Computer Science and Application 计算机科学与应用, 2019, 9(3), 613-621
Published Online March 2019 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2019.93069

Hans汉斯

Improvement of the Recommended Click Prediction Model Based on Text Similarity

Bin Zhan, Xiaoling Wu*, Jie Ling

School of Computers, Guangdong University of Technology, Guangzhou Guangdong
Email: *xl.wu@giat.ac.cn, 807874967@qq.com

Received: Feb. 25th, 2019; accepted: Mar. 13th, 2019; published: Mar. 20th, 2019

Abstract

In order to further improve the accuracy of the recommended content click prediction in search engine, a method based on the similarity feature of search content is proposed. The structure of the method is composed of multiple decision tree models. The Hierarchical softmax is used to convert the result to binary classification results. In order to understand the semantics of the user's search text, the user input text similarity with the recommended content title, high-frequency related content, and recommended content tags is used to increase the accuracy of the click prediction model. The segmentation of words in the text is performed using the jieba segmentation, and word2cev is used to train all the words and construct a word vector model. Finally, Light GBM is used to build the prediction model. Then, 50,000 of the 2.05 million users' search records are taken as the verification set and the rest as the training set. Experimental results show that the accuracy of the model is improved after adding similarity features.

Keywords

Search Recommendation, Search Click Prediction, Term Vectors, Click Model, Keyword Semantics

基于文本相似度的搜索推荐点击预测模型

詹　彬，吴晓鸽*，凌　捷

广东工业大学计算机学院，广东 广州
Email: *xl.wu@giat.ac.cn, 807874967@qq.com

收稿日期：2019年2月25日；录用日期：2019年3月13日；发布日期：2019年3月20日

摘　要

为了进一步提高用户在搜索引擎中的推荐内容点击预测准确率，本文采用了一种包含搜索内容相似度特

*通讯作者。

文章引用: 詹彬, 吴晓鸽, 凌捷. 基于文本相似度的搜索推荐点击预测模型[J]. 计算机科学与应用, 2019, 9(3): 613-621.
DOI: 10.12677/csa.2019.93069

征的方法。该方法的结构是由多个决策树构成的预测模型，使用了层次化softmax (Hierarchical softmax)将结果转换为二分类结果。为了理解用户搜索文本的语义，使用用户输入与推荐内容标题、高频相关内容以及推荐内容标签的文本相似度来增加点击预测模型的准确率。采用jieba分词将文本中的词汇切分出来，使用word2cev对所有词汇进行训练，构建词向量模型。最后使用LightGBM进行预测模型的构建。从205万条用户搜索记录中取出5万条作为验证集，剩下的作为训练集。实验结果表明，添加相似度特征之后模型的点击预测准确率得到了提升。

关键词

搜索推荐，搜索点击预测，词向量，点击模型，关键字语义

1. 引言

随着计算机的发展，现在已经进入了信息时代。特别是互联网产业的飞速发展，使得网络资源呈现出庞大而杂乱的特点。面对大量的文本、音频、视频信息，如果要全部都接收到人的大脑里是不实际的。在这个信息大爆炸的时代如何才能在海量的数据中找到自己需要的信息，成为一个亟待解决的问题。

搜索引擎是日常生活中常用的一种解决信息过载的技术，最开始的搜索引擎是通过对信息进行人工或半自动化分类来提供搜索服务的，最早由加拿大麦吉尔大学于 1990 年开发[1]。第二代搜索引擎技术是基于文本匹配的搜索技术，也就是通过建立关键词库来进行查找和匹配。第三代搜索引擎结合人工智能技术，对用户需求理解更加准确，不仅能提供文本上相似的内容，还可以给出基于语义的内容。文献[2]提出了利用语义嵌入空间来表示词语的语义和逻辑，解决了问答系统中对用户需求的语义理解。

