一种改进的社交媒体文本规范化方法

宋亚军[1]，于中华[2]

（1.四川大学，610000；2. 四川大学，610000）

摘　要：社交媒体文本书写不规范的特点，使得现有的自然语言处理工具直接应用于社交媒体文本时效果不甚理想，并且基于关键词的算法和应用也达不到预期的效果。因此，研究如何更好的对社交媒体文本进行规范化是非常有意义和价值的。本文基于社交媒体文本中非规范词与其规范形式具有相似上下文的假设，引入词嵌入模型更好地刻画上下文的相似性，提出了一种改进的基于图的社交媒体文本规范化方法，该方法是无监督并且语言无关的，可以处理不同类型语言的大规模无标注社交媒体文本。实验结果表明，该方法能够改进前人方法的不足，并且在与相关方法的对比实验中取得了最好的 F 值。

关键词：社交媒体；文本规范化；自然语言处理；词嵌入

中图分类号：TP391　　**文献标识码：**A

An improving Method for Social Media Text Normalization

Song Yajun[1], Yu Zhonghua[2]

(1. *Sichuan University, 610000* ; *2. Sichuan University, 610000*)

Abstract: The informal nature of social media text makes it difficult for many natural language processing tools to process social media text effective, and many keywords-based methods proposed on social media text works not as well as expected. Therefore, Social media text normalization is very meaningful and valuable. Based on the assumption that lexical variants occur in similar contexts to their standard forms, we proposed an improved graph-based social media text normalization method by introducing word embedding model to better capture the context similarity, the proposed method is unsupervised and language independent, and can be used to process large-scale different language social media text. Experimental results show that the proposed method improved the shortcomings of previous methods, and achieves state-of-the-art F-score.

Keywords: social media; text normalization; natural language process; word embedding

1 引言

微博、Twitter 等社交媒体网站每时每刻都在产生大量的短文本，这些用户实时产生的、数量众多的社交媒体短文本是非常具有研究和应用价值的，它们被广泛用于疾病检测、情感分析和事件发现等。

然而，和普通文本相比，社交媒体文本的书写通常不规范，包含很多符号、缩写、音节替代和俚语等，例如英文推特中经常将“tomorrow”写成“tmrw”或者“2morrow”，而中文微博中也有类似情况，比如“同学”写为“童鞋”、“稀饭”代替“喜欢”等。社交媒体文本书写不规范的特点，使得现有的自然语言处理工具直接应用于社交媒体文本时效果不甚理想[1]，比如词性标注器、依存分析工具等，而很多基于关键词的算法和应用也经常达不到预期的效果，比如情感分析、事件发现等。因此，研究如何更好的分析处理这种不规范的社交媒体文本是非常有意义和价值的。

解决这个问题通常有两种主流的方式：第一种，针对社交媒体语言和文本的特点，设计符合其特点的新的算法和工具[2,3]；另一种则是设计算法将社交媒体文本中不规范的用法转换为其规范的形式，即社交媒体文本规范化，例如将“2mo- rrow”和“童鞋”等不规范形式，分别转换为其规范形式“tomorrow” 和“同学”。社交媒体文本规范化通常作为预处理步骤，将非规范的社交媒体文本进行规范化处理后，交给现有的自然语言工具进行分析处理，而不用重新设计算法和工具。

虽然本文主要关注的是英文推特文本的规范化，但是本文提出的方法是语言无关的，可以方便地应用于其他语言的社交媒体文本，比如中文等。和其他很多最近的相关工作一样，我们只关注于一对一的规范化，即将一个非规范化词规范化为一个对应的规范形式，比如“tmrw”规范为“tomorrow”，而不考虑一对多或多对一的情况,比如“idk”规范为“I don't know”等。

基于“非规范词和它的规范形式通常出现在相似的上下文中”的假设，我们提出了一种改进的基于图的社交媒体文本规范化方法，该方法可以自动从大规模无标注的社交媒体文本中构建规范化词典，应用于社交媒体规范化。

