MoreThanSentiments: A text analysis package

Jinhang Jiang, Karthik Srinivasan

A R T I C L E I N F O
Keywords:
Text mining
Natural language processing
Information extraction
Text complexity measures
Business analytics
Accounting

A B S T R A C T
Text mining on a large corpus of data has gained utility and popularity over recent years owing to advancements in information retrieval and machine learning methods. However, popular text mining software packages mainly focus on either sentiment analysis or semantic meaning extraction, requiring pretraining on a large corpus of text data. In comparison, MoreThanSentiments provides computation of newer text attribution measures, including boiler score, specificity, redundancy, and hard info, which have been proposed in accounting analytics literature. Our software package, available in Python, is flexible in terms of parameter setting and is adaptable to different applications. Through this package, we seek to simplify the process of deploying nontrivial information extraction techniques published in domain-specific text analysis research into domain-agnostic analytics applications.

1. Motivation and significance

Natural language processing, one of the major data mining methods, has been expanded and applied in many research fields in the past decades. Researchers have already paid lots of effort into studying unstructured text data by leveraging information extraction and retrieval techniques. And using the pre-trained models, such as the sentiment analysis model provided by TextBlob [1] or VADER [2] allows the users to deploy powerful models to tackle those tasks with less training time and resources. Recent studies have also proposed newer methods, such as Bidirectional Encoder Representations from Transformers (BERT) [3], for extracting the semantic meaning of the text. However, text features from existing software packages in Python and R either use pre-trained, high-performance models that result in features that cannot be directly interpreted or propose features that have a limited scope of explaining the nature of the text content. Sentiment analysis software typically provides the polarity or aggregate negative/positive sentiments expressed in text. Text analysis software such as py-readability-metrics provides metrics, including Gunning Fog, SMOG, and Flexh-Kincaid that focus on discerning the readability of the text. On the other hand, deep learning methods have been used to summarize longer text into short paragraphs [4].

MoreThanSentiments [5] was motivated by the fact that the users are eager to seek more nontrivial methods to quantify the text and
summarize the structure of the text. Currently, the package supports the following text complexity metrics: Boilerplate [6], a measurement of informativeness; Redundancy [7], a measurement of usefulness; Specificity [8], a measurement of the quality of relating uniquely to a particular subject; and Relative Prevalence [9], a measurement of hard information. This domain-agnostic package can easily be implemented for text quantification tasks in various projects. Additionally, we expect the novel adoption of the features in this package can serve as an enabler for different downstream works.

2. Software description

In this section, we discuss the functionality of MoreThanSentiments, followed by the demonstrations of the main functions.

2.1. Software architecture

The package, MoreThanSentiments, is implemented in Python. Currently, it is composed of one major module that supports all of the features:

- Read raw txt. format data into pandas dataframe
- Clean and preprocess the text corpora
- Calculate Boilerplate, Redundancy, Specificity, Relative Prevalence

2.2. Software functionalities

A boilerplate is a group of words (e.g., tetragram) that may be omitted from a statement without altering its semantic meaning in textual analysis. It is a measurement of informativeness, in other words. The boiler score [6] is determined by comparing the number of sentences that use boilerplate language to the total number of words. Thus, the higher the boiler scores, the lower the informativeness for the given corpora. To identify a boilerplate, the users need first to set the length. The default is four words, which is a tetragram. Then the whole corpora will be scanned, and the frequency of the boilerplate per document will be captured. By default, only the boilerplates appearing at least in five documents and less than 75% of the total documents will be used to calculate the boiler scores. The frequency threshold is used as a bias control. And the formula of the Boilerplate is as follows:

\[
Boilerplate = \frac{W_1}{W_d}
\]

(1)

\(W_1\) is the word count of the sentence that has a boilerplate.
\(W_d\) is the word count of the whole document.

The degree of Redundancy indicates how useful a corpus is. It is the proportion of really large sentences or phrases (e.g., 10 grams) that appear more than once in a given document. If a super-long statement or phrase is used repeatedly, that means the author tries to impose the duplicated information. Hence, this piece of information shall be marked as not useful. Similar to the Boilerplate, the higher the Redundancy, the less useful the given corpus is.

Specificity is a measure of the ability to relate specifically to a particular subject. It is defined as the number of specific entity names, numerical values, and times/dates scaled by the overall word count of a document. Currently, the Named Entity Recognizer from spaCy serves as the foundation for the Specificity function.

