

Pucong Han  
Frontier Computational Journalism Final Project  
Professor: Jonathan Stray and Emily Bell

## Chinese Text Modeling Tools and Applications for Computational Journalism

### Introduction

Statistical topic models – one of the sub fields of machine learning and natural language processing – provide a data-driven framework for analyzing collections of text documents. It has become one of the most frequently used tools for computational journalism used to investigate abstract topics and keywords that occur in a collection of text documents. Digital journalists can use such tools to extract frequently appearing terms, and to analyze the trend of a particular news brand or stories about a social event. Articles, analyses and documents written in Chinese have become increasingly important for multimedia stories about China. Available Chinese archives on the Internet might contain stories that require digital journalists to apply appropriate topic modeling tools.

Unlike English and other alphabetic languages, the basic structural unit of Chinese language is character encoded in Guobiao GB18030 or Unicode. Since there are no spaces between words in Chinese documents, topic-modeling tools for analyzing English text documents, such as [gensim](#) library, cannot be fully applied to analyze Chinese text documents. The gensim library can neither separate Chinese characters into segments nor convert Chinese characters to vectors using the bag-of-words approach. It only accepts pre-compiled documents with Chinese word segmentations with UTF-8 encoding. Digital journalists can take advantage of gensim statistical analyses, such as Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) and Term Frequency–Inverse Document Frequency (TF-IDF) to analyze existing corpus created by other tools. In this paper, I will analyze one of the Chinese topic-modeling tools [jieba](#) and its application to computational journalism. I will analyze articles from the opinion archive of the People’s Daily and generate a list of frequently appearing words using the keyword extraction tool provided by the jieba library. I will also apply gensim TF-IDF analysis to existing Chinese segmentation documents represented as bag-of-words counts and apply a weighting, which discounts common terms. I will compare the jieba keyword extraction results with the gensim TF-IDF results.

The People’s Daily is one of the most influential state-owned presses in China. The opinion section of the newspaper provides direct information on the policies and viewpoints of the Party. By analyzing the two lists of frequently appearing words, I want to predict the viewpoint of the government and its emphasis during the period of leadership transition. By analyzing the two lists of words, each from the same month, but in a different year, I want to learn whether the voice of the Party repeat in the same month of each year.

### Previous Studies and Researches

Previous research about Chinese topic modeling emphasizes text segmentation and semantic analyzing. “Word segmentation is considered an important first step for Chinese natural language processing tasks,” explained Professor Pi-Chuan Chang from the Computer Science Department of Stanford University. “The task of Chinese word

segmentation starts with designing a segmentation standard based on linguistic and task intuitions, and then aim to building segmenters that output words that conform to the standard” (Chang 1). Existing Chinese text segmentation tools have been applied to natural language processing research, in the areas of improving translation performance and text modeling.

In order to improve the accuracy of Chinese segmentation, researchers put emphasis on semantic analysis of the Chinese language. “The structure of the Chinese language is different from that of most western languages,” explained Professor Yunkai Zhang from Beihang University. “[Current] Chinese segmentation techniques failed to provide an efficient and highly accurate method to identify words from sequences of characters” (Zhang 1). As a result, Chinese topic modeling tools modified from existing topic modeling tools in other language might count meaningless terms or eliminate meaningful terms combining more than two Chinese characters. These topic models lack knowledge about Chinese semantics. “Existing topical models that involve collocations have a common limitation,” explained Professor Wei Hu from the Department of Computer Science of Shanghai Jiao Tong University. “Instead of directly assigning a topic to a collocation, they take the topic of a word within the collocation as the topic of the whole collocation” (Hu 1). Despite the difference between Chinese terms and English words, there are a number of competitive topic models that have “the ability to analyze Chinese documents with a high accuracy rate and the ability to understand the documents from the perspective of semantic meaning” (Zhang 4). Based upon these algorithms and topic models, researchers have developed a number of applicable text-segmentation tools and text-modeling tools, such as the jieba text-modeling tool.

According to Professor Zhang, future work is to train more models on a larger corpus to compare different models’ performance. In this paper, I will apply both the Jieba topic-modeling library and the gensim topic-modeling library to analyze The People’s Daily Opinion. By comparing the similarities and differences between the two results, I will evaluate the performance of these two libraries.

### **Data Mining and Cleaning**

The raw data text files are downloaded from the opinion archive of [the People’s Daily](#), as shown in figure 1. The archive contains opinion articles between 2010 and 2012. In order to fetch these articles using Python, I imported urllib2 and BeautifulSoup libraries. I also imported the csv library to save articles in the jieba\_analysis/raw\_data folder. The People’s Daily opinion archive does not have consistent tags for stories in recent years. I have to manually download stories in 2012 and save them in the jieba\_analysis/raw\_data folder, as shown in figure 2.

These downloaded articles contain “voices” from the state-owned press. To learn the interests of the Chinese government and its emphasis on social issues, digital journalists can analyze the frequently appearing keywords. In order to visualize the variation of frequently appearing words from 2010 to 2012, I reorganize the opinion articles to monthly-based text files stored in the jieba\_analysis/monthly\_raw\_data folder, as shown in figure 3.



Figure 1: The opinion archive of [the People's Daily](http://www.people.com.cn).

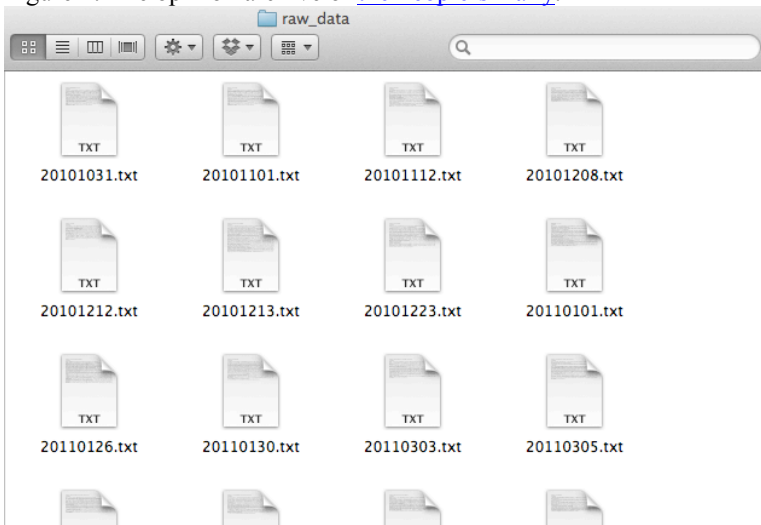


Figure 2: Opinion articles saved in txt files.