文章接下来的内容组织如下：第 2 节讨论相关工作，第 3 节详细介绍本文提出的方法，第 4 节描述算法的实现细节和相关数据，第 5 节对实验结果进行分析和讨论，第 6 节是文章总结和未来工作展望。

2 相关工作

早期的文本规范化工作大多使用噪声信道模型。文献[4]首先将噪声信道模型应用于文本规范化工作，他们提出了一种新的基于字符串编辑的噪声信道模型，该模型对子串转换的概率建模，极大地提高了文本规范化的效果。文献[5]通过扩展噪声信道模型中的错误模型（将词之间的语音相似性加入错误模型），改进了上述方法，该方法通过学习规则来预测每一个字符的发音，并且预测依赖于词中的相邻其他字符。文献[6]针对 SMS 文本规范化，提出了一种基于隐马尔可夫模型的文本规范化方法，该方法也是一对一的规范化方法，通过构造常用缩写和非规范用法的词典，可以解决部分一对多的规范化（例如 “howz” 规范化为“how are”或者“aint” 规范化为 “are not”）。文献[7]引入无监督的噪声信道模型对文献[6]提出的模型进行了扩展，模型对常用缩写形式和各种不同的拼写错误类型进行了概率建模。

以上方法都存在一定的局限性,因为它们不考虑上下文的特性并且假设每个非规范词都具有唯一的规范化形式。在文本规范化任务中，相同的非规范词可能有不同的规范化形式(例如“2”可以规范化为“two”、“to”或“too”）,没有上下文信息是不可能正确地构建模型和消除歧义的。

还有一些研究人员使用统计机器翻译方法进行文本规范化，这种方法把问题形式化为将词的非规范形式翻译为规范化的形式。文献[8]中基于字符水平的短语对齐的 SMT 方法，将非规范的 SMS 文本转换为规范形式。文献[9]提出一种基于字符的 SMS 文本规范化方法，对新出现缩写的规范化非常有效。

但是基于统计机器翻译模型的规范化方法是有监督的方法，需要大量的标注数据。然而我们没有现成的标注数据可以使用，而创建标注数据也是非常困难的，尤其是在社交媒体文本中，其变化迅速的特点使得标注好的数据很快就会变得不适用[10].

最近提出的很多方法通过构建规范化词典用于文本规范化任务。例如，文献[11]首先训练分类器用于识别非规范词候选，然后使用词音相似度得到规范化候选，最后利用字面相似度和上下文特征找出最可能的规范化候选;文献[12]通过考察以用户为中心的信息包括用户所处地理位置、推特客户端的类型（比如网页端、移动端、第三方客户端等）等对推特书写习惯的影响，提出了一种针对不同人群的社交媒体文本规范化方法。

文献[13]提出了一个类似的方法，基于上下文相似性和字面相似性构建规范化词典进行推特文本的规范化，该方法使用词袋模型表示上下文分布，然后两两之间计算上下文分布相似度。

但是文献[13]提出的方法存在很多不足：首先，用词袋模型（bag-of-words）表示上下文分布容易产生高维稀疏问题，因为社交媒体文本中存在大量的不规范词、新词、实体名词

等；第二，该方法中使用两两计算相似度的方法选择候选，如果两个词之间没有共享的上下文，那么它们的相似度将会为 0，很难得到全局最优的规范化结果。

另一个非常相关的工作是由文献[14]提出的，针对文献[13]提出方法不能得到全局最优的规范化结果，提出了一种基于二部图随机游走的方法，该方法首先通过随机游走得到全局优化的基于上下文相似性的规范化候选列表，然后利用非规范词与规范词之间的字面相似度，对规范化候选列表进行重排序，得到最终的规范化结果加入到规范化词典中。

文献[14]同样也存在不足，因为这篇文章将每个词的上下文定义为前后各两个词组成的有序四元组，并且要求上下文中的每个词都为 IV 词，这就容易产生上下文稀疏性问题，特别是在社交媒体文本这种以书写不规范为特点的文本中，从算法评测结果的低召回率我们也可以看出这个问题。

文献[15]提出了一种新的无监督的社交媒体规范化方法，他们的方法中综合使用了字面特征、上下文特征和语法特征，其中上下文特征和语法特征是从构建好的词关联图中得到。但是他们的方法中使用了俚语词典和音译表等多种外部资源，并且非常依赖于使用的词性标注器的效果。