随着人工智能的发展，生活中涌现出越来越多人机交互和智能对话系统，搜索推荐系统面临着新的机遇和挑战。已有的基于 CTR (Click Through Rate)的模型[3]以及引入用户的偏好信息[4]等技术已经远远不能满足当前搜索推荐系统的需求。而基于 CTR 的搜索推荐给用户的关键词都是采用字符匹配技术来进行检索，往往缺乏语义上的理解，对用户的文本无法进行语义相关度的计算，甚至是对语义一无所知。加了用户偏好信息也因用户行为数据过于稀疏，无法对模型进行改进。自然语言处理作为计算机科学领域与人工智能领域的一个重要分支在信息检索、机器翻译、文档分类、文本挖掘等领域中被广泛应用。将自然语言处理应用于搜索推荐的点击预测中，可以进一步理解用户的需求，来预测用户是否会点击，从而提高预测结果的准确率。

2. 相关工作

2.1. 分词

词是“最小的能独立运用的语言单位”[5]。对于英文来说，每句话里的单词都是由空格隔开的，分词非常简单。由于中文具有大字符集连续书写的特点，如果不进行分词，计算机则无法得知中文词的确切边界，从而很难理解文本中所包含的语义信息。然而对于中文来说词与词之间一般都没有任何间隔，这就需要对一句话进行分词。

目前比较流行的分词方法有三种：第一种是基于字符串匹配的分词方法，这种方法是按一定策略将汉字串与机器字典里面的词条进行匹配，找到某个字符串则认为识别出一个词。第二种方法是基于理解

的方法，这种方法是通过模拟人对句子的理解，在分词的同时也分析句法、语义，用于解决歧义现象。这种方法包含分词子系统、句法语义子系统和总控部分。这种方法需要大量的语言语法知识。最后一种方法是基于统计的分词方法，这个方法是利用统计学的方法学习词语切分规律从而对句子进行分词。

中文一般较常使用 jieba 分词对句子进行分词。jieba 分词算法使用了基于前缀词典的词扫描，生成句子中汉字所有可能生成的词语情况所构成的有向无环图(DAG)，再采用动态规划查找最大概率路径，找出基于词频的最大切分组合。对于没有见过的词，采用基于汉字成词的 HMM 模型，或者使用 Viterbi [6] 算法。

2.2. 词向量

在计算机中通常有两种方式表示词汇，第一种是使用离散化表示(one-hot representation)，第二种是使用分布式表示(distribution representation)。离散化表示是将每一个词都表示为一个向量，这个向量的维度是词表的大小，向量中只有一个维度的值为 1。例如：西瓜：[0, 1, 0, 0, 0, ……]。但这样会导致词与词之间的关系不能表示，另外这个方法在词表很大的时候会导致维度非常的大，从而造成维度灾难，在使用中也难以计算。分布式表示是将词转换成一种分布式的表示形式，又称作词向量，是将每个词表示为一个向量。分布式表示与离散化表示不一样的地方是向量里的值可以是任意实数。例如：中国：[1.2, 3.5, 3, 5, ……]。其中的每一维都是有意义的，这样就解决了维度灾难。而且向量之间的距离可以表示词之间的相似度。目前用得最广泛的是分布式表示，其思想最早由 Hinton 于 1986 年提出[7]。本文用到的词向量也是基于分布式表示的词向量。

搜索引擎中的搜索文本包含一个及其以上的词语，标题一般也不止一个词语。所以对于多个单词的相似度而言要计算的不仅仅是一个词之间的相似度，而是两段文本之间的相似度。现在文本表示方法有词集和词袋法(Bag-of-word，BOW) [8]。本文用的是的是词袋法，此方法是将文本看成一些词的集合，在该集合中，每个词的出现是相互独立的。利用词袋法表示一组词可以计算出两个句子或者文档之间的相似度。

2.3. 神经网络语言模型(NNLM)

