计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(11):159−164 [doi: 10.15888/j.cnki.csa.006055]

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

基于 Word2vec 的文档分类方法①

陈　杰, 陈　彩, 梁　毅

(北京工业大学 信息学部, 北京 100124)

摘　要: 文档的特征提取和文档的向量表示是文档分类中的关键, 本文针对这两个关键点提出一种基于 word2vec 的文档分类方法. 该方法根据 DF 采集特征词袋, 以尽可能的保留文档集中的重要特征词, 并且利用 word2vec 的潜在语义分析特性, 将语义相关的特征词用一个主题词乘以合适的系数来代替, 有效地浓缩了特征词袋, 降低了文档向量的维度; 该方法还结合了 TF-IDF 算法, 对特征词进行加权, 给每个特征词赋予更合适的权重. 本文与另外两种文档分类方法进行了对比实验, 实验结果表明, 本文提出的基于 word2vec 的文档分类方法在分类效果上较其他两种方法均有所提高.

关键词: 文档向量; 文档特征提取; 文档分类; TF-IDF; word2vec

引用格式: 陈杰,陈彩,梁毅.基于 Word2vec 的文档分类方法.计算机系统应用,2017,26(11):159–164. http://www.c-s-a.org.cn/1003-3254/6055.html

Document Classification Method Based on Word2vec

CHEN Jie, CHEN Cai, LIANG Yi

(Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China)

Abstract: The feature extraction and the vector representation are the key points in document classification. In this paper, we propose a classification method based on word2vec for the two key points. This method builds the bag of feature words by Document Frequency (DF) to retain the important feature of the document as much as possible. It takes advantage of the Latent Semantic Analysis of word2vec thus to reduce the size of bag of feature words and the dimension of document vector effectively, which replaces the semantically relevant words with the product of a topic word and proper parameters. Besides, it also gives each feature word the optimal weight by combining with the TF-IDF algorithm. Finally, compared with two other document classification methods, the method presented in this paper has made some significant progress, and the experimental result has proved its effectiveness.

Key words: document vector; feature extraction of document; document classification; TF-IDF; word2vec

1　概述

随着互联网的发展, 越来越多的信息来自于互联网, 如何有效的挖掘、利用这些海量信息, 尤其海量文档信息, 将成为关键, 对文档进行有效分类可以缩小信息的规模, 因此, 精准的文档分类方法依然是当前众多科研工作者所研究的热点问题[1-3].

对海量文档进行分类涉及两个难点问题: 文档的特征提取和文档的向量表示[4]. 要想对不同格式的海量文档进行有效分类, 首先要提取出每篇文档的主题信息, 通常的做法是提取文档的主题特征词并赋予这些特征词合适的权重. 要想对文档进行分类, 单纯的文字信息是无法完成分类工作的, 最常用的方法是将文档映射为一个高维数字向量, 进而再对向量进行分类[5].

目前, 众多科研工作者对文本分类都尝试做了很多改进工作, 比如, 文献[6]中, 李学明等人提出 TFIDFIGE 算法, 在实验中选取信息增益值前 1000 的

① 收稿时间: 2017-02-23; 修改时间: 2017-03-09; 采用时间: 2017-03-20

词做特征词袋, 用 TFIDFIGE 算法计算权重, 为每篇文档建立一个 1000 维的数字向量, 分类器选用为 KNN, 进而完成海量文档的分类工作. 该方法很好的解决了文档中特征词的分布权重问题, 但容易引起维度灾难, 比如: 文档集规模非常庞大, 那么, 特征词袋也需要更加庞大, 信息增益值前 1000 的词将不能覆盖整个文档集的特征, 随之建立的文档向量的维度也将大幅提升, 且该方法形成的文档向量将是高维的、稀疏的, 对后期分类的时间复杂度将有较大影响; 文献[7]中, 唐明等人将分词结果直接作为特征词袋, 用 word2vec 分析得到词袋中每个特征词的词向量, 并把每篇文档中的特征词向量通过 TF-IDF 加权累加而成文档向量, 最终把这些文档向量作为分类器的输入, 进而完成海量文档的分类工作. 该方法很好的解决了文档向量的维度灾难, 但忽略了特征词的语义特征, 比如: 有两篇文档, 一篇文档有 A、B、C 三个特证词, 另一篇有 D、E 两个特征词, 其中 A、B、C、D、E 五个特征词主题且语义均不相关, 使用文献[7]中的方法有可能出现 Vec(A) * TFIDF(A) + Vec(B) * TFIDF(B) + Vec(C) * TFIDF(C) = Vec(D) * TFIDF(D) + Vec(E) * TFIDF(E) 的情况, 这样两篇主题不相关的文档就会被分到一个类别中.