本文针对文献[14]中存在的上下文稀疏问题，通过引入词嵌入模型[16,17,18,19]缓解上下文稀疏问题，并且通过实验验证了改进方法的可行性和有效性。我们将在第 3 节中详细描述我们提出的改进方法。

3 方法

本文通过分析文献[14]工作的不足，引入词嵌入模型对其进行改进，提出了一种改进的基于二部图的社交媒体文本规范化方法，命名为 BiGraph+。

3.1 概述

本文的改进主要有两个方面：第一，取消了文献[14]中上下文词均为 IV 词的限制，因为社交媒体文本中非规范词的上下文也倾向于使用非规范词，因此要求上下文全为规范词是不合理的，从后面的实验结果可以看出，这个改进在基本保证精确率的前提下可以大幅地提高算法的召回率；第二，针对文献[14]提出的二部图中存在大量单独的上下文节点的情况（本文中将仅和一个词节点连接的上下文节点定义为 **single** 节点），通过使用词嵌入模型，找出与这些 **single** 节点语义相似的其他非单独上下文节点，在图中将它们连接起来，然后通过随机游走的方法得到全局优化的规范化结果。

3.2 二部图的构造

BiGraph+基于“非规范词与其对应的规范形式具有相似的上下文分布”的假设，每个词的上下文定义为前后固定窗口大小的词序列，比如给定五元组序列 *word1word2word3word4word5*，定义词 *word3* 的上下文为 *word1word2word4word5*。如果另一个词 *word3`* 是 *word3* 的规范化形式，那么 *word3* 和 *word3`* 将会有相同的上下文，这种上下文相似性可以用二部图进行表示。

介绍 Bigraph+方法前，我们先看文献[14]中定义的二部图，如图 3-1 左图所示，图中左边节点为上下文节点，右边节点为词节点，其中词节点可以是规范词也可以是非规范词，上下文节点为规范词序列，图中两个词节点直接或间接连接的上下文节点越多，两个词的上下文相似度越大。

但是这种上下文的定义形式容易产生上下文稀疏的问题，因为在上下文词序列中，只要有一个字母不同那么两个上下文就会当成不同的上下文，这使得很多上下文节点成为 **single** 节点，如图 3-1 左图中深色节点 C1 和 C4 所示。

因此针对文献[14]提出的二部图中存在大量单独的上下文节点的情况，通过使用词嵌入模型，找出与这些 single 节点语义相似的其他非单独上下文节点，在图中将它们连接起来，定义了一种新的二部图，如图 3-1 右图所示为 Bigraph+方法中定义的二部图，其中词节点

