基于情感词典的情感分析

崔健聪

中国科学院信息工程研究所

202128018670043 jcqueue@gmail.com

1 主要贡献

张三、李四同学与我共同完成作业题目 2（情感分析系统的实现），其中我主要负责：

- 基于情感词典的情感分析的领域调研报告
- 基于情感词典的情感分析的实验设计与实现
- 基于情感词典的可解释性情感分析系统的设计与实现

2 情感分析领域调研报告

随着互联网的普及以及社交网络的成熟，越来越多的评价性文本涌入到互联网中，这也极大的影响了人们的决策方式 [Rao et al.(2014)]：比如，人们现在通常都会在买一些物品或服务前，先去了解他人对这一事物的意见或评价，并再消费后发表自己的评价或看法 [Das and Chen(2001), Hardeniya and Borikar(2016)]。这些评价文本中蕴含着人们对待各种事物的态度或看法，对企业或政府了解公众舆论存在重大的研究价值与现实意义 [Van Atteveldt et al.(2021), Pang and Lee(2004)]。因此能够高效且精确的从给定的评论性文本中识别出作者对于目标实体的情感极性表达，成为了当今热点的研究方向 [Prabowo and Thelwall(2009)]。

2.1 概述

首先文本情感分析以文本为基础，这些文本大都以评论性的非结构化数据为主体，分为文档级，锻炼级，句子级，词语级；其次采用不同的方法对文本进行情感分析，主要有：基于情感词典的情感分析、基于机器学习的情感分析、基于神经网络的情感分析。这三种不同的方法会将目标文本作为输入，随后输出预测的文本情感极性。

2.2 基于情感词典的情感分析领域调研

基于情感词典的情感分析其核心思想就是将文本情感分析任务转化为文本中多个词的情感分析任务，通过预先构造好的积极情感词典与消极情感词典，完成对目标词汇集合的情感极性标注，最后采用加权求和等方法求解出文本的情感极性。

早在 1997 年 Hatzivassiloglou & McKeown [Hatzivassiloglou and McKeown(1997)] 就关注了形容词的极性分析，通过对数线性模型与聚类分析重点划分了形容词的词性，并指出看似同义的词可能表达着不同的极性, 如 simple 与 simplistic, simple 就是积极的简单，simplistic 常常用于表示消极的过度简单, 比如：Her explanation of the complex political problem was simplistic, 它把这个复杂的问题解释的过于简单了；此外作者还表示，在划分极性时也需要注意连词的影响, 如 and, but 等。最终作者的情感极性划分达到了高于 90% 的准确率; Das 和 Chen [Das and Chen(2001)] 则很早的就将情感词典应用到股市之中，利用情感词典对股票留言板上的评论文本进行情绪分类，以决定是否买入或卖出股票。但是，当时的应用面还太窄，一方面词性方面考虑不周，限制了情感词典的大小并忽略了常见的否定型情感表达；另一方面情感词典的构建还需要大量的人力，并且常与某一领域相关，无法进行很好的扩展。

随后各种各样的情感分类应用进行了各方面的创新与改进: Turney [Turney(2002)] 在词典引入了短语以及更多的词性, 如副词、动词、连词等, 丰富了词典的表达并且为每个情感词分配了不一样的权值：通过计算文档短语与“excellent”和“poor”两个词之间的互信息，得到情感词典中短语的语义方向；最后采用文档中所匹配短语的平均语义方向进行情感分类任务, 得到了平均 74% 的准确率; Huettner and Subasic [Maas et al.(2011)] 则关注于细粒度的情感分类, 在 4000 个词表中生成了面向多领域, 多情感的情感词典, 并计算了每个词与情感之间的相关程度。

随着研究的进展，情感分析的应用面越来越广, 刘兵等人 [Hu and Liu(2004)] 针对网络中数量庞大的评论信息进行了产品特征的挖掘、评论的情感分类以及产品评论的总结, 将应用范围进一步扩大。其中针对评论文本的情感二分类任务, 他们只关注于文本中的形容词, 并借助 WordNet [Miller(1995), Miller et al.(1990)] 中所列出的每个单词的同义词和反义词, 实现通过少量的标注样本构建了庞大的领域无关的情感词典, 并且支持人工添加种子形容词, 实现领域词汇的快速融合。

在 05 年左右, 随着机器学习的发展, 很多研究者都将情感分类任务与机器学习模型相结合, AC König 等人 [König and Brill(2006)] 将人工的情感词典或模式的提取与机器学习相结合, 利用文本文档处理理论作出合理假设:

1. 通常只需要知道文本文档中的一小部分就足以作出正确的分类决策 [Pang and Lee(2004)];

2. 人类擅长提取文本中潜在的模式或分类规则 [Liu et al.(2004), Raghavan et al.(2005)];

因此一方面大大降低了人共标注的成本, 另一方面在当时深度学习还未兴起的情况下, 利用人类的推理出的规则辅助机器学习达到更好的分类准确率。

然而到了近几年, 即使在深度学习的热潮下, 基于情感词典的情感分析依的发展也逐渐转移到情感词典的发展上面, 最后的分类任务则更多的与深度学习相结合。饶洋辉等人 [Rao et al.(2014)] 将研究中心放在了情感词典的细粒度以及规模上。通过最大似然估计以及 Jensen 不等式构造情感词典的生成模型, 将每一个情感词与多个细粒度情感相映射, 并利用统计概率计算出权值来表示词与情感的相互关系；因此这样生成的词典

具有语言无关性并且规模理论上可以无限扩充, 但是前提需要有细粒度的用户评分数据。在这一背景下也产生了许多著名的情感词典，将在后续小节进行总结。

当然也出现了许多与深度学习相结合的工作, 并大都应用到了对原始著名情感词典的扩充上面: Alshari 等人 [Alshari et al.(2018)] 指出 SentiWordNet 忽略了文本中的大量的情感无关词汇, 并认为如果能够学习到情感无关词汇中的潜在情感表达, 对情感分析任务将有很大帮助。因此他们提出了 Senti2Vec 模型, 旨在利用 Word2Vec [Mikolov et al.(2013)] 方法扩充 SentiWordNet 的情感词表, 即将大量的情感无关词汇与 SentiWordNet 中的原始情感词汇映射到高维空间中, 然后利用 Frobenius 距离公式，分配情感权重。实验证明, 情感词典被扩充后，大大的提升了情感分析的表现; Zhang 等人 [Zhang et al.(2018)] 则将应用领域迁移到了以微博文本为主的中文情感分析中, 并利用大量的微博数据，构造了丰富类别的中文情感词典，包括: 基本情感词典，程度副词词典，否定词词典，网络词词典，表情词典，关联词词典，并赋予了它们相应的权重。最后利用中文分词系统（ICTCLAS）匹配词典中的词汇完成情感分析任务；Xu 等人 [Xu et al.(2019)] 针对现有中文情感词典的缺陷:

1. 领域单一性;
2. 多义词表达问题;

进行了情感词典的扩充，花费了一定的人力去调研收集如酒店、数码、水果、服装、洗发水等领域的领域情感词以及多义情感词。最后利用机器学习方法解决文本中多义词的释义问题，并进行了实验，证明了所构建扩展情感词典的有效性。

