利用自然语言处理技术
构建中国市场金融舆情因子

江俊锋*†, 李家豪*†
*似然科技
†中山大学

{jiangjf6, lijh76}@mail2.sysu.edu.cn

***Abstract*---**在本文中我们设计了一种综合的算法来计算中国市场的舆情因子。首先，我们利用浏览器自动化操作自动地抓取几大有影响力的财经网站中的新闻与评论。然后再运用中文语境中的自然语言处理技术，如分词、**word2vec** 词嵌入算法，以及语义数据库 **WordNet** 对它们进行情感分析。由于场景的特殊性，我们要自定义情感词典，使得输出的舆情判断是舆论对中国市场看好与否的判断，而非以往的褒贬喜悲的情感判断。其次，在计算出原始舆情因子后，还要对其进行修正。实验结果表明，我们的舆情因子与中国市场有较显著的相关性，且修正后的舆情因子与中国市场有很显著的相关性，因此该因子可以作为投资决策中的重要参考。特别在 **2015** 中国股灾期间，修正后的舆情因子与上证指数的皮尔逊相关系数为 **0.5844**，由此可以看出我们的模型，特别是对于特别时期，即市场受舆论影响较大的时期有很好的指导作用。

***Index Terms*---**自然语言处理;**Word2Vec** 词嵌入;**WordNet;** 情感分析;

I. 简介

自然语言处理是机器学习中最有前途的领域之一，而相关的研究在最近也取得了重大的进展。许多研究也将自然语言处理的技术应用到金融市场当中。在利用自然语言处理进行投资决策的过程中，人们面临的主要难题是自然语言是非结构化的数据。自然语言处理的一个主要任务也就是要处理自然语言这一非结构化的数据。许多模型能够很好地将自然语言数据转换为便于处理的数值数据。应用这种方法，利用自然语言数据就变得可能并且更简单了。

在这其中的一种模型是基于朴素贝叶斯的。[1] 这种模型背后的逻辑是：反映出同一类情感的词总是会频繁地同时出现。这些研究往往会选择一些词作为标签词。通过研究大量文本中使用的词并着眼于标签词与其他词的出现频率之间的关系，就可以将许多不同的词进行分类。对于任意给定的文本，就能够用这些分类过的词来评价文本反映的情感。研究表明这种方法能够很好地分析推特或者新闻所反映出来的情感，而基于这种舆情因子，投资者可以相应地做出投资决策。

然而，这种方法有其局限性。最主要的局限性在于它们只关注很少量的词。一些新的词可能会表现出相似的情感，但是却由于出现频率太低而被忽视。有些时候这些词语在分析文本情感的时候却会起到十分重要的作用。那么这些的信息的丢失就会对情感分析的准确性带来巨大的破坏。

此研究主要着眼于利用更多的词进行情感分析，而非只关注少数几个典型的词。具体的研究目的包括以下几个方面：

- 自动地从几个具有影响力的财经网站下载新闻；
- 对从网站中爬取的新闻进行预处理；
- 采用一种方法或一些算法对预处理文本数据进行分析，并最终利用每天的新闻计算当天的舆情得分；
- 选择合适的标准对舆情因子与市场趋势的相关性进行分析，评价舆情因子是否对金融投资有帮助。

其中用到的完整代码已经在 Github 上开源 [1]

文章剩下的部分安排如下：第二部分描述研究的背景和相关研究，包括结巴分词，Word2vec 和 WordNet。第三部分则展示研究方法与使用到的数据。第四部分包括实验结果和讨论。最后第五部分则是我们的结论。

II. 相关研究

A. 结巴分词

中文分词比英文分词要复杂的多。进行英文分词，只需要将词语按照空格和标点符号分开即可。而中文词与词之间没有空格。因此对中文进行文本分析时，需要额外增加分词这一步骤。

结巴中文文本分词是一个中文分词模块。其算法是基于概率语言模型。它通过一个预先准备好的词典生成一棵 trie 树，并且计算字典中的词的词频。当处理一个需要分词的句子的时候，它会生成一个 DAG（有向无环图）来记录每一种可能的分词的情况。这样一个 DAG 就是一个字典，字典的键是一个词起始的位置，而字典的值则是由一个词可能的结尾的位置组成的列表。

对于 DAG 中每种可能的分词，需要基于预先准备好的词典计算概率。然后结巴会通过从右向左的方向计算概率最大的路径。这一条概率最大的路径就给了我们一种最大可能性的分词。

