融合形态和语义相似度的对话短文本聚类

陈国梁,贺 樑,胡琴敏,杨 静

(华东师范大学 信息科学技术学院,上海 200241)
E-mail: glchen@ ica. stc. sh. cn

摘 要: 智能对话系统是一种人机交互系统,其产生的对话文本是一种特殊的短文本并蕴含着丰富的信息. 这类对话短文本具有口语化、输入错误、同音不同字以及同义不同字等特点,导致现有的经典聚类算法无法进行有效的处理. 为了对这类对话短文本进行有效的聚类,提出一种形态和语义相似度相结合的短文本聚类算法,其中形态相似度采用字符串相似度,语义相似度基于 HowNet 和 WordNet 词语知识库. 通过在多种短文本数据集上实验结果表明,本文短文本聚类算法在中英文短文本数据集上均有较好的聚类效果. 在小 i 机器人对话文本数据集上的聚类纯度指标相对于 Kmeans 算法和 gcluto 工具包中的算法分别有 20% 和 7% 的提高.

关 键 词: 智能对话系统; 短文本; 聚类算法; 形态相似度; 语义相似度
中图分类号: TP311 **文献标识码:** A **文 章 编 号:** 1000-1220(2015)09-1963-05

Improve Dialogue Short Text Clustering by Fusion Form and Semantic Similarity

CHEN Guo-liang, HE Liang, HU Qin-ming, YANG Jing
(College of Information Science and Technology, East China Normal University, Shanghai 200241, China)

Abstract: Intelligent dialogue system is a human-computer interaction system, which produces a special kind of short texts containing a wealth of information. Such dialogue short texts with colloquial, input errors, homonyms and synonyms characteristics, so that existing classical clustering algorithms can't be effectively processed. This paper proposes a short text clustering algorithm based on combination of form and semantic similarity for such dialogue short text, the form similarity using string similarity, the semantic similarity based on HowNet and WordNet words knowledge databasess. The experimental results of several short text data sets show that the short text clustering algorithm of this paper has good performance on Chinese and English short texts. On the Xiaoi dialogue text data set, the clustering purity of this paper compared with Kmeans and the gCluto toolkit respectively 20% and 7% increase.

Key words: intelligent dialogue system; short text; clustering algorithm; form similarity; semantic similarity

1 引 言

随着移动通信和移动互联网的飞速发展,出现各种智能对话系统,例如 Siri、google now、小 i 机器人等. 以小 i 机器人为例,其用户数已超过 1 亿,每年有 100 亿次的对话访问并产生大量有价值的对话文本数据. 针对这些对话文本数据进行聚类分析,可以将相似的对话文本聚集起来并形成若干个重要的聚类中心,从而挖掘用户兴趣并提炼出知识来更新和完善知识库. 因此,针对对话文本进行聚类研究具有巨大的价值.

然而,这类对话文本是一种特殊的短文本,通常只有一两句话,甚至只有几个词语或指令. 通过对小 i 机器人约 15 万条对话短文本的分析和统计,我们发现这类短文本具有以下几个特点:

1) 信息稀疏: 特征信息少,只包含少量的词,小 i 机器人对话文本的平均字数为 7. 91.

2) 主题单一,一个对话短文本通常只讨论一件事情.

3) 形式不规则,口语化,大小写字母混用以及拼写错误特征明显. 例如 "。,。,666 5 元", "看看我的话费剩多少呀!", "kTWlan"与"KTWLAN", "余额"变成"余饿".

4) 同义词、语义鸿沟现象突出. 例如"查询话费余额"与"查看剩多少钱".

由于对话类短文本的这些特点,现有的一些相似度计算方法和聚类算法不能很好处理这类短文本数据. 本文针对这类对话短文本,提出一种基于形态和语义相似度相结合的聚类算法,其中形态相似度是字符串相似度,语义相似度是基于词库知识库的词语语义相似度. 字符串相似度能较好适应口语化、输入错误噪音的短文本; 语义相似度能较好解决对话短文本中的同义词、语义鸿沟问题.