NNLM (Neural network language model)是在 2003 年由 Bengio 提出来的[9]，现在已被广泛应用语音识别系统[10]和上下文分析[11]等。NNLM 的原理是使用前 n 个词来预测最后一个词。神经网络模型使用 Distributed Representation 词向量来表示词语并作为输入。神经网络模型分为 4 层：输入层、投影层、隐藏层、输出层(如图 1 所示)。

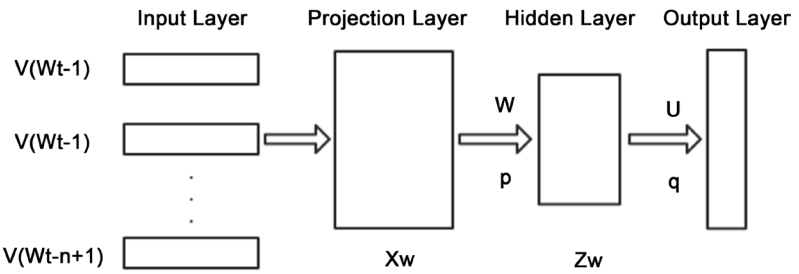

Figure 1. Neural network language model structure diagram
图 1. 神经网络语言模型结构示意图

假如 NNLM 输入是一组词序列 $w_t, w_{t-1}, \cdots, w_{t-n+2}, w_{t-n+1} \in V$。($V$ 是所有词汇集合) NNLM 的目标输出如公式(1)：

$$f\left(w_t, w_{t-1}, \cdots, w_{t-n+2}, w_{t-n+1}\right) = p\left(w_t \left| w_1^{t-1}\right.\right) \tag{1}$$

训练过程则是最大化以下函数(2)：

$$L = \frac{1}{T}\sum_t \log f\left(w_t, w_{t-1}, \cdots, w_{t-n+2}, w_{t-n+1}; \theta\right) + R\left(\theta\right) \tag{2}$$

其中 θ 为模型的所有参数，$R(\theta)$为正则项。

2.4. word2vec

word2vec 是谷歌公司在 2013 年开放的一款词向量训练模型，可以根据给定的语料库，通过优化后的模型将单词训练成向量的形式。再使用 word2vec 计算出关键字的语义相似度[12]。word2vec 依赖 skip-grams 或者 CBOW 来建立词嵌入。在 word2vec 中，使用是层次化 softmax (Hierarchical softmax)进行归一化[13]，改善了传统 softamx 的运算效率。

CBOW 与 Skip-Grams

在 word2vec 中，CBOW 是通过给定上下文来预测 input word。而 skip-gram 则是与 CBOW 相反，通过 input word 预测上下文。两个模型过程如图 2 所示，w(t)为句子中的第 t 个词语。

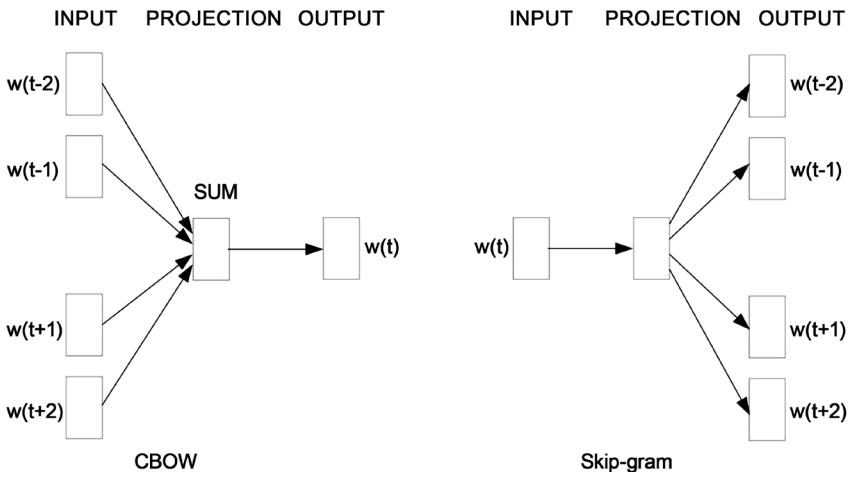

Figure 2. Schematic diagram of CBOW and Skip-gram
图 2. CBOW 和 Skip-gram 示意图

3. 本文工作

3.1. 分词纠错

这里用 N-gram 来对分词进行纠错，将切分过小的词进行合并。例如：“清华大学”可能会被分成“清华”和“大学”，因此要对这些分词进行合并。这里需要将句子分词分成 3 个词的多个子串。子串如图 3 所示：