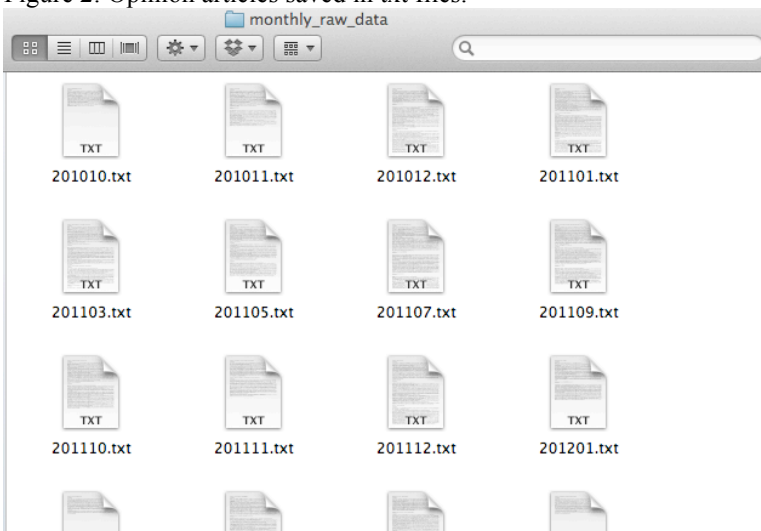


Figure 3: Opinion articles reorganized in monthly-based txt files.

## The jieba Library and its Application

According to the [documentation](#), the jieba library is a Python Chinese word segmentation module that supports three types of segmentation:

1. Accurate Mode: attempt to cut the sentence into the most accurate segmentation.
2. Full Mode: break the words of the sentence into words scanned.
3. Search Engine Mode: attempt to cut the long words into several short words.

The jieba library has three functions for text segmentation and analysis:

1. Cut function: The jieba.cut() method returns an iterative generator and takes two parameters. The first parameter is the string that requires segmentation, and the second parameter controls the segmentation pattern. The jieba.cut\_for\_search() method returns an iterative generator and takes one parameter: the string that requires segmentation. This method cuts the sentence into short words.
2. Add a custom dictionary: Developers can include their own custom dictionary to jieba and train the system to identify new words using the jieba.load\_userdict() function. Adding new words ensures a higher rate of correct segmentation.
3. Keyword extraction: The jieba library can extract keywords from a sentence using jieba.analyse.extract\_tags() method. The method takes two parameters. The first parameter is the text to be extracted. The second parameter is the TF-IDF weights controlling the number of extracted keywords (default value is 20).

## Chinese Text Segmentation

The three Python applications in the jieba\_analysis folder, including PeoplesDaily\_Jieba\_all\_segmatation.py, PeoplesDaily\_Jieba\_monthly\_segmatation.py and PeoplesDaily\_Jieba\_TFIDF.py, import jieba library and set the encoding to UTF-8 by declaring #encoding=utf-8. In this section, I will emphasize the segmentation function of the jieba library.