2.3 情感词典存在的缺陷

- 情感词典所包含的情感词汇通常具有领域相关性, 并且规模有限;
- 情感词典无法解决情感多义的线性;
- 单类情感词典无法处理否定句的形式;
- 引文文本的多样性, 设计俚语和形态变化, 因此情感词典的匹配度较低;

2.4 常用情感词典

- SentiWordNet(SWN): 是用于确定文本极性最常用的情感词典 [Alshari et al.(2018)]。其基于 WordNet，令 WordNet 中的每个同义词集都与正面、负面和中性的情感分数相关联。其 3.0 版本共有 117,659 条记录，每条记录由 POS(词性), ID, PosScore(正向情感权重), NegScore(负向情感权重), SynsetTerms(同义词词条名), Gloss(注释) 组成。它不仅被应用于传统的统计型基于情感词典的情感分析任务, 因其具有独特的权重, 还能够很好的与机器学习中相关模型结合。
- SentiTFIDF [Ghag and Shah(2014)]: SentiTFIDF 基于词频 (term frequency, TF) 与逆文本频率 (inverse document frequence, IDF) 对正负文档中的词进行统计计算, 并分配极性与权值。其认为如果

一个词在正文档集中的 if-idf 值大于其在负文档中的 tf-idf 值, 则被认为是积极情感词, 反之亦然。但是如果出现在正文档和负文档的 tf-idf 值相同, 则将该词忽略。

- Dictionary from [Hu and Liu(2004)], 该词典是由作者不断更新并手动整理的, 仅仅是对情感词的极性划分, 并没有相关的权重分布。但是在实验中, 发现其效果并不亚于上述的 SentiWordNet 与 SentiTFIDF。

3 实验设计与实现

本实验采用的是 IMDB 影评数据集 [Maas et al.(2011)], 共含有 5 万条 IMDB 影评, 本系统将用其展示基于情感词典的情绪分类的效果。

3.1 实验环境

- Python: 3.9.7
- nltk: 3.6.5
- Pyqt5: 5.15.4

实验代码需要在上述环境下通过 `python main.py` 执行, 情感分析系统可以直接通过双击 `JCSA System.exe` 执行

3.2 数据集分析

IMDB 数据集是专门用于情绪分析的数据集，所提供的评论情绪是二元的。其中训练集测试集均有 25,000 个评论, 并且评论样本中的积极评论的个数与消极评论的个数一致，是一个非常平衡的数据集。

3.2.1 数量分析

将给定训练集中的数据按照 9:1 的比例划分成训练集与验证集, 用于后续的基于深度学习的相关实验。最终得到了 22,500 个训练样本, 2,500 个验证样本, 25,000 个测试样本。如图 1所示, 其中 label 表示评论样本的标签, 0 表示为消极的评论, 1 表示为积极的评论样本。

3.2.2 长度分析

对数据集中评论样本的长度分布进行了统计。直观的长度分布如图 2 所示, 图 3 尝试进行拟合后, 可以明显的看出训练集与测试机集合的评论样本长度分布基本一致。最后图 4通过样本长度的散点分布查看噪音数据的分布情况：因为刚刚的柱状统计粒度较细，难以发现离群的数据长度信息，因此采用长度三点分布以对样本长度的分布密度进行一定的认知。

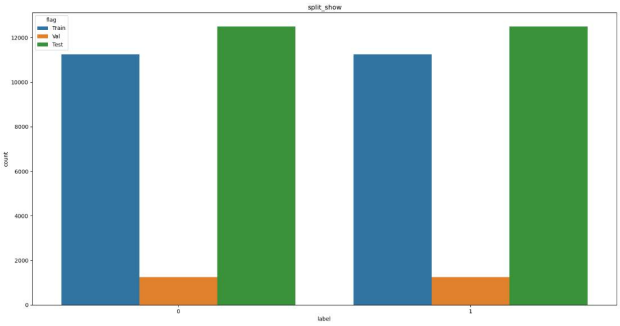


Figure 1: 训练, 验证与测试集的数量分布

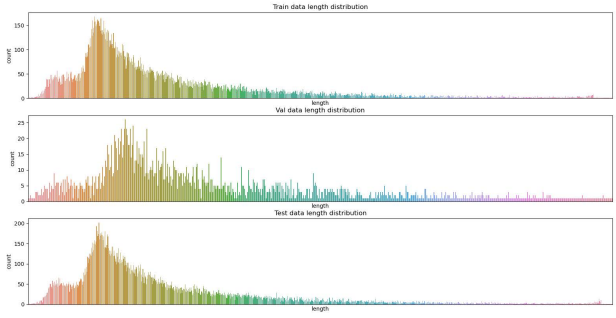


Figure 2: 评论样本的长度分布。

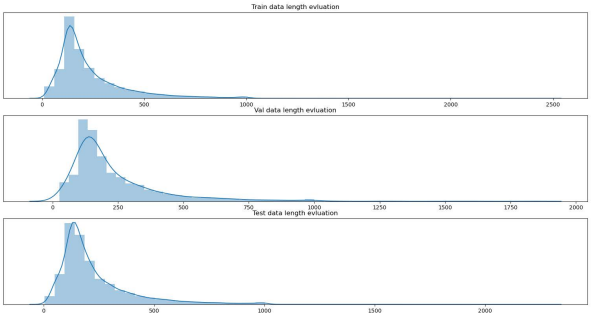


Figure 3: 评论样本的长度拟合。

3.2.3 词数统计

在进行必要的数据预处理前，我想先大致的看一下 IMDB 数据集的词表长度，实验中采用的是 Jieba 分词，但是针对英文分词，本质就相当于根据空格进行分词。最终得到的结果如表 1所示：

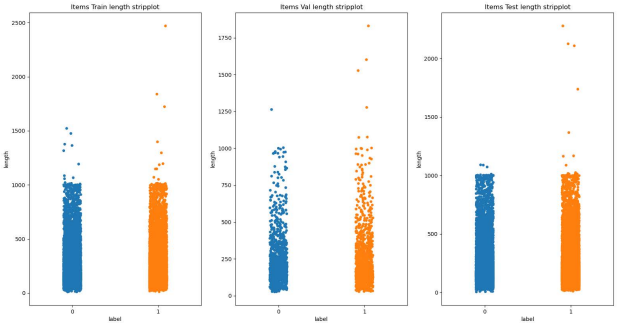


Figure 4: 评论样本长度的散点分布。

Table 1: 数据集词数统计

Data Type	Word Count
Train	89887
Validation	33370
Train + Validation	94083
Test	93533

3.2.4 词云分布

最后按照 SentiWordNet 中的情感词汇的词性类型: 名词，形容词，动词和副词，我们分别针对积极情感的样本与消极情感的样本制作了高频词的词云 (如图 5, 图 6和图 7所示)，以检验基于情感词典进行情感分析的有效性。

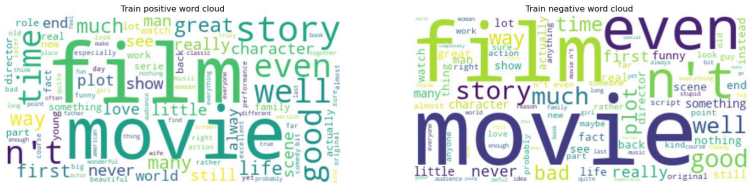


Figure 5: 训练集合的词云图。

Figure 6: 验证集和的词云图。

从上述的词云结果来看:

- 首先切分后训练集，验证机，测试机中的词汇分布并没有较为明显的不同;
- 其次，让我有些怀疑名词作为情感词汇的有效性，可以看出积极和消极样本的词云中均有高频名词,

Figure 7: 测试集和的词云图。

film 与 movie 因此这两个高频词应当不具备情感区分性, 当然这可以在情感词典中通过较为中性的权值来体现;

- 最后，也是最有意思的一幕，就是消极样本生成的词云中，有着我们所熟知的积极情感词汇，如 good, great 等, 但是有了前文的理论分析, 这一点也就很好解释了：评论文本中否定句是其常见表达，因此在消极样本中，可能会有较多的 not good 形式的情感表达;

3.3 词典分析

本次实验总共涉及四大词典: SentiWordNet [Alshari et al.(2018)], Dict_Liub(Dictionary from [Hu and Liu(2004)]), Dict_IMDB(IMDB Dictionary from [nproellochs(2018)]), SentiTFIDF [Ghag and Shah(2014)]。

3.3.1 词典数据分析

下面简单的对词典中积极情感词数与消极情感词数进行统计, 如图 8 所示。

- SentiWordNet 中共存在 117,659 条记录, 但是有较多的记录其正极性分数与负极性分数的分值均为 0, 将其舍去后, 共有 13,128 条正极性记录, 14,726 条负极性记录;
- Dict_Liub 中共有 2,003 条正极性记录, 4,780 条负极性记录;
- Dict_IMDB 中共有 294 条正极性记录, 255 条负极性记录;
- SentiTFIDF for IMDB 是对所有训练集和进行分词后筛选的词数, 其中共有 12,666 条正极性记录, 10,635 条负极性记录;

3.3.2 词典应用分析

下面将依次介绍其在实验中如何使用的:

- SentiWordNet: 由 Python 的 nltk 包封装完成, 其中一个词会对应多种词性, 一个词性会对应多个词义, 一个词义会对应两个得分: 积极情感得分与消极情感得分。在实验中我们将以句子为最小单元进行词性的标注，缩小 SentiWordNet 的索引范围，然后对目标词性的多个词义进行迭代降权求和的方式得到该词的情感贡献, 如下方代码 1所示。

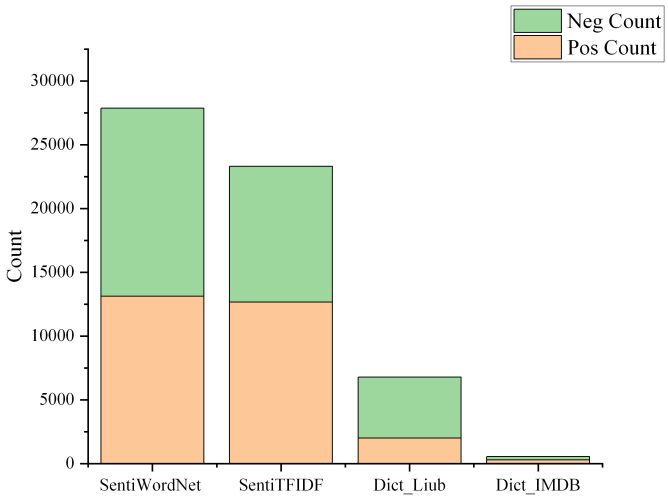


Figure 8: 各个词典的极性分布统计。

Listing 1: SentiWordNet 情感词典针对情感词计算情感得分的代码。

```
score = 0
for i, (pos_score, neg_score) in enumerate(sentiWordNet(word)):
    score += (pos_score - neg_score) * (1 / (2 ** i))
return score
```

- Dict_Liub: 目前能搜集到的网上公开的最新版本为 (04) 年的版本, 初步应用其的目的是为了与 SentiWordNet 作对比, 查看同样是基于统计的面向多领域的词典随研究发展的对比效果。
- Dict_IMDB: 这也是在网上找到的公开词典, 主要是用于对比实验, 判断面向数据集所在领域词典的效果, 是否比领域无关性词典的效果出色。
- SentiTFIDF: 一个简单容易实现的词典构造方法, 本文将使用 SentiTFIDF 通过训练集与验证机构造 IMDB 的专属词典, 去除了专有名词并根据实验结果只考虑三类词, 即名词, 形容词与副词。

3.4 情感分析

虽然经过初步的数据分析表明 IMDB 是一个质量非常高的情感分析数据集, 但是我们还是需要对其中的用户表达进行一定的统一化处理后进行情感词典的匹配与情感分类工作，具体过程介绍如下。

3.4.1 数据清洗

这一步是根据数据集的特性来应用的, 对 IMDB 数据集进行抽样查看, 发现其中出现了 HTML 标签, 标点的重复, 超链接等, 这些内容与所用的情感分析方法无关, 是必要的数据清洗。具体的清洗方法如代码 2 所示。

Listing 2: 评论文本中数据清洗代码。

```
def jc_basic_clean(self, opinion):
    opinion = opinion.replace("<br />", "")
    opinion = opinion.replace('"', ' ')

    # 替换网址为 <URL>
    url_regex = r'(https?:\/\/(?:www\.|(?!www))[a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.
    [^\s]{2,}|www\.[a-zA-Z0-9][a-zA-Z0-9-]+[a-zA-Z0-9]\.[^\s]{2,}
    |https?:\/\/(?:www\.|(?!www))[a-zA-Z0-9]+\.[^\s]{2,}|www\.[a-zA-Z0-9]+\.[^\s]{2,})'
    opinion = re.sub(url_regex, "<URL>", opinion)

    # 去除重复的标点，如 `!!!`
    opinion = re.sub(r'([!?,;])\1+', r'\1', opinion)
    opinion = re.sub(r'\.{3,}', r'...', opinion)
    return opinion
```

3.4.2 分句

后续的预处理操作单元当以句为单位, 因此先进行分句操作同时提取句末的标点符号, 用于后续的对比分析。具体的分句方法如代码 3 所示。

Listing 3: 评论文本中分句代码。

```
def jc_sent_tokenize(self, opinion):
    opinion_sents = sent_tokenize(opinion)
    params_list = []

    for opinion_sent in opinion_sents:
        if(len(opinion_sent.strip()) == 0): continue
        endmark = opinion_sent[-1]
        if(endmark in self._support_endmarks):
            params_list.append([endmark, ])
        else:
            params_list.append(['.', ])

    return opinion_sents, params_list
```

3.4.3 缩写词处理

在评论文本中存在着大量的缩写词, 我觉得即使情感词典中将缩写词纳入情感词典也应当将缩写词展开, 因为部分缩写词中包含着否定, 如果不处理, 将对后续的否定词作用的分析产生影响, 如 I don't 将转换为 I do not; 2moro 将变为 tomorrow 等。经过上一阶段的领域了解，本实验采用维基百科公开的缩写词库 abbreviations.json [Motamedi(2018)] 进行转换。具体转换方式如代码 4 所示。