本文剩余部分的组织结构如下: 第二部分介绍相关工作,第三部分介绍本文提出的方法,第四部分介绍实验搭建,第五部分为实验结果与分析,第六部分为总结和未来研究工作.

2 相关工作

目前总体有三大类型的短文本聚类方法. 第一类利用搜索引擎获取辅助数据来扩充短文本信息. 例如 Sahami 等[1] 利用搜索引擎获取辅助数据来计算短文本相似度,从而可以提

收稿日期: 2014-06-27 收修改稿日期: 2014-11-12 基金项目: 国家科技支撑项目(2012BAH93F03) 资助; 上海市科委项目(13511506201) 资助. 作者简介: 陈国梁,男,1990 年生,硕士,研究方向为数据挖掘与大数据处理; 贺 樑,男,1973 年生,博士,教授,研究方向为用户行为分析与语义信息处理; 胡琴敏,女,1980 年生,博士,副教授,研究方向为信息检索与大数据处理; 杨 静,女,1976 年生,博士,副教授, 研究方向为多媒体技术、知识管理、自然语言处理.

高短文本聚类效果. 第二类利用主题模型从短文本集合中挖掘隐含主题,然后基于隐含主题进行聚类处理. 例如 Quan 等[2] 利用 LDA 提取短文本集合的主题,然后基于这些主题计算短文本之间的相似度,最后基于这种相似度对短文本集合进行聚类. 第三类利用知识库或维基百科扩展词特征信息. 例如 Banerjee 等[3] 利用维基百科来扩充短文本的向量空间,然后利用扩展后的向量空间对短文本进行聚类. Hu 等[4] 充分利用 WordNet 和维基百科来扩充内部和外部的语义信息来提高短文本聚类效果. Xin 等[5] 基于 HowNet 知识库对中文微博文本数据进行聚类研究,从语义层面对向量空间进行补充来提高聚类效果. 另外一些利用字符串相似性和核心词进行短文本聚类. Yang 等[6] 利用字符串相似性对网络短文本进行聚类来挖掘舆情热点. Ni 等[7] 通过发现的核心词对短文本进行聚类. Islam 等[8] 结合基于语料的词语语义和字符串相似度计算短文本相似度.

3 FS-STC 算法

本节首先介绍短文本和类中心的数据表示形式,然后介绍短文本之间、短文本与类中心以及类中心之间的相似度计算方法; 最后介绍 FS-STC 算法流程.

3.1 数据表示形式

对话短文本中存在一些无效字符、形式不规则以及大小写字母混杂的多种噪音信息,本文利用正则过滤、字符串处理方法去除这些噪音信息,形成格式较为规整的字符串文本,例如 "。,。,666 5 元"将被转化为"666 5 元", "kTWlan"和"KT-WLAN"均被转换为"ktwlan". 口语化严重的对话短文本中每个词语的重要程度不尽相同,例如"看看我的话费剩多少呀!"中的"话费"和"剩"的重要程度显然要大于其他的词. 下面给出对话短文本和类中心的定义.

定义 1. (短文本) 短文本(ST) 由原始短文本(RST)、规整化字符串文本(FST) 和关键词集合(KWL) 三部分组成. $ST=\{RST,FST/WT,KWL=[KW_1/wt_1,\cdots,KW_i/wt_i,\cdots]\}$,$WT$ 表示 FST 的权重,wt_i 表示 KW_i 的权重.

WT 值与 ST 在类中的地位或重要程度成正比,可以将该权重信息融入相似度计算方法,提高权重值较大 ST 的作用. wt_i 值体现了不同词对 ST 的贡献能力,同样我们也可以将该权重融入相似度计算方法,从而提高权重较大词的作用.

定义 2. (类中心) 类中心($Center$) 由类标号(CID)、类权重(CWT)、类成员数目(CMN)、原始短文本集合($CRSTL$)、规整化字符串文本集合($CFSTL$) 和关键词集合($CKWL$) 组成.

$$Center=\{CID,CWT,CMN,CRSTL=[RST_1,\cdots,RST_i,\cdots],$$
$$CFSTL=[FST_1/WT_1,\cdots,FST_j/WT_j,\cdots],$$
$$CKWL=[KW_1/wt_1,\cdots,KW_k/wt_k,\cdots]\}$$

WT_j 表示 FST_j 的权重,即体现了 FST_j 对 $Center$ 的贡献能力或影响程度; wt_k 表示 KW_k 的权重,即 KW_k 在 $Center$ 中的地位或重要程度.