The PeoplesDaily\_Jieba\_all\_segmatation.py application from the jieba\_analysis folder read the monthly-based txt files from the monthly\_raw\_data folder and save the texts of each monthly article collection to a multiline string called multi\_line\_text, as shown in figure 4.

```
raw_data_path = 'monthly_raw_data/'
file_name = ["201010", "201011", "201012", "201101", "201103", "201105", "201107", "201109", "201110", "201111", "201112", "201201", "201202", "201203", "201204", "201205", "201206", "201207", "201208", "201209", "201210", "201211", "201212", "201301", "201302", "201303", "201304", "201305", "201306", "201307", "201308", "201309", "201310", "201311", "201312", "201401", "201402", "201403", "201404", "201405", "201406", "201407", "201408", "201409", "201410", "201411", "201412", "201501", "201502", "201503", "201504", "201505", "201506", "201507", "201508", "201509", "201510", "201511", "201512", "201601", "201602", "201603", "201604", "201605", "201606", "201607", "201608", "201609", "201610", "201611", "201612", "201701", "201702", "201703", "201704", "201705", "201706", "201707", "201708", "201709", "201710", "201711", "201712", "201801", "201802", "201803", "201804", "201805", "201806", "201807", "201808", "201809", "201810", "201811", "201812", "201901", "201902", "201903", "201904", "201905", "201906", "201907", "201908", "201909", "201910", "201911", "201912", "202001", "202002", "202003", "202004", "202005", "202006", "202007", "202008", "202009", "202010", "202011", "202012", "202101", "202102", "202103", "202104", "202105", "202106", "202107", "202108", "202109", "202110", "202111", "202112", "202201", "202202", "202203", "202204", "202205", "202206", "202207", "202208", "202209", "202210", "202211", "202212", "202301", "202302", "202303", "202304", "202305", "202306", "202307", "202308", "202309", "202310", "202311", "202312", "202401", "202402", "202403", "202404", "202405", "202406", "202407", "202408", "202409", "202410", "202411", "202412", "202501", "202502", "202503", "202504", "202505", "202506", "202507", "202508", "202509", "202510", "202511", "202512", "202601", "202602", "202603", "202604", "202605", "202606", "202607", "202608", "202609", "202610", "202611", "202612", "202701", "202702", "202703", "202704", "202705", "202706", "202707", "202708", "202709", "202710", "202711", "202712", "202801", "202802", "202803", "202804", "202805", "202806", "202807", "202808", "202809", "202810", "202811", "202812", "202901", "202902", "202903", "202904", "202905", "202906", "202907", "202908", "202909", "202910", "202911", "202912", "203001", "203002", "203003", "203004", "203005", "203006", "203007", "203008", "203009", "203010", "203011", "203012", "203101", "203102", "203103", "203104", "203105", "203106", "203107", "203108", "203109", "203110", "203111", "203112", "203201", "203202", "203203", "203204", "203205", "203206", "203207", "203208", "203209", "203210", "203211", "203212", "203301", "203302", "203303", "203304", "203305", "203306", "203307", "203308", "203309", "203310", "203311", "203312", "203401", "203402", "203403", "203404", "203405", "203406", "203407", "203408", "203409", "203410", "203411", "203412", "203501", "203502", "203503", "203504", "203505", "203506", "203507", "203508", "203509", "203510", "203511", "203512", "203601", "203602", "203603", "203604", "203605", "203606", "203607", "203608", "203609", "203610", "203611", "203612", "203701", "203702", "203703", "203704", "203705", "203706", "203707", "203708", "203709", "203710", "203711", "203712", "203801", "203802", "203803", "203804", "203805", "203806", "203807", "203808", "203809", "203810", "203811", "203812", "203901", "203902", "203903", "203904", "203905", "203906", "203907", "203908", "203909", "203910", "203911", "203912", "204001", "204002", "204003", "204004", "204005", "204006", "204007", "204008", "204009", "204010", "204011", "204012", "204101", "204102", "204103", "204104", "204105", "204106", "204107", "204108", "204109", "204110", "204111", "204112", "204201", "204202", "204203", "204204", "204205", "204206", "204207", "204208", "204209", "204210", "204211", "204212", "204301", "204302", "204303", "204304", "204305", "204306", "204307", "204308", "204309", "204310", "204311", "204312", "204401", "204402", "204403", "204404", "204405", "204406", "204407", "204408", "204409", "204410", "204411", "204412", "204501", "204502", "204503", "204504", "204505", "204506", "204507", "204508", "204509", "204510", "204511", "204512", "204601", "204602", "204603", "204604", "204605", "204606", "204607", "204608", "204609", "204610", "204611", "204612", "204701", "204702", "204703", "204704", "204705", "204706", "204707", "204708", "204709", "204710", "204711", "204712", "204801", "204802", "204803", "204804", "204805", "204806", "204807", "204808", "204809", "204810", "204811", "204812", "204901", "204902", "204903", "204904", "204905", "204906", "204907", "204908", "204909", "204910", "204911", "204912", "205001", "205002", "205003", "205004", "205005", "205006", "205007", "205008", "205009", "205010", "205011", "205012", "205101", "205102", "205103", "205104", "205105", "205106", "205107", "205108", "205109", "205110", "205111", "205112", "205201", "205202", "205203", "205204", "205205", "205206", "205207", "205208", "205209", "205210", "205211", "205212", "205301", "205302", "205303", "205304", "205305", "205306", "205307", "205308", "205309", "205310", "205311", "205312", "205401", "205402", "205403", "205404", "205405", "205406", "205407", "205408", "205409", "205410", "205411", "205412", "205501", "205502", "205503", "205504", "205505", "205506", "205507", "205508", "205509", "205510", "205511", "205512", "205601", "205602", "205603", "205604", "205605", "205606", "205607", "205608", "205609", "205610", "205611", "205612", "205701", "205702", "205703", "205704", "205705", "205706", "205707", "205708", "205709", "205710", "205711", "205712", "205801", "205802", "205803", "205804", "205805", "205806", "205807", "205808", "205809", "205810", "205811", "205812", "205901", "205902", "205903", "205904", "205905", "205906", "205907", "205908", "205909", "205910", "205911", "205912", "206001", "206002", "206003", "206004", "206005", "206006", "206007", "206008", "206009", "206010", "206011", "206012", "206101", "206102", "206103", "206104", "206105", "206106", "206107", "206108", "206109", "206110", "206111", "206112", "206201", "206202", "206203", "206204", "206205", "206206", "206207", "206208", "206209", "206210", "206211", "206212", "206301", "206302", "206303", "206304", "206305", "206306", "206307", "206308", "206309", "206310", "206311", "206312", "206401", "206402", "206403", "206404", "206405", "206406", "206407", "206408", "206409", "206410", "206411", "206412", "206501", "206502", "206503", "206504", "206505", "206506", "206507", "206508", "206509", "206510", "206511", "206512", "206601", "206602", "206603", "206604", "206605", "206606", "206607", "206608", "206609", "206610", "206611", "206612", "206701", "206702", "206703", "206704", "206705", "206706", "206707", "206708", "206709", "206710", "206711", "206712", "206801", "206802", "206803", "206804", "206805", "206806", "206807", "206808", "206809", "206810", "206811", "206812", "206901", "206902", "206903", "206904", "206905", "206906", "206907", "206908", "206909", "206910", "206911", "206912", "207001", "207002", "207003", "207004", "207005", "207006", "207007", "207008", "207009", "207010", "207011", "207012", "207101", "207102", "207103", "207104", "207105", "207106", "207107", "207108", "207109", "207110", "207111", "207112", "207201", "207202", "207203", "207204", "207205", "207206", "207207", "207208", "207209", "207210", "207211", "207212", "207301", "207302", "207303", "207304", "207305", "207306", "207307", "207308", "207309", "207310", "207311", "207312", "207401", "207402", "207403", "207404", "207405", "207406", "207407", "207408", "207409", "207410", "207411", "207412", "207501", "207502", "207503", "207504", "207505", "207506", "207507", "207508", "207509", "207510", "207511", "207512", "207601", "207602", "207603", "207604", "207605", "207606", "207607", "207608", "207609", "207610", "207611", "207612", "207701", "207702", "207703", "207704", "207705", "207706", "207707", "207708", "207709", "207710", "207711", "207712", "207801", "207802", "207803", "207804", "207805", "207806", "207807", "207808", "207809", "207810", "207811", "207812", "207901", "207902", "207903", "207904", "207905", "207906", "207907", "207908", "207909", "207910", "207911", "207912", "208001", "208002", "208003", "208004", "208005", "208006", "208007", "208008", "208009", "208010", "208011", "208012", "208101", "208102", "208103", "208104", "208105", "208106", "208107", "208108", "208109", "208110", "208111", "208112", "208201", "208202", "208203", "208204", "208205", "208206", "208207", "208208", "208209", "208210", "208211", "208212", "208301", "208302", "208303", "208304", "208305", "208306", "208307", "208308", "208309", "208310", "208311", "208312", "208401", "208402", "208403", "208404", "208405", "208406", "208407", "208408", "208409", "208410", "208411", "208412", "208501", "208502", "208503", "208504", "208505", "208506", "208507", "208508", "208509", "208510", "208511", "208512", "208601", "208602", "208603", "208604", "208605", "208606", "208607", "208608", "208609", "208610", "208611", "208612", "208701", "208702", "208703", "208704", "208705", "208706", "208707", "208708", "208709", "208710", "208711", "208712", "208801", "208802", "208803", "208804", "208805", "208806", "208807", "208808", "208809", "208810", "208811", "208812", "208901", "208902", "208903", "208904", "208905", "208906", "208907", "208908", "208909", "208910", "208911", "208912", "209001", "209002", "209003", "209004", "209005", "209006", "209007", "209008", "209009", "209010", "209011", "209012", "209101", "209102", "209103", "209104", "209105", "209106", "209107", "209108", "209109", "209110", "209111", "209112", "209201", "209202", "209203", "209204", "209205", "209206", "209207", "209208", "209209", "209210", "209211", "209212", "209301", "209302", "209303", "209304", "209305", "209306", "209307", "209308", "209309", "209310", "209311", "209312", "209401", "209402", "209403", "209404", "209405", "209406", "209407", "209408", "209409", "209410", "209411", "209412", "209501", "209502", "209503", "209504", "209505", "209506", "209507", "209508", "209509", "209510", "209511", "209512", "209601", "209602", "209603", "209604", "209605", "209606", "209607", "209608", "209609", "209610", "209611", "209612", "209701", "209702", "209703", "209704", "209705", "209706", "209707", "209708", "209709", "209710", "209711", "209712", "209801", "209802", "209803", "209804", "209805", "209806", "209807", "209808", "209809", "209810", "209811", "209812", "209901", "209902", "209903", "209904", "209905", "209906", "209907", "209908", "209909", "209910", "209911", "209912", "210001", "210002", "210003", "210004", "210005", "210006", "210007", "210008", "210009", "210010", "210011", "210012", "210101", "210102", "210103", "210104", "210105", "210106", "210107", "210108", "210109", "210110", "210111", "210112", "210201", "210202", "210203", "210204", "210205", "210206", "210207", "210208", "210209", "210210", "210211", "210212", "210301", "210302", "210303", "210304", "210305", "210306", "210307", "210308", "210309", "210310", "210311", "210312", "210401", "210402", "210403", "210404", "210405", "210406", "210407", "210408", "210409", "210410", "210411", "210412", "210501", "210502", "210503", "210504", "210505", "210506", "210507", "210508", "210509", "210510", "210511", "210512", "210601", "210602", "210603", "210604", "210605", "210606", "210607", "210608", "210609", "210610", "210611", "210612", "210701", "210702", "210703", "210704", "210705", "210706", "210707", "210708", "210709", "210710", "210711", "210712", "210801", "210802", "210803", "210804", "210805", "210806", "210807", "210808", "210809", "210810", "210811", "210812", "210901", "210902", "210903", "210904", "210905", "210906", "210907", "210908", "210909", "210910", "210911", "210912", "211001", "211002", "211003", "211004", "211005", "211006", "211007", "211008", "211009", "211010", "211011", "211012", "211101", "211102", "211103", "211104", "211105", "211106", "211107", "211108", "211109", "211110", "211111", "211112", "211201", "211202", "211203", "211204", "211205", "211206", "211207", "211208", "211209", "211210", "211211", "211212", "211301", "211302", "211303", "211304", "211305", "211306", "211307", "211308", "211309", "211310", "211311", "211312", "211401", "211402", "211403", "211404", "211405", "211406", "211407", "211408", "211409", "211410", "211411", "211412", "211501", "211502", "211503", "211504", "211505", "211506", "211507", "211508", "211509", "211510", "211511", "211512", "211601", "211602", "211603", "211604", "211605", "211606", "211607", "211608", "211609", "211610", "211611", "211612", "211701", "211702", "211703", "211704", "211705", "211706", "211707", "211708", "211709", "211710", "211711", "211712", "211801", "211802", "211803", "211804", "211805", "211806", "211807", "211808", "211809", "211810", "211811", "211812", "211901", "211902", "211903", "211904", "211905", "211906", "211907", "211908", "211909", "211910", "211911", "211912", "212001", "212002", "212003", "212004", "212005", "212006", "212007", "212008", "212009", "212010", "212011", "212012", "212101", "212102", "212103", "212104", "212105", "212106", "212107", "212108", "212109", "212110", "212111", "212112", "212201", "212202", "212203", "212204", "212205", "212206", "212207", "212208", "212209", "212210", "212211", "212212", "212301", "212302", "212303", "212304", "212305", "212306", "212307", "212308", "212309", "212310", "212311", "212312", "212401", "212402", "212403", "212404", "212405", "212406", "212407", "212408", "212409", "212410", "212411", "212412", "212501", "212502", "212503", "212504", "212505", "212506", "212507", "212508", "212509", "212510", "212511", "212512", "212601", "212602", "212603", "212604", "212605", "212606", "212607", "212608", "212609", "212610", "212611", "212612", "212701", "212702", "212703", "212704", "212705", "212706", "212707", "212708", "212709", "212710", "212711", "212712", "212801", "212802", "212803", "212804", "212805", "212806", "212807", "212808", "212809", "212810", "212811", "212812", "212901", "212902", "212903", "212904", "212905", "212906", "212907", "212908", "212909", "212910", "212911", "212912", "213001", "213002", "213003", "213004", "213005", "213006", "213007", "213008", "213009", "213010", "213011", "213012", "213101", "213102", "213103", "213104", "213105", "213106", "213107", "213108", "213109", "213110", "213111", "213112", "213201", "213202", "213203",
```