Listing 4: 评论文本中的缩写词处理代码。

```
def jc_replace_abbreviations(self, opinion_sents):
    if(self.replace_abbreviation is not True):
        return opinion_sents

    opinion = " @@ ".join(opinion_sents)

    opinion_words = opinion.split(" ")
    for index in range(len(opinion_words)):
        word = opinion_words[index]
        if(not len(word)): continue
        first_upper = True if word[0].isupper() else False

        if(word.lower() in self._support_abbreviations_map.keys()):
            target = self._support_abbreviations_map[word.lower()]
            if(first_upper):
                target = target[0].upper() + target[1:]
            opinion_words[index] = target
    opinion = " ".join(opinion_words)
    opinion_sents = opinion.split(" @@ ")
    return opinion_sents
```

3.4.4 命名实体识别

在情感分析中, 特别是针对电影等的情感评论中, 常常会进行电影名, 人名等专有名词中的引用, 这类词中可能包含有情感词典中的词汇, 如 The Pursuit of Happyness, 这时我们需要将其识别出来并改写为一个实体单词, 使其不影响后续的情感分析任务。具体识别方法如代码 5 所示。主要的原理就是: 检查是否有多个连续首字母大写的单词出现, 当然连词等虚词可以连接这些首字母大写的单词, 如果存在则用下划线 (_) 将其拼接为一个单词, 就能忽略其对情感词典的影响了。算法的效果如图 9 所示。

Listing 5: 评论文本中的命名实体识别的处理代码。

```
def jc_replace_propernouns(self, opinion_sents):
```

```
opinion_tokens_list = []
for opinion_sent in opinion_sents:
    # opinion_sent = re.sub(r'[^\w]', '', opinion_sent)
    opinion_tokens = []
    # tokens = opinion_sent.split()
    tokens = word_tokenize(opinion_sent)

    if(len(tokens) <= 1): # 句子里就一个词
        opinion_tokens = [tokens[0], ]
    else:
        start_index = 0
        while start_index < len(tokens):
            token = tokens[start_index]

            if(len(token) == 0):
              start_index += 1
              continue
            else: token = token.strip()

            if(token[0].isupper()):
                # 尝试寻找专有名词
                target_index = start_index + 1
                union_index = start_index # 虚词能够向下寻找，但是只有虚词不能认定为专有名词
                while target_index < len(tokens):
                    target_token = tokens[target_index]
                    if(len(target_token) != 0):
                        if(target_token not in self._support_function_words and
                            target_token[0].islower()):
                           break
                    target_index += 1
                    if(len(target_token) == 0 or target_token[0].isupper()): union_index =
                        target_index - 1

                if(start_index < union_index):
                    opinion_tokens.append("_".join(tokens[start_index:union_index + 1]))
                    start_index = union_index + 1
                    continue
            opinion_tokens.append(token)
```

```
            start_index += 1
        opinion_tokens_list.append(opinion_tokens)
    return opinion_tokens_list
```

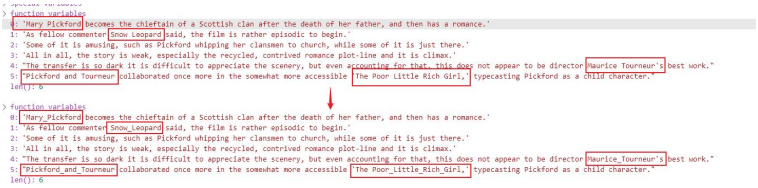


Figure 9: 实验中命名实体识别的效果图。

3.4.5 词形统一

英文中的动词有过去式, 过去分词, 进行时或动名词等多种形式; 名词有单数复数; 形容词有比较级, 最高级等; 在这里如果发现情感词典中无法匹配目标词形时, 则会将这些不同的词形还原为基本的词形, 再次尝试进行匹配。

词形的统一是一个可选的处理过程, 也将在后续实验部分用消融实验的方式验证其的有效性, 但是词形小写化的处理是必须的, 这也是为了增加情感词典的匹配率。词形统一的具体方法如代码 6 所示, 示例的统一效果如图 10 所示。

Listing 6: 评论文本中词形统一的处理代码。

```
def jc_union_morphology(self, opinion_tokens_list):

    wnl = WordNetLemmatizer()
    for tokens in opinion_tokens_list:
        pos_tags = pos_tag(tokens)
        for token_index in range(len(tokens)):
            token = tokens[token_index]
            jc_pos_tag = pos_tags[token_index]

            token = token.lower()

            if("_" in token): continue # 专有名词不需要词形归一化也不需要词性标注

            token_tag = get_wordnet_pos(token, jc_pos_tag)

            if(token_tag is None): # 未知词性
                tokens[token_index] = token
                continue
```

```
        if(self.union_morphology is False): # 不进行词性归一化只进行词性标注
            union_token = token
        else:
            union_token = wnl.lemmatize(token, token_tag)
        tokens[token_index] = union_token + self.tag_split + token_tag
    return opinion_tokens_list
```

Figure 10: 实验中词形统一的效果图。

3.4.6 情感无关词处理

句子单元在上一步转为了词单元, 因此可以单独的处理与情感无关的词汇，如分词后单独分出的标点，公认的停用词表等。

但是我发先停用词中存在较多的含否定形式的缩写, 这些缩写在之前的步骤中被展开了, 因此去除停用词也变为了可选, 并且其对最终的结果影响应当是有限的。情感无关此处理的具体方法如代码 7 所示。

Listing 7: 评论文本中情感无关词的处理代码。

```
def jc_remove_punctuations_and_stops(self, opinion_tokens_list):
    english_punctuations = [',', '.', ':', ';', '?', '(', ')', '[', ']', '&', '!', '*', '@', '#',
        '$', '%', '\'']
    stops = list(set(stopwords.words("english"))) if self.remove_stops else []
    remove_set = english_punctuations + stops

    ret_opinion_tokens_list = []
    for tokens_list in opinion_tokens_list:
        tokens_list = [ token for token in tokens_list if token not in remove_set or token in
            ['not'] ]
        ret_opinion_tokens_list.append(tokens_list)

    return ret_opinion_tokens_list
```

3.4.7 结尾标点的影响

这是一个可选的影响最终评分的规则, 认为最终标点作为! 能加重情感, 能将本句的情感扩大。最终标点作为? 能反转情感, 将本句的情感反转。这一点并没有在相关文献中查看到，是在相关博文中看到的，只是用于作个对比试验进行效用的分析。将结尾标点引入情感分析的方法如代码 8 所示。

Listing 8: 情感分析中考量结尾标点影响的代码。

```
scores_list = self.jc_get_scores_list(opinion_tokens_list, params_list)

sent_scores = []
for scores, param in zip(scores_list, params_list):
    sent_score = sum(scores)
    if(self.consider_endmarks):
        if('!' in param): sent_score = sent_score * 2
        elif('?' in param): sent_score = sent_score * -2
```

3.4.8 否定词的影响

否定词的作用在多篇文献中都又强调，因此否定词的影响也被选择用于后续的对比分析。在这里我们将检测句子中是否存在否定词，如果存在则将句子的极性得分进行反转。将否定词引入情感分析的方法如代码 9 所示。