原句：广东工业大学是以工为主的多学科协调发展的大学。

分词后：{广东，工业，大学，是，以，工，为主，的，多学科，协调，发展，的，大学}

N-gram 子串：{[广东，工业，大学]，[工业，大学，是]，[大学，是，以]，[是，以，工]，[以，工，为主]，[为主，的，多学科]，[的，多学科，协调]，[多学科，协调，发展]，[协调，发展，的]，[发展，的，大学]}

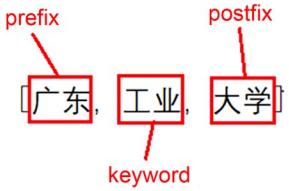

Figure 3. Substring of n-gram
图 3. n-gram 子串

将每一句话都进行这样的切分，然后将每个子串中的中间的词作为 keyword 去遍历整个文本集合的 keyword，然后在 keyword 相同的子集里面寻找 prefix 和 postfix，计算出现频率高的 prefix 或 postfix，将其与 keyword 合并。比如广东工业大学应该是一个词，而分词的时候将其分得过细了，而它们在整个语料库中会同时出现，其同时出现频率会比较高，因此可以将其合并。

3.2. 通过点击率和词相似度特征构建点击预测模型

基于对用户搜索记录的分析来预测用户对于搜索引擎的提示的点击行为。搜索记录包括 prefix (用户在输入框的输入)、query_prediction (包含用户输入信息且出现频度最高的前几个词条)、title (搜索引擎推荐或提示词条)、tag (提示词条的标签)、label (用户是否点击提示词条(0 或 1))。

本文用到的是点击率和词条相似度的特征，如表 1。

Table 1. Feature
表 1. 用到的特征

特征	描述	特征	描述
Prefix_ctr	用户所输入内容点击率	PQ_min_sim	用户输入与高频相关词条的最小相似度
Title_ctr	搜索提示内容点击率	PQ_mean_sim	用户输入与高频相关词条的平均相似度
Tag_ctr	标签点击率	QTi_max_sim	提示词条与高频相关词条的最高相似度
PTi_sim	用户输入与提示词条相似度	QTi_min_sim	提示词条与高频相关词条的最小相似度
PTa_sim	用户输入与提示词条标签相似度	Qti_mean_sim	提示词条与高频相关词条的平均相似度
PQ_max_sim	用户输入与高频相关词条的最高相似度	QTa_max_sim	提示词条标签与高频相关词条的最高相似度
QTa_min_sim	提示词条标签与高频相关词条的最小相似度	QTa_mean_sim	提示词条标签与高频相关词条的平均相似度

上面的相似性特征都是可以作为判断用户是否会采纳搜索提示的特征。比如用户输入与提示词条之间的相似度，相似度越高而用户的点击概率则会越高。这里增加了 11 个相似性特征，在预测的时候只需通过训练好的词向量模型多计算 11 个相似度即可。图 4 为点击率模型使用特征，图 5 为添加相似度之后的特征。