ST 和 $Center$ 表达式中的关键词集合以及相应的权重值是采用关键词提取算法 TextRank[10]. TextRank 算法是一种基于图的排序模型,基本原理同 PageRank 类似. 首先利用上下文窗口将已分词的短文本集合构建成词语与词语的图结构,其中词语作为图的顶点,词语与词语的上下文共现关系作为图的边; 然后利用图排序模型的迭代算法进行更新图中的顶点的权重直到收敛; 最后根据图的顶点权重值选取权重值较大的前 N 个词语作为关键词集合.

3.2 相似度计算方法

基于 ST、$Center$ 表示形式,本文结合字符串相似度[9] 和词语语义相似度[10,11] 来计算短文本相似度 $Simi$,其中字符串相似度表示为 $FSimi$,词语语义相似度表示为 $SSimi$. $Simi=\partial*FSimi+(1-\partial)*SSimi$,($\partial\in[0,1]$),其中 ∂ 为 $FSimi$ 的权重因子,即形态因素在相似度计算中所占比例.

$FSimi$ 计算方法主要采用字符串之间的编辑距离进行计算. $SSimi$ 计算方法主要采用 HowNet 或 WordNet 知识库[10],其中两个主要的概念 "概念"与"义元". "概念"是对词语语义的一种描述,每一个词可以表达为几个"概念"; "义原"是用来描述一个"概念"的最小意义单元,即一个"概念"由一系列的"义原"进行描述. 对于两个词语 $Word1$ 和 $Word2$,如果 $Word1$ 有 n 个"概念": $S_{11},S_{12},\cdots,S_{1n}$,$Word2$ 有 m 个"概念": $S_{21},S_{22},\cdots,S_{2m}$. $Word1$ 和 $Word2$ 的相似度为各个"概念"的相似度最大值. 由于所有的"概念"均由若干个"义原"组成,因此"义原"的相似度计算是"概念"相似度计算的基础. 由于所有的"义原"根据上下位关系构成了一个树状的"义原"层次体系,通过"义原"的路径距离来计算相似度,可以得到两个"义原"之间的语义相似度.

定义 3. (ST 之间相似度)

$Simi(ST_A,ST_B)$ 表示 ST_A 和 ST_B 的相似度,$FSimi(ST_A.FST,ST_B.FST)$ 表示规整化文本的形态相似度,$SSimi(ST_A.KWL,ST_B.KWL)$ 表示 ST_A 和 ST_B 的语义相似度.

$$FSimi(ST_A.FST,ST_B.FST)=Levenshtein(ST_A.FST,ST_B.FST) \quad (1)$$

$$SSimi(ST_A.KWL,ST_B.KWL)=\frac{1}{2}\left\{\frac{\sum_{w_A\in ST_A.KWL}(\max SSimi(w_A,ST_B.KWL)*wt_A)}{\sum_{w_A\in ST_A.KWL}wt_A}+\frac{\sum_{w_B\in ST_B.KWL}(\max SSimi(w_B,ST_A.KWL)*wt_B)}{\sum_{w_B\in ST_B.KWL}wt_B}\right\} \quad (2)$$

公式(2) 中 wt_i 作为词语的权重因素融入计算语义相似度计算方法中,会提高权重较大的词语对相似度的影响程度.

定义 4. (ST 与 $Center$ 之间相似度)

$Simi(ST,Center)$ 表示 ST 与 $Center$ 的相似度,$FSimi(ST.FST,Center.CFSTL)$ 表示 ST 与 $Center$ 的形态相似度,$SSimi(ST.KWL,Center.CKWL)$ 表示 ST 与 $Center$ 的语义相似度.

