 的具体计算方式如下:

$$Q_{BSCORE} = f\{\sum_{i=1}^{n}(Query_{EM} \cdot URL_{BEM})\} \tag{7}$$

$$Q_{WSCORE} = f\{\sum_{i=1}^{n}(Query_{EM} \cdot URL_{WEM})\} \tag{8}$$

其中, f 为 Sigmoid 激活函数; n 为向量的维度大小, $\cdot$符号为向量对应位置相乘. Q_{BSCORE} 与 Q_{WSCORE} 之间的差越大, 即认为在当前搜索词下 URL1 的权威性比 URL2 权威性高.

5) 定义损失函数并记为 $Loss$, 并以此优化模型, 公式 (9) 为其计算方式:

$$Loss = \sum_{i=1}^{n} \max((Q_{BSCORE} - Q_{WSCORE}), 0) \quad (9)$$

其中, n 为 Batchsize 的大小.

6) 训练过程中, Q_{BSCORE} 与 Q_{WSCORE} 之间的差大于 0 的样本为正例, 小于 0 为负例. 每个 Batchsize 的平均准确率的计算方式为当前批次中正例的个数除以当前批次的总样本数.

具体流程如图 2 所示.

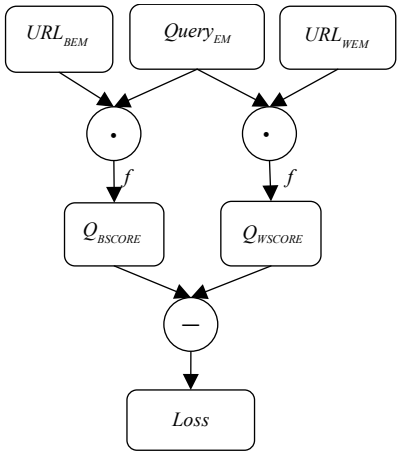

图 2 Word2vec 模型实验的训练流程图

2.2 使用 LSTM 模型的实验设计

同样设置词向量与网址向量的维度大小为 256; 词向量表与网址表中网址的向量随机初始化的, 初始权重范围为: [–1.0, 1.0]; 使用 TensorFlow 搭建 LSTM 模型, 设置其网络层次为 1 层, 时间维度的最大长度为 20, 隐层维度为 256; 设置其初始的学习速率为 0.1, 然后使用 tensorflow 提供的指数衰减函数 tf.train.exponential_decay 动态的更新学习率; 为了防止模型在训练过程中过拟合, 采用了 Dropout 方式, 其相关系数设置为 0.5, 并结合使用 early_stopping 的方式. Batchsize 的大小设置为 256; 模型训练的最大迭代次数设为 100; 与 Word2vec 模型不同的是, 这里词汇表和网址表都是随着模型训练实时更新的. 具体的训练步骤除了第一、二步外, 其他步骤与 Word2vec 模型实验训练步骤一致. 这里模型训练的前二个步骤为:

1) 使用 tf.nn.embedding_lookup 函数查随机初始化的词汇表, 然后获取搜索词切词后不同词的向量表达, 将其结果作为 LSTM 的输入, 记为 *QueryEm_init*, 并将其值设置为可训练的;

2) 取 LSTM 模型最后一个时刻隐层输出为整个搜索词的向量表达, 即第二十个时刻隐层的输出为这个搜索词的向量表达, 记为 Q_{EM}.

具体流程如图 3 所示.