与上下文节点之间的连接权重为它们的共现次数，而上下文节点之间的连接权重为它们的语义相似度。

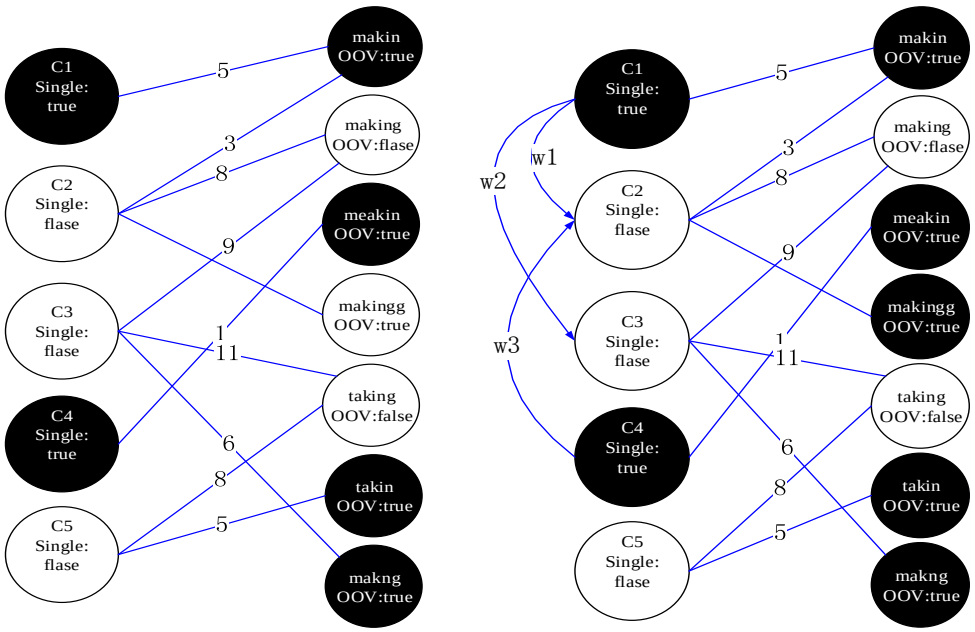

图 3-1：Bgraph(左)和 Bigraph+(右)示例图

BiGraph+方法的二部图的构造如算法3.1 所示。

算法 3.1: ConstructBiGraphplus(corpus)

Input: Ngram corpus

注释：W 为词节点，C 为上下文节点，E1 为词与上文节点之间的边，E2 为上下文节点之间的边

Output: (G(W, C,E1,E2))

for each (ngram,count) in (Ngrams,Count)

do

注释：添加词节点

word = GETCENTER(ngram)

if IS_NOISY (word)

W ← ADDWORD (word, false)

else

W ← ADDWORD (word,true)

注释：添加 word-context 边到 E1

E1 ← A DDCONTEXT (context, word, count)

注释：map 存储上下文对应的不同中心词的个数，用于判断 single 节点与非 single 节点

map.put(context, CURRENT+1)

注释：添加上下文节点

for each(context,count) in map

注释：count>1 时为非 single 节点

if count>1

C ← A DDCONTEXT (Context, false)

注释：count=1 时为 single 节点

else

C ← A DDCONTEXT (Context,true)

注释：添加 context-context 边到 E2

for each C_i in SingleContext

for each C_j in noSingleContext

w_{ij} = Similarity(Ci,Cj)

If wij>Threshold

E2 ← A DD (Ci, Cj, wij)

算法 3.1 中，我们使用 Aspell 词典(v0.60.0)判断一个词是规范词（IV）还是非规范词(OOV)。图中词节点与上下文节点之间边的权重定义为它们的共现次数，而上下文节点之间边的权重定义为它们的语义相似度，通过词嵌入模型进行计算得到。

3.3 规范化词典构造

构建好二部图之后，按照算法 3.2 的过程构造规范化词典。

算法 3.2: INDUCE LEXICON (G)

注释： W 为词节点，C 为上下文节点，E1 为词与上文节点之间的边，E2 为上下文节点之间的边

Input (G(W, C,E1,E2))

Output (Lexicon)

INIT ((Lexicon))

for each word in W in G(W, E)

do

If ISNOISY (word)

INIT (Rn)

注释: 进行 k 次随机游走

for i ← 0 to K

do

注释：Bigaph+与（Hassan 2013）方法的最大区别体现在这一随机游走过程中

Rn ← RANDOMWALK (word)

注释：计算平均的 hittingtime 并且归一化,作为上下文相似度存储于 Ln 中

Ln ← NORMALIZE (Rn)

注释: 计算字面相似度并且和上下文 相似度综合，对候选列表重排序

Ln ← SIMCOST(Ln)

注释：剪枝并将 topN 个规范化结果加入规范化词典

Ln ← PRUME(Ln)

LEXICAL ← ADDLEXICAL(Ln)