Listing 9: 情感分析中引入否定词影响的代码。

```
def jc_get_scores_list(self, opinion_tokens_list, params_list):

    scores_list = []

    for tokens_list, param in zip(opinion_tokens_list, params_list):
        scores = []
        for token_index, token in enumerate(tokens_list):
            score = self.jc_get_token_polarity_score(token)
            scores.append(score)
        if(self.consider_negative_word):
            reverse_flag = self.jc_check_negative_word_and_update(tokens_list)
            if(reverse_flag):
                scores = [ score * -1 for score in scores ]
        scores_list.append(scores)
    return scores_list

def jc_check_negative_word_and_update(self, tokens_list):
```

```
    for token in tokens_list:
        if(self.tag_split in token): token = token.split(self.tag_split)[0]
        if(token in self._support_negative_words): return True
    return False
```

3.4.9 首尾句的作用

在领域调研中, 发现有多篇文献强调首尾句或文本中的部分句即可代表整个文本的情感, 因此简单的, 我采用两个具有有效情感的首尾句进行最终的情感分类。

当然这也是一个可选的用于对比分析的机制。其具体的实施方法如代码 10 所示。

Listing 10: 情感分析中只考虑有效首尾句的代码。

```
# 在得到所有句子的情感得分列表后
if(self.partial_sent):
    sent_scores = [score for score in sent_scores if score != 0]
    if(len(sent_scores) == 0):
        sent_scores = [0., ]
    total = sent_scores[0] + sent_scores[-1]
else:
    total = sum(sent_scores)

predict = 1 if total > 0. else 0
```

3.4.10 词性的影响

在领域调研中, 多篇基于情感词典的分析都只采用形容词的形式进行情感分析，因此有人指出大量非形容词也可以用于情感分析工作，应当进行情感词典的扩充，引入更多的词性。

正好 SentiWordNet 中具有四类词性, 分别是名词，动词，形容词和副词。因此我们对 SentiWordNet 情感词典, 额外进行了对比实验, 用于验证词性的多样性对情感分析任务的影响。其余词典没有对词性进行标注, 而一个词又具有多个词性, 因此并没有展开此对比工作。将词形引入到最终的情感分类任务的具体方法如代码 11 所示。

Listing 11: 情感分类中分析词性影响的代码。

```
def jc_get_token_polarity_score_swn(self, token):
    score = 0.
    # 获取单词与词性
    if(self.tag_split in token):
        token, tag = token.split(self.tag_split)
```

```
        if(tag not in self.part_of_speech):
            tag = None
    else:
        tag = None

    if(tag is None): return 0.

    items = swn.senti_synsets(token, tag)
    # 一个词性多种意思
    for index, item in enumerate(items, 0):
        score += (item.pos_score() - item.neg_score()) * (1 / (2 ** index))

    return score
```

3.5 实验结果的对比与讨论

3.5.1 各类情感词典的对比分析

首先针对每一个情感词典, 我们都进行了如下的对比实验:

1. Basic: 按上述过程处理后基础的实验结果;
2. Endmarks: 引入句末标点后的实验结果;
3. No_Morphology: 不进行词形统一后的实验结果;
4. No_Stopwords: 不删除停用词的实验结果;
5. Negative: 引入否定词后的实验结果;
6. Partial: 只考虑有效首尾句后的实验结果;

A: SentiWordNet 情感词典的实验结果如表 2所示。

Table 2: SentiWordNet 情感词典的实验结果。

Method	TP	TN	FN	FP	Precision	Recall	F1-score	Acc
Basic	18955	15689	6045	9311	67.06%	75.82%	71.17%	69.29%
Endmarks	18830	15285	6170	9715	65.97%	75.32%	70.33%	68.23%
No_Morphology	19108	15414	5892	9586	66.59%	76.43%	71.17%	69.04%
No_Steopwords	18955	15689	6045	9311	67.06%	75.82%	71.17%	69.29%
Negative	21146	7478	3854	17522	54.69%	84.58%	66.43%	57.25%
Partial	17386	14193	7614	10807	61.67%	69.54%	65.37%	63.16%

B: Dict_Liub 情感词典的实验结果如表 3所示。

C: Dict_IMDB 情感词典的实验结果如表 4所示。

Table 3: Dict_Liub 情感词典的实验结果。

Method	TP	TN	FN	FP	Precision	Recall	F1-score	Acc
Basic	18113	18445	6887	6555	73.43%	72.45%	72.94%	73.12%
Endmarks	18060	17821	6940	7179	71.56%	72.24%	71.90%	71.76%
No_Morphology	18845	17813	6155	7187	72.39%	75.38%	73.86%	73.32%
No_Steopwords	18113	18445	6887	6555	73.43%	72.45%	72.94%	73.12%
Negative	15223	16407	9777	8593	63.92%	60.89%	62.37%	63.26%
Partial	16527	17706	8473	7294	69.38%	66.11%	67.70%	68.47%

Table 4: Dict_IMDB 情感词典的实验结果。

Method	TP	TN	FN	FP	Precision	Recall	F1-score	Acc
Basic	9516	23021	15484	1979	82.78%	38.06%	52.15%	65.07%
Endmarks	9826	22152	15174	2848	77.53%	39.30%	52.16%	63.96%
No_Morphology	11656	21918	13344	3082	79.09%	46.62%	58.66%	67.15%
No_Steopwords	8397	23403	16603	1597	84.02%	33.59%	47.99%	63.60%
Negative	12863	15522	12137	9478	57.58%	51.45%	54.34%	56.77%
Partial	10570	20000	14430	5000	67.89%	42.28%	52.11%	61.14%

D: SentiTFIDF 情感词典的实验结果如表 5所示。

全部的实验结果如下方的多因子组柱状图 (图 11) 所示：

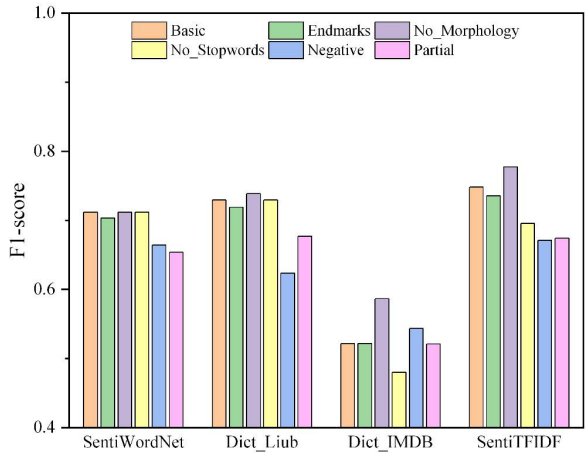


Figure 11: 全部的实验结果如下方的多因子组柱状图。

可以看出依照 SentiTFIDF 算法基于 IMDB 训练集构造的词典在测试集上模型的综合评价 (F1-score) 是最高的 (74.80%), 而在网上搜寻的针对 IMDB 构造的情感词典 Dict_IMDB 的与其它情感词典相比较差, 究其原因可能是情感词典的规模太小了, 总共才包含了 549 个情感词, 而我自己根据 SentiTFIDF 在一定的 TFIDF 阈值筛选, 词性筛选后还保留了 23,301 个情感词。

在单独来看各组中的对比, 可以看出考虑结尾标点符号会对最终的情感分析造成一定的误导, 因为叹号和问好并非一定的表示情感的积极面或消极面; 此外, 不进行词形的统一普遍都会让情感分析的结果有所上升,