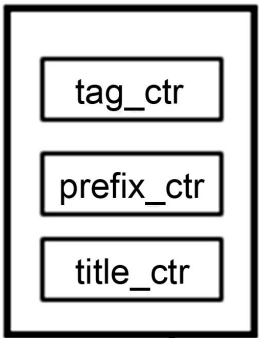

Figure 4. Feature of click rate
图 4. 点击率特征

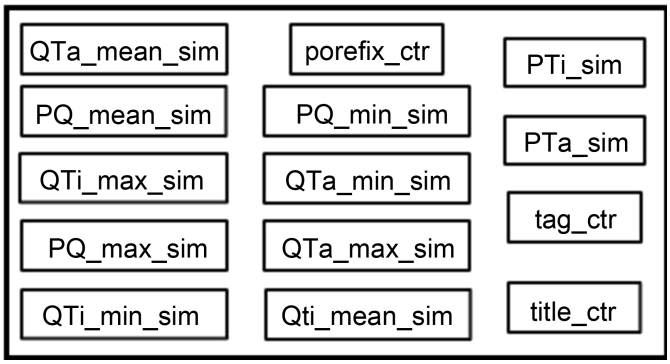

Figure 5. Features with added similarity
图 5. 添加相似度之后特征

基于这些特征，可以使用 Boosting 来提高预测算法的准确度。Boosting 是一种提高任意给定学习算法准确度的方法[14]，图 6 即为 Boosting 模型结构。它的思想起源于 Valiant 提出的 PAC (Probably Approximately Correct)学习模型[15]。Valiant 和 Kearns 提出了弱学习和强学习的概念，识别错误率小于 1/2，也即准确率仅比随机猜测略高的学习算法称为弱学习算法；识别准确率很高并能在多项式时间内完成的学习算法称为强学习算法。

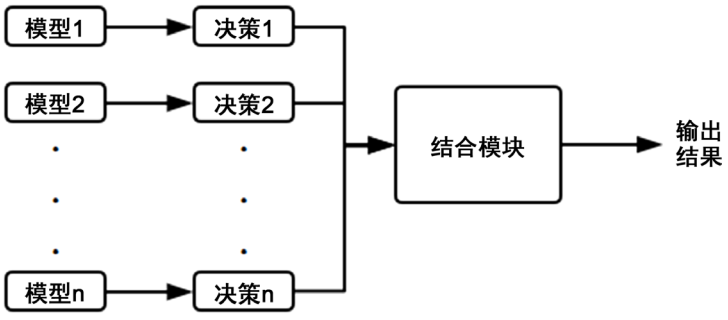

Figure 6. A model for the Boosting
图 6. Boosting 模型示意图

这里使用 GBDT (Gradient Boosting Decision Tree)来构建模型。GBDT 是通过多棵决策树共同影响结果的模型，GBDT 学习的过程是每一步都将用到前面的所有树的结论的残差，这个残差就是真实值与预测值之间的差。

3.3. 实验结果分析与比较

实验使用微软的 Lihght GBM [16]训练模型。Light GBM 采用了 Leaf-wise (Best-first)的决策树生长策略，可以比 level-wise 算法[17]减少更多损失。实验硬件平台为 Intel 酷睿 i5-4200U，主频 1.6 GHz，内存 8G。本实验数据集是使用天池 OGeek 大数据竞赛提供的数据集，数据样例和各字段的说明如表 2、表 3 所示。实验数据集是用户的搜索记录，训练集包含 200 万条搜索记录，验证集包含 5 万条搜索记录。将训练集和验证集合并，使用 StratifiedKFold 将数据集按五次不同的切分规则切分训练集和验证集，最后计算均值来做比较。

Table 2. Sample data
表 2. 样例数据

挂号{“挂号信是什么”：“0.023”，“挂号网上预约”：“0.029”，“挂号网官网”：“0.015”，“挂号信”：“0.082”，“挂号”：“0.066”，“挂号信单号查询”：“0.075”，“挂号平台”：“0.025”，“挂号网”：“0.225”，“挂号信查询”：“0.201”，“挂号信查询中国邮政”：“0.020”，“挂号预约”：“0.021”}预约挂号网应用 1
挂号{“挂号信是什么”：“0.023”，“挂号网上预约”：“0.029”，“挂号网官网”：“0.015”，“挂号信”：“0.082”，“挂号”：“0.066”，“挂号信单号查询”：“0.075”，“挂号平台”：“0.025”，“挂号网”：“0.225”，“挂号信查询”：“0.201”，“挂号信查询中国邮政”：“0.020”，“挂号预约”：“0.021”}挂号网网站 0