综上, 本文在总结前人研究经验的基础上, 提出一种基于 word2vec 的文档分类方法, 该方法的优势在于: ① 采用 DF 采集特征词袋, 尽可能的保留文档集中的重要特征词; ② 结合 word2vec, 利用其潜在的语义分析特性浓缩特征词袋, 将语义相关的特征词用一个主题词乘以合适的系数来代替, 有效降低了文档向量的维度; ③ 结合 TF-IDF 算法进行特征词加权, 给每个特征词赋予更合适的权重.

2 相关技术

2.1 词袋模型

到目前为止, 在文档分类和自然语言处理领域, 最直观也是最常用的词的表示方法就是词袋模型. 构建词袋模型之前, 往往会收集一个忽略词顺序的特征词袋, 并以特征词袋中词的个数作维数, 使向量的每一维代表一个特征词, 构建高维词向量, 并辅以特征词的出现次数或特征词的其他特征权重作为该维向量的值[8].

举例说明, 如果收集的特征词袋为{西红柿、玉米、小麦、番茄……}, 词袋大小为 100, 用词的出现次数做权重. 假设某篇文档中只出现过“西红柿”一词, 且“西红柿”出现过 10 次, 则该文档可表示为[10, 0, 0, 0, 0, 0, 0……]; 假设某篇文档中只出现过“番茄”一词, 且“番茄”出现过 8 次, 则该篇文档可表示为[0, 0, 0, 8, 0, 0, 0……].

词袋模型会把每篇文档表示为一个维度统一但长度很长的向量, 其中绝大多数元素为 0, 向量中的非 0 维度就代表当前文档中出现过该特征词. 这种表示方法除了形成向量过于稀疏的问题, 还存在一个重要的问题: 任意两个词之间都是孤立的, 仅从向量中看不出两个词是否有关系, 即使是“西红柿”和“番茄”这样的同义词也不能幸免于难.

2.2 TF-IDF 词权算法

首先, 介绍三个与 TF-IDF 算法相关的概念: TF、DF、IDF.

TF 即 Term Frequency, 是指某个特征词在一篇文档中出现的频率, TF 可以很好的表示某个特征词对一篇文档的重要程度, 其计算公式可描述为:

$$TF(word) = \frac{Count(word)}{\sum_{i=0}^{n} Count(word_i)} \tag{1}$$

式 (1) 中, 分子为特征词 *word* 在本篇文档中出现的次数, 分母为本篇文档一共包含的词的个数.

DF 即 Document Frequency, 是指某个特征词在文档集中出现的频率, DF 可以很好的表示某个特征词在文档集中的分布特征, 其计算公式可描述为:

$$DF(word) = \frac{Count(word, docs)}{Count(docs)} \tag{2}$$

式 (2) 中, 分子为文档集中包含特征词 *word* 的文档数量, 分母为文档集中文档的总篇数.

IDF 即 Inverse Document Frequency, 是指逆向文档频率, 它可以很好的表示某个特征词的类别区分能力, 其计算公式可描述为:

$$IDF(word) = \log\left(\frac{Count(docs)}{Count(word, docs)} + 0.01\right) \tag{3}$$

因此, Salton 在 1973 年提出了 TF-IDF (Term Frequency-Inverse Documentation Frequency) 算法[9], 并被论证了在信息检索领域的有效性[10]. TF-IDF 算法是目前最常用的特征词权重计算方法, 其计算公式可描述为:

$$Weight(word) = TF(word) * IDF(word) \tag{4}$$

2.3 word2vec

Word2vec 是由 Mikolov 提出的一种可以快速有效训练词向量的模型, word2vec 吸收了 Bengio 在文献[11]中提出的 NNLM 模型 (Neural Network Language Model) 和 Hinton 在文献[12]中提出的 logLinear 模型的优点, 使用 Distributed Representation 作为词向量的表示方式[13]. 其基本思想是: 通过训练, 将每个词映射成 K 维实数向量 (K 一般为模型中的超参数), 通过词向量之间的距离来判断它们之间的语义相似度, 采用一个三层的神经网络, 分别为: 输入层-投影层-输出层. 并根据词频生成 Huffman 编码 , 使得所有词频相似的词隐藏层激活的内容基本一致, 使出现频率越高的词语激活的隐藏层数目越少, 有效的降低了计算的复杂度, 从而大大提高了 word2vec 处理效率, Mikolov 曾在在文献[14]中指出, 一个优化的单机版本一天可训练上千亿词.

