WDS:基于词向量的文本相似函数

王路琪[1] 龙 军[1] 袁鑫攀[2]

(中南大学软件学院 长沙 410075)[1] (湖南工业大学计算机与通信学院 湖南 株洲 412000)[2]

摘 要 为进一步提高文本相似度计算的准确性,在系统相似函数的架构下,提出了基于词向量的文本相似函数WDS(Word Documents Similarity)及其优化算法 FWDS(Fast Word Documents Similarity)。该函数将文本词语集合对应的词向量集合看作系统,将词语对应的词向量看作系统的元素,则两个文本相似度就是两个向量集合的相似度。在具体计算时,以第一个向量集合为标准进行两个向量集合的对齐操作,同时计算相似元与非相似元的多个参数。实验结果表明,随着文本长度的增加,与 WMD 和 WJ 算法相比,WDS 表现出了较高的命中率。

关键词 文本相似,词向量,系统相似函数,相似元,权值

中图法分类号 TP301.6 **文献标识码** A

WDS:Word Documents Similarity Based on Word Embedding

WANG Lu-qi[1] LONG Jun[1] YUAN Xin-pan[2]

(School of Software,Central South University,Changsha 410075,China)[1]

(School of Computer and Communication,Hunan University of Technology,Zhuzhou,Hunan 412000,China)[2]

Abstract In order to further improve the accuracy of document similarity,under the framework of system similarity function,this paper presented Word Documents Similarity (WDS) based on word embedding,and its optimization algorithm FWDS (Fast Word Documents Similarity). WDS regards the set of word embedding corresponding to the words set of documents as the system,and regards the word embedding corresponding to the word as the element of the system. So,the similarity of the documents is the similarity of the two word embedding sets. In the concrete calculation,the first vector set is used as the standard,the alignment operation of the two vector sets is carried out,and the multiple parameters of the sets that are in and not in MOPs are calculated. The experimental results show that compared with WMD and WJ,WDS always keep better hit rate with documents' length increase.

Keywords Document similarity,Word embedding,System similarity function,MOP,Weight

1 引言

在自然语言处理中,文本相似度计算是一项基础且应用十分广泛的工作。例如:在推荐系统中,将图书内容用于图书推荐[1];在问答系统中,查找潜在的答案[2];在文本分类中,判断文档类别[3];在机器翻译系统中,匹配最优的翻译结果[4]等。文本相似度可以定义为两个文本的语义相似程度,通常是被映射到[0,1]之间的一个实数。

随着自然语言处理的发展,文本相似度计算的相关方法不断演进。近年来,利用词向量[5-6]与其他相似性度量函数相结合的方法得到了广泛的关注。

徐昇提出了基于文献[7]的相似度计算函数和词向量的方法来计算文本相似度,并称之为 WJ 算法。WJ 算法虽然考虑了词语的权值对文本相似度的影响,但是由于自身缺陷,只能计算同义词替换等经过简单变换的文本的相似度。

Kusner 等[8]提出基于 EMD 和词向量的 WMD (Word Mover's Distance)算法来计算文本距离,并通过分类实验证明了该方法的有效性。由于 WMD 算法缺少惩罚项及词语的权值,只能计算短文本(10 个元素内)的距离[9]。

在关毅等[10]提出的系统相似函数的框架下,本文提出基于词向量的文本相似函数(WDS),从系统论的角度分析文本相似度。

本文方法首先经过分词把文本转换为词语集合,通过词向量把词语集合转换得到的向量集合看作系统,把词语对应的词向量看作系统元素,则两个文本转换为两个词向量集合;然后通过构建相似元并计算相似元词语的权值、个数、相似度、非相似元词语的个数和权值等多个参数,计算文本相似度。

2 基于词向量的文本相似函数

WDS 的定义为:给定词语集合 $A=\{a_1,a_2,a_3,\cdots,a_m\}$,$m=|A|$ 与词语集合 $B=\{b_1,b_2,b_3,\cdots,b_n\}$,$n=|B|$,其中,$a_i$ $(1\leqslant i\leqslant m)$和 $b_j(1\leqslant j\leqslant n)$分别是词语集合 A 和词语集合 B 经过分词及去掉停用词得到的词语,令 $x_i>0$ 表示词语 $a_i(1\leqslant i\leqslant m)$的权值,$y_j>0$ 表示词语 $b_j(1\leqslant j\leqslant n)$的权值,令 $\mu=Similarity(a,b)$表示词语 a 和词语 b 的相似度,并且约定 $0\leqslant\mu\leqslant1$,当且仅当$a\equiv b$ 时 $\mu=1$。