对于句子中出现词典里没有的词的情况，结巴采用 HMM（隐马尔科夫模型）和 Viterbi 算法（维特比算法）来进行分词。对于不在词典中的字，它们会有四种可能的状态：B（起始字），M（中间字），E（结束字）和 S（单字）。这些状态表明了这个字在词中的位置。而分词的过程主要就是基于这些状态。结巴分词的作者通过对大量文本的训练得到了三个概率表，并利用 Viterbi 算法来计算一个字最有可能的状态，并依据这种一个句子每个词的状态构成的状态链来进行分词。

[1] https://github.com/Coldog2333/Financial-NLP

B. Word2vec[2]

在 2013 年，Google 团队发表了 word2vec 工具。word2vec 工具主要包含两个模型：跳字模型（skip-gram）和连续词袋模型（continuous bag of words，简称 CBOW）。值得一提的是，word2vec 的词向量可以较好地表达不同词之间的相似和类比关系。

word2vec 自提出后被广泛应用在自然语言处理任务中。它的模型和训练方法也启发了很多后续的词嵌入模型。下面以中文为例介绍 word2vec 的模型。

1) 跳字模型*:* 在跳字模型中，我们用一个词来预测它在文本序列周围的词。

举个例子，“他很喜欢她呢”，这一句话可以划分为“他”，“很”，“喜欢”，“她”，“呢”。若以“喜欢”作为中心词，设窗口为 2，那么在跳字模型中，我们关心的是，给定的中心词“喜欢”生成与它距离不超过两个词的每一个背景词“他”，“很”，“她”，“呢”的条件概率。

我们用数学语言来严格地描述跳字模型。

假设词典索引集 D的大小为 $|\mathrm{D}|$, 且记 $D=\{1,2,...,|D|\}$. 给定一个长度为 T的文本序列，其中第 t个词记为 $w^{(t)}$. 当窗口大小为 m时，跳字模型要求最大化任一中心词生成距离不超过 m个词的背景词的总概率

$$\prod_{t=1}^{T} \prod_{-m\le j\le m, j\neq 0, 1\le t+j\le |T|} P(w^{(t+j)}|w^{(t)}) \tag{1}$$

所以似然函数为，

$$\sum_{t=1}^{T} \sum_{-m\le j\le m, j\neq 0, 1\le t+j\le |T|} logP(w^{(t+j)}|w^{(t)}) \tag{2}$$

最大化似然函数等价于最小化下面的损失函数，

$$-\frac{1}{T}\sum_{t=1}^{T} \sum_{-m\le j\le m, j\neq 0, 1\le t+j\le |T|} logP(w^{(t+j)}|w^{(t)}) \tag{3}$$

我们可以用 $\mathbf{v}$和 $\mathbf{u}$分别表示中心词和背景词的向量，也就是说，对于索引为 i的词，它作为中心词和背景词时的向量表示分别为 $\mathbf{v}_i$ 和 $\mathbf{u}_i$. 而我们要训练的模型参数就是词典中所有词的这两种向量。为了将模型参数植入损失函数，我们需要使用模型参数表达损失函数中的给定中心词生成背景词的条件概率。给定中心词，假设生成各个背景词是相互独立的，那么对于中心词 w_c，背景词 w_b,b,c为这两个词在词典中的索引。那么给定中心词 w_b 生成背景词 w_c 的条件概率可以通过 softmax 函数定义为

$$P(w_b|w_c) = \frac{exp(\mathbf{u}_b^T\mathbf{v}_c)}{\sum_{i\in D} exp(\mathbf{u}_i^T\mathbf{v}_c)} \tag{4}$$

通过微分我们可以得到上述条件概率的梯度

$$\frac{\partial logP(w_b|w_c)}{\partial \mathbf{v}_c} = \mathbf{u}_b - \sum_{j\in D}\frac{exp(\mathbf{u}_j^T\mathbf{v}_c)}{\sum_{i\in D} exp(\mathbf{u}_i^T\mathbf{v}_c)}\mathbf{u}_j \tag{5}$$

上式也可以写作

$$\frac{\partial logP(w_b|w_c)}{\partial \mathbf{v}_c} = \mathbf{u}_b - \sum_{j\in D} P(w_j|w_c)\mathbf{u}_j \tag{6}$$

我们可以用梯度下降法或随机梯度下降法来进行迭代求解，最终求得使得损失函数最小时，词典中所有词的中心词和背景词的词向量 v_i 和 u_i $i=1,2,...,|D|$.