$SSimi(ST.KWL,Center.CKWL)$ 计算方法同公式(2) 一样. $FSimi(ST.FST,Center.CFSTL)$ 计算方法如下:

$$FSimi(ST.FST,Center.CFSTL)=\frac{\sum_{FST_i\in Center.CFSTL}WT_i*FSimi(ST.FST,FST_i)}{\sum_{FST_i\in Center.CFSTL}WT_i} \quad (3)$$

公式(3) 中 WT_i 作为短文本的权重因素融入形态相似度计算方法中,会提高权重较大的短文本对相似度的影响程度.

定义 5. ($Center$ 之间的相似度)

$Simi(Center_A,Center_B)$ 表示 $Center_A$ 与 $Center_B$ 的相似

度，$FSimi(Center_A.CFSTL, Center_B CFSTL)$ 表示 $Center_A$ 与 $Center_B$ 的形态相似度，$SSimi(Center_A.CKWL, Center_B.CKWL)$ 表示 $Center_A$ 与 $Center_B$ 的语义相似度.

$SSimi(Center_A.CKWL, Center_B.CKWL)$ 计算方法同公式(2)一样. $FSimi(Center_A.CFSTL, Center_B CFSTL)$ 计算公式如下:

$$FSimi(Center_A.CFSTL, Center_B CFSTL) = \sum_{FST_{Aj} \in Center_A.CFSTL} WT_{Aj} * \frac{\sum_{FST_{Bi} \in Center_B.CFSTL} WT_{Bi} * FSimi(FST_{Aj}, FST_{Bi})}{\sum_{FST_{Bi} \in Center_B.CFSTL} WT_{Bi}} \Big/ \sum_{FST_{Aj} \in Center_A.CFSTL} WT_{Aj} \tag{4}$$

3.3　聚类算法过程

这部分首先给出聚类问题描述，然后介绍本文提出的 FS-STC 算法聚类流程和类中心更新过程.

3.3.1　问题描述

问题定义: 将输入的短文本集 $STList = \{\cdots, ST_i, \cdots\}$ 聚成若干个类中心集 $CenterList = \{\cdots, Center_j, \cdots\}$，其中 ST_i 和 $Center_j$ 分别为定义 1 和定义 2 给出的短文本和类中心表示形式.

3.3.2　FS-STC 算法流程和类中心更新

FS-STC 的是一种基于相似度的聚类算法，其中的相似度采用定义 3、4、5 给出的相似度计算方法. FS-STC 算法的核心过程是基于相似度距离将输入短文本集合进行聚类形成类中心集合和短文本加入类中心和类中心更新的算法流程的伪代码分别见图 1、图 2.

Algorithm 1 Cluster short text list

Input: *cL* Center list, *stL* Short text list, *T* Similarity distance threshold, *maxNum* maximum number of clustering centers
Output: *cL* Center list

```
1: Initialize cL as an empty list;
2: Set FLAG as the flag to represent whether added into existed center;
3: Set D as the similarity distance;
4: function Cluster ST List(stL,T)
5:     while (stL,has more st) do
6:         FLAG ← false;
7:         if (cL is empty) then
8:             create center based on st;
9:             add center into cL;
10:        else
11:            while (cL has more center) do
12:                D ← calculate the similarity between st and center;
13                 if (D<T) then
14:                    add st into center and update center;
15:                    FLAG ← true;
16:            if (FLAG is false) then
17:                create center based on st;
18:                add center into cL;
19:    if (cL.size()>maxNum) then
20:        cL ← select and retain the top maxNum clustering centers;
    return cL;
```

图 1　短文本集合聚类算法流程

Fig. 1　Clustering process of short text collection

4　实验搭建

这部分主要介绍实验实验数据集、评测指标、FS-STC 实验过程和对比算法.