相似元(MOP)由成对的词语组成,构造方法是:对于任意 $a_i\in A$,令 $b_j=\arg\max_{b_j\in B}(Similarity(a_i,b_j))$,若 $Similarity$

本文受国家自然科学基金资助项目(61402165,S1651002),湖南省重点研发计划(2016JC2018)资助。

王路琪(1990-),男,硕士生,主要研究方向为自然语言处理,E-mail:wangluqinet@163.com;**龙 军**(1972-),男,博士,教授,博士生导师,主要研究方向为网构化软件;**袁鑫攀**(1982-),男,博士,讲师,主要研究方向为信息检索、数据挖掘,E-mail:xpyuanfly@163.com(通信作者)。

(a_i,b_j)大于某个阈值 μ_0，则$\langle a_i,b_j\rangle$构成一个相似元，记为 s_i，其相似度为 $\mu_i(1\leqslant i\leqslant p)$。假定词语集合 A 和词语集合 B 之间的相似元有 $p(p\leqslant\min\{m,n\})$个，记为 $s_1,s_2,\cdots,s_p\in A\times B$，设其分别为$\langle a_1,b_1\rangle,\langle a_2,b_2\rangle,\cdots,\langle a_p,b_p\rangle$，则其相似度分别为 $\mu_1,\mu_2,\cdots,\mu_p$。

在计算 $WDS(A,B)$时，以词语集合 A 对应的词向量集合 A'为标准。WDS 可以认为是 $N(N\geqslant m+n)$维向量 $X=(x_1,x_2,\cdots,x_m,0,\cdots,0)_N$，$Y=(x_1\mu_1,x_2\mu_2,\cdots,x_p\mu_p,0,\cdots,0,y_{p+1},y_{p+2},\cdots,y_n)_N$ 的夹角 α 的余弦值 $\cos\alpha$。向量 X 前 m 项表示词语集合 A 前 m 项词语的权值，向量 Y 前 p 项表示词语集合 B 与词语集合 A 对应位置的相似权值，后$(p+1)$项表示非相似元的权值。$WDS(A,B)$可以表示为：

$$WDS(A,B)=\frac{\sum_{i=1}^{p}\mu_i x_i^2}{\sqrt{\sum_{i=1}^{m}x_i^2}\sqrt{\sum_{i=1}^{p}\mu_i^2x_i^2+\sum_{i=p+1}^{n}y_i^2}} \tag{1}$$

能够证明式(1)满足以下 1)，2)，3) 3 个条件，在阈值的条件下满足以下 4)，5)两个条件。

1)$WDS(A,B)$是关于相似元词语权值的单调增函数；

2)$WDS(A,B)$是关于非相似元词语权值的单调减函数；

3)$WDS(A,B)$是关于相似元相似度的单调增函数；

4)$WDS(A,B)$是关于相似元个数的单调增函数；

5)$WDS(A,B)$是关于非相似元个数的单调减函数。

Similarity 函数：构造相似元的 *Similarity* 函数是计算词语之间的相似度，本文基于词向量计算词语之间的相似度，故 *Similarity* 函数的计算方式如式(2)所示：

$$\mu=Similarity(a_i,b_j)=\cos\theta=\frac{a\cdot b}{\|a\|\times\|b\|} \tag{2}$$

其中，a,b 分别表示词语 a_i 和词语 b_j 的词向量。

权值：相似元和非相似元词语的权值计算方式如式(3)所示：

$$w_i=\frac{c_i}{\sum_{i=1}^{n}c_i} \tag{3}$$

其中，c_i 表示词语 i 在文本出现的次数。

在文本中，不同的词语对文本的重要程度并不完全一致，根据词语的重要程度给词语赋予权值，进而计算文本相似度是十分必要的，本文利用词频计算词语的权值。

相似元的阈值 μ_0：根据词向量的精度进行词语相似度阈值的判定，文献[11]给出了以下结论：对于汉语，[0,0.25)为不相似，[0.25,0.4)为相似，[0.4,0.5)为非常相似，[0.5,1]为基本等同。WDS 中在阈值的情况下满足上述 4)和 5)两个条件。综合以上结论，用词向量计算词语的相似度能够满足 WDS 对阈值的基本要求。

