Stock Financial News Automatic Summarizer

Group Member: Xiang Li, Luhan Shen, Jiawang Zhou

Preface

In the modern business world, security analyzers need to read thousands of news for their portfolio stocks, to figure out what's going on to those companies and hows that influence the market. In an efficient market, the stock price can change in hours after new information releases. Therefore, to analyze news content and sentiment as quick as possible, being able to classify and summarize financial news and get as much valuable information as possible, become important for equity researchers.

Our project is mainly focused on analyzing financial news for publicly traded stock news. The purpose is that by summarising financial news, which sometimes can be long and tedious, we can get a quick understanding of what has happened without spending too much time reading over the news.

In this project, Xiang Li and Luhan mainly focused on writing codes, and Jiawang mainly focused on writing reports.

The outline of our report will include instructions, procedures and python file introductions.

1. Instructions

You need to take the following steps to get prepared for running the code. We suggest you use a Mac because some required packages are not stable on Windows. To run the module it requires additional packages installed correctly as follow: cleanco, pandas, spacy, nltk, polyglot, datetime, numpy, scrapy, keras.

 # STEP 1: install python3.7 and python3.7-dev
sudo apt-get install python3.7-dev
# if return apt-get: command not found error:
* First, you need to install the Xcode command-line tool by using the following command:
xcode-select –install
*After the Xcode tool installation, now type/copy the following command to install Homebrew on macOS:
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
* The installation will ask for Return (Enter) key and password for confirmation.
* brew install python3

 # STEP 2: install ML packages
pip3 install pycld2
pip3 install pyicu
pip3 install spacy / or pip install spacy
if pip not working, run
$CFLAGS=-stdlib=libc++ pip install pycld2
brew install icu4c
ls /usr/local/Cellar/icu4c/
export ICU_VERSION=64
export PYICU_INCLUDES=/usr/local/Cellar/icu4c/64.2/include
export PYICU_LFLAGS=-L/usr/local/Cellar/icu4c/64.2/lib
export PATH="/usr/local/opt/icu4c/bin:$PATH"
pip install pyicu

 # STEP 3: install spacy packages
$ YOUR_INTERPRETER_PATH -m spacy download en(You can find the path by opening the terminal and typing “which python3” in Mac and “where python” in Windows)
# i.e.
$ C:\Users\Li Xiang\Anaconda3\python -m spacy download en

 # STEP 4: install nltk packages

```
>import nltk
>nltk.download('vader_lexicon')
>nltk.download('punkt')
```

 # STEP 5: install polyglot packages

```
$ polyglot download embeddings2.en
$ polyglot download ner2.en
```

#STEP 6: run the scraper

```
$ scrapy crawl nlp_news
```

Notification: it may take about 10 minutes to scrape all the news

Screenshot of finished spider:

```
2019-12-22 18:58:50 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
{'downloader/exception_count': 3,
 'downloader/exception_type_count/twisted.internet.error.TimeoutError': 3,
 'downloader/request_bytes': 83958,
 'downloader/request_count': 71,
 'downloader/request_method_count/GET': 71,
 'downloader/response_bytes': 7079383,
 'downloader/response_count': 68,
 'downloader/response_status_count/200': 68,
 'finish_reason': 'finished',
 'finish_time': datetime.datetime(2019, 12, 22, 23, 58, 50, 203748),
 'item_scraped_count': 303,
 'log_count/DEBUG': 376,
 'log_count/ERROR': 8,
 'log_count/INFO': 1454,
 'log_count/WARNING': 313,
 'memusage/max': 2621317120,
 'memusage/startup': 2367401984,
 'response_received_count': 68,
 'retry/count': 3,
 'retry/reason_count/twisted.internet.error.TimeoutError': 3,
 'scheduler/dequeued': 71,
 'scheduler/dequeued/memory': 71,
 'scheduler/enqueued': 71,
 'scheduler/enqueued/memory': 71,
 'start_time': datetime.datetime(2019, 12, 22, 23, 40, 23, 128518)}
I1222 18:58:50.205340 4554712512 engine.py:326] Spider closed (finished)
2019-12-22 18:58:50 [scrapy.core.engine] INFO: Spider closed (finished)
```

2. Procedures

2.1 Data Scrape (Scrapy and Request)

We applied a ‘Scrapy’ framework to run the news scraper. All the news that we scraped is from the website newsapi.org, which provides easy APIs for users to get a large bulk of wanted company news with many parameters, including ticker and time. In our case, we tried to get the news for the stocks that are stored in the file ‘companies.xlsx’ that happened in the past two hours from now.