In order to separate the string to Chinese segments, I pass the string that contains texts of a monthly article collection to the jieba segmentation tool:

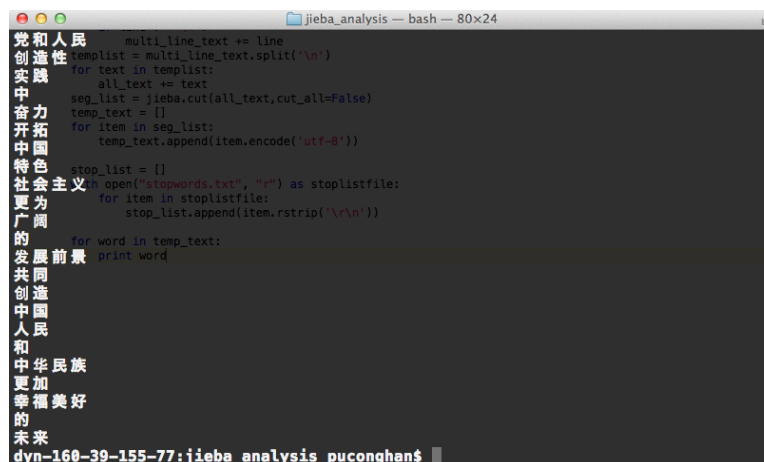
```
seg_list = jieba.cut(all_text, cut_all=False)
temp_text = []
for item in seg_list:
    temp_text.append(item.encode('utf-8'))
```

The jieba.cut() function returns a generator expression that creates all segments of the multiline string on the fly. Developers need to loop through the generator seg\_list once and create a copy in memory. The temp\_text list is the copy of the generator. It contains all segments of the articles, which includes a number of meaningless Chinese words commonly appearing in all text files. To clean such noises, the application reads a stop word txt file and saves the list of Chinese words to a stop\_list. Looping through the segment words in temp\_text, I generate a new list called text\_without\_stopwords contained segment words that were not appeared in the stop\_list:

```
stop_list = []
with open("stopwords.txt", "r") as stoplistfile:
    for item in stoplistfile:
        stop_list.append(item.rstrip('\r\n'))
```

```
text_without_stopwords = []
for word in temp_text:
    if word not in stop_list:
        text_without_stopwords.append(word)
```

As shown in figure 4 and figure 5, the stop list removes all meaningless words from the text list, such as “的”(Of), “在”(In), “中”(Middle) and “和”(And). The text\_without\_stopwords list contains Chinese segment words needed for statistical analyses.



```
dyn-160-39-155-77:jieba_analysis puconghan$
```

Figure 4: Text list before running stop list.

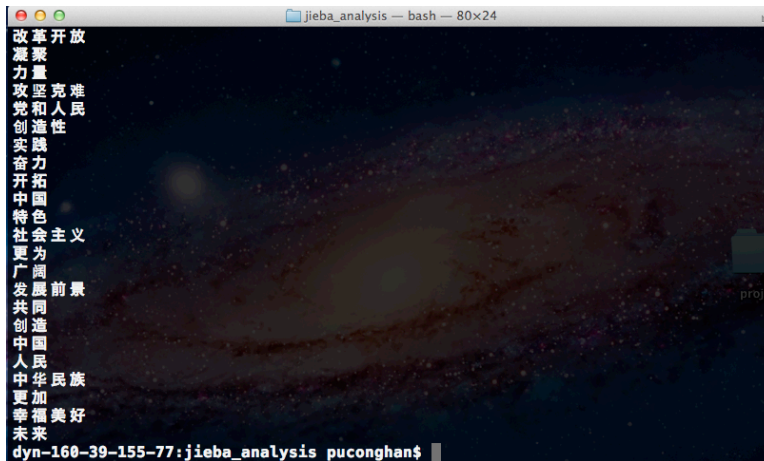


Figure 5: Text list after running stop list.