Table 3. Data field description
表 3. 数据字段说明

字段	说明	数据示例
prefix	用户输入(query 前缀)	刘德
query_prediction	根据当前前缀，预测的用户完整需求查询词，最多 10 条；预测的查询词可能是前缀本身，数字为统计概率	{“刘德华”:“0.5”,“刘德华的歌”:“0.3”,…}
title	文章标题	刘德华
tag	文章内容标签	百科
label	是否点击	0 或 1

这里分别使用加了相似性特征和未加相似性特征的模型对比，使用了准确率、召回率和 F1-Score。计算方法如公式(3) (4) (5)。

TP (True Positive)真阳性：预测为正，实际也为正

FP (False Positive)假阳性：预测为正，实际为负

FN (False Negative)假阴性：预测与负、实际为正

TN (True Negative)真阴性：预测为负、实际也为负

$$\text{准确率} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{召回率} = \frac{TP}{TP + FN} \tag{4}$$

$$F1 - Score = 2 * \frac{\text{准确率} * \text{召回率}}{\text{准确率} + \text{召回率}} \tag{5}$$

如表 4 和表 5 所示，这里使用 5 组数据的准确率、召回率、F1_SCORE 的平均值来对比各个方法。

通过对比表 4 和表 5 的结果可以看到在这三方面确实是得到了进一步提升。语义相关度可以将一些比较个性化的词语联系到常用的搜索词语，以便于更好的对用户行为进行预测。

Table 4. Results of using only CTR (Click Through Rate)
表 4. 仅仅使用点击率(CTR)的结果

	准确率	召回率	f1_score
1	0.7807175	0.6398994887	0.6847090371
2	0.7819975	0.6484725105	0.6888312392
3	0.78199	0.644300217	0.6874386196
4	0.7829825	0.648163451	0.6896990538
5	0.7813425	0.64750502	0.687867357
MEAN	0.781806	0.645668	0.687709

Table 5. Result after adding semantic similarity
表 5. 增加语义相似度之后结果

	准确率	召回率	f1_score
1	0.84155	0.7516175196	0.7792560601
2	0.8402575	0.7512815861	0.7777781642
3	0.839735	0.750206599	0.7769652014
4	0.8406225	0.7536532763	0.7787168904
5	0.8406225	0.7550104475	0.7790267624
MEAN	0.840558	0.752354	0.778349

从图 7 可以看到，增加了文本相似性特征之后准确率提升了，通过上下文信息训练出词向量从而可以计算文本之间的相似度。如果有足够的数据集。训练出来的词向量会更加准确，从而对准确率的提升会更加明显。

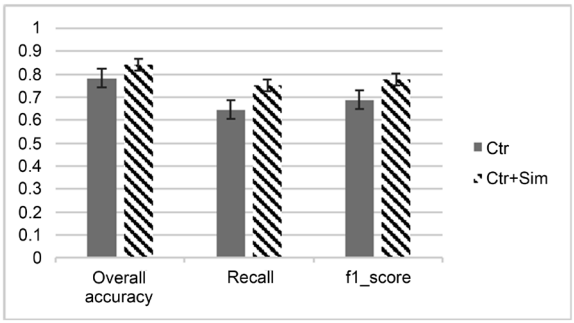

Figure 7. Only click-through rate and similarity were used for the model comparison
图 7. 只使用点击率和加入相似度的模型对比

4. 结语

基于点击率的点击预测模型在实际问题中被广泛应用，在大多数情况下的效果较好。但是在点击率比较接近 50%的时候，效果较差。为增加基于点击率的模型的准确率，本文提出在点击率的基础上增加