In ‘nlp_news.py’, we defined the spider class with parsing methods to extract and parse the need information from the raw news. The ‘parse’ and ‘parse_article’ function are the core of our parsing procedure. By using a pre-trained sentiment analysis model and tagging model, we are able to get information such as sentiment, named entities, locations, genres that can give the users a really easy way to understand what the news is about. All the parsed attributes are then stored in the ‘item’ object and pass into the ‘pipelines.py’. In ‘pipelines.py’, we assign a tag telling whether the news is relevant to the ticker or not, and then store all the information to an Excel file directly.

2.2 Text Pre-Classification and Drop Ads

2.2.1 Block Ads Sources

Firstly, we would love to remove news from specific sources, of which are mostly irrelevant ads and can influence further NER and summarize. Here are some sample sources of the blocked sources: "Boardingarea.com", "Checkpoint.com", "Collider.com", "9to5google.com".

2.2.2 News Relevant Score and Relevance Judgement

After delete news from bad sources, we then dropped unnecessary news by calculating relevance score. The relevance score was calculated by checking if the right company name(or part of company name) appears in News Title. The detailed information of how we built the relevance score algorithm will be addressed later.

After determined how relevant this news is to our company, in the file “items.py”, we created three categories: “NewsScraperUpdateItem”, “IrrelevantNewsItem” and “AdsItem”. We only focus on scraper updated news and cast out irrelevant news and Ads.

2.3 Feature engineering (Tokenization and Vectorization)

We applied preprocess() by removing punctuation and clean bad symbols that can influence further Name Entity Recognition such as 's, .com, :, ', ", (,), -. Name Entity such as people’s name with those bad symbols can be mis recognized as company and vise versa.

We applied NLTK word tokenizer to tokenize and lemmarize. After unified the word format, we applied TF-IDF vectorizer to project text to word vectors, so the whole text is converted into a data frame with each content word as a feature and TF-IDF score as the values.

2.4 Name Entity Recognition

The NER includes the names of people and organizations, organization address and zip code. Entity recognition is to identify entities with specific meaning in text, which includes Entity class(person name, place name, organization name),Time class(date) and Digital class(phone number, zip code).

There are three methods of entity recognition:

(1) Linguistic grammar-based techniques : The Linguistic grammar-based techniques mainly based on grammar, and its application in the engineering implementation is to write a lot of regex, which can solve the recognition of time class and digital class named entities.

(2) Statistical models: At present, the statistical methods are mainly HMM and CRF models, which are also relatively mature at present.

(3) Deep learning models: The method of deep learning is the most popular way at present, especially the DL model of RNN series, which can absorb more text semantic information, and its effect is the best at present.

Here we adopted three NER models: Stanford NER, polyglot and spacy entities. StanfordNER is a java implementation of NER (named entity recognizer), it can mark the sequence of words in the text, such as person name, company name, gene name or protein name. It comes with a well-designed feature extractor for NER and many options for defining the feature extractor. There are many good English named entity recognizers, especially for person name, organization name, place name(Locations).

Polyglot language detection relies on pycld2 and cld2, among which cld2 is a multilingual detection application developed by Google. The training corpus of polyglot entity recognition comes from Wikipedia (wiki). The trained model has not been installed for the first time, so it needs to download the corresponding model. Polyglot supports the identification of entity classes (person name, place name, organization name) in 40 languages.

Spacy includes a fast entity recognition model – “spacy entities”, which can recognize entity phrases in documents. There are many types of entities, such as

people, places, organizations, dates, numbers. You can access these entities through the ents property of the document.

We created weighted function vote() that can pick the most appropriate NER result from this three algrothiums. After recognized the Person's Name, sometimes there's more than 1 person's name compacted together in the "Person Name" dictionary. Then we split different person's name by the name length

2.5 The classifier: News Category.

The classifier works like this: The keywords are searched in a dictionary of categories when a keyword is found then the class of text is stored and a score is attributed to the class. The same keyword often belongs to several classes but with different probabilities (the probability is computed as a score). Finally, a grade for each class is computed, and the class with the highest grade will be labeled as the most probable class for the text. We will assign tags to the news with categories like "person", "organization", "location", "event", "product terminology", etc.