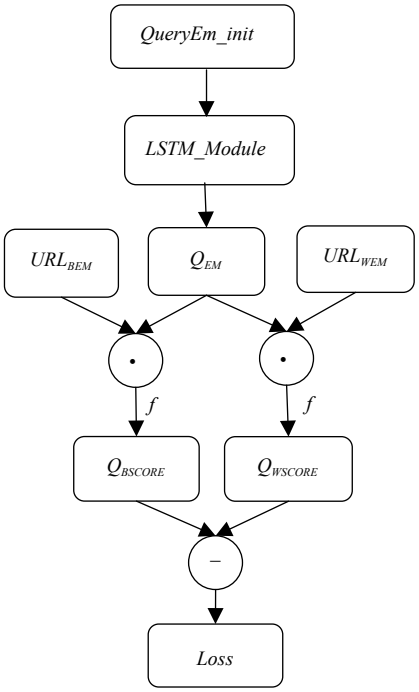

图 3 LSTM 模型实验的训练流程图

3 实验结果及分析

3.1 数据集、数据格式及评价指标

本文使用的日志数据集来源于搜狗公开的数据集 SogouQ, 数据包括了约一千万用户输入的大约九百万的搜索词以及大约一千五百万的 URL 展现结果. 词向量表是所有搜索词切词后的集合, 网址表是所有 URL 的集合; 通过使用计算点击率的方式计算出查询词下出现的 URL 的点击率, 然后将样本的格式整理成为: 搜索词、URL1、 URL2 的格式, 它们之间以 tab 键隔开, 其中 URL1 的点击率比 URL2 的点击率高. 通过随机打乱样本顺序, 取其中 4/5 作为训练集, 剩下的为测试集.

本文采用了 P(准确率) 为主要评测指标.

3.2 实验结果

图 4 和图 5 给出了使用两种不同模型实验在训练集、测试集上随着迭代次数的变化平均准确率的变化曲线.

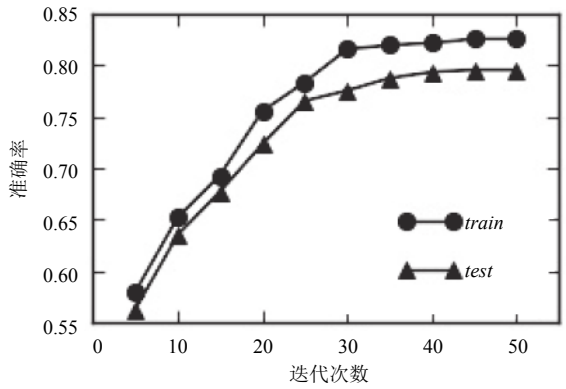

图 4 Word2vec 模型实验

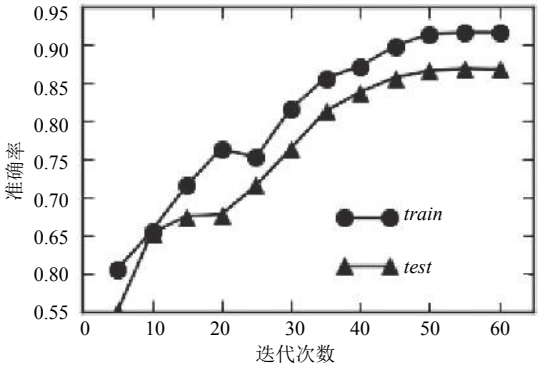

图 5 LSTM 模型实验

两组不同的实验在训练集、测试集上的平均准确率值如表 1 所示.

表 1 平均准确率表

模型	训练集准确率	测试集准确率
Word2vec	0.825	0.795
LSTM	0.915	0.863

由图 4、图 5 和表 1 可知, 使用 Word2vec 模型的实验比使用 LSTM 模型的实验在训练集、测试集上的平均准确率低, 但是模型收敛的时间比后者快. 分析原因主要有以下两点:

(1) 由于前者是先利用 Word2vec 模型训练得到了最终的词向量表, 所以在后续的训练过程中只需要针对网址表更新训练, 所以收敛时间会比后者快.

(2) 由于用户输入的查询词是具有先后的语义关系, 而 LSTM 模型本身的时序特性能够很好的结合查询词的这种语义关系, 而 Word2vec 考虑更多的是在维度空间上词与词之间的相似度, 所以通过 LSTM 模型获得的搜索词向量表达比 Word2vec 要更符合这本文中的应用场景, 导致平均准确率比使用 Word2vec 的实验要高.

4 结束语

本文分别使用 Word2vec 模型与 LSTM 模型获取用户搜索词的向量表达, 在通过与相应的网址的向量表达计算相似度作为对应网址的权威性. 实验结果表明两种方式的有效性, 其中 LSTM 模型试验的效果突出.