Word2vec 除了拥有很高的处理效率外, 经 word2vec 训练出的词向量还有一个重要特征: 可以揭示特征词之间的潜在联系. 使用 word2vec 训练出的词向量, 其每一维可以表示该特征词的一个潜在特征, 该特征包含但不限于该特征词所在的句子结构、上下文语义. 通过 word2vec 训练得到的词向量, 根据余弦夹角公式, 如式 (5), 可以很容易地推算出在语义上与某个特征词最为相近的其他特征词.

$$\cos\theta = \frac{x_1y_1 + x_2y_2 + \ldots + x_{n-1}y_{n-1} + x_ny_{n-1}}{\sqrt{x_1^2 + x_2^2 \ldots + x_n^2}\sqrt{y_1^2 + y_2^2 \ldots + y_n^2}} \quad (5)$$

3 基于 word2vec 的文档分类方法

本文提出的基于 word2vec 的文档分类方法, 共可分为三个阶段: 预处理阶段、特征提取-建向量阶段、分类阶段, 总流程图如图 1 所示.

3.1 预处理阶段

预处理阶段主要有两项任务: 去掉文档中无用的格式和停词、进行文档分词.

从互联网上采集到的文档, 大部分是格式不一、排版不齐的文档, 这些文档中往往会包含大量 html 标签、空行、标点, 为提高后续任务的效率与精准度, 必须把这些无用的格式过滤掉. 众所周知, 文档的行文中往往还会含有一些“的”、“了”、“吗”、“等”……这些出现频率极高但又毫无语义的词汇, 即停词[15,16], 停词对后续任务的执行效率和精准度也是有很大影响的, 所有也必须过滤掉.

经由前一个步骤的处理, 文档集的规模会缩小三分之一左右, 大大提高了文档分词的效率. 接下来便是采用分词器对文档进行分词操作, 文档分词对于下一个阶段的特征词袋建立和建立文档向量至关重要, 文档分词是将文档集变为数字向量集的先决条件.

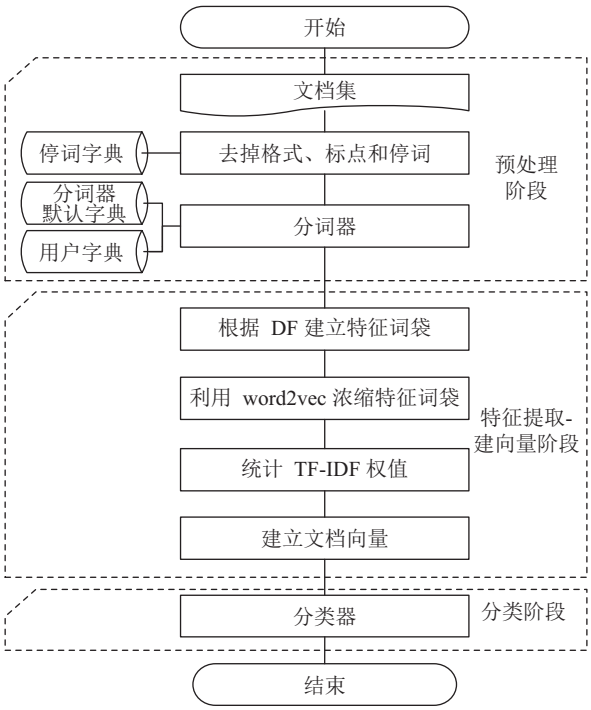

图 1 总流程图

3.2 特征提取-建向量阶段

经由上一个阶段处理, 文档集中的每一篇文档都变成了一个词集 $doc = \{word|word \in doc\} = \{word_1、word_2、word_3……\}$. 本阶段主要有三项任务: 建立特征词袋、浓缩特征词袋、建立文档向量.