2.6 Perform sentiment analysis

The result of sentiment analysis will be either "pos", which represents positive news, or "neg", which represents negative news. This part is very similar to what we did in assignment 5. Therefore, we applied the models we trained in assignment 5 to this summarizer. We tested all models and picked the one having the highest accuracy, which is exactly the one did best in assignment 5 - naive bayes classifier with raw counts of word tokens. Therefore, when we do sentiment analysis for a financials news, we would first use count vectorizer to transform it to a matrix with the same dimensions with what we generated from the movie review corpora, and then use naive bayes classifier with raw counts to predict its sentiment. For example, "China is about to start operation on its 'artificial sun' a nuclear fusion device that produces energy by replicating the reactions that take place at the center of the sun. If successful, the device could edge scientists closer to achieving the ultimate goal of nuclear fusion: near limitless, cheap clean energy." will be classified as "pos", which is consistent with its literal sentiment.

2.7 Result Validation and Conclusion

The result of our news summarizer comes out well, we don't have enough tagged and proved news summarize results to train or examine our final result, so instead we took random samples. For example, we scrapped news from the link: https://seekingalpha.com/news/3526800-freeportplus-1_5-bmo-turns-bullish / Freeport +1.5% as BMO turns bullish The result of our news summarizer will show the author name : Brandy Betz, this news is tagged as ['Analyst Estimates'] genre, the event sector

appears as “Basic Materials” mostly because Freeport-McMoRan Inc. (FCX) is a mineral Producer company based in Phoenix. All information mentioned above indicates pretty accurate results, however, the event organization was not well identified. Mostly because there are too many organizations in the news. This news is a stock price raise estimation from BMO(Organization1) about FCX(Organization2) because the transition of The Grasberg Mine (Organization3) is progressing ahead of schedule.

It would be better if we can train our own model on NER specifically for financial and business news and create classification tagger by training our own Naïve Bayes Model. It is a huge pain to yield accurate enough result when we have deficient confirmed summarized results for training. We believe as people who used the summarizer trained the model with more and more consolidated data, this summarizer can be more functional and well-performed in the future. Our summarizer can be used as a prototype.

3. Key Python Files Introduction

3.1 Get_companies.py:

This main role of this module is to perform a pretty print of the names of a company with the help of “cleanco” package. It can strip out redundant symbols in a company name. For example, “XXX L.L.C.” will become “XXX”. The ticker of the company will also be printed.

3.2 Relevance_ score.py:

The main role of this module is to check the relevance score of a news, which represents how relevant the news is as for the company. We first defined the function “preprocess”, which can strip symbols like '.com', '.', ':', '!', '?', ',', "\'", '\"', '(', ')', 'nyse', 'nasdaq', 'business wire'. Then we defined the function “get_tags” because we wanted to tag the key words in a news. We downloaded from Yahoo a file, which lists all financial news categories and words belonging to each category. And we use the “get_tags” function to create a dictionary that mapped each word with its belonging category so that when we look at a financial news, we can tag all keywords by referring to the dictionary. Then we defined the function “is_good_doc”, which is a judgement function that test whether a news mentioned keywords. Finally, we defined the “relevance_score” function. The algorithm behind is: if the news contains the ticker of the company or mentions the name of the company more than twice, the relevance score is assigned to 2; if the news mentions the name of the company only once, the relevance score is assigned to 1; if the news does not mention the name of the company, the relevance score is assigned to 0.

3.3 Get_sentiment.py:

Financial news can also have sentiment - is it positive news or negative news? The main role of this module is exactly performing a sentiment analysis on the news. Here, we took a shortcut - we looked at the pre-trained model in assignment 5 and picked the one with the highest accuracy, the raw count naive bayes classifier. Since there should be no significant linguistic differences between movie reviews and news, we thought this model can also be applied on this project. We used it again so that we can predict the sentiment of financial news.