算法 3.2 的核心是随机游走过程 **Rn ← RANDOMWALK (word)**，算法中每个非规范词进行 K 轮随机游走，每轮随机游走按照这样一个过程进行：从给定的非规范词节点开始，按照状态转移概率游走至相邻的上下文节点，然后从上下文节点随机游走至另一个词节点，到达规范的词节点或者游走步数达到设定的阈值停止，当然本文的随机游走中允许上下文节点之间的跳转，这也是 Bigraph+方法与文献[14]的最大区别。其中随机游走状态转移概率（从一个节点 i 转移到另一个节点 j 的概率）p 为:

$$P_{ij} = \frac{w_{ij}}{\sum_k w_{ik}} \quad (1)$$

通过 K 轮随机游走我们可以得到一个规范化候选列表 Ln，其中每个候选规范词 n 与非规范词 m 都存在一个上下文相似度 L(m,n)，定义为 K 轮随机游走中从 m 随机游走至 n 的步数的平均值 H(m,n)归一化的结果，H(m,n)定义如公式(2)所示：

$$\mathrm{H}(\mathrm{m},\mathrm{n})=\sum_{\forall r} h_r(\mathrm{m},\mathrm{n})/\mathrm{R} \tag{2}$$

其中, $H_r(m,n)$是第 r 次随机游走的步数，R 为 K 轮随机游走中到达 n 的次数。

以图 3-2 为例，假设某轮随机游走的起点为非规范词节点“makin”，那么我们的随机游走路径可以为[“makin”->C1->C2->“making”],这时随机游走路径长度为 4（r=4）,也有可能为[“makin”->C2->“making”],这时路径长度为 3（r=3），2 轮随机游走后平均路径长度为 3.5，词 m 和 n 的上下文相似度 L(m，n)定义如公式(3)所示：

$$\mathrm{L}(\mathrm{m},\mathrm{n})=H(\mathrm{m},\mathrm{n})/\sum_{i} H(\mathrm{m},\mathrm{n}) \tag{3}$$

最终的得分为上下文相似度和字面相似度的加权和：

$$\mathrm{Score}(\mathrm{m},\mathrm{n})=\lambda L(\mathrm{m},\mathrm{n})+(1-\lambda)\,\mathrm{LexSim}(\mathrm{m},\mathrm{n}) \tag{4}$$

3.4 上下文语义相似度计算

本文采用词嵌入模型计算上下文的语义相似度，主要步骤如下：

一、训练词嵌入模型；

二、从训练好的词嵌入模型得到上下文中每个词的词向量；

三、连接所有词向量为一个上下文向量；

四、通过余弦夹角公式计算上下文之间的语义相似度：

$$\mathrm{S}emanticSim(\mathrm{C1},C2)=\frac{C1\bullet \mathrm{C2}}{\|C1\|\|C2\|} \tag{5}$$

其中C1和C2均为上下文的向量表示形式。

3.5 字面相似度计算

字面相似度的计算使用文献[20]提出的方法，该方法基于最大相同字串率和编辑距离，计算公式如下：

$$LexSim(\mathrm{m},\mathrm{n})=\frac{LCSR(\mathrm{m},\mathrm{n})}{ED(m,n)} \tag{6}$$

$$LCSR(\mathrm{m},\mathrm{n})=\frac{LCS(\mathrm{m},\mathrm{n})}{MaxLength(\mathrm{m},\mathrm{n})} \tag{7}$$

4 实现与数据

4.1 训练数据

我们从 Stanford’s 476 million Twitter Dataset[21] 中随机抽取了 1.5GB 的英文推特文本作为训练语料。文本的语言识别使用 langid.py Python library [22, 23]完成。

CMU Ark Tagger (v0.3.2)是一个专门针对社交媒体文本进行词性标注的工具，其在社交媒体文本上进行词性标注的准确率达到了 95% [2, 3]，这里我们使用 CMU Ark Tagger (v0.3.2)进行词汇单元化和词性标注。

词汇单元化和词性标注之后，我们将文本中词性被标注为提及(例如 @brendon),语篇标记（例如 RT)，URL，邮箱地址，表情符号和标点的词汇去除，用得到的数据构造二部图、训练词嵌入模型和语言模型。

4.2 词向量模型

word2vec[24,25]是 2013 年由 Google 研究人员提出的非常高效的基于神经网络的词嵌入模型，我们训练 word2vec 得到每个词的词向量表示，模型参数设定见 4.5 节,训练好词嵌入模型后直接应用于 BiGraph+中上下文的语义相似度计算。