当序列长度 T较长时，我们可以在每次迭代时随机采样一个较短的子序列来计算有关该子序列的损失，以求得近似解。

在自然语言处理的应用中，我们会采用跳字模型的中心词向量作为每一个词的词向量。

2) 连续词袋模型*:* 连续词袋模型与跳字模型类似，它是用中心词在文本序列中前后的背景词来预测该中心词。举个例子，“他很喜欢她呢”，这一句话可以划分为“他”，“很”，“喜欢”，“她”，“呢”。若仍以“喜欢”作为中心词，设窗口为 2，那么连续词袋模型关心的是，给定与中心词距离不超过两个词的背景词“他”，“很”，“她”，“呢”这四个词生成中心词“喜欢”的条件概率。

假设词典索引集 D的大小为 $|\mathrm{D}|$, 且记 $D=\{1,2,...,|D|\}$. 给定一个长度为 T的文本序列，其中第 t个词记为 $w^{(t)}$. 当窗口大小为 m时, 连续词袋模型要求最大化由距离不超过 m个词的背景词生成中心词的总概率.

$$\prod_{t=1}^{T} P(w^{(t)}|w^{(t-m)},...,w^{(t-1)},w^{(t+1)},...,w^{(t+m)}) \tag{7}$$

其中，m为窗口大小，且要确保 $(t-m+j)\in[1,|\mathrm{T}|]$,$j\in[0,2\mathrm{m}]$.

所以似然函数为，

$$\sum_{t=1}^{T} logP(w^{(t)}|w^{(t-m)},...,w^{(t-1)},w^{(t+1)},...,w^{(t+m)}) \tag{8}$$

最大化似然函数等价于最小化下面的损失函数，

$$-\sum_{t=1}^{T} logP(w^{(t)}|w^{(t-m)},...,w^{(t-1)},w^{(t+1)},...,w^{(t+m)}) \tag{9}$$

我们仍用跳字模型中表示中心词和背景词的记号，此时，对于中心词 w_c 以及它所对应的背景词 $w_{b0},w_{b1},...,w_{b\cdot 2m}$, 那么给定背景词 $w_{b1},w_{b2},...,w_{b\cdot 2m}$ 生成中心词 w_c 生成的条件概率可以通过 softmax 函数定义为,

$$P(w_c|w_{b0},w_{b1},...,w_{b\cdot 2m}) = \frac{exp(\frac{\mathbf{v}_c^T(\mathbf{u}_{b0}+\mathbf{u}_{b1}+...+\mathbf{u}_{b\cdot 2m})}{2m})}{\sum_{i\in D} exp(\frac{\mathbf{v}_i^T(\mathbf{u}_{b0}+\mathbf{u}_{b1}+...+\mathbf{u}_{b\cdot 2m})}{2m})} \tag{10}$$

通过微分我们可以得到上述条件概率的梯度,

$$\frac{\partial log P(w_c|w_{b0},w_{b1},...,w_{b\cdot 2m})}{\partial \mathbf{u}_{bi}} = \frac{1}{2m}(\mathbf{v}_c - \sum_{j\in D}\frac{exp(\frac{\mathbf{v}_c^T(\mathbf{u}_{b0}+\mathbf{u}_{b1}+...+\mathbf{u}_{b\cdot 2m})}{2m})}{\sum_{i\in D} exp(\frac{\mathbf{v}_i^T(\mathbf{u}_{b0}+\mathbf{u}_{b1}+...+\mathbf{u}_{b\cdot 2m})}{2m})}\cdot \mathbf{v}_j) \tag{11}$$

上式也可以写作,

$$\frac{\partial log P(w_c|w_{b0},w_{b1},...,w_{b\cdot 2m})}{\partial \mathbf{u}_{bi}} = \frac{1}{2m}(\mathbf{v}_c - \sum_{j\in D} P(w_c|w_{b0},w_{b1},...,w_{b\cdot 2m})\cdot \mathbf{v}_j) \tag{12}$$

与跳字模型一样，我们也可以用梯度下降法或随机梯度下降法来进行迭代求解，最终求得使得损失函数最小时，词典中所有词的中心词和背景词的词向量 v_i 和 u_i (i=1,2,...,|D|).

当序列长度 T较长时，我们可以在每次迭代时随机采样一个较短的子序列来计算有关该子序列的损失，以求得近似解。

在自然语言处理的应用中，我们会采用连续词袋模型的背景词向量作为每一个词的词向量。

C. *WordNet*

WordNet 是一个大型英文词典数据库。在 WordNet 中，词汇之间的关系是根据语义来相互联系的。在 WordNet 中词与词之间的关系主要是近义词 [3]。通过利用一个网络来表现词之间的关系，WordNet 帮助我们找到词的近义词，并表现了两个词在语义的意义上有多相似。下图就是一个 WordNet 结构的例子。

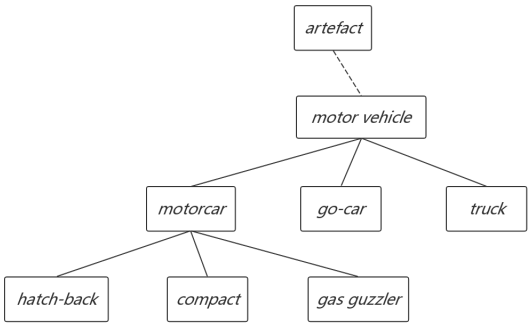

图 1. WordNet 结构示例 [4]