WDS 取值范围：式(1)可以认为是向量 X 和向量 Y 夹角 α 的余弦值 $\cos\alpha$，结合柯西不等式易知，WDS 的取值区间为[0,1]。当词语集合 A 和词语集合 B 没有任何相似度大于阈值 $\mu_0(0<\mu_0\leqslant0.5)$的相似元时，它们的相似度值为 0；当词语集合 A 和词语集合 B 完全等同，它们的相似度达到最大值 1。

WDS 的特性如下：

1)不满足交换律。易验证 WDS 并不满足交换律，即存在词语集合 A 和词语集合 B 使得 $WDS(A,B)\neq WDS(B,A)$，也不满足三角不等式，即存在词语集合 A、词语集合 B 和词语集合 C 使得 $WDS(A,B)+WDS(B,C)\leqslant WDS(A,C)$。关于不满足三角不等式的系统相似性，关毅等[10]已经进行了详细的阐述。

2)单调性。WDS 是关于相似元词语权值、个数、相似度的单调递增函数，也是关于非相似元词语权值、个数的单调递减函数。WDS 这一特性确保文本相似度的最终结果随着相似元权值、个数等参数的线性变化而变化，保证了 WDS 的可行性和准确性。

为降低 WDS 的时间复杂度，本文提出了 FWDS(Fast Word Documents Similarity)。

词向量模型训练完成后，对于任意词语的词向量在模型中的位置即确定，与相似度大于阈值的词语同样是确定的。基于此，本文提出的构建潜在相似元的方法是可行的。

构建潜在相似元的过程如图 1 所示。以词语“关键”为例，给定阈值 μ_0 和已训练完成的词向量模型的条件下，在 P_1 阶段选出与“关键”相似的若干相似度大于 μ_0 的词语；在 P_3 阶段，对选出的词语进行字典排序。

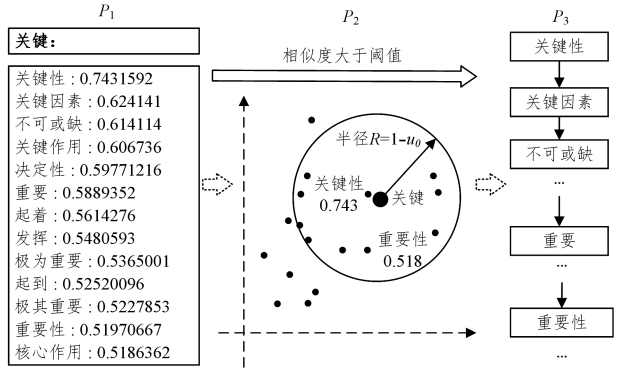

图 1 构建潜在相似元的过程

如图 1 所示，P_1 阶段选出相似度大于 μ_0 的词语全部在 P_2 阶段以“关键”所在的位置为圆心，以 $R=1-\mu_0$ 为半径的圆内，若有其他词语和“关键”构成相似元，则该词语存在于圆内，也存在于 P_3 阶段排序后的数组中。

潜在相似元存储：在词向量[5-6]训练过程中，层次 Softmax 策略使用的 Huffman 树存储词语及其词向量。本文借助同样的思想，把训练词向量词语的潜在相似元用 Huffman 树存储，即 Huffman 树的叶子节点由词语与词向量替换为词语与潜在相似元。

查找潜在相似元：在原词向量模型中输入词语获得词向量，借助同样的方法，在已存储潜在词语相似元的 Huffman 树中，输入词语即可获得词语潜在相似元。

相似元匹配：在训练词向量模型后，需构建所有词语的潜在相似元，在计算词语集合 A 和词语集合 B 的相似度时，词语集合 A 中词语的所有潜在相似元可以构成一个矩阵。遍历词语集合 B 的词语 b_j，使用二分查找的方法在该矩阵中查找 b_j 是否存在，之后再计算 A 和 B 之间的相似元个数等参数。

WDS 的时间复杂度是 $O(mnp)$，p 表示词向量的维度；FWDS 利用在字典有序的潜在相似元中匹配相似元的方式代替了词向量的距离计算，时间复杂度为 $O(mn\log d)$，d 表示词语潜在相似元的个数。