Table 5: SentiTFIDF 情感词典的实验结果。

Method	TP	TN	FN	FP	Precision	Recall	F1-score	Acc
Basic	8522	10735	3978	1765	82.84%	68.18%	74.80%	77.03%
Endmarks	8648	10140	3852	2360	78.56%	69.18%	73.57%	75.15%
No_Morphology	9279	10402	3221	2098	81.56%	74.23%	77.72%	78.72%
No_Steopwords	7393	11134	5107	1366	84.40%	59.14%	69.55%	74.11%
Negative	9694	5805	2806	6695	59.15%	77.55%	67.11%	62.00%
Partial	7751	9755	4749	2745	73.85%	62.01%	67.41%	70.02%

这可能是因为像 SentiWordNet, Dict_Liub, SentiTFIDF 在词典中已经考虑了词形多样的问题, 词形统一时可能还会受到词性识别的影响, 因此导致结果受到一定的影响; 让我有点吃惊的是否定词对情感的影响竟然是负面的, 代表考虑否定词可能会误导最终的情感极性, 这可能就和人们的表达习惯有关了, 或者存在较多的一句话中先否后肯, 先肯后否的情况, 导致情感反转。其它的, 验证了停用词对最终结果的影响不大的猜测, 并且用有效首尾剧表达整个文本的情感也势必会将其分析的准确率。

在综合看, SentiTFIDF 与 Dict_Liub 的 Precision 与 Recall 较为接近, 表明其对积极情感与消极情感的敏感程度一致。其次再看 SentiWordNet 的 Recall 值明显高于其 Precision 值, 这表明 SentiWordNet 对积极情感较为敏感, 能够识别出大部分的积极情感文本, 但是存在较多的误报; 相反的 Dict_IMDB 的 Precision 值明显高于其 Recall 值, 这表明 Dict_IMDB 对消极情感较为敏感, 能够识别出大部分的消极情感文本, 但是存在较多的误报。因此我想利用集成学习的思想, 将这两个极性敏感的情感词典 (SentiWordNet, Dict_IMDB) 与一个综合评价较高的词典 (SentiTFIDF) 进行实验, 来查看是否能提高最终的结果。

3.5.2 SentiWordNet 中情感词典词性的影响

本实验以形容词词性为基准, 逐渐扩充词性, 其中名词为 n, 动词为 v, 形容词为 a, 副词为 r, 组合不同的词性所得到的实验结果如表 6 所示。

Table 6: 组合不同的词性所得到的实验结果。

Part of Speech	TP	TN	FN	FP	Precision	Recall	F1-score	Acc
a	19428	14844	5572	10156	65.67%	77.71%	71.19%	68.54%
a, r	17695	16870	7305	8130	68.52%	70.78%	69.63%	69.13%
a, n	20311	13630	4689	11370	64.11%	81.24%	71.67%	67.88%
a, v	23773	6120	1227	18880	55.74%	95.09%	70.28%	59.79%
a, r, n	18955	15689	6045	9311	67.06%	75.82%	71.17%	69.29%
a, r, v	22943	8447	2057	16553	58.09%	91.77%	71.15%	62.78%
a, n, v	23748	6212	1252	18788	55.83%	94.99%	70.33%	59.92%
a, r, n, v	23078	8266	1922	16734	57.97%	92.31%	71.22%	62.69%

可以看出形容词词形对情感分析的重要程度, 只考虑最终的识别准确率的话, 副词和名词对能够略微提升情感分析的准确率。相反动词则可能会起到反作用。

3.5.3 集成词典的影响

在第一个实验中可以看出不同的词典存在不同的极性偏向, 因此本实验将验证将不同词典 (SentiWordNet, Dict_IMDB, SentiTFIDF) 集成起来最终对极性结果进行投票对最终预测准确率的影响。实验结果如

表 7所示。

Table 7: 使用集成词典进行情感分析的实验结果。

Method	TP	TN	FN	FP	Precision	Recall	F1-score	Acc
Basic	17734	21711	7266	3289	84.36%	70.94%	77.07%	78.89%
Endmarks	17948	20493	7052	4507	79.93%	71.79%	75.64%	76.88%
No_Morphology	19196	21091	5804	3909	83.08%	76.78%	79.81%	80.57%
No_Steopwords	15519	22419	9481	2581	85.74%	62.08%	72.01%	75.88%
Negative	19618	11831	5382	13169	59.83%	78.47%	67.90%	62.90%
Partial	15928	19721	9072	5279	75.11%	63.71%	68.94%	71.30%

可以看出, 集成学习思想的应用, 让基于情感词典的情感分类任务准确率突破了 80% 这已经算是一个很好的成绩了, 而且 SentiTFIDF 在这其中贡献巨大, 再加上其构造简单快速, 因此为构建一个快速可解释的情感分析系统/平台打下了基础。

4 可解释性情感分析系统

基于情感的词典分析并没有涉及到复杂的模型, 但也因此拥有着良好的可解释性, 即对最终情感分析结果有贡献的词都可以定位到, 因此基于上述实验的工作, 我开发了一个简单的基于情感词典的可视化分析系统, 其以一个评论文本为输入, 通过选定的情感词典进行情感分类与情感标注工作。

4.1 基本功能

1. 选择情感词典 (图 12): 情感词典内置在系统中, 支持 SentiWordNet, Dict_Liub, Dict_IMDB, SentiTFIDF;

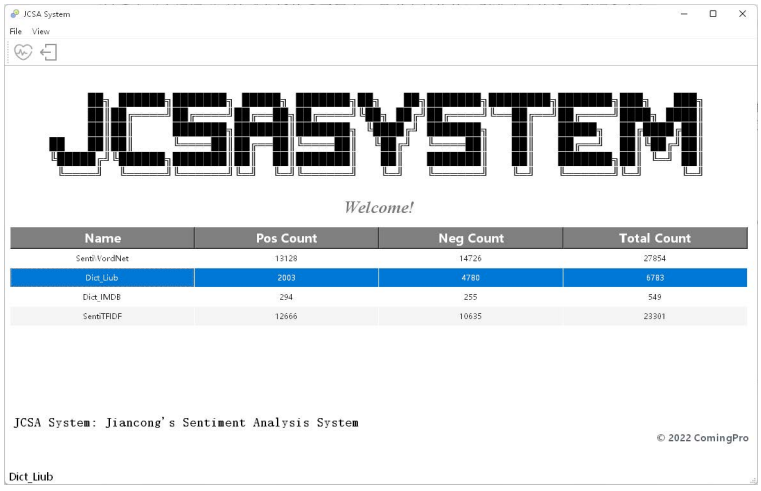


Figure 12: 情感分析系统基本功能之选择情感词典。

2. 情感分析系统 (图 13): 双击目标情感词典即可进入到情感分析界面, 底部的状态栏会显示出当前的配置信息;