由于 WordNet 以上的一些性质，它在进行文本的情感分析时会更可靠。汉语开放词网（COW）是由 WordNet 启发的大规模的、能够免费获取的汉语语义词典 [5]。它有着和 WordNet 相同的结构，并且是基于中文的。它包含 42315 个近义词集，79812 个词义以及 61532 个词语，并且仍然在建设当中。我们的研究应用 COW 来计算每个词的情感得分。

III. 研究方法

A. 数据获取

为了获取足够数量的财经新闻，我们使用 selenium 库[2]进行浏览器自动化操作，自动化获取历史财经新闻数据，用作之后的舆情得分计算。

我们抓取了雪球网、北京证券、中国银河证券等各大财经网站的文章，其中中国银河证券网在时间的深度以及板块的广度上都远超其他网站，并且资讯渠道也包含了各大财经网，所以我们就选用中国银河证券网进行文本数据的获取。

为了便于我们进行回测，在抓取数据时按日期分类管理。

B. 预处理

汉语，作为一种孤立语，它不是通过词形变化来表达语法，而是通过独立的虚词和固定的词序来表达语法意义，与英语相比有其特殊性，比如英语天生就具有自动分词的特性，而汉语，在一句话中并没有像英语里的空格那样划分词的符号，所以在进行中文自然语言处理之前，我们需要对其进行分词，将一篇文章切分为词袋。这是英文自然语言处理与中文自然语言处理间的一个重要的差别。对任意给定的新闻或文本，我们采用结巴分词来进行预处理。除了需要将整篇文章转换为词袋以外，实际上，还会出现许多类似“的”、“是”、“可能”……这样的助词、副词等不具有具体含义或与情感分析无关的词，这些词和符号称为停用词，它们通常会频繁地在许多新闻或文本中出现然而却没有太多的意义。为了提高精度和减少运算，在进行预处理时，我们会对其进行去停用词操作，将注意力放在对文本的情感影响大的词上面去。

要注意的是，载入语料库训练模型时，不能对语料库进行去停用词操作，因为停用词的丢失会影响正确的词向量生成，在下面的文本分析部分中会对于这个问题做具体分析。

另外，我们还可以自定义一些应该去掉的字符来减少计算量并去噪，比如去掉英文单词、数字、标点符号等。[6]

C. 文本分析

1) 语境相似度*:* 首先，我们要训练 word2vec 模型以获得足够多的词的词向量（理论上，只要用于训练的语料库足够大，我们是可以获得所有词的词向量，但实际上我们不会有这样的语料库）。为了保证一般性，我们采用中文维基百科语料库进行训练。

要注意的是我们不能对用于训练的语料库进行去停用词操作，因为停用词的存在对描述中心词也会起到很关键的作用。

为了尽可能地使每个词的词向量在其所在的超平面上线性可分，我们要设置一个合适的词向量长度。另外我们可以设置一个恰当的词频阈值，使得可以忽略一些生僻词，在不影响我们关注的词向量计算的前提下以节省内存。

[2]https://www.seleniumhq.org/

训练结束后我们可以得到一个 word2vec 模型，与此同时，我们也就得到了所有词对应的词向量——我们只要像查字典那样在模型中查找对应的词向量即可。

现在，我们可以计算任意两个词之间的语境相似度了。比如，考虑两个词 w_1 和 w_2，标准化后为 $w_1^{'}$ 和 $w_2^{'}$，于是 w_1 和 w_2 的语境相似度为，

$$distance = \frac{w_1 \cdot w_2}{\|w_1\|\|w_2\|} = w_1^{'} \cdot w_2^{'} \tag{13}$$

注意：虽然我们现在可以用 word2vec 模型估计两个词的相似度，但是我们得到的只是它们的语境相似度。也就是说，我们只能知道某个词与哪些词在语境上类似，但并不知道它们的实际语义。我们可以举一个简单的例子，考虑两个词“上涨”“下跌”，我们也经常在新闻中能看到这样的句子——“今天上证指数上涨 5%”。然后你就会发现，我们可以很轻易地把“上涨”改为“下跌”而不会有任何的问题。根据 **2.2**所说的原理，这两个词的词向量基本上是一样的，这就会影响到我们去判定它们各自的情感得分。但是值得一提的是，word2vec 的确可以帮助我们找到与给定的词在某种意义上比较接近的一些词。