4.1　实验数据集

本文实验评测的中文短文本数据集分别为: 小 i 智能机器人对话数据集(Xiaoi DataSet)和百度搜索关键词数据集(Baidu DataSet); 英文短文本数据来自 Zekikovitz(2002)[12]的 3CPhys 和 7CNetv. Xiaoi DataSet 正是 FS-STC 算法需要解决的一类对话短文本. Baidu DataSet 作为一种中文搜索短文本，用来测试 FS-STC 算法对中文其他短文本的适用情况. 3CPhys 和 7CNetv 作为英文类型的短文本，用来测试 FS-STC 算法对英文短文本的适应情况. 这四种实验评测数据集的统计信息见表 1.

表 1　四种实验评测数据集的统计信息

Table 1　Statistical information of the four evaluation datasets

数据集	类别数	文本条数	最小字数	最多字数	平均字数
Xiaoi DataSet	10	4729	4	30	7.91
Baidu DataSet	10	4853	2	15	6.53
3CPhys	3	1066	3	17	9.39
7CNetv	7	1495	1	9	3.25

Algorithm 2 Add short text into center and Update center

Input: *center* Previous clustering center, *st* Formalized short text
Output: *center* Updated clustering center

```
1: Set K0 as the number of formalized text to retain;
2: Set N0 as the number of word to retain;
3: function AddAndUpdateCenter(center, st)
4:     cTL←the CFSTL of center;
5:     cWL←the CKWL of center;
6:     stT←the FST of st;
7:     stWL←the KWL of st;
8:     if(cTL contain stT) then
9:         Update the weight stT in cTL;
10:        Add the weight of stT upto centerWeight;
11:    else
12:        Put stT and its weight into cTL;
13:        Add the weight of stT upto centerWeight;
14:    while(stWL has more stW) do
15:        if(stWL contain stW) then
16:            Update the weight of stW in stWL;
17:        else
18:            Put stW and its weight into stWL;
19:    if(cTL.size()>K0) then
20:        cTL←Select Top K0 stT by its weight;
21:    if(cWL.size()>N0) then
22:        cWL←Select Top N0 stW by its weights;
23:    center.CFSTL←cTL;
24:    center.CKWL←cWL;
25:    center.CWT←center.CWT+1; return center;
```

图 2　短文本加入类中心和类中心更新算法流程

Fig. 2　Short text adding into class-center and class-center update algorithm

4.2　实验评测

本文采用真实的分类数据来评测各种聚类算法的结果，将聚类结果构建成一个混淆矩阵，根据该混淆矩阵可以计算每个类的熵与纯度指标[14]. 熵和纯度指标是聚类效果的两个重要的指标，文献 Rasmussen Matt[13] 和 Yu Yong[15] 中均采用熵和纯度作为实验结果的评价指标. 熵指标可以度量聚类结果的混杂程度，其值越小表明聚类越好. 纯度: 纯度指标度量的是一个聚类中仅包含一个类别成员的程度，其值越大表明聚类越好. 对于聚类结果的每个类($Center_i$)，熵和纯度指标的公式为:

$$entropy(Center_i) = -\sum_{j=1}^{k} \Pr_i(c_j)\log_2 \Pr_i(c_j)$$
$$purity(Center_i) = \max_j(\Pr_i(c_j)) \quad (5)$$

其中 $\Pr_j(c_j)$ 是 $Center_i$ 中属于类别 c_j 的成员数所占的比例. 对于整个聚类结果($Centers$)的熵和纯度为:

$$entropy_{total}(Centers) = \frac{\sum_{i=1}^{k} \frac{|Center_i|}{|Centers|} entropy(Center_i)}{-\sum_{j=1}^{k} \frac{1}{k}\log_2\left(\frac{1}{k}\right)}$$
$$purity_{total}(Centers) = -\sum_{i=1}^{k} \frac{|Center_i|}{|Centers|} \times purity(Center_i) \quad (6)$$

4.3 FS-STC 实验过程

这部分主要介绍本文实验过程. 实验过程主要包括二个阶段,第一阶段首先针对原始文本数据进行过滤和预处理,例如文本长度过滤、中文分词、英文字符串的统一化等处理,然后利用关键词提取算法获取关键词和其权重值; 第二阶段利用 FS-STC 算法对短文本集合进行聚类处理. 为了阐述整个实验过程,图 3 以 Xiaoi DataSet 为例进行描述. 、