在第二节中已经介绍到, DF 即某个特征词在文档集中出现的频率, 在该阶段的第一个任务便是根据特征词的 DF 建立特征词袋, 以尽可能的保留文档集中的重要特征. 某个特征词的 DF 值越大说明该特征词在文档集中分布越广泛, 同时也说明该特征词的个性描述能力越低. DF 的取值范围为[1/n, 1], 其中:

$$\lim_{n\to\infty}\frac{1}{n} = 0 \quad (6)$$

DF 取值为 1 时, 表示文档集中每篇文档都包含该特征词; DF 不可能取 0 值, 只可能无限接近于 0, 若某个特征词的 DF 等于 1/n, 其中 n 为文档集中文档的总篇数, 则表示该特征词只在一篇文档中出现过. 显然,

若要选取即能兼顾覆盖率又能保证特征描述力的特征词袋, DF=1 和 DF=1/n 的特征词是均不能放入词袋的. 根据经验值及实验论证, 能放入特征词袋的特征词 DF 的取值范围一般为: [0.1/CN, 1/2], 其中 CN 为训练集数据分类结果中类簇的个数, 即放入特征词袋的特征词至少保证在某类文档集的十分之一的文档中都出现过, 同时又不会在全部文档集的一半以上的文档中出现过, 这样既可以保证特征词的“个性”, 也可以很好兼顾特征词的“共性”.

该阶段的第二个任务是浓缩特征词袋. 将上一个步骤得到的词袋中的特征词输入进 word2vec, 利用 word2vec 将特征词集训练成词向量, 并将这些词向量进行划分聚类, 选取词向量之间的夹角余弦做相似度度量. 当两个词向量夹角余弦等于 1 时, 这两个特征词完全重复; 当两个词向量夹角的余弦值接近于 1 时, 两个特征词相似; 两个词向量夹角的余弦越小, 两个特征词越不相关. 通过 word2vec 的训练和聚类划分, 语义相似或相近的特征词会聚到一个类簇中, 因此, 几个语义相关的特征词, 可以从中选取一个特征词乘以与其的夹角余弦值来表示, 如公式 (7) 所示:

$$word_a = word_b * \cos(\vec{a}, \vec{b}) \tag{7}$$

举例说明, 经 word2vec 训练, “番茄”的词向量与“西红柿”的词向量夹角余弦为 0.73(训练结果与输入词集有关). 显然, 特征词袋以及文档集中的“番茄”可以用“西红柿”乘以 0.73 来代替, 以此类推, 利用 word2vec 这一优良特性可以大幅缩减特征词袋的大小, 有效预防后期文档向量维度灾难的发生.

该阶段的最后一个任务的是建立文档向量. 首先, 依据 TF-IDF 词权算法计算出特征词袋中每个特征词在每篇文档中的权重, 然后以每个特征词为维度建立文档向量, 文档向量的维度等于特征词袋中特征词的个数. 至此, 文档不再由文字或词语来描述, 而是由数字来描述, 文档集由文字的集合变成了数字向量的集合, 如式 (8) 所示, 方便了后期的分类计算.

$$\overrightarrow{doc_i} = (tfidf_{w_1}, tfidf_{w_2} \cdots\cdots tfidf_{w_{n-2}}, tfidf_{w_{n-1}}, tfidf_{w_n}) \tag{8}$$

3.3 分类阶段

本阶段的主要任务是对文档向量集进行分类.

分类阶段的相似度计算公式为:

$$Sim(\overrightarrow{doc_1}, \overrightarrow{doc_2}) = \left| \frac{Eu(\overrightarrow{doc_1}, \overrightarrow{doc_2})}{\mathrm{Cos}(\overrightarrow{doc_1}, \overrightarrow{doc_2})} \right| = \left| \frac{\sqrt{\sum_{k=1}^{n}(x_k{}^2 - y_k{}^2)} * \sqrt{\sum_{k=1}^{n} x_k{}^2} * \sqrt{\sum_{k=1}^{n} y_k{}^2}}{\sum_{k=1}^{n} x_k y_k} \right| \tag{9}$$

相似度计算公式采用欧式距离乘以夹角余弦的倒数, 这样既考虑了向量间的空间距离大小又兼顾了向量间的夹角方向问题, 防止了距离小但方向反向的向量被分类到一个类簇中.

该阶段完成后, 相似度高的文档便会被分到一个类簇中. 至此, 便完成了对文档的分类工作.

4 实验设计及分析

4.1 实验设计

针对上文提出的基于 word2vec 的文档分类方法进行了实验设计, 本文实验选用的数据集为搜狗中文实验室的全网中文新闻数据集——“SogouCA”精简版[17], 该数据集采集了来自多家新闻站点 9 个栏目的分类新闻数据, 共 17910 篇文档, 实验数据集中的文档分类情况如表 1 所示.