2) 语义相似度*:* 在 **2.3**中我们提到，WordNet 通过一种树状结构来储存词语，这种树状结构就天然地形成一种距离。在利用 WordNet 计算两个词的语义相似度时，就是利用这两个词在 WordNet 中的结点之间的最短路径来进行计算的。我们取两个词的结点之间的最短路径的长度的倒数作为语义相似度。每个词和自身的相似度是 1，而如果两个词完全没有路径将它们的结点相连，那么这个两个词的语义相似度定义为 0。这样定义得到的语义相似度是一个介于 0 到 1 之间的值，这个值越大，表明两个词之间在语义上越相似。

D. 情感得分计算

1) 情感词典*:* 我们定义一个情感词典作为小标签集来评估一个词的情感。其思想在于我们可以通过计算一个词与已知情感的标签词的相似度可以反映它的情感。我们选择了 100 个在财经新闻中经常出现、并且明显反映了对待市场的情绪的词语作为标签词。为了使计算更加公平，我们选择了 50 个积极词和 50 个消极词。在计算情感得分之前，我们首先计算常用词和情感词典中的每个词的语境相似度和语义相似度。

2) 词语情感得分计算*:* 通过上面的计算，对于每一个词语，我们都可以得到一个 200 维的向量，这个向量的前 100 维是这个词语与情感词典中的 100 个词的 Word2vec 相似度，反映的是这个词和每个情感词典中的词在语境上的相似度，而向量的后 100 维则是这个词语与情感词典中的 100 个词的 WordNet 相似度，反映的是这个词和每个情感词典中的词在语义上的相似度。由于情感词典中的词就是由反映对市场的态度的词语组成，因而根据与这些词的相似度，就能够代表每一个词对市场的态度，也即情感。

在我们的研究中，我们通过每个词的这一由相似度构成的向量来计算这个词的情感得分。具体而言包括以下几个步骤：（1）先后利用 Word2vec 相似度与 WordNet 相似度，通过协同过滤的方法判断词语属于积极词还是消极词；（2）利用 Word2vec 相似度计算词语情感得分。具体操作过程与原因叙述如下。

对于一个词，它和情感词典里的哪些词更像，有两个层面上的含义。一个是在语境层面上的相似，另一个是在语义层面上的相似。采用协同过滤的方法，可以找到与这个词语在语境和语义上最相似的几个词。首先根据前 100 维上的 Word2vec 相似度，在情感词典中找出与目标词在语境上最相似的 *n*个词，这个过程其实就是从前 100 维的相似度中找出最大的 *n*个。这 *n*个词就是在语境上与目标词最接近的 *n*个词。然后在后 100 维中找出目标词与这 *n*个词的 WordNet 相似度，并取出这 *n*个 WordNet 相似度最大的前 *m*个词。那么这 *m*个词就是在语境和语义两个层面都与目标词最相似的词。

有了这 *m*个词我们本可以直接判断词语属于积极词还是消极词，然而仅仅判断是积极词还是消极词是不够的，事实上，即使同样是积极词，不同的词反映的积极程度是不同的。为此，对于每个目标词，我们需要计算它们具体的情感得分。计算方法是记积极词为 +1，消极词为 -1，并利用这 *m*个词的 Word2vec 相似度大小作为权值，求加权平均值作为目标词的情感得分。

在实际操作中如前文所描述的那样，相比通过训练得到的 Word2vec 模型中的词语数目，COW 中的词语数目仍然不够充足，因此有些时候会出现一些词语与标签词集的 WordNet 相似度无法计算的情况，对于这些词，我们可以认为暂时无法判断其与标签词的语义相似度，因而仅仅对语境相似度进行判断。所以对这些词的协同过滤的过程，第二步就不能在 *n*个词中选出 WordNet 相似度最高的 *m*个词，而只能取 *n*个词中 Word2vec 相似度最高的 *m*个词了。最终仍采用 Word2vec 相似度作为权值进行情感得分值的计算。

在这样的计算方法下，我们就可以计算出每个词语的情感得分。

3) 文章情感得分计算与舆情因子*:* 利用上述计算得到的情感得分，我们就可以计算出每篇新闻的情感得分，并根据一天内所有网站所有新闻的情感得分，就能够构造市场当天的舆情因子。这一过程的具体描述在这一部分阐述。

对于经过预处理的新闻，我们得到的是一个词袋。固然我们可以对词袋中的每个词计算 200 维的相似度向量，并依据这个向量进行这个词的情感得分计算。然而这样会有大量的重复计算，事实上预先计算好足够多的词语的情感得分，往往就已经能够涵盖任意一篇文章的绝大多数词语，能够给我们对文章的情感作出足够准确的判断了。