3.4 Get_genres.py:

Each financial news discusses a topic or genre, which is very useful if a user wants to look at news according to a certain topic. To tag a document, we used two algorithms simultaneously. One is to get the tag directly from the “get_tag” module we built in relevance_score.py, the other is to predict out the tag from a pre-trained tagging model. Since a financial news may involve several topics, we used the function “argsort” in pandas to get the indices of tags with a reasonable predicted probability(here we set the threshold as 0.19). However, when we find only one or two genres have the predicted probability higher than the threshold, we will back off a little bit to check whether other genres still have acceptable predicted probabilities. For example, if we only get one predicted genre with the predicted probability higher than the threshold, we will check whether the second highest one has the predicted probability higher than half of the threshold. If it does, we will still include it as our predicted genre. Additionally, knowing the fact that the predicted results of two algorithms may have conflicts, we created a list called “small topics”, which includes the top 20 hot topics of financial news - 'Cloud Computing', 'Automotive', 'Machine Learning', 'Intellectual Property', 'Startup', 'Trade', 'Corruption', 'Taxation', 'Credit Market', 'Fraud', 'Blockchain', 'Recession', 'Economic data', 'Fixed Income', 'Emerging market', 'Technical Analysis', 'Antitrust' and 'Tariff'. If the predicted results of the second algorithm does not fall into these small topics but the first one does, we will manually add it into the predicted genres.

3.5 Get_tags.py:

Each financial news will describe an event, and each event will have its main character(here we label it as person), its happening place(here we label it as location) and its interest group(here we label it as organization), which are the keywords. To manually extract those keywords may involve model training, which is quite complex. Therefore, we applied a module called “StanfordNERTagger”, which is like a pre-trained model that can identify characters, place names and organization names. Similarly, we defined functions to remove redundant symbols like "\'s", '.com', ':', "\"", '\"', '(', ')', '-' and useless digits. To prevent the module return entities that are substrings of other entities,

we defined the function “vote”. Finally, we created our output functions “get_named_entities”, which can extract event persons, event location and event organizations of a financial news.

4. Summary

In this project, we mainly focused on crawling financial news from certain websites and digging useful information from the news. An excel is attached in the root directory, which is our final result after crawling. Below is a glimpse on our result excel table.