3 实验结果及分析

3.1 命中率实验

3.1.1 词向量模型

本文采用 Word2Vec 的 Skip-gram 模型及 Hierachy Soft-

max 策略；词向量的维度设定为 200 维，其他参数采用默认值；采用开源的 Java 版本的工具包[1]训练数据集；采用维基百科中文语料[2]作为训练词向量的数据集。

3.1.2　命中率实验的相关算法

1）WMD（Word Mover's Distance）

WMD 算法是将词向量和 EMD（Earth Mover's Distance）相结合的方法，它是计算一个文本转移到另一个文本最短距离的最优化解，计算时需要构建一个稀疏矩阵 T，如式（4）所示：

$$\min_{T\geqslant 0}\sum_{i,j=1}^{n}T_{ij}c(i,j)$$

$$\text{s.t.}\begin{cases}\sum_{j=1}^{n}T_{ij}=d_i, & \forall i\in\{1,\cdots,n\}\\ \sum_{i=1}^{n}T_{ij}=d_j', & \forall j\in\{1,\cdots,n\}\end{cases}\tag{4}$$

其中，$c(i,j)$是词语 i 和词语 j 的距离，$T_{i,j}$为相对应的非负权值矩阵，d_i，d_j'分别表示词语的词频。词语 i 和词语 j 距离的计算公式为：$c(i,j)=\|a-b\|_2$，其中 a，b 分别表示词语 i 和词语 j 的词向量，用工具包[3]实现 WMD 算法。

2）WDS（Word Document Similarity）

本文提出的基于词向量的文本相似函数[4]。

3）WJ

WJ 基于式（5）和词向量计算文本相似度。

$$Sim(A,B)=\frac{\sum_{i=1}^{m}w_i Max(a_i,b_j)+\sum_{j=1}^{n}w_j Max(b_j,a_i)}{\sum_{i=1}^{m}w_i+\sum_{j=1}^{n}w_j}\tag{5}$$

其中，m 和 n 分别表示词语集合 A 和词语集合 B 的个数，w_i 表示词语 i 的权值。WJ 算法把名词、动词等实词的权值设为 1，将其他词性的权值设为 0.8。$Max(a_i,b_j)$表示词语集合 A 中一个词语 a_i 和词语集合 B 中任意一个词语 b_j 的相似度中的最大值，a_i 和 b_j 的相似度基于词向量按照式（2）计算，用工具包[5]实现 WJ 算法。

3.1.3　实验方法及测试数据集

本文的实验方法与文献[12]保持一致。实验方法如下：在新闻类的文本中一般包含标题和正文两个部分，标题是把最重要的文字呈现给读者，正文中一般有一个或多个句子的内容与标题相似。因此，实验把新闻类文本的标题作为目标句，以人工标记的句子作为标准与算法返回的结果作对比，最终比较相关算法的命中率。

测试数据集：从人民日报、光明日报等新闻类文本中选取测试数据并分为 5 组，每组 50 篇新闻文本，包含人民日报 30 篇、光明日报 20 篇。

3.1.4　评价指标

在测试数据集中，首先文本按照标点符号“。!?。!?”划分得到一个句子集合 S，遍历句子 s_i，计算 s_i 和标题的相似度或距离，把得出的最优结果和标准文本作对比。

WMD 算法是计算文本的距离，距离越小则相似度越高。实验把标题作为目标，WMD 算法返回与目标距离最小的文本，WJ 和 WDS 返回相似度最高的文本，将其与标准文本作比较。最后统计每组测试数据命中标准文本的个数与总数之比的平均值，作为比较 WJ，WDS，WMD 命中率的依据。

3.1.5　命中率实验结果及分析

测试数据集的实验结果如图 2 所示，随着标题长度的增加，WJ，WDS，WMD 算法的命中率都在降低，但是相对于 WJ 算法和 WMD 算法，WDS 算法一直保持着较高的命中率。

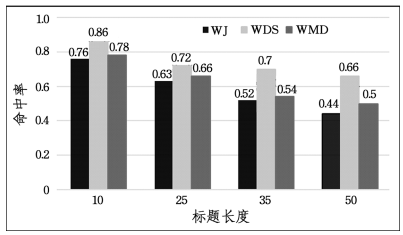

图 2　新闻类文本上 WJ，WDS，WMD 的命中率

互为复述的两个句子中的大部分词语都是相同的，只有部分词语被同义词所替代，WJ 算法能够计算互为复述等简单变化的句子相似度。如图 2 所示，随着标题长度及文本语义复杂度的增加，WJ 算法的命中率快速下降。