表 1 “SogouCA”数据集分布

序号	主题类别	文档数量
1	财经	1990
2	IT	1990
3	健康	1990
4	体育	1990
5	旅游	1990
6	教育	1990
7	招聘	1990
8	文化	1990
9	军事	1990

本文实验的预处理阶段, 分词器选用的是中国科学院的汉语词法分析系统 ICTCLAS(Institute of Computing Technology Chinese Lexical Analysis System)[18]. 在最终的分类阶段, 选用 KNN(K=10) 和 SVM 两种分类器分别进行了分类实验, 以消除不同分类器对分类结果的影响, 并且采用五分交叉验证法, 把数据量随机分成 5 份, 每次取其中 4 份作为训练集, 剩余 1 分做测试集, 最终取 5 次实验结果的平均值.

本文实验采用 C++作为算法的实现语言, 开发环境为 Visual Studio 2013, 将本文提出的文档分类方法

同文献[6]中的方法和文献[7]中的方法进行了对比实验.

4.2 评价指标

本文实验引入三个文本分类领域常用的评价指标, 即: 召回率、精准率和 F-measure 值[19].

其中, 召回率 (Recall) 是指某个类簇内同属于某类别文档的数量与文档集中本属于该类别文档的数量的比值, 一般用字母 R 表示; 准确率 (Precision) 是指某个类簇内同属于某类别文档的数量与该类簇内所有文档的数量的比值, 一般用字母 P 表示; F-measure 值是召回率 (R) 和准确率 (P) 的几何平均值, 是用来综合评价文档的分类效果的一种指标, 其计算公式可描述为:

$$F-measure=\frac{2RP}{R+P} \tag{10}$$

4.3 实验结果分析

17910 篇文档经过去停词、分词器分词后, 特征词袋中不重复的特征词有 84874 个, 根据 DF 排序, 提取 DF 大于 0.01 且小于 0.5 的特征词, 共提取特征词 2493 个, 经 word2vec 浓缩后, 特征词袋还有 340 个特征词, 大大消除了后期文档向量的维度灾难隐患.

表 2 文献[6]分类方法效果 (单位: %)

类别	KNN			SVM		
	R	P	F	R	P	F
财经	79.36	75.94	77.61	79.74	82.44	81.07
IT	87.49	74.67	80.57	84.68	79.31	81.91
健康	80.67	79.62	80.14	83.42	83.88	83.65
体育	79.61	84.56	82.01	88.37	75.28	81.30
旅游	83.79	75.54	79.45	80.31	82.71	81.49
教育	76.98	75.25	76.11	79.05	79.84	79.44
招聘	86.91	73.99	79.93	84.00	78.68	81.25
文化	80.09	78.93	79.51	80.61	82.08	81.34
军事	79.03	83.30	81.11	79.35	83.24	81.25
avg	81.55	77.98	79.60	82.17	80.83	81.41

表 3 文献[7]分类方法效果 (单位: %)

类别	KNN			SVM		
	R	P	F	R	P	F
财经	80.34	83.49	81.88	90.45	80.48	85.17
IT	84.53	74.86	79.40	82.39	87.08	84.67
健康	83.46	78.83	81.08	81.13	84.51	82.79
体育	76.64	82.91	79.65	86.08	87.91	86.99
旅游	86.58	79.80	83.05	84.81	87.64	86.20
教育	79.76	78.54	79.15	89.76	79.31	84.21
招聘	83.95	79.23	81.52	88.50	83.88	86.13
文化	82.88	80.10	81.47	80.44	87.28	83.72
军事	80.01	84.17	82.04	90.05	79.85	84.64
avg	82.02	80.21	81.03	85.96	84.22	84.95

表 4 本文分类方法效果 (单位: %)

类别	KNN			SVM		
	R	P	F	R	P	F
财经	84.63	84.52	84.57	91.24	84.12	87.54
IT	91.81	76.46	83.43	89.97	86.68	88.29
健康	87.74	81.41	84.46	83.92	89.55	86.64
体育	80.93	85.78	83.28	86.66	88.92	87.78
旅游	85.11	85.09	85.10	85.60	85.52	85.56
教育	84.05	81.70	82.86	90.55	83.48	86.87
招聘	90.23	80.44	85.05	87.16	86.05	86.60
文化	87.16	85.39	86.27	87.90	82.92	85.34
军事	88.48	81.15	84.66	90.84	81.76	86.06
avg	86.68	82.44	84.41	88.20	85.44	86.74

从表 2、表 3、表 4 中可以看出, 排除分类器因素, 本文提出的文档分类方法在召回率上, 较文献[6]中的分类方法提高了 6.82%, 较文献[7]中的分类方法提高了 4.15%; 在准确率上, 较文献[6]中的分类方法提高了 5.71%, 较文献[7] 中的分类方法提高了 2.12%; 在 F-measure 值上, 较文献[6]中的分类方法提高了 6.29%, 较文献[7]中的分类方法提高了 3.14%. 因此, 本文提出的基于 word2vec 的文档分类方法在分类效果上均优于其他两种方法, 证明了本文提出的方法在文档分类方面的有效性.