In order to test the performance of the jieba segmentation tool, I print all Chinese segments stored in the text\_without\_stopwords list, as shown in figure 6.

人民日报社,论,落幕,上海,世博,祝贺,世博会,闭幕,2010,10,31,刚刚,过去,东道主,中国,邀约,四海,宾朋,上海,浦江两岸,主办,一场,人类文明,盛大,聚会,一届,成功,精彩,难忘,世界,博览会  
人民日报社,论,落幕,上海,世博,2010,11,01,刚刚,过去,东道主,中国,邀约,四海,宾朋,上海,浦江两岸,主办,一场,人类文明,盛大,聚会,一届,成功,精彩,难忘,世界,博览会  
人民日报社,论,风雨,见证,伟大,精神,2010,12,08,伟大,国家,灾难,挑战,历练,国家,能力,坚强,民族,灾难,意味,风险,砥砺,民族,精神,今年以来,面对,历史,罕见,特  
把握,机遇,迎接,十年,社论,2011,01,01,新世纪,第一个,十年,过去,历经,自然灾害,各种,风险,考验,穿越,国际,金融危机,冲击,波澜,中国,社会主义,现代化,巨轮,驶入,历  
人民日报社,论,汇聚,推动,科学,发展,强大,合力,热烈祝贺,全国政协,十一届,四次会议,开幕,2011,03,03,全国政协,十一届,四次会议,开幕,充满希望,春天里,各党派,团体,  
人民日报社,论,勤奋,劳动,诚实,劳动,创新,劳动,庆祝,五一,国际,劳动节,2011,05,01,光荣,属于,伟大,劳动者,我国,工人阶级,劳动,群众,迎来,节日,五一,国际,劳动节,5  
人民日报社,论,永远,人民,奋斗,热烈庆祝,中国共产党,成立,九十,周年,2011,07,01,年前,中华民族,存亡绝续,关键时刻,中国共产党,诞生,这一,开天辟地,事变,开创,中国,  
人民日报社,论,铭记,历史,汲取,复兴,力量,纪念,中国,人民,抗日战争,暨,世界反法西斯战争,胜利,六十六,周年,2011,09,03,有些,重大,历史,时刻,往往,因其,影响,深远,  
人民日报社,论,中华民族,伟大,复兴,共同奋斗,纪念,辛亥革命,一百周年,2011,10,10,历史,一面镜子,一部,教科书,秋天,武昌,城头,震惊,世界,一声,枪响,中国,大地,日新月异  
人民日报社,论,空间,探索,重大,跨越,热烈祝贺,天宫,一号,神舟,八号,交会,对接,任务,圆满完成,2011,11,18,天宫,神舟,顺利,对接,神舟,八号,安全,返回,标志,天宫,一号,  
迈向,充满希望,2012,社论,元旦,献词,2012,01,01,随着,新年,钟声,敲响,告别,充满,挑战,奋发有为,迎来,充满希望,奋发进取,辞旧迎新,时刻,回顾过去,一年,国内,改革,  
人民日报社,论,书写,时代,精神,史诗,论,深入开展,学雷锋,活动,2012,03,02,春天,来到,大地,一片,勃勃生机,公民,道德,实践,呈现,图景,无论是,志愿者,这是,应该,积极  
改革,发展,稳定,主力军,社论,写,五一,国际,劳动节,2012,05,01,劳动,生活,美好,劳动者,美丽,劳动者,节日,全国,工人阶级,各族,各界,劳动,群众,致以,诚挚,祝福,广大,  
人民日报社,论,携手,共建,和谐,美好,家园,热烈祝贺,上海,合作,组织,北京,峰会,圆满成功,2012,06,08,2012,6,6,日至,7,北京,见证,历史性,盛会,上海,合作,组织,隆重  
人民日报社,论,中国,特色,社会主义,坚强,柱石,2012,08,01,金星,熠熠生辉,军旗,猎猎,飘扬,党,十八,召开,之际,中国人民解放军,迎来,建军,85,周年,全军,武警部队,官兵  
人民日报社,论,奋进,充满希望,中国,道路,庆祝,中华人民共和国,成立,63,周年,2012,10,01,不同寻常,一年,机遇,挑战,交织,而来,时代,征程,社会主义,中国,前行,步伐,伟  
人民日报社,论,夺取,中国,特色,社会主义,胜利,热烈祝贺,中国共产党,第十八次,全国代表大会,开幕,2012,11,08,每当,历史,发展,关键时刻,党,都,会,集中,全党全国,各族

Figure 6: Chinese segments before adding customized dictionary.

As we can see from the list of printed words, a number of Chinese terms, such as “人民日报社论”(People’s Daily Opinions), “国际劳动节”(The International Labor Day), “中国特色社会主义”(Socialism with Chinese Characteristics) and “中国共产党第十八次全国代表大会/十八大”(The 18<sup>th</sup> National Congress of Communist Party of China), are not accurately divided. In order to train the system to recognize these new terms, I created a new customized dictionary including these new terms and added it to the jieba module using the load\_userdict() method:

```
jieba.load_userdict("customized_dict.txt")
```

After adding this customized dictionary, the jieba segmentation tool can recognize such new terms and correctly cut the string into segments, as shown in figure 7. This word list can be exported to a txt file using csv library. The exported text file stores words in comma-separated values, which can be imported to statistical libraries, such as gensim library and jieba analysis library, for further analyses.