随着标题长度的增加，文本经过分词和去掉停用词后出现了更多单字的个数及其对应权值的可能性也在增加。例如以下文本 T：

书记在文艺工作座谈会上的讲话，令我深受震动，其中“浮躁”二字，直击当下文艺创作的命门。

经过分词及去掉停用词，其中单字是“中/二/字/直/击”。通常，汉语表达多以词语为基本单元，单字并没有过多语义信息，训练语料得到的词向量模型中单字词向量同样没有隐含过多的语义信息。

WMD 算法[8]缺少惩罚项使得单字造成的“单字距离”降低了算法的命中率。在 WDS 中单字很难构成相似元，根据 WDS 定义中的 2）和 5）两个条件，单字使文本之间的相似度单调递减。WDS 把非相似元的个数及权重作为惩罚项，降低了单字对文本整体相似度的影响，提高了算法的命中率。

WMD 算法满足交换律，即对于任意词语集合 A 和词语集合 B，满足 $WMD(A,B)=WMD(B,A)$。若标题与句子 s_i 长度之差增加，WMD 计算一个短文本“转移”到一个长文本的“不平衡转移距离”增加，导致 WMD 算法的命中率快速下降。WDS 不满足交换律，其以“左”为准的特点计算文本的相似度，在两个文本长度差比较大的情况下准确率较高。

3.2　WDS 与 FWDS 的对比实验

WDS 与 FWDS 对比实验的数据集和实验方法与 3.1 节保持一致。经过实验，FWDS 与 WDS 命中率保持一致的情况下，两者每组数据的耗时对比如图 3 所示。从图 3 可知，与 WDS 对比，在 4 组数据上 FWDS 计算文本相似度的运算时间均有降低，平均节省 17.5％的运算时间。

FWDS 在构建潜在相似元及查找相似元的过程中，认为单字即是非相似元，使用空间换取时间的方法提高了计算文本相似度的速度。

1）https://github.com/NLPchina/Word2VEC_java

2）https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2

3）https://github.com/crtomirmajer/wmd4j

4）http://download.csdn.net/detail/u011001835/9849524

5）https://github.com/jsksxs360/Word2Vec

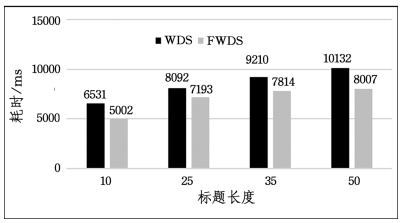

图 3 WDS 与 FWDS 的耗时对比

3.3 分析与总结

WDS 的创新在于认为相似度较低的词语对文本整体相似度有负面影响，并具有不满足交换律的特性；FWDS 通过先构建潜在相似元、后查找相似元的方式降低了 WDS 的时间复杂度。

WJ 算法简单地把词语权值和其相似度最高词语的相似度相乘作为基本计算单元，存在一定缺陷，可用于计算互为复述等经过简单变换的文本相似度；WMD 算法在计算文本之间距离时容易受到单字距离和不平衡转移距离的影响，可用于计算短文本距离；WDS 把相似度较低且不能构成相似元的词语作为惩罚项，同时具有不满足交换律的特性，可用于计算“长-长”文本或“短-长”文本的文本相似度。经过理论分析及实验，证明了 FWDS 对 WDS 优化的有效性，并且可以提高运算速度。

结束语 文本相似度计算一直是自然语言处理中多个领域的基础和核心。本文首先介绍了通过若干词向量与相似函数相结合的方式计算文本相似度或距离的算法，并提出了基于词向量的文本相似函数(WDS)及其优化算法 FWDS，讨论分析了 WDS 的相关特性。随后，实验对比基于词向量的 WJ，WDS，WMD 3 种算法在新闻类文本中的命中率，对比 FWDS 与 WDS 在运算时间的优化效果。最后，通过分析实验结果证明了本文方法的有效性及可行性。

在本文的基础上，未来的工作有：把系统数学符号化表示为集合，集合不仅可以包含元素还可以包含集合，因此 WDS 可以递归计算，可以测试 WDS 计算更大文本相似度的效果；FWDS 中构建潜在相似元后，可以使用多线程查找相似元，进一步提高运算速度；WDS 的难点在于：在相似元构建的过程中，即在向量的相似度匹配过程中，除了本文提出的 FWDS 方法外，还可以尝试通过 FAISS[13] 对向量建立索引或使用基于全文搜索引擎的向量相似性搜索[14] 等其他方法解决。

参 考 文 献

[1] GOPALAN P, CHARLIN L, BLEI D M. Content-based recommendations with Poisson factorization [J]. Advances in Neural Information Processing Systems, 2014, 4(31): 76-84.