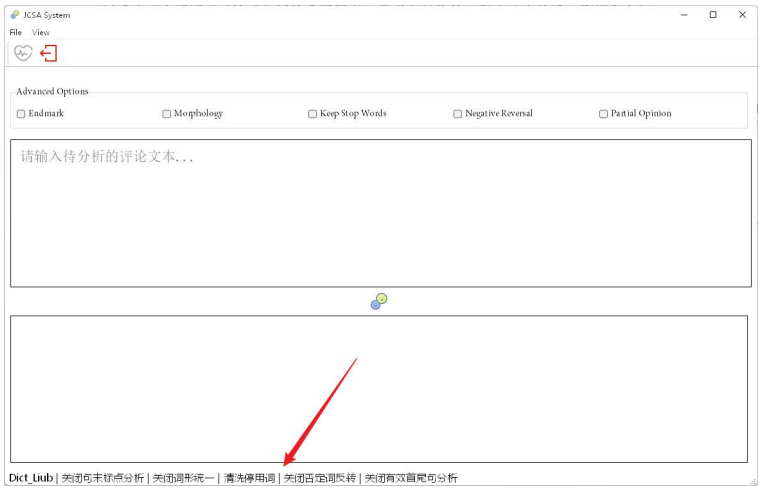


Figure 13: 情感分析系统基本功能之系统信息展示。

3. 开始情感分析 (图 14): 输入评论文本后, 能够自动识别, 开放情感分析功能;

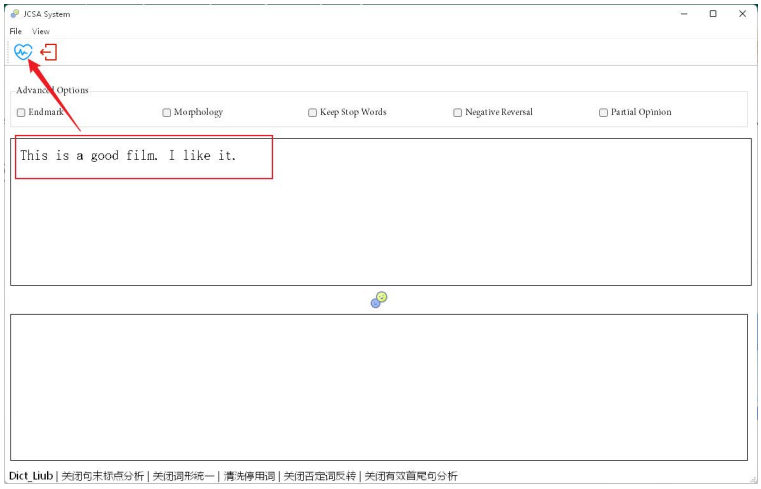


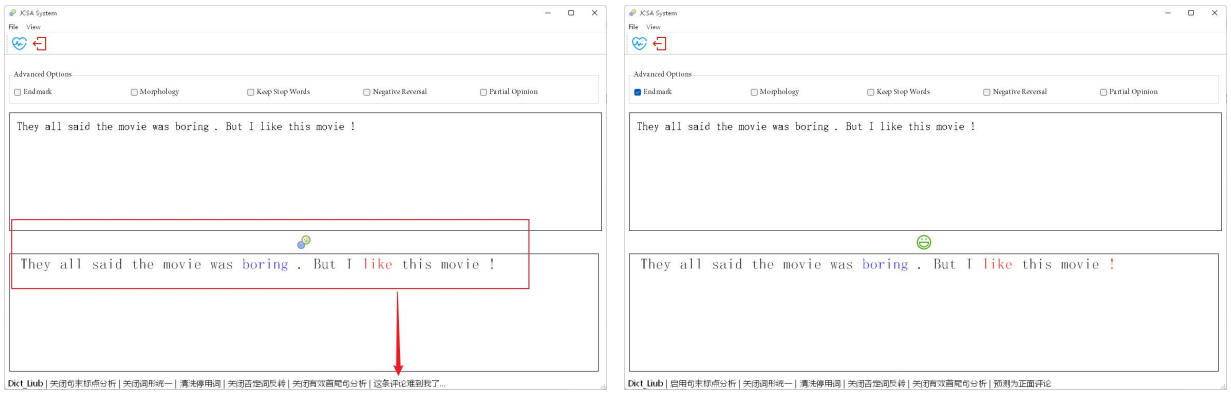
Figure 14: 情感分析系统基本功能之开始情感分析。

4. 解释分析结果 (图 15): 点击分析按钮后会展示分析结果, 并进行结果的解释;

4.2 扩展功能

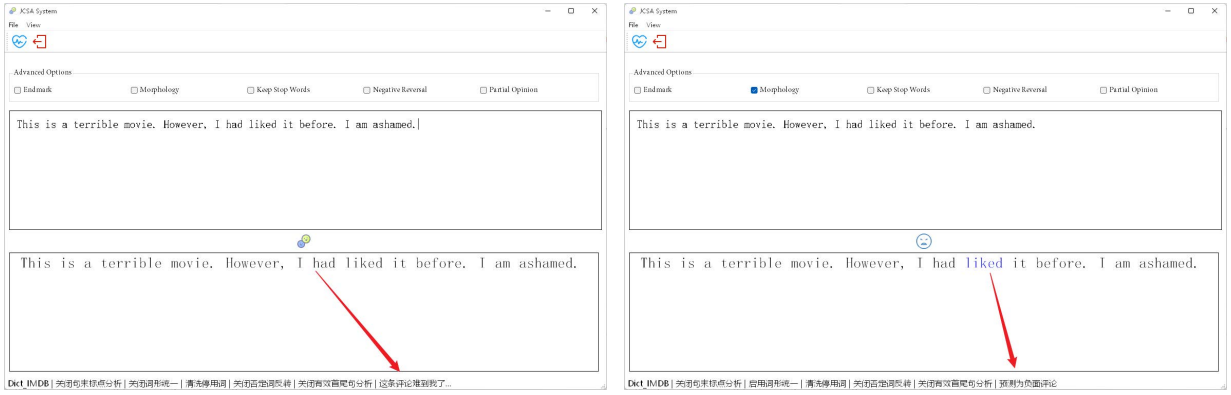
1. 结尾标点符号的考察 (图 16);
2. 词形统一 (图 17);
3. 保留停用词: 经过上述实验验证, 对实验影响不大;
4. 否定词反转 (图 18);
5. 有效首尾句分析 (图 19);