然而, 本方法中只考虑了搜索词对应的网址, 并未考虑具体网站标题和内容等信息; 由于网站的标题和内容是具体的文字, 所以理论上是可以获取很好的向量表达, 从而计算用户查询词和网站标题、内容之间的相似度; 如此就可以从多个方面更加全面的来评判网站的权威性. 所以评判用户查询词和网站标题等信息之间的相似度, 并将其与本文中提出方法相结合是本文下一步研究的重点.

参考文献

1 Page L, Brin S, Winograd T. The PageRank citation ranking: Bringing order to the web. Stanford: Stanford InfoLab, 1998. 1–14.

2 余凯, 贾磊, 陈雨强, 等. 深度学习的昨天、今天和明天. 计算机研究与发展, 2013, 50(9): 1799–1804. [doi: 10.7544/issn1000-1239.2013.20131180]

3 孙志军, 薛磊, 许阳明, 等. 深度学习研究综述. 计算机应用研究, 2012, 29(8): 2806–2810. [doi: 10.3969/j.issn.1001-3695.2012.08.002]

4 Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe, NV, USA. 2013. 3111–3119.

5 Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver, Canada. 2013. 6645–6649.

6 Bengio Y, Ducharme R, Vincent P, *et al*. A neural probabilistic language model. Journal of Machine Learning Research, 2003, 3: 1137–1155.

7 Mnih A, Hinton G. Three new graphical models for statistical language modelling. Proceedings of the 24th International Conference on Machine Learning. Corvalis, OR, USA. 2007. 641–648.

8 王勤勤, 张玉红, 李培培, 等. 基于 word2vec 的跨领域情感分类方法. 计算机应用研究, 2018, 35(10). [在线出版] http://www.arocmag.com/article/02-2018-10-004.html.

9 Xue B, Fu C, Zhan SB. A Study on sentiment computing and classification of Sina Weibo with Word2vec. Proceedings of 2014 IEEE International Congress on Big Data. Anchorage, AK, USA. 2014. 358–363.

10 Sharma K, Kumar AC, Bhandarkar SM. Action recognition in still images using word embeddings from natural language descriptions. Proceedings of 2017 IEEE Winter Applications of Computer Vision Workshops. Santa Rosa, CA, USA. 2017. 58–66.

11 Lilleberg J, Zhu Y, Zhang YQ. Support vector machines and word2vec for text classification with semantic features. Proceedings of the 14th International Conference on Cognitive Informatics & Cognitive Computing. Beijing, China. 2015. 136–140.

12 Venugopalan S, Rohrbach M, Donahue J, *et al*. Sequence to sequence-video to text. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 4534–4542.

13 Byeon W, Breuel TM, Raue F, *et al*. Scene labeling with LSTM recurrent neural networks. Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA. 2015. 3547–3555.

14 Vinyals O, Toshev A, Bengio S, *et al*. Show and tell: A neural image caption generator. Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, MA, USA. 2015. 3156–3164.

15 马胜蓝. 基于深度学习的文本检测算法在银行运维中应用. 计算机系统应用, 2017, 26(2): 184–188. [doi: 10.15888/j.cnki.csa.005628]

16 Zhao Z, Chen WH, Wu XM, *et al*. LSTM network: A deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems, 2017, 11(2): 68–75. [doi: 10.1049/iet-its.2016.0208]

17 Liu PF, Qiu XP, Chen XC, *et al*. Multi-timescale long short-term memory neural network for modelling sentences and doucuments. Proceedings of Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal. 2005. 2326–2335.

18 Wang X, Liu YC, Sun CJ, *et al*. Predicting polarities of tweets by composing word Embeddings with long short-term memory. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing, China. 2015. 1343–1353.

19 Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems. Montreal, QB, Canada. 2014. 3104–3112.

20 Ghosh S, Vinyals O, Strope B, *et al*. Contextual LSTM (CLSTM) models for large scale NLP tasks. arXiv: 1602.06291, 2016.

21 Wang D, Nyberg E. A long short-term memory model for answer sentence selection in question answering. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing, China. 2015. 707–712.