4.3 语言模型

为了将构建的规范化词典用于测试，我们使用 SRILM 工具[26]在 1.5GB 英文推特文本上训练了一个 5-gram 语言模型，测试中我们使用维特比解码器，选择出符合当前上下文的最佳规范化候选，作为我们的规范化结果。

4.4 参数设置

实验中有多个参数需要人工设置，首先是 word2vec 模型中词向量的维数，通过实验发现该参数与语料库的大小有关系，在我们的实验中将其设置为 300，上下文语义相似度阈值设置为经验值 0.85，上下文窗口大小、随机游走模型中随机游走的最大步数和随机游走次数都按照文献[14]的实验分析进行设置，其中上下文窗口大小设置为 5，随机游走的最大步数设置为 4，随机游走次数设置为 100 次。

5 实验

5.1 测试数据集

我们使用 LexNorm1.1[11]作为算法评价的数据集. LexNorm1.1 包括 549 条英文推特，其中包含 1184 个人工标注的非规范词。这个数据集在文本规范化研究中广泛用于算法评价测试，这使得我们可以直接在这个数据集上和其他先前的方法进行比较[11,15].

5.2 实验结果与分析

5.2.1 评价方法

对于实验结果的评价,本文采用标准的精确率(P)、召回率(R)和 F 度量值作为评价标准。精确率（Precision）衡量的是在所有被算法规范化的词中，正确规范化的词所占的比率；召回率(Recall)衡量的是在所有需要被规范化的词中,算法进行了正确规范化的比率；F 度量值(FScore)是对上述两个指标的综合考虑。三个指标的计算公式如下:

$$Precision(\mathrm{P}) = \frac{\text{正确规范词数}}{\text{算法规范总词数}} \tag{8}$$

$$Recall(\mathrm{R}) = \frac{\text{正确规范词数}}{\text{需要规范总词数}} \tag{9}$$

$$Fscore(\mathrm{F}) = \frac{2PR}{P + R} \tag{10}$$

5.2.2 结果与分析

（1）规范化词典构造

规范化词典就是每个非规范词(OOV)与其规范形式的映射，相同的非规范词在不同的上下文中可能具有不同的规范形式，因此在构建规范化词典时保留 Top-N 个规范化候选，而不是仅仅保留一个，这样就可以避免传统方法中，每个非规范词都规范化为相同规范形式的缺点。因此构造规范化词典之前，一个很重要的步骤就是确定每个非规范词规范化候选的数目，以往的方法中往往都是根据经验设定，本文我们通过实验来设定，如图 5-1 所示。

图 5-1 中横坐标为规范化词典中每个非规范词的规范化候选的个数，纵坐标为百分比，图中三条曲线蓝色为精确率，红色为召回率，绿色为 F 值，从图中可以看出，当横坐标值达到一定值得时候，三条曲线的值都会趋于稳定，因此我们可以从稳定之后的值中选择一个 N，文章中为了得到更高的召回率选取 N=10。