5 结语

本文在分析、总结前人研究经验的基础上, 针对文档分类中的两个难点——文档的特征提取和文档的向量表示, 提出了一种基于 word2vec 的文档分类方法. 该方法根据 DF 采集特征词袋, 以尽可能的保留文档集中的重要特征词, 利用 word2vec 的潜在语义分析特性, 浓缩了特征词袋的大小, 将语义相关的特征词用一个主题词乘以合适的系数来代替, 降低了文档向量的维度, 节约了分类阶段的耗时, 并且该方法还结合 TF-IDF 改进算法, 对特征词进行加权, 赋予每个特征词合适的权重. 最后, 本文设计了三组对比实验, 与另外两种文档分类方法相比, 本文提出的基于 word2vec 的文档分类方法在分类效果上较其他两种方法均有所提高.

参考文献

1 Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. arXiv: 1301.3781, 2013.

2 Hwang M, Choi C, Youn B, *et al*. Word sense disambiguation based on relation structure. Proc. of the 2008 International

Conference on Advanced Language Processing and Web Information Technology. Dalian, Liaoning, China. 2008. 15–20.
3 苏金树, 张博锋, 徐昕. 基于机器学习的文本分类技术研究进展. 软件学报, 2006, 17(9): 1848–1859.
4 孙建涛. Web 挖掘中的降维和分类方法研究[博士学位论文]. 北京: 清华大学, 2005.
5 胡承成. 基于文本向量的微博情感分析[硕士学位论文]. 北京: 中国科学院大学, 2015.
6 李学明, 李海瑞, 薛亮, 等. 基于信息增益与信息熵的 TFIDF 算法. 计算机工程, 2012, 38(8): 37–40.
7 唐明, 朱磊, 邹显春. 基于 Word2Vec 的一种文档向量表示. 计算机科学, 2016, 43(6): 214–217, 269. [doi: 10.11896/j.issn.1002-137X.2016.06.043]
8 Lauly S, Boulanger A, Larochelle H. Learning multilingual word representations using a bag-of-words autoencoder. Computer Science, 2014.
9 Salton G, Yu CT. On the Construction of Effective Vocabularies for Information Retrieval. Proc. of the 1973 Meeting on Programming Languages and Information Retrieval. New York, NY, USA. 1973. 48–60.
10 Salton G, Fox EA, Wu H. Extended boolean information retrieval. Communications of the ACM, 1983, 26(11): 1022–1036. [doi: 10.1145/182.358466]
11 Bengio Y, Ducharme R, Vincent P, *et al*. A neural probabilistic language model. The Journal of Machine Learning Research, 2003, 3(6): 1137–1155.
12 Mnih A, Hinton G. Three new graphical models for statistical language modelling. Proc. of the 24th International Conference on Machine Learning. Corvalis, Oregon, USA. 2007. 641–648.
13 Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. arXiv: 1301.3781, 2013.
14 Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. Proc. of Advances in Neural Information Processing Systems 26. 2013.
15 熊文新, 宋柔. 信息检索用户查询语句的停用词过滤. 计算机工程, 2007, 33(6): 195–197.
16 Lo RTW, He B, Ounis I. Automatically building a stopword list for an information retrieval system. Journal of Digital Information Management, 2005, (3): 3–8.
17 “SogouCA”语料库. http://www.sogou.com/labs/resource/ca.php, 2012.
18 Zhang HP, Yu HK, Xiong DY, *et al*. HHMM-based Chinese lexical analyzer ICTCLAS. Proc. of the 2nd SIGHAN Workshop on Chinese Language Processing. Sapporo, Japan. 2003. 184–187.
19 Anoual H, Aboutajdine D, Elfkihi S, *et al*. Features extraction for text detection and localization. Proc. of the 5th International Symposium on I/V Communications and Mobile Network (ISVC). Rabat, Morocco. 2010. 1–4.