Hard information of a given corpus is measured by Relative Prevalence. It compares the number of numerical values to the overall length of the text. It aids in assessing the amount of quantitative data in a particular text.

3. Illustrative examples

In this section, we illustrate three usage examples of MoreThanSentiments. For a full usage guide, please refer to the library documentation (https://pypi.org/project/MoreThanSentiments/). The dataset we used to experiment with is the BBC Business news dataset [10].

The codes below demonstrate how to read the raw text data (“read_txt_files”) and perform the text cleaning (“clean_data”) as needed.

```python
import MoreThanSentiments as mts
my_dir_path = "E:\Package\Boilerplate\moranthansentiments\Data\bbc\business"
df = mts.read_txt_rules(PATH = my_dir_path)
display(df)
df['sent_tok'] = df.text.apply(mts.sent_tok)
df['cleaned_data'] = pd.Series()
for i in range(len(df['sent_tok'])):
    df['cleaned_data'][i] = mts.clean_data(x, lower=True, number=False,
                punctuations=True, unicode=True, stop_words=False) for x in df['sent_tok'][i]]
```

For the data cleaning function (“clean_data”), we offer the following options:

- **lower**: make all the words lowercase
- **punctuations**: remove all the punctuations in the corpus
- **number**: remove all the digits in the corpus
- **unicode**: remove all the types of unicode in the corpus
- **stop_words**: remove the stopwords in the corpus

The following codes illustrate how to calculate the boiler score. It needs to be applied to the whole corpora instead of a single document.

```python
df['Boilerplate'] = mts.Boilerplate(input_data = df['cleaned_data'],
    n = 4, min_doc = 5, get_ngram = False)
```

For the (“Boilerplate”) function, we offer the following options:

- **input_data**: this function requires tokenized documents.
- **n**: number of the ngrams to use. The default is 4.
- **min_doc**: when building the ngram list, ignore the ngrams that have a document frequency strictly lower than the given threshold. The default is 5 document. 30% of the number of documents is recommended. When the parameters are given as decimals (e.g., 0.3), it will be read as a percentage.
- **get_ngram**: if this parameter is set to “True”, it will return a dataframe with all the ngrams and the corresponding frequency, and “min_doc” parameter will become ineffective.

4. Impact

Our proposed software package contains functions that can be beneficial to research conducted in multiple disciplines, including but not limited to accounting, finance, information systems, marketing, management science, information sciences, applied computer science, and applied linguistics. For example, the boiler scores of financial disclosures could indicate the extent to which firms tend to adapt common phrases from other firms or reuse statements from their previous disclosures. The trends in disclosure reporting behavior regarding boilerplate, specificity, and hardinfo could help understand the role of disclosure scripting in the firm image, performance, and market behavior. Another application that examines Specificity, Boilerplate, and Redundancy
could be the impact of pre-written bot responses to customer queries. What textual characteristics of bot response are most appreciated by end users and how it contributes to problem resolution can be an interesting research question to address. A third potential application is a comparison of text characteristics of spam mail versus regular email. One could expect that boiler scores for regular emails might be much lesser than boiler scores for spam emails as often these emails tend to use common ‘key’ phrases geared towards fear-mongering and click-baiting of end users. Though measures such as Boilerplate and Redundancy have been widely used in accounting literature [6–9], recent studies have considered such measures in applications such as predicting crowdfunding success [11]. Our software code has been widely downloaded from the GitHub repository. Therefore, we decided to convert the code to a software package for ease of use and program replicability.

5. Conclusions

We propose a new software package called MoreThanSentiments that includes a list of text characterization features. The textual features we present via the python software package are unavailable elsewhere as reproducible code or software for general-purpose applications. These features originate from multiple studies in the accounting analytics discipline focusing on gleaning a variety of quantifiable information about the financial disclosure of firms. We make these quantitative features available for general-purpose applications by allowing the flexibility to generate the features using simple functions with user-defined parameters. Our package facilitates text characterization beyond sentiment analysis, counting of words, and computing readability.

The future development of MoreThanSentiments will focus on expanding its capabilities in terms of the text attribution measures it can compute [12–14], as well as making the software more user-friendly and adaptable to different applications. This will involve continuing to improve the underlying machine learning algorithms and information retrieval methods, as well as incorporating user feedback to ensure that the software meets the needs of a wide range of users. By providing a flexible and easy-to-use package for text mining on large corpora of data, MoreThanSentiments has the potential to become an essential tool for researchers and practitioners in a variety of fields.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References