人民日报社论,落幕,上海,世博,祝贺,世博会,闭幕,2010,10,31,刚刚,过去,东道主,中国,邀约,四海,宾朋,上海,浦江两岸,主办,一场,人类文明,盛大,聚会,一届,成功,精彩,人民日报社论,落幕,上海,世博,2010,11,01,刚刚,过去,东道主,中国,邀约,四海,宾朋,上海,浦江两岸,主办,一场,人类文明,盛大,聚会,一届,成功,精彩,难忘,世界,博览会,人民日报社论,风雨,见证,伟大,精神,2010,12,08,伟大,国家,灾难,带来,挑战,历练,国家,能力,坚强,民族,灾难,意味,风险,砥砺,民族,精神,今年以来,面对,历史,罕见,特把握,机遇,迎接,十年,社论,2011,01,01,新世纪,第一个,十年,过去,历经,自然灾害,各种,风险,考验,穿越,国际,金融危机,冲击,波涛,中国,社会主义现代化,巨轮,驶入,历史人民日报社论,汇聚,推动,科学,发展,强大,合力,热烈祝贺,全国政协,十一届四次会议,开幕,2011,03,03,全国政协,十一届四次会议,开幕,充满希望,春天里,各党派,团体,各界人民日报社论,勤奋,劳动,创新,发展,劳动,庆祝,五一,国际劳动节,2011,05,01,光荣,属于,伟大,劳动者,我国,工人阶级,劳动,群众,迎来,节日,五一,国际劳动节,全国,人民日报社论,永远,人民,奋斗,热烈庆祝,中国共产党,成立,九十,周年,2011,07,01,年前,中华民族,存亡绝续,关键时刻,中国共产党,诞生,这一,开天辟地,事变,开创,中国,人民日报社论,铭记,历史,汲取,复兴,力量,中国,人民,抗日战争,暨,世界反法西斯战争,胜利,六十六,周年,2011,09,03,有些,重大,历史,时刻,往往,因其,影响,深远,总人民日报社论,中华民族伟大复兴,共同奋斗,纪念,辛亥革命,一百周年,2011,10,10,历史,一面镜子,一部,教科书,秋天,武昌,城头,震惊,世界,一声,枪响,中国,大地,日新月异人民日报社论,空间,探索,重大,跨越,热烈祝贺,天宫一号,神舟八号,交会,对接,任务,圆满成功,2011,11,18,天宫,神舟,顺利,对接,神舟八号,安全,返回,标志,天宫一号,神舟人民日报社论,吹响,十年,扶贫开发,号角,2011,12,02,十二五,时期,顺利,开局,全面,建设,小康社会,进入,关键时期,重要,时刻,中央,扶贫开发,工作,会议,京,召开,全面,迈向,充满希望,2012,社论,元旦,献词,2012,01,01,随着,新年,钟声,敲响,告别,充满,挑战,奋发有为,迎来,充满希望,奋发进取,辞旧迎新,时刻,回顾过去,一年,国内,改革,人民日报社论,强化,农业,科技,创新,确保,农产品,有效,供给,2012,02,02,日龙,伊始,万象更新,本报,全文,发表,中共中央国务院,关于,加快,推进,农业,科技,创新,持续,增人民日报社论,书写,时代,精神,论,深入开展,学雷锋,活动,2012,03,02,春天,来到,大地,一片,勃勃生机,公民,道德,实践,呈现,图景,无论,是,志愿者,这是,应该,积极,改革,发展,稳定,主力军,社论,号,五一,国际劳动节,2012,05,01,劳动,生活,美好,劳动者,美丽,劳动者,节日,全国,工人阶级,各族,各界,劳动,群众,致以,诚挚,祝福,广大,劳人民日报社论,携手,共建,和谐,美好,家园,热烈祝贺,上海合作组织,北京,峰会,圆满成功,2012,06,08,2012,6,6,日至,7,北京,见证,历史性,盛会,上海合作组织,隆重,开启人民日报社论,中国特色社会主义,坚强,柱石,2012,08,01,金星,熠熠生辉,军旗,猎猎,飘扬,党,十八大,召开,之际,中国人民解放军,迎来,建军,85,周年,全军,武警部队,官兵,人民日报社论,奋进,充满希望,中国,道路,庆祝,中华人民共和国,成立,63,周年,2012,10,01,不同寻常,一年,机遇,挑战,交织,而来,时代,征程,社会主义,中国,前行,步伐,格人民日报社论,夺取,中国特色社会主义,胜利,热烈祝贺,中国共产党第十八次全国代表大会,开幕,2012,11,08,每当,历史,发展,关键时刻,党,都,会,集中,全党全国,各族人民,

Figure 7: Chinese segments before adding customized dictionary.

## Keyword Extraction using Jieba Library

The jieba library has an analysis module that supports keyword extraction. The method takes two parameters, including the text that needs to be extracted and the number of frequently appearing keywords. It returns a list of keywords. Both `PeoplesDaily_Jieba_all_TFIDF.py` and `PeoplesDaily_Jieba_monthly_segmatation.py` applications read the previously generated csv file and save the lists of word segments to `all_text` string. These two apps import jieba.analyse module and pass the `all_text` string to the `jieba.analyse.extract_tags()` method:

```
for item in jieba.analyse.extract_tags(all_text, 10):
    text_temp.append(item.encode('utf-8'))
```

Both applications need to export the results to txt files using the csv library. Developers need to encode Chinese text segments to UTF-8 and save them to a temporary list. The csv writer can save the list of words using this particular encoding to Chinese characters.

The `PeoplesDaily_Jieba_all_segmatation.py` application extracts keywords from the entire text string and returns 20 frequently appearing keywords. The application saves the keywords to the `tfidf-jieba-all-result.csv` file in the results folder. Here is the list of 20 frequently appearing keywords in the entire corpus:

发展 (Development)  
 中国特色社会主义 (Socialism with Chinese Characteristics)  
 建设 (Construction)  
 人民 (People)  
 工作 (Word)  
 中国 (China)  
 我国 (Our Country)  
 小康社会 (Well-off Society)  
 人民日报社论 (People's Daily Opinion)  
 社会 (Society)  
 经济 (Economics)  
 文化 (Culture)  
 改革 (Reform)  
 社会主义 (Socialist)  
 水利 (Water Resources)  
 科学发展观 (Scientific Outlook on Development)  
 历史 (History)

中华民族伟大复兴 (The Great Rejuvenation of the Chinese Nation)  
经济社会 (Economics & Society)  
农业 (Agriculture)

The PeoplesDaily\_Jieba\_monthly\_segmatation.py application extracts keywords from monthly article collections and returns 10 frequently appearing keywords for each month. The application saves the monthly keywords to the tfidf-jieba-monthly-result.csv file in the results folder. Here is the list of frequently appearing keywords for each month:

2010/10 上海世博会,世博,落幕,中国,人类,世博会,世博园,低碳,沟通,城市  
2010/11 亚运,亚运会,上海世博会,世博,广州,中国,落幕,城市,亚洲,发展  
2010/12 发展,农村,农业,工作,残疾人,残运会,经济,精神,推进,防汛  
2011/01 中国特色社会主义,水利,发展,法律,建设,立法,体系,一号文件,坚持,我国  
2011/03 中国特色社会主义,发展,工作,十二五规划,法律,人民政协,十二五,立法,热烈祝贺,体系  
2011/05 劳动,发展,科技,青年,创新,工人阶级,青春,创造,中国,90  
2011/07 水利,发展,人民,西藏,建设,60,水资源,改革,中国特色社会主义,加快  
2011/09 中华民族,人民,中国,中华民族伟大复兴,抗日战争,民族,中国共产党,历史,复兴,伟大  
2011/10 文化,中国特色社会主义,中华民族伟大复兴,发展,建设,社会主义,辛亥革命,坚持,繁荣,推动  
2011/11 文化,文艺工作者,神舟八号,对接,交会,天宫一号,广大,发展,任务,圆满成功  
2011/12 发展,扶贫开发,经济,农业,环境保护,工作,农村,我国,经济社会,加快  
2012/01 金融,发展,经济,工作,我国,国际金融,把握,改革,金融业,经济社会  
2012/02 农业,科技,农村,发展,农产品,供给,创新,稳定,保障,加快  
2012/03 人民政协,发展,工作,社会,建设,民政,人民,中国特色社会主义,会议,发挥  
2012/05 青年,共青团,广大青年,中国特色社会主义,劳动,事业,90,共青团员,发展,群众  
2012/06 上海合作组织,成员国,发展,峰会,合作,地区,共同,北京,元首,携手  
2012/08 奥运会,中国特色社会主义,奥运,奥林匹克精神,伦敦,奥林匹克,奥林匹克运动,体育健儿,见证,赛场  
2012/10 中国特色社会主义,发展,社会主义,十年,社会主义现代化,十八大,中国,道路,社会,构建  
2012/11 中国特色社会主义,十八大,小康社会,科学发展观,社会主义现代化,全面,党和国家,人民,发展,大会