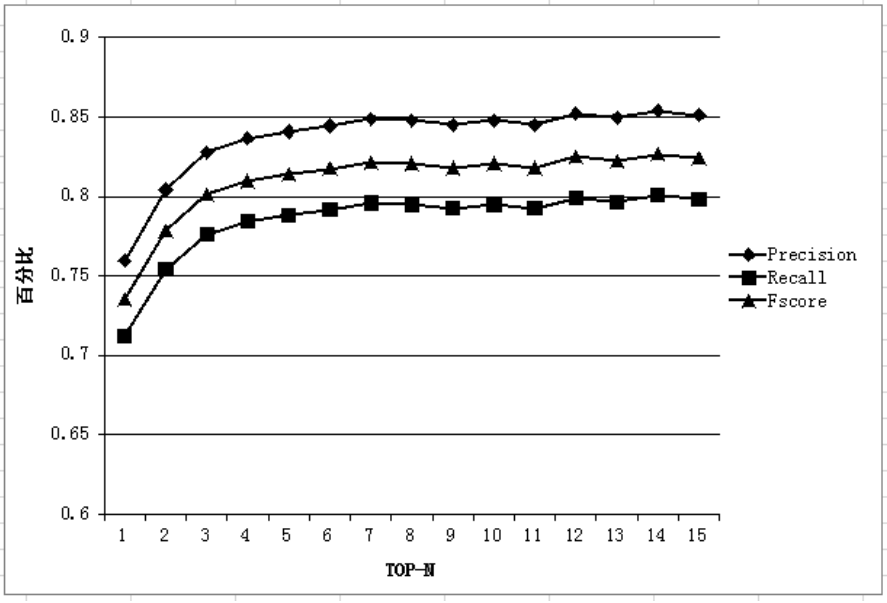

图 5-1:规范化词典的构造

（2）实验对比与分析

构造好规范化词典后，利用训练好的语言模型根据当前上下文从词典选择出每个非规范词最佳规范形式。表 1 是本文提出的方法与文献[13]和文献[14]中的方法的实验对比，表 2 中将本文提出的方法与其他相关工作进行了实验对比，从实验对比结果可以看出本文提出的方法在精确率和召回率都有很好的结果。

从表 1 中可以看出，与 Han(2012)方法相比，我们提出的 BiGraph+方法在精确率、召回率和 F 值三个指标上都有很大的提高，而和 Hassan(2013)方法相比，BiGraph+在精确率稍微降低的情况下，召回率和 F 值都大幅地提高了，从而证明了我们所提出方法的有效性。

从表 2 可以看出与 Han(2011)方法相比，我们所提出的方法无论在精确率、召回率,还是 F 值，效果都要更好，而与 Sönmez (2014)方法相比，虽然我们的精确率稍低，但是我们的召回率和 F 值都更高，而相比 Sönmez (2014)我们的方法不依赖于外部资源，并且 Sönmez (2014)方法非常依赖于其使用的词性标注器的效果。

表1：本文提出方法与文献[13]和文献[14]实验对比

方法	Precision	Recall	Fscore
Han (2012)[13]	70	17.90	28.50
Hassan(2013)[14]	**85.37**	56.4	69.93
BiGraph+	83.65	**78.87**	**81.19**

表2：本文提出方法与其他相关方法实验对比

方法	Precision	Recall	Fscore
Han(2011) [11]	75.30	75.30	75.30
Sönmez(2014)[15]	**85.87**	76.52	80.92
BiGraph+	83.65	**78.87**	**81.19**

（3）错误分析

这里，讨论一下我们在实验中发现的一些问题。首先试验中对精确率影响最大的是一些长度较短的非规范词，比如“dn't”的规范化形式可以是 “don't”、“doesn't” 或者“didn't”，并且它们出现的上下文也是类似的，这就产生了模糊性，从而导致错误；另外我们的方法对于新出现的非规范词也是无法处理的。

6 总结与展望

本文基于社交媒体文本中非规范词与其规范形式具有相似上下文的假设，引入词嵌入模型更好地刻画上下文的相似性，提出了一种改进的基于图的社交媒体文本规范化方法，我们提出的方法是无监督且语言无关的，能够方便地应用于其他语言。但是本文方法只能一对一地规范化，无法处理新出现的非规范词等，因此下一步工作将尝试将模型进行扩展和改进。

参考文献

[1] A.Ritter,C.Cherry,and B.Dolan. Unsupervised modeling of twitter conversations. Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010. pages 172–180.

[2] O.Owoputi, B.O'Connor,C.Dyer,et.al. Improved Part-of- Speech Tagging for Online Conversational Text with Word Clusters. Human Language Technologies : Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2013.pages 380–390.

[3] K.Gimpel, N.Schneider, B.O'Connor, et.al. Part-of-spee- ch Tagging for Twitter: Annotation, Features, and Experiments. Human Language Technologies :Proceedi-
ngs of the 49th Annual Meeting of the Association for Computational Linguistics,2011.pages 42–47.

[4] E. Brill and R.C. Moore. An improved error model for noisy channel spelling correction. Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Englewood Cliffs, NJ, USA,2000.pages 286-293.

[5] K. Toutanova and R.C. Moore. Pronunciation modeling for improved spelling correction. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL, Philadelphia, USA, 2002. pages 144 – 151.