Figure 15: 情感分析系统基本功能之解释分析结果。



(a) 结尾标点符号应用前。 (b) 结尾标点符号应用后。

Figure 16: 情感分析系统扩展功能之对结尾标点符号的考察。

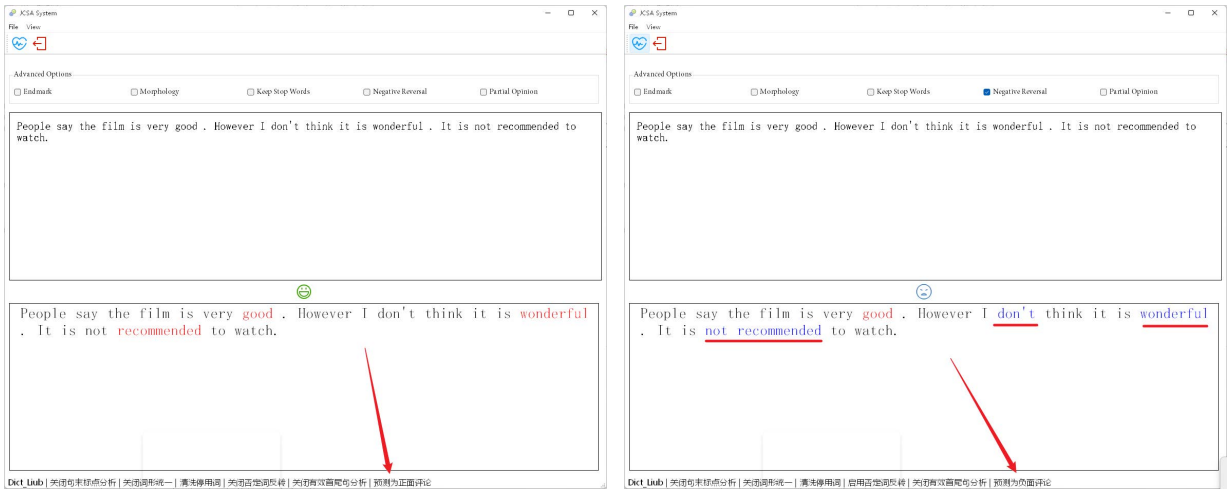


(a) 词形统一应用前。 (b) 词形统一应用后。

Figure 17: 情感分析系统扩展功能之词形统一。

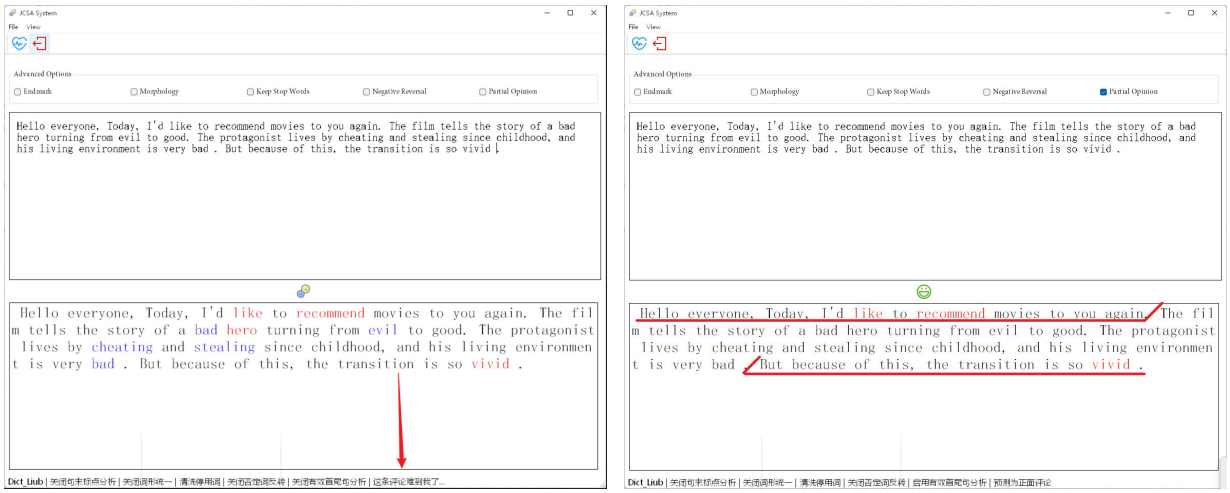
References

[Alshari et al.(2018)] Eissa M Alshari, Azreen Azman, Shyamala Doraisamy, Norwati Mustapha, and Mostafa Alkeshr. 2018. Effective method for sentiment lexical dictionary enrichment based on Word2Vec for



(a) 否定词反转功能开启前。 (b) 否定词反转功能开启后。

Figure 18: 情感分析系统扩展功能之否定词反转。



(a) 有效首尾句分析应用前。 (b) 有效首尾句分析应用后。

Figure 19: 情感分析系统扩展功能之有效首尾句分析。

sentiment analysis. In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*. IEEE, 1–5.

[Das and Chen(2001)] Sanjiv Das and Mike Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)*, Vol. 35. Bangkok, Thailand, 43.

[Ghag and Shah(2014)] Kranti Ghag and Ketan Shah. 2014. SentiTFIDF–Sentiment classification using relative term frequency inverse document frequency. *International Journal of Advanced Computer Science and Applications* 5, 2 (2014).

[Hardeniya and Borikar(2016)] Tanvi Hardeniya and Dilipkumar A Borikar. 2016. Dictionary based approach to sentiment analysis-a review. *International Journal of Advanced Engineering, Management and Science* 2, 5 (2016), 239438.

[Hatzivassiloglou and McKeown(1997)] Vasileios Hatzivassiloglou and Kathleen McKeown. 1997. Predicting the semantic orientation of adjectives. In *35th annual meeting of the association for computational linguistics and 8th conference of the european chapter of the association for computational linguistics*. 174–181.

[Hu and Liu(2004)] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 168–177.

[König and Brill(2006)] Arnd Christian König and Eric Brill. 2006. Reducing the human overhead in text categorization. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 598–603.

[Liu et al.(2004)] Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S Yu. 2004. Text classification by labeling words. In *Aaai*, Vol. 4. 425–430.

[Maas et al.(2011)] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 142–150.

[Mikolov et al.(2013)] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[Miller(1995)] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[Miller et al.(1990)] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography* 3, 4 (1990), 235–244.

[Motamedi(2018)] Hadi Motamedi. 2018. english constractions. `https://www.kaggle.com/datasets/hadimotamedi/english-contractions?resource=download`. [Online; accessed 19-june-2022].

[nproellochs(2018)] nproellochs. 2018. SentimentDictionaries. `https://github.com/nproellochs/SentimentDictionaries`. [Online; accessed 19-june-2022].

[Pang and Lee(2004)] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058* (2004).

[Prabowo and Thelwall(2009)] Rudy Prabowo and Mike Thelwall. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics* 3, 2 (2009), 143–157.

[Raghavan et al.(2005)] Hema Raghavan, Omid Madani, and Rosie Jones. 2005. InterActive Feature Selection.. In *IJCAI*, Vol. 5. Citeseer, 841–846.

[Rao et al.(2014)] Yanghui Rao, Jingsheng Lei, Liu Wenyin, Qing Li, and Mingliang Chen. 2014. Building emotional dictionary for sentiment analysis of online news. *World Wide Web* 17, 4 (2014), 723–742.

[Turney(2002)] Peter D Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *arXiv preprint cs/0212032* (2002).

[Van Atteveldt et al.(2021)] Wouter Van Atteveldt, Mariken ACG Van der Velden, and Mark Boukes. 2021. The validity of sentiment analysis: Comparing manual annotation, crowd-coding, dictionary approaches, and machine learning algorithms. *Communication Methods and Measures* 15, 2 (2021), 121–140.

[Xu et al.(2019)] Guixian Xu, Ziheng Yu, Haishen Yao, Fan Li, Yueting Meng, and Xu Wu. 2019. Chinese text sentiment analysis based on extended sentiment dictionary. *IEEE Access* 7 (2019), 43749–43762.

[Zhang et al.(2018)] Shunxiang Zhang, Zhongliang Wei, Yin Wang, and Tao Liao. 2018. Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary. *Future Generation Computer Systems* 81 (2018), 395–403.