## TF-IDF Keywords Analysis using Gensim Library

Although the gensim library does not have segmentation tools for Chinese content, it has statistical analysis tools that can be applied to an existing corpus. I copied the existing segmentation result files from the jieba\_analysis/results folder to the gensim\_analysis folder. The gensim\_Dictionary.py application reads this existing segmentation file and generates a gensim dictionary, as shown in figure 8, and a corpus, as shown in figure 9. The dictionary is a document file that maps each word with its ID, as shown in figure 10. The corpus represented the original documents as sparse vectors, as shown in figure 11.

```
diclist = []
texts = []
with open('segmentation-jieba-result.csv', 'rU') as f:
    reader = csv.reader(f)
    for row in reader:
        for word in row:
            texts.append(word)
dictionary = corpora.Dictionary(line.split() for line in texts)
once_ids = [tokenid for tokenid, docfreq in dictionary.dfs.iteritems() if docfreq == 1]
dictionary.filter_tokens(once_ids)
dictionary.compactify()

valuelist = dictionary.token2id.values()
keylist = dictionary.token2id.keys()

for item in range(0, len(dictionary)):
    diclist.append([valuelist[item], keylist[item]])
#print diclist
w = csv.writer(open("PeoplesDaily_gensim_dic.csv", "wb"))
w.writerows(diclist)
```

Figure 8: Code for generating the gensim dictionary.



```

corpus1012 = []
with open('segmentation-jieba-result.csv', 'rU') as ff:
    reader2 = csv.reader(ff)
    for row2 in reader2:
        corpus1012.append(row2)
class MyCorpus(object):
    def __iter__(self):
        for line in corpus1012:
            yield dictionary.doc2bow(line)

corpus_memory_friendly = MyCorpus()
print corpus_memory_friendly
for vector in corpus_memory_friendly:
    print vector

corpora.MmCorpus.serialize('corpus.mm', corpus_memory_friendly)

```

Figure 9: Code for generating the gensim corpus.

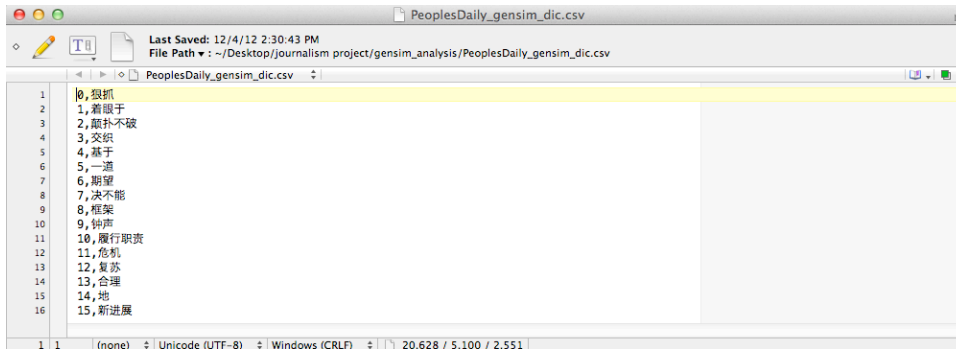


Figure 10: The generated gensim dictionary.

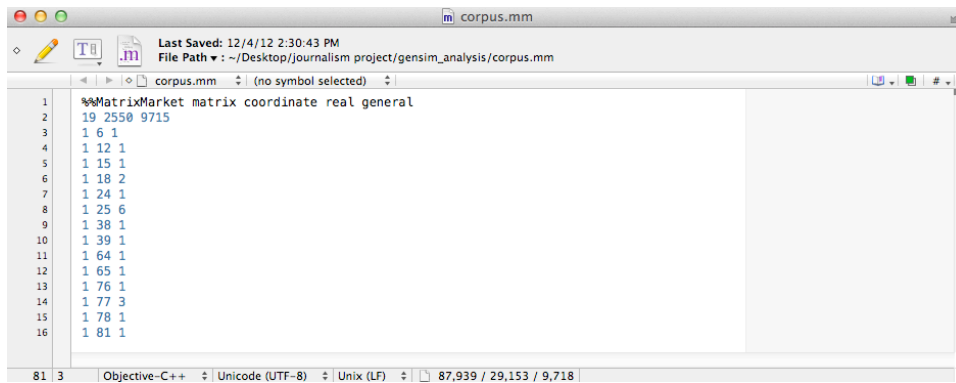


Figure 11: the generated gensim corpus.

Using the gensim dictionary and the corpus, developers can perform statistical analyses. The TopicModel.py application runs the TF-IDF analysis on the existing corpus:

```

corpus = corpora.MmCorpus('corpus.mm')
tfidf = models.TfidfModel(corpus)

```

It saves the TF-IDF results in the gensim\_TFIDF.csv file in the resultlist folder. The find\_TFIDF\_all\_keywords.py application reads gensim\_TEIDF.csv file and returns 20 word IDs with highest TF-IDF score, as shown in figure 12. Developers usually emphasize words with low TF-IDF scores. Words with low IF-IDF score frequently appear in one document, but they are not frequently appear in all documents. Since we want to make comparisons with the keywords generated by jieba, these 20 TF-IDF words must be frequently appearing in all documents.

```

NUM = 20
sumlist = numpy.zeros(38398)
lists = []
resultlist = []
with open('resultlist/gensim_TFIDF.csv', 'rU') as f:
    reader = csv.reader(f)
    for row in reader:
        for word in row:
            sumlist[int(word.find("(")+1:word.find(","))] = sumlist[int(word.find("(")+1:word.find(","))] + float(word[word.find(",")+1:word.find(")"]])

count = 0
for item in sumlist:
    lists.append([count, item])
    count += 1
sortedlist = sorted(lists, key=itemgetter(1), reverse=True)

for i in range(0, NUM-1):
    resultlist.append(sortedlist[i])

with open('PeoplesDaily_gensim_dic.csv', 'rU') as f:
    reader = csv.reader(f)
    for row in reader:
        for item in resultlist:
            if str(item[0]) == str(row[0]):
                item[0] = str(row[1])

print resultlist
w = csv.writer(open("resultlist/gensim_TFIDF_word.csv", "wb"))
w.writerows(resultlist)

```

Figure 12: TF-IDF analysis for all keyword.