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
author	content	date	escriptiv	ent_geni	nt_locat	organiz	ent_pers	ent_sect	ent_tick	minute	blished_	relevanc	entimen	source	ticker	timestam	title	url
Hannah O	China is	2019-12-	It is ho	['Commod	[]	[]	[]	Energy	SWN	16:34:55	2019-12-	irreleva	pos	Newsweek	SWN	1.58E+12	China Is	https://ww
SA Edito	BMO upgr	2019-12-	BMO upgr	['Analyst	[]	[]	[]	Basic Ma	FCX	14:55:57	2019-12-	relevant	pos	Seekinga	FCX	1.58E+12	Freeport	https://see
	WASHINGT	2019-12-	Sunday, I	['Politic	[]	['Trump A	[]	Energy	HAL	16:35:35	2019-12-	irreleva	pos	Commondr	HAL	1.58E+12	Sunday M	https://ww
Will Sch	Along th	2019-12-	Hundreds	['Politic	[]	[]	[]	Energy	CHK	15:15:00	2019-12-	irreleva	neg	Ggwash.o	CHK	1.58E+12	By 2025,	https://ggw
Michael	Pfizer (	2019-12-	New Pfize	['Pharma	[]	[]	['Albert	Healthca	PFE	16:27:59	2019-12-	relevant	pos	Seekinga	PFE	1.58E+12	Pfizer:	https://see
By ALAN	RICHMOND,	2019-12-	RICHMOND,	['Politic	['Richmo	[]	[]	Energy	CHK	14:57:57	2019-12-	irreleva	neg	Associat	CHK	1.58E+12	Northam	https://apn
	SHANGHAI,	2019-12-	SHANGHAI,	['Corpor	['Shangha	['Discov	['Jie Ja	Healthca	PFE	16:00:00	2019-12-	irreleva	neg	Prnewswi	PFE	1.58E+12	Dr. Jie	https://ww
	WASHINGT	2019-12-	Sunday, I	['Politic	[]	['Trump A	[]	Consumer	M	16:35:35	2019-12-	irreleva	pos	Commondr	M	1.58E+12	Sunday M	https://ww
Paul Aus	After tal	2019-12-	PG&E has	['Partner	[]	[]	['Gavin N	Utilities	PCG	16:40:36	2019-12-	irreleva	pos	247wallst	PCG	1.58E+12	PG&E Pla	https://247
Kazunori	December	2019-12-	December	['Product	['San Jos	[]	[]	Consumer	F	15:30:13	2019-12-	irreleva	neg	Playstat	F	1.58E+12	GT Sport	https://blo
David Ka	Author's	2019-12-	The S&P	['Politic	['China'	[]	[]	Healthca	DHR	16:00:03	2019-12-	irreleva	pos	Seekinga	DHR	1.58E+12	10 Stock	https://see
PR Newsw	SHANGHAI,	2019-12-	Shanghai	['Corpor	['Shangha	[]	['Jie Ja	Healthca	PFE	16:00:00	2019-12-	irreleva	neg	Yahoo.co	PFE	1.58E+12	Dr. Jie	https://nev
Ted Andersen		2019-12-	PG&E (NY	['Politic	[]	[]	['Gavin N	Utilities	PCG	16:40:27	2019-12-	relevant	pos	Bizjourna	PCG	1.58E+12	PG&E rem	https://ww
Kazunori	December	2019-12-	WeatherTe	['Product	['Norther	[]	[]	Consumer	F	15:30:08	2019-12-	irreleva	pos	Playstat	F	1.58E+12	Gran Tur	https://blo
PR Newsw	DUBLIN, I	2019-12-	The "Next	['Analyst	[]	[]	[]	Healthca	DHR	16:00:00	2019-12-	irreleva	pos	Yahoo.co	DHR	1.58E+12	Next Gen	https://nev
Anomali	The inte	2019-12-	The inte	['Politic	[]	[]	[]	Communic	T	15:52:00	2019-12-	relevant	neg	Anomali.	T	1.58E+12	Weekly Th	https://ww
Chris We	Being a	2019-12-	Embrace	[]	[]	[]	[]	Basic Ma	X	16:17:55	2019-12-	irreleva	neg	Hiconsum	X	1.58E+12	20 Last	https://hic
	DENVER--	2019-12-	DENVER--	['Pharma	[]	['Pfizer'	[]	Healthca	PFE	15:06:39	2019-12-	relevant	pos	Business	PFE	1.58E+12	Colorado	https://ww
Ted Andersen		2019-12-	PG&E (NY	['Politic	[]	[]	['Gavin N	Utilities	PCG	16:40:27	2019-12-	relevant	pos	Bizjourna	PCG	1.58E+12	PG&E rem	https://ww
Jeremy Bo	A pilot	2019-12-	Startup	['Partner	[]	[]	[]	Consumer	F	15:17:58	2019-12-	irreleva	pos	Forbes.co	F	1.58E+12	SkyRyse	https://ww
	DUBLIN, I	2019-12-	DUBLIN, I	['Analyst	[]	[]	[]	Healthca	DHR	16:00:00	2019-12-	irreleva	pos	Prnewswi	DHR	1.58E+12	Next Gen	https://ww
Jeff Pol	The Deca	2019-12-	The Deca	['Politic	[]	[]	[]	Communic	T	15:01:51	2019-12-	irreleva	pos	Forreste	T	1.58E+12	Decade R	https://go.
The Ener	The utte	2019-12-	The utte	['Politic	[]	[]	[]	Basic Ma	X	15:46:57	2019-12-	irreleva	neg	Energyce	X	1.58E+12	Victor: A	https://ww
	NEW YORK-	2019-12-	NEW YORK-	['Earning	[]	['Nyse Pi	[]	Healthca	PFE	15:06:39	2019-12-	relevant	pos	Business	PFE	1.58E+12	Pfizer In	https://ww

We can see that, although most columns can be filled with according values, event location, event organization and event person contain lots of NA. This is partially due to the fact that not every news has such values, but it is also because the name entity extraction part in our project is not good enough. If we can have more accurate name entity extraction model, more of the blanks can be filled.

Another problem we found during this project is that, some required packages, for example, “pyicu” – a package dealing with unicodes, is not stable in Windows, although it is in Mac. This may cause potential inconvenience for users who use Windows as their operating systems. We hope the creator of this package could upgrade it to make it stable in Windows.

Finally, the time takes for crawling the 450 financials news is long, which is not very good because finance world changes quickly, and finance people may want to look at news every minute. We think that this defect may derive from my codes – we are still immature coders. If we become more and more experienced in coding, we could streamline our algorithms much better.