[2] MINCHEVA S. FBK-HLT: An Application of Semantic Textual Similarity for Answer Selection in Community Question Answering[C]// Proceedings of the International Workshop on Semantic Evaluation. 2015.

[3] KIM Y. Convolutional neural networks for sentence classification[C] // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014: 1746-1751.

[4] LIN C Y, OCH F J. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics [J]. Proceedings of Annual Meeting of the Association for Computational Linguistics, 2004: 605-612.

[5] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient Estimation of Word Representations in Vector Space[C]// Proceedings of the International Conference on Learning Representations. 2013.

[6] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26(311): 1-9.

[7] 徐帅. 面向问答系统的复述识别技术研究与实现 [D]. 哈尔滨：哈尔滨工业大学，2009.

[8] KUSNER M J, SUN Y, KOLKIN N I, et al. From word embeddings to document distances[C] // International Conference on Machine Learning. 2015: 957-966.

[9] JASON. Document Similarity With Word Movers Distance[EB/OL]. [2016-06-13]. http://jxieeducation.com/2016-06-13/Document Similarity-With-Word-Movers-Distance.

[10] GUAN Y, WANG X, WANG Q. A New Measurement of Systematic Similarity [J]. IEEE Transactions on Systems Man & Cybernetics Part A Systems & Humans, 2008, 38(4): 743-758.

[11] 郭胜国，邢丹丹. 基于词向量的句子相似度计算及其应用研究 [J]. 现代电子技术，2016，39(13)：99-102.

[12] 李峰，侯加英，曾荣仁，等. 融合词向量的多特征句子相似度计算方法研究 [J]. 计算机科学与探索，2017，11(4)：608-618.

[13] JOHNSON J, DOUZE M, JÉGOU H. Billion-scale similarity search with GPUs [J]. arXiv preprintarXiv: 1702.08734, 2017.

[14] RYGL J, POMIKÁLEK J, ŘEHŮŘEK R, et al. Semantic Vector Encoding and Similarity Search Using Fulltext Search Engines [C]// The Workshop on Representation Learning for Nlp. 2017: 81-90.

(上接第 96 页)

[7] HUTSON C, VENAYAGAMOORTHY G K, CORZINE K A, et al. Intelligent scheduling of hybrid and electric vehicle storage capacity in a parking lot for profit maximization in grid power transactions[C] // IEEE-SA, IEEE-NTDC, IEEE-PELS, IEEE-PES, Proceedings of Energy 2030 Conference. Atlanta, GA, United States, 2008.

[8] 吴红斌，侯小凡，赵波，等. 计及可入网电动汽车的微网系统经济调度[J]. 电力系统自动化，2014，38(9)：77-84.

[9] 周天沛，孙伟. 基于微网的电动汽车与电网互动技术[J]. 电力系统自动化，2018，42(3)：98-104.

[10] HONARMAND M, ZAKARIAZADEH A, JADID S. Optimal scheduling of electric vehicles in an intelligent parking lot considering vehicle-to-grid concept and battery condition[J]. Energy, 2014, 65(2): 572-579.

[11] 刘利兵，刘天琪，张涛，等. 计及电池动态损耗的电动汽车有序充放电策略优化[J]. 电力系统自动化，2016，40(5)：83-90.

[12] 卢志刚，王荟敬，赵号，等. 含 V2G 的虚拟电厂双层逆鲁棒优化调度策略[J]. 电网技术，2017，41(4)：1245-1252.

[13] 葛少云，王龙，刘洪，等. 计及电动汽车入网的峰谷电价时段优化模型研究[J]. 电网技术，2013，37(8)：2316-2321.

[14] KATHLEEN S, LESTER B L. Demand response and electricity market efficiency[J]. The Electricity Journal, 2007, 20(3): 69-85.

[15] PJM INT. Hourly integrated real-time LMP values for 201708 [EB/OL]. [2018-06-12]. http://www.pjm.com/markets-and-operations/energy/real-time/monthlylmp.aspx.