为此，我们首先利用了商务印书馆出版社出版的《现代汉语常用词表》的 50000 词，然后我们对经过预处理的任意 3000 篇新闻中的词进行判断，提取出不属于《现代汉语常用词表》中的词，最终我们整合出约 100000 个常用词。我们预先计算好这 100000 词的情感得分，这样每次进行新闻的情感得分计算时，对每个词的得分就不需要进行计算的过程，而仅仅只要进行检索即可。这样的操作可能会丢失一些词语，然而实验结果表明对于大多数文章而言，不在常用词集中的词只占词量的 5% 左右，丢弃这 5% 的词对文章情感得分的影响微不足道。

那么，每篇文章就利用这一常用词集中的情感得分，对文章中词的情感得分进行求均值，从而得到整篇文章的情感得分。对于一天中所有文章的情感得分再进行求均值，就得到当天的舆情因子（舆情得分）。而这一舆情因子就能够很好地反映市场的情绪了。

4) 模型修正*:* 至此我们计算出了每一天的舆情得分，我们用当天的舆情得分与当天的大盘指数进行时间序列分析时发现，即使大盘指数变化不大，但舆情得分却有剧烈的波动，这与实际相违。于是我们在应用时还需要对模型进行修正，对舆情得分进行平滑化处理，即评价某一天的舆情时并非直接应用当天的得分，而是取某个时间窗口大小内的舆情得分的平均值。这样做是由于信息的时效性，某一天的舆情不仅会对当天的市场走势有所影响，它还会接连地影响之后几天的市场走势。在后面我们会用原始模型与修正后的模型进行实验结果的比较。

E. 相关性分析

我们需要建立一个标准来评价舆情因子是否能够很好的反映市场情绪。我们采用的主要方法是利用舆情因子与市场的一些指标进行线性回归。假设市场情绪与市场走势是相关的，那么舆情因子与市场指标做回归的结果的显著性就能反映舆情因子效果的好坏。另外，在评价它们之间的相关性时，也可以用下面的 Pearson 相关系数（皮尔逊相关系数）来进行评估。

$$\rho_{X,Y} = \frac{E[X\dot{Y}] - E[X]E[Y]}{\sigma_X \dot{\sigma}_Y} \tag{14}$$

IV. 实验结果和讨论

我们计算了从 2012 年 11 月 6 日至今 1379 个交易日的舆情得分，并用这个舆情得分与一些大盘指数比如上证指数和深证成指进行相关性分析。

对它们进行线性回归 [7]，我们可以看到回归结果很好地通过了显著性检验。

表 I
上证指数与舆情因子回归结果

舆情因子	5161.70 ***
p	(2.52e-12)

表 II
深成指数与舆情因子回归结果

舆情因子	18071.5 ***
p	(1.01e-11)

此时的皮尔逊相关系数为 0.1851，说明原始的舆情因子与大盘指数呈较弱的相关性。当我们采用上面所说的修正后的舆情因子再进行同样类似的相关性分析，计算得到的皮尔逊相关系数为 0.3152，相关性显著提高。

特别地，我们对中国股灾前后从 2015 年 2 月 11 日到 2015 年 9 月 11 日进行实验，计算这时期里 139 个交易日的舆情得分，并同样地利用这个舆情得分与一些大盘指数比如上证指数和深证成指进行相关性分析。

此时的皮尔逊相关系数为 0.3424，说明原始的舆情因子与大盘指数呈较弱的相关性。当我们采用上面所说的修正后的舆情因子再进行同样类似的相关性分析，计算得到的皮尔逊相关系数为 0.5844，呈较强的相关性。另外再对其进行时间序列分析可以看到，特别是在股灾发生前后，舆情因子与大盘走势几乎一致。

由此可以看出我们的模型，特别是对于特别时期，即市场受舆论影响较大的时期有很好的指导作用。

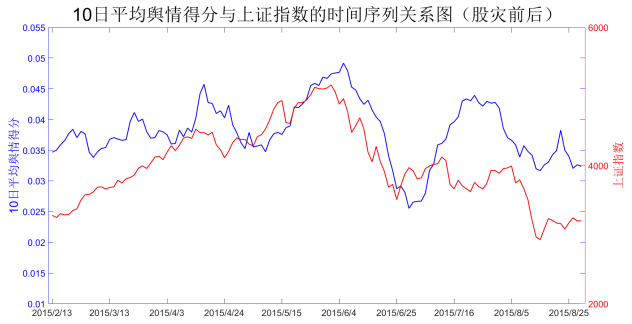

图 2. 10 日平均舆情得分与上证指数的时间序列关系图（股灾前后）

V. 结论

本文中我们构建了一种算法来计算中国市场的舆情因子，这一舆情因子与中国市场有着较好的相关性。这个因子能够为我们作投资决策提供一种新的方法。

本文的主要贡献在于建立了计算市场舆情因子的方法。如果我们能够计算各种金融产品的舆情因子，那将会更加有帮助。同时，将舆情因子与传统金融因子相结合能够帮助我们作出更好的投资决策。我们非常期待能看到相关的研究。