[6] M. Choudhury, R. Saraf, V. Jain, et.al. Investigation and modeling of the structure of texting language. International Journal of Document Analysis and Recognition, vol. 10, 2007. Pages 157 - 174.

[7] P. Cook and S. Stevenson. An unsupervised model for text message normalization. Proceedings of the Workshop on Computational Approaches to Linguistic Creativity, , Boulder, USA. 2009. pages 71 – 78.

[8] A. Aw, M. Zhang and J. Xiao. A phrase-based statistical model for SMS text normalization. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, 2006. pages 33– 40

[9] D. Pennell and Y. Liu. 2011. A Character-Level Machine Translation Approach for Normalization of SMS Abbreviations. Fifth International Joint Conference on Natural Language Processing, pages 974–982.

[10] Y. Yang and J. Eisenstein. A Log-Linear Model for Unsupervised Text Normalization. Proceedings of the Empirical Methods on Natural Language Processing, 2013. pages 61–72

[11] B. Han and T. Baldwin. Lexical Normalization of Short Text Messages: Makn Sens a #Twitter. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,2011. Volume 1, pages 368–378.

[12] S. Gouws ,S. Metzler, C. Cai, and E. Hovy. Contextual Bearing on Linguistic Variation in Social Media. Proceedings of the Workshop on Languages in Social Media, 2011.pages 20–29.

[13] B. Han, P. Cook, and T. Baldwin. Automatically constructing a normalisation dictionary for microblogs. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012.pages 421–432.

[14] H. Hassan and A. Menezes. Social Text Normalization Using Contextual Graph Random Walks. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 2013. pages 1577–1586.

[15] C. Sönmez, A. Ozgür. A Graph-based Approach for Contextual Text Normalization. Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014.pages 313–324.

[16] Y. Bengio, R. Ducharme, Vincent, and C. Jauvin. A neural probabilistic language model. The Journal of Machine Learning Research, 2003, 3: pages 1137-1155.

[17] A. Mnih and G.E. Hinton. A scalable hierarchical distributed language model. Advances in neural information processing systems, 2009. 21, pages 1081–1088.

[18] T. Mikolov, A. Deoras, D. Povey , et al. Strategies for Training Large Scale Neural Network Language Models. Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on. IEEE, 2011.pages 196-201.

[19] T. Mikolov, I. Sutskever, et al. Exploiting similarities among languages for machine translation. CoRR, abs/1309.4168, 2013b.

[20] D. Contractor and T. Faruquie and V. Subramaniam . Unsupervised cleansing of noisy text. Proceedings of the 23rd International Conference on Computational Linguistics, 2010. pages 189 - 196.

[21] J. Yang and J. Leskovec. Patterns of Temporal Variation in Online Media. Proceedings of the Forth International Conference on Web Search and Web Data Mining, 2011.pages 177–186.

[22] M. Lui and T. Baldwin. Langid.Py: An Off-the-shelf Language Identification Tool. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2012.pages 25–30.

[23] T. Baldwin and M. Lui. Language Identification: The Long and the Short of the Matter. Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010.pages 229–237.

[24] T. Mikolov, I. Sutskever, K. Chen, et al. Distributed representations of words and phrases and their compositionality[C] Advances in Neural Information Processing Systems.2013.pages 3111-3119.

[25] Q. Le, T. Mikolov.2014.Distributed Representations of Sentences and Documents. Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014.pages 1188-1196.

[26] A. Stolcke. 2002. SRILM-an extensible language mode- ling toolkit. INTERSPEECH. 2002.pages 901-904.

作者简介：作者一

宋亚军（1990-）， 男，硕士研究生

主要研究方向为自然语言处理（主要为社交媒体文本处理，包括社交媒体文本规范化及其应用）与文本数据挖掘。

Email: songyajun90@163.com

作者简介：作者二

于中华（1967-），男，副教授，硕士研究生导师

主要从事自然语言处理（自然语言理解、自然语言处理在 Internet 上的应用、生物医学文献信息抽取）和中药数据挖掘的研究工作。在国内外核心刊物及 SCI、EI 检索发表学术论文 60 余篇。

Email: yuzhonghua@scu.edu.cn