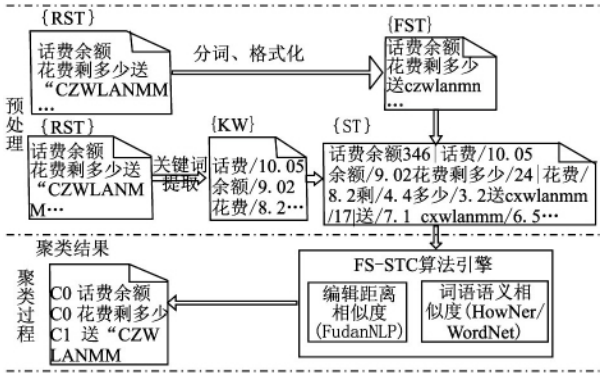

图 3 FS-STC 实验过程示意图

Fig. 3 Schematic diagram of FS-STC´s experiment process

4.4 对比算法实验

本文基于开源工具包实现 5 种对比实验,其中包括 gCLUTO [13] 工具包中的 4 种聚类算法: rb,direct,agglo,graph, Mahout [14] 工具包中 Kmeans 聚类算法. rb 算法是重复二分法进行聚类,direct 算法是直接计算进行聚类,agglo 算法是典型的凝聚聚类,graph 算法是基于图形分割优化进行聚类. Kmeans 算法是基于划分的经典聚类算法.

5 实验结果与分析

FS-STC 算法和其他 5 种对比算法在 4 种短文本数据集上的聚类结果的纯度和熵指标的详细结果分别见表 2 和表 3.

这 5 种对比聚类算法,除了调整聚类个数之外,其他采用默认参数. FS-STC 算法中的融合系数 ∂ 取 0. 5,相似度距离阈值 T 取 0. 55. 上述所有聚类算法中,对于可以指定聚类个数的算法,其聚类个数均设置成相应数据集的类别数.

表 2 中的值表示算法聚类结果的纯度指标,其值越大越好; 表 3 中的值表示算法聚类结果的熵指标,其值越小越好. 我们观察表 2、3 发现 FS-STC 算法在 Xiaoi DataSet 数据集上聚类结果的纯度和熵指标都要优于其他 5 种聚类算法的聚类结果相应的指标,其聚类结果纯度指标比最好的算法 rb 有 7% 的提高. FS-STC 算法在 Baidu DataSet、7CNetv 二个数据集上的聚类结果纯度指标比最好的算法分别有 1% 和 6% 的提高. 综合上述比较分析可以发现 FS-STC 算法对于中英文的短文本均有较好的聚类效果,尤其是像 Xiaoi DataSet 这类对话短文本.

表 2 各种聚类算法的纯度指标结果

Table 2 Purity results of clustering algorithms

数据集	rb	direct	agglo	graph	Kmeans	FS-STC
Xiaoi DataSet	0. 902	0. 841	0. 763	0. 801	0. 678	0. 963 (+0. 061) (1)
Baidu DataSet	0. 520	0. 535	0. 435	0. 415	0. 286	0. 540 (+0. 005) (1)
3CPhys	0. 606	0. 530	0. 625	0. 663	0. 486	0. 611 (-0. 052) (3)
7CNetv	0. 374	0. 369	0. 329	0. 371	0. 308	0. 401 (+0. 027) (1)

表 3 各种聚类算法的熵指标结果

Table 3 Entropy results of clustering algorithms

数据集	rb	direct	agglo	graph	Kmeans	FS-STC
Xiaoi DataSet	0. 152	0. 192	0. 368	0. 278	0. 395	0. 135 (-0. 017) (1)
Baidu DataSet	0. 607	0. 616	0. 737	0. 725	0. 838	0. 626 (+0. 019) (3)
3CPhys	0. 796	0. 849	0. 769	0. 728	0. 863	0. 783 (+0. 055) (3)
7CNetv	0. 803	0. 820	0. 871	0. 838	0. 932	0. 825 (+0. 022) (3)

下面针对聚类过程中的三个重要因素: 形态语义二种因素的融合系数 ∂、相似度距离阈值 T 以及相似度计算中的权重信息进行实验分析. 这里选取 Baidu DataSet、Xiaoi DataSet 进行实验,实验结果见图 4、图 5.