致谢

十分感谢来自朝旭投资管理有限公司的刘铭文在项目进行过程中给予的帮助与支持。同时，感谢来自中山大学的付星宇在前期研究中给予我们的引导和帮助。因为他们的热心帮助，我们这个课题的研究得以顺利完成。

参考文献

[1] Alec Go, Richa Bhayani, Lei Huang, 2009 [R]. Twitter Sentiment Classification Using Distant Supervision.

[2] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

[3] George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

[4] Bird S, Klein E, Loper E. Natural language processing with Python: analyzing text with the natural language toolkit[M]. " O'Reilly Media, Inc.", 2009.

[5] Building the Chinese Wordnet (COW): Starting from Core Synsets. In Proceedings of the 11th Workshop on Asian Language Resources: ALR-2013 a Workshop of The 6th International Joint Conference on Natural Language Processing (IJCNLP-6). Nagoya. pp.10-18.

[6] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[7] Rao T, Srivastava S. Analyzing Stock Market Movements Using Twitter Sentiment Analysis[C]// International Conference on Advances in Social Networks Analysis and Mining. IEEE Computer Society, 2012:119-123.

VI. 附录

A. 图片

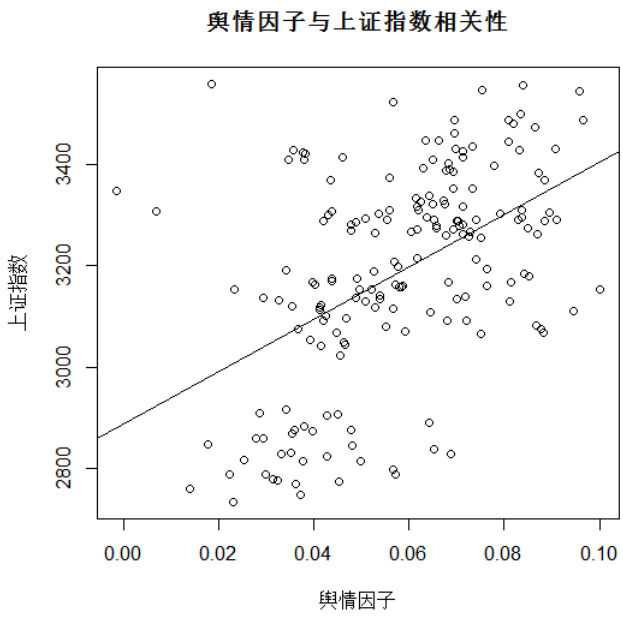

图 3. fig1

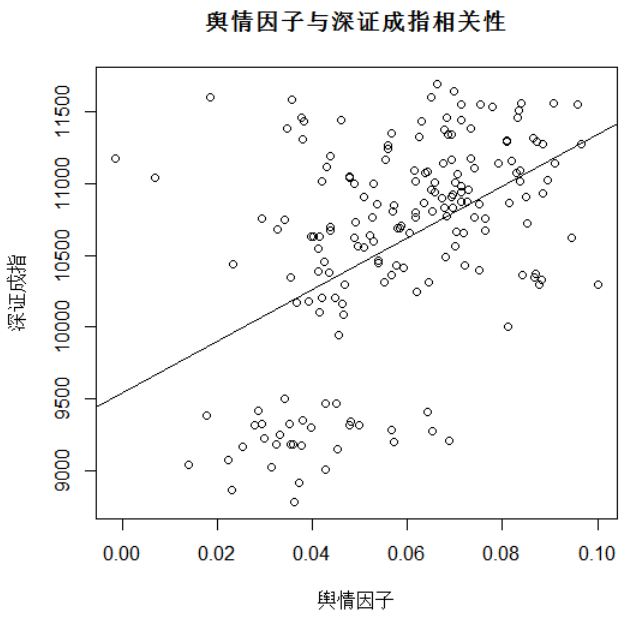

图 4. fig2

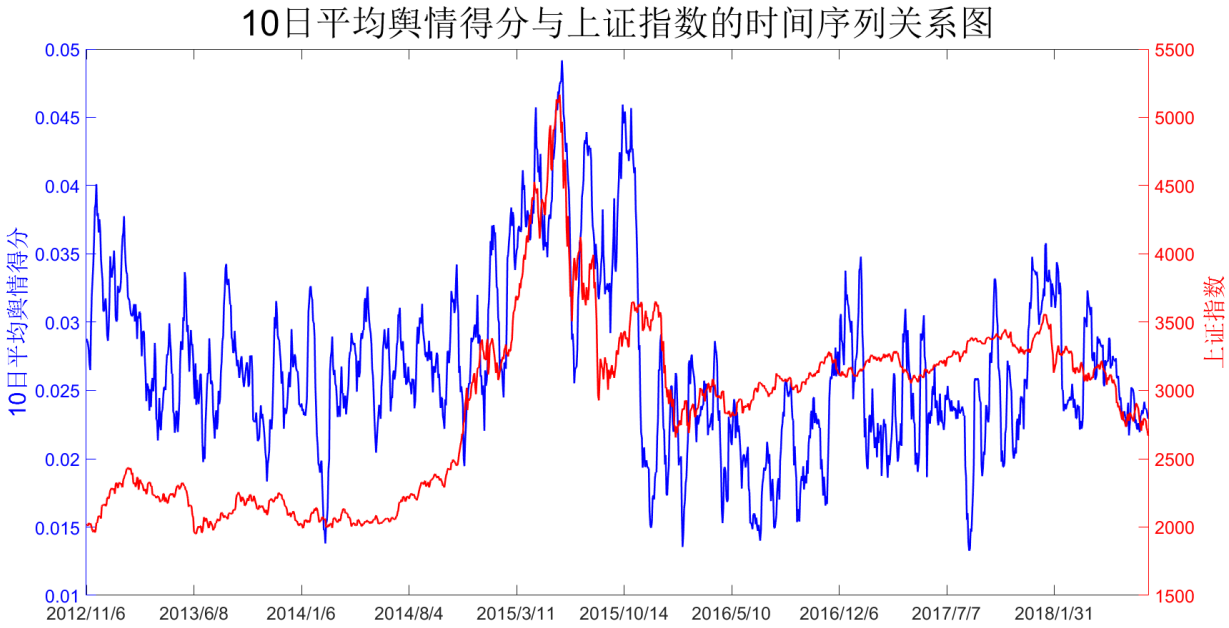

图 5.　10 日平均舆情得分与上证指数的时间序列关系图

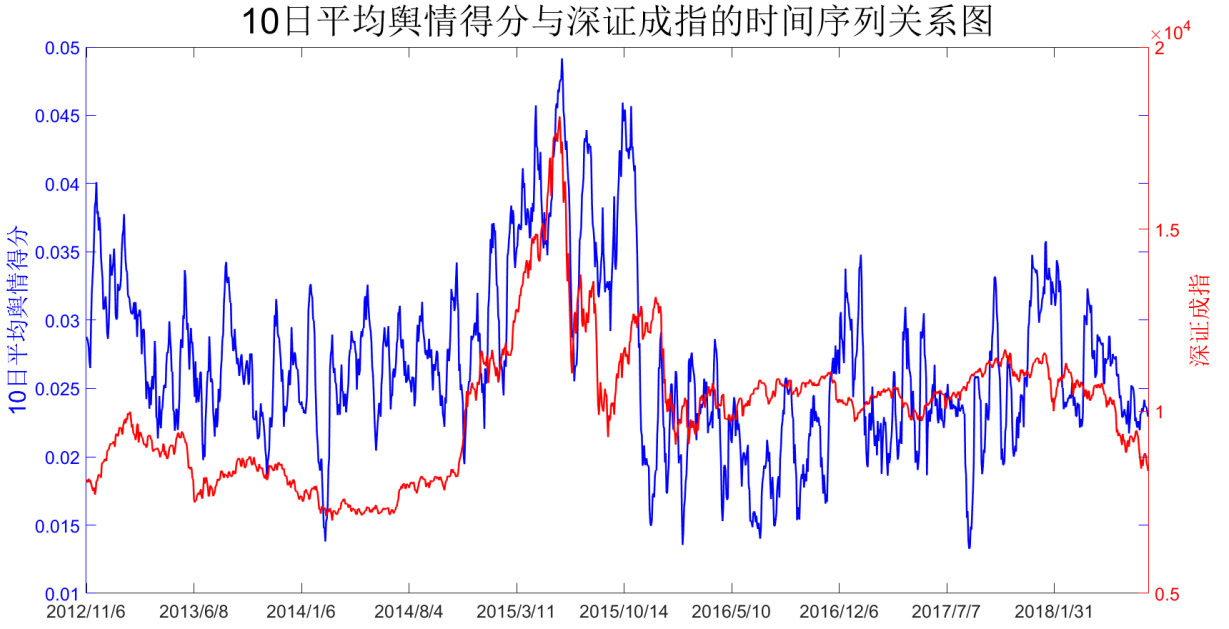

图 6.　10 日平均舆情得分与深成指数的时间序列关系图