The application maps each returned ID to a particular Chinese word using the gensim dictionary file. The application saves the keywords to the gensim\_TFIDF\_word.csv file in the resultlists folder. Here is the list of 20 words along with the TF-IDF score:

农业 (Agriculture)	1.13101010579
法律 (Law)	0.965040840409
劳动 (Work)	0.957638193359
青年 (Youth)	0.942623989634
水利 (Water Resources)	0.907716774625
金融 (Finance)	0.872824577986
世博 (Expo 2010 Shanghai China)	0.718687279642
上海合作组织 (The Shanghai Cooperation Organisation)	0.718505717462
农村 (Countryside)	0.695366468602
文化 (Culture)	0.693982350176
十年 (Ten years)	0.661320834741
立法 (Legislation)	0.637700637339
中国特色社会主义 (Socialism with Chinese Characteristics)	0.582919379105
工作 (Work)	0.581953680087
人民政协 (Chinese People's Political Consultative Conference)	0.578274475464
西藏 (Tibet)	0.556786191245
辛亥革命 (The Xinhai Revolution/ The Revolution of 1911)	0.537749641255
落幕 (Ring down the curtain)	0.522466708471
供给 (Supply)	0.497780006814

The find\_TFIDF\_monthly\_keywords.py application reads gensim\_TEIDF.csv file and returns the top 10 IDs with the highest scores for each month. The application also maps the list of IDs to their Chinese words using the gensim dictionary file. The application saves the monthly keywords to a number of csv files in the resultlists folder. Here is the list of keywords for each month:

2010/10 世博, 落幕, 上海世博会, 世博会, 沟通, 上海, 世博园, 人类, 城市  
2010/11 亚运, 亚运会, 广州, 世博, 亚洲, 落幕, 上海, 上海世博会, 沟通  
2010/12 残疾人, 明年, 农业, 农村, 亚, 抗旱, 残运会, 防汛, 亚洲  
2011/01 法律, 水利, 立法, 一号文件, 十年, 社会主义民主法制, 宪法, 体系, 水资源  
2011/03 法律, 人民政协, 立法, 报告, 十一届四次会议, 牢固, 规划, 委员, 监督  
2011/05 劳动, 青年, 工人阶级, 青春, 紧密结合, 科技, 人才, 工作者, 劳动者  
2011/07 西藏, 水利, 水资源, 代表, 政党, 谨记, 须, 全党同志, 07  
2011/09 抗日战争, 日本, 侵略者, 觉醒, 九一八事变, 抗日, 世界反法西斯战争, 这场, 侵略

2011/10 辛亥革命,文化,先生,孙中山,百年,先驱,中华民族伟大复兴,繁荣,没有  
2011/11 文艺工作者,交会,神舟八号,对接,天宫一号,航天,创作,文化,航天事业  
2011/12 环境保护,扶贫开发,农业,扶贫,明年,农村,贫困地区,12,力度  
2012/01 金融,金融业,金融机构,防范,实体,金融监管,监管,稳健,动荡  
2012/02 农业,供给,农产品,农村,科技,绝不能,因为,约束,强  
2012/03 民政,人民政协,雷锋,五次,学雷锋,思想道德,民政工作,雷锋精神,十一届  
2012/05 青年,共青团,劳动,广大青年,工人阶级,共青团员,团组织,劳动者,主力军  
2012/06 上海合作组织,成员国,峰会,地区,本,元首,合作,互信,携手  
2012/08 奥运会,伦敦,奥运,奥林匹克,奥林匹克运动,人民解放军,军队,我军,体育健儿  
2012/10 十年,构建,回顾,越,关键环节,伟大祖国,奋勇前进,社会主义,十八大  
2012/11 中央委员会,十八大,党和人民,新一届,中国共产党第十八次全国代表大会,选举,中国特色社会主义,表现,党的建设

### Comparison of Jieba Keyword Extraction Results with Gensim TF-IDF Results

The 20 frequently appearing words generated by the jieba library does not well match the TF-IDF results generated by the gensim library. Only 5 out of 20 Chinese words match in these two results. However, the monthly keyword results of the jieba library match the monthly TF-IDF results of the gensim library. Although some words are not exactly the same, these two monthly results contain synonyms.

The 20 frequently appearing words generated by jieba library are nouns. This is because the jieba keyword extraction takes advantages of its default Chinese dictionary to identify word types. This list of words can be found in the monthly keyword results of the jieba library. The list of words with high TF-IDF scores generated by the gensim library picks up all words that are frequently appearing in all documents. It does not avoid picking up meaningless words, such as 供给 (Supply). In extracting keywords from documents, users should use the jieba library. The gensim library might only be good for getting words with low TF-IDF scores. Such words frequently appear in one document, but they are not frequently appear in all documents

### Stories in the Keywords Extraction of People's Daily Opinion

If we look at the monthly keywords generated by the jieba library, we can find some interesting patterns. Comparing the keywords in Dec 2010 and the keywords in Dec 2011, we can find a number of words are repeated in the same month: 发展 (Development),农村 (Countryside),农业 (Agriculture),工作 (Works), and 经济 (Economics).

2010/12 发展,农村,农业,工作,残疾人,残运会,经济,精神,推进,防汛  
2011/12 发展,扶贫开发,经济,农业,环境保护,工作,农村,我国,经济社会,加快

The same pattern repeats in almost all months. The keywords in March 2011 and the keywords in March 2012 contain the following highlighted repeated words: 中国特色社会主义 (Socialism with Chinese Characteristics), 发展 (Development), 工作 (Work), 人民政协 (Chinese People's Political Consultative Conference).

2011/03 中国特色社会主义,发展,工作,十二五规划,法律,人民政协,十二五,立法,热烈祝贺,体系  
2012/03 人民政协,发展,工作,社会,建设,民政,人民,中国特色社会主义,会议,发挥

The same pattern changes in Nov 2012 when the political transition was underway during the 18th National Congress of the Communist Party of China. Almost all content emphasized the Party and the Congress:

Keywords in 2012/11

中央委员会 (The Central Committee of the Communist Party of China)

十八大 (18th National Congress of the Communist Party of China)

党和人民 (Party and People)

新一届 (New)

中国共产党第十八次全国代表大会 (18th National Congress of the Communist Party of China)

选举 (Election)

中国特色社会主义 (Socialism with Chinese Characteristics)

表现 (Express)

党的建设 (Development of the Party)

As we can see from this analysis, content in the People's Daily Online is closely related to the party. Most opinion stories published by the People's Daily Online repeats content of the same month from previous years. The political transition of the 18th National Congress of the Communist Party of China breaks the repetitive pattern and contributes new content to the opinion pieces.

### Work Cited

- Chang, Pi-Chuan, Michel Galley, and Chris Manning. "Optimizing Chinese Word Segmentation for Machine Translation Performance." (2008): n. page. Web. 9 Dec. 2012. <<http://nlp.stanford.edu/pubs/acl-wmt08-cws.pdf>>.
- Hu, Wei, Nobuyuki Shimizu, Hiroshi Nakagawa, and Huanye Sheng. "Modeling Chinese Documents with Modeling Chinese Documents with Topical Word-Character Models." *COLING '08 Proceedings of the 22nd International Conference on Computational Linguistics*. 1.345-352 (2008): n. page. Web. 9 Dec. 2012. <<http://dl.acm.org/citation.cfm?id=1599125>>.
- Zhang, Yunkai, and Zengchang Qin. "A Topic Model of Observing Chinese Characters." *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics*. 2. (2010): 7-10. Web. 9 Dec. 2012. <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5591057](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5591057)>.