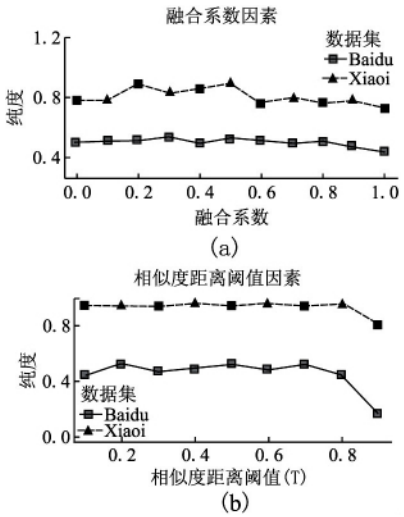

图 4 不同融合系数和相似度阈值 T 的情况下 FS-STC 算法聚类纯度

Fig. 4 Clustering purity results of FS-STC algorithm based on different fusion coefficients and similarity distance thresholds

图 4(a) 中当融合系数为 0 的时候,说明相似度只包含语

义相似度因素; 当融合系数为 1 的时候,说明相似度只包含形态相似度因素. 从图 4 实验结果中发现,当融合系数在 0.3 和 0.5 的时候聚类结果达到最好,显然要比融合系数为 0 和 1 的时候要好,即比单纯的语义或形态相似度的聚类效果要好. 其中基于 HowNet 或 WordNet 知识库的语义相似度可以在一定程度上解决短文本的同义词、语义鸿沟的现象,从而相对与基于词袋向量的传统聚类算法有一定程度的提升; 但是像小 i 机器人对话短文本中包含一些形式不规则、输入错误等噪音信息以及一些特殊指令信息的情况下,词语语义相似度将受到影响,然后字符串相似度可以较好适应这些短文本,因此将这二种相似度因素进行结合起来可以共同提高聚类的效果.

FS-STC 算法中聚类过程的一个重要参数为相似度距离阈值 T,其决定该短文本是否加入当前类中心. 从图 4(b) 实验结果发现,本文 FS-STC 在相似度距离阈值 T 在 0.2 到 0.7 之间聚类效果比较稳定. 然而实验运行过程发现相似度距离阈值 T 越小算法运行时间越长,即算法的计算复杂度越大.

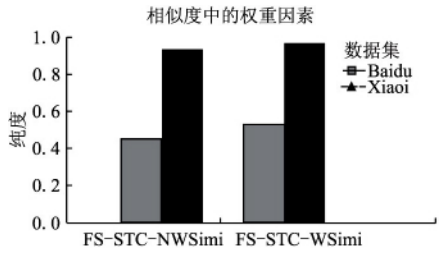

图 5　相似度计算的权重因素对 FS-STC 算法聚类纯度影响

Fig. 5　Weight factor of similarity effects on FS-STC´s clustering purity

图 5FS-STC-NWSimi 表示相似度计算方法中不考虑词语、短文本和类中心的权重信息因素,FS-STC-WSimi 表示相似度计算方法中考虑词语、短文本和类中心的权重信息因素. 图 5 实验结果表明,FS-STC-WSimi 的聚类效果较好. 由于 FS-STC-WSimi 在计算相似度的时候,考虑了词语、短文本和类中心的权重值信息,从而提高了权重较大的词语、短文本的影响作用,因此可以提升聚类效果.

6　结束语

本文针对智能对话系统产生的对话短文本提出一种基于字符串编辑距离相似度和词语语义相似度的聚类算法. 字符串编辑距离相似度的算法符合用户输入信息的过程,并对于包含输入错误噪音信息和特殊指令符号的对话短文本有较好的适应. 词语语义相似度可以在一定程度上解决同义词、语义鸿沟的问题,对于中文的短文本需要利用 HowNet 知识库的词语语义相似度,对于英文短文本需要利用 WordNet 知识库的词语语义相似度. 实验表明 FS-STC 算法在多种中英文短文本数据集上聚类纯度指标比所有的对比算法都有一定程度的提高,在 Xiaoi 对话类数据集上的提升效果尤其明显. FS-STC 算法在中英文的短文本测试数据集上聚类效果都比较好,说明 FS-STC 算法可以应用于中英文二种语言的对话短文本的聚类分析任务.