搜索文本与高频搜索标题、推荐搜索结果之间的相似度来对用户进行点击预测。文本相似度在模型中结合点击率提高了用户行为预测的准确率，进一步对用户意图进行了预测，让模型更准确地预测用户的点击行为。实验结果表明，添加了相似性的特征之后，预测的准确率得到了提升。通过文本语义可以发现文本之间更多的联系，让机器更加“人性化”。

基金项目

本文得到广东省科技计划项目(2017B090906003)、广州市科技计划项目(201802010043、201807010058)和机器智能与先进计算教育部重点实验室开放课题基金(MSC-201604A)的资助。

参考文献

[1] 李晓明, 闫宏飞, 王继民. 搜索引擎: 原理、技术与系统[J]. 2012.

[2] Yang, M.C., Lee, D.G., Park, S.Y., *et al*. (2015) Knowledge-Based Question Answering Using the Semantic Embedding Space. *Expert Systems with Applications*, **42**, 9086-9104. https://doi.org/10.1016/j.eswa.2015.07.009

[3] Joachims, T. (2002) Optimizing Search Engines Using Clickthrough Data. *ACM Conference on Knowledge Discovery & Data Mining*, Edmonton, 23-26 July 2002, 1-21.

[4] Xing, Q., Liu, Y., Nie, J.Y., *et al*. (2013) Incorporating User Preferences into Click Models.

[5] 汉语信息处理词汇 01 部分: 基本术语(GB12200.1-90)6 [S]. 北京: 中国标准出版社, 1991.

[6] Forney, G.D. (1993) The Viterbi Algorithm. *Proceedings of the IEEE*, **61**, 268-278.

[7] Hinton, G.E. (1989) Learning Distributed Representations of Concepts. 8*th Conference of the Cognitive Science Society*, Ann Arbor, 1989, 1-11.

[8] Manning, C.D. (1999) Foundations of Statistical Natural Language Processing. MIT Press, Cambridge.

[9] Bengio, Y., Schwenk, H., Senécal, J., *et al*. (2003) Neural Probabilistic Language Models. *Journal of Machine Learning Research*, **3**, 1137-1155.

[10] Lee, K., Park, C., Kim, N., *et al*. (2018) Accelerating Recurrent Neural Network Language Model Based Online Speech Recognition System.

[11] Deng, H., Lei, Z. and Wang, L. (2017) Global Context-Dependent Recurrent Neural Network Language Model with Sparse Feature Learning. *Neural Computing & Applications*, No. 6, 1-13.

[12] Shao, T., Chen, H. and Chen, W. (2018) Query Auto-Completion Based on Word2vec Semantic Similarity. *Journal of Physics Conference Series*, **1004**, Article ID: 012018. https://doi.org/10.1088/1742-6596/1004/1/012018

[13] 周练. Word2vec 的工作原理及应用探究[J]. 图书情报导刊, 2015(2): 145-148.

[14] Kearns, M.J. and Valiant, L.G. (1993) Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. Springer-Verlag, Berlin. https://doi.org/10.1007/3-540-56483-7_21

[15] Valiant, L. (2015) Probably Approximately Correct: Nature’s Algorithms for Learning and Prospering in a Complex World. *Common Knowledge*, **21**, 340. https://doi.org/10.1215/0961754X-2872666

[16] Guolin, K., Qing, M. and Thomas, F. (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. 31*st Conference on Neural Information Processing Systems*, Long Beach, 2017, 1-11.

[17] Shi, H. (2007) Best-First Decision Tree Learning. The University of Waikato, Hillcrest.

知网检索的两种方式：

1. 打开知网页面 http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD
   下拉列表框选择：[ISSN]，输入期刊 ISSN：2161-8801，即可查询
2. 打开知网首页 http://cnki.net/
   左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：http://www.hanspub.org/Submission.aspx
期刊邮箱：csa@hanspub.org