还有许多与本文研究相关的拓展性工作可以去做,比如寻找一种相似度融合系数自适应估计方法; 将本文的 FS-STC 算法应用到短文本数据集合的热点或主题挖掘等场景.

References:

[1] Mehran Sahami and Timothy D. Heilman. A web-based kernel function for measuring the similarity of short text snippets [C]. In Proceedings of the 15th International Conference on World Wide Web, WWW '06, ACM, 2006: 377-386.

[2] Quan Xiao-jun, Liu Gang, Lu Zhi, et al. Short text similarity based on probabilistic topics [J]. Knowledge and Information Systems, 2010, 25(3): 473-491.

[3] Banerjee Somnath, Ramanathan Krishnan, Gupta Ajay. Clustering short text using Wikipedia [C]. Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007), 2007: 787-788.

[4] Hu Xia, Sun Nan, Zhang Chao, et al. Exploiting internal and external semantics for the clustering of short texts using world knowledge [C]. Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM, 2009: 919-928.

[5] Chen Xin, Zhang Yu-qing, Cao Long, et al. An improved feature selection method for Chinese short texts clustering based on HowNet [M]. Computer Engineering and Networking, Springer International Publishing, 2014: 635-642.

[6] Yang Zhen, Duan Li-juan, Lai Ying-xu. Online public opinion hotspot detection and analysis based on short text clustering using string distance [J]. Journal of Beijing University of Technology, 2010, 36(5): 669-673.

[7] Ni Xing-liang, Quan Xiao-jun, Lu Zhi, et al. Short text clustering by finding core terms [J]. Knowledge and Information Systems, 2011, 27(3): 345-365.

[8] Islam Aminul, Inkpen Diana-zaiu. Semantic text similarity using corpus-based word similarity and string similarity [J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2008, 2(2): 10.

[9] Navarro, Gonzalo. A guided tour to approximate string matching [C]. ACM Computing Surveys (CSUR) 33, 2001, 1: 31-88.

[10] Liu Qun, Li Su-jian. Computational lexical semantic similarity based on HowNet [J]. Chinese Computational Linguistics, 2002, 7(2): 59-76.

[11] Wu Ben-bin, Jing Yang, Liang He. Chinese hownet-based multi-factor word similarity algorithm integrated of result modification [C]. Neural Information Processing, Springer Berlin Heidelberg, 2012.

[12] Zelikovitz, Sarah. Using background knowledge to improve text classification [C]. Diss. Rutgers, The State University of New Jersey, 2002.

[13] Rasmussen M, Karypis G. gcluto: An interactive clustering, visualization, and analysis system [J/OL]. UMN. http://citeseerx.ist.psu.edu/viewdoc/summary? doi=10.1.1.4.2246, 2011.

[14] Sean Owen, Robin Anil, Ted Dunning, et al. Mahout in action [M]. Manning Publications, 2011.

[15] Yu Yong, Xue Gui-rong, Han Ding-yi. Web data mining [M]. Beijing: Tsinghua University Press, 2009.

[16] Mihalcea R, Tarau P. TextRank: bringing order into texts [D]. UNT Digital Library. http://digital.library.unt.edu/ark:/67531/metadc30962/, 2014., 2004.

附中文参考文献:

[6] 杨　震,段立娟,赖英旭. 基于字符串相似性聚类的网络短文本舆情热点发现技术 [J]. 北京工业大学学报, 2010, 36(5): 669-673.

[10] 刘　群,李素建. 基于《知网》的词汇语义相似度计算 [J]. 中文计算语言学, 2002, 7(2): 59-76.

[15] 俞　勇,薛贵荣,韩定一. Web 数据挖掘 [M]. 北京: 清华大学出版社, 2009.