Named Entity Recognition with Context Extraction for News Texts

Aditya Panikkar
Auckland University of Technology,
jkc2084@aut.ac.nz

Rudra Sakhardande
Auckland University of Technology,
swg7857@aut.ac.nz

***Abstract*—News articles generate tremendous amount of data everyday. The enormous size of this data not only make it difficult to analyze but also incurs huge costs to businesses. The paper proposes a system that not only extracts the names of top 5 trending companies but also extracts the reasons for their mentions in the news articles. The extracted context is represented in two ways. One is a graphical representation using word clouds and the other is a textual representation with company context data contained in text files.**

I. Introduction

News articles are one the most common and ubiquitous media through which people exchange information and gain knowledge. The amount of information the news articles generate on a daily basis is tremendous and hold great significance for everyday decision making. Knowledge extracted from such texts not only influences the lives of common people but also form the basis for many business decisions and research. For instance, news articles reporting various market trends and stock patterns are used by enterprises and investors to make crucial business decisions. The content reported in news articles is seen to influence the psyche of an individual and in turn drive trends in the market based on which important decisions are taken by investors and entrepreneurs. The sheer amount of the content created from news articles is one of the aspects that make it even more difficult for analysis. In order to physically analyse such articles for a particular topic, the individual has to sift through hundreds of articles in order, filtering through irrelevant information and extracting information that is relevant to what he is looking for. Such a process is not only time consuming but may also affect the productivity of the analyst performing the job. Also for a business to perform such analysis for decision making would mean employing a large amount of people to get the work done in lesser time thus implying huge costs for the company. This where text mining plays a huge role in extracting relevant information and meaning from large quantities of text. A company using text mining techniques to analyze data such as news articles would not only save time but also a significant amount of cost with respect to the inefficiency that manually scrutinizing each news article brings.

One of the use cases of particular significance is the extraction of company names and their context in a body of news text. This information is of particular interest to investors and entrepreneurs who consider performing a detailed research before executing professional decisions. For instance, and investor can use a text mining system that gives it company names with their context in order to gain an insight about the working of the company before making a decision to invest in its operations. If a company comes in the news too many times for the wrong reasons the investor can immediately be assured of the background of the company and decide not make an investment in such a venture. Such a system is also of great significance to venture capitalists that have to perform a detailed risk assessment of the companies they would like to invest in. Text analysis of news articles for company name could also be indicative of brand images even corporate image of the company in the minds of the general public. This can prove extremely useful for a relatively smaller sized company who would want to market itself and create its own image through mergers and associations. Also there exist companies that show particular interest in endorsing certain initiatives or types events. For instance, companies like Coca Cola, Pepsi, Nike, Adidas and so on are more likely to endorse sporting events. Identifying such information could particularly aid smaller sporting companies that organize events to get sponsorships from the larger corporates that endorse their sport. Text mining can similarly be used by Non Profit Organizations to identify initiatives and businesses that are particularly interested in providing donations organising events for certain causes as a part of their Corporate Social Responsibility (CSR) or for other reasons. This would help them approach a company that would be genuinely be willing to invest in their cause. An analysis of the top companies that appear in the articles could also be indicative of the current world trends with respect to business sectors and their popularity. Such trends could also be compared over the years to get an overall view of the consistency or variation in top trends. For instance, a case where the top 5 companies in the news are related to the Information Technology sector could be indicative of the fact that the current trend in industry is primarily related to the IT domain.

Thus it can be observed that the extraction of company names as named entities along with their context is of significant importance to businesses and can provide valuable insights based on which important decisions can be made. Such a system if designed would provide a fundamental framework that would in turn act as a knowledge base on which businesses can build their decision making infrastructure.

II. Literature Review

Two of the preliminary steps in the direction of extracting information from texts are Part of Speech Tagging and Named Entity Recognition:

A. Part of Speech Tagging

POS tagging is one of the fundamental steps in mining text. The process consists of assigning Part of speech tags to an input text. POS tagging starts with the process of tokenization. Word tokenization is a process that separates words from an input text for analysis. In other words the process creates a list of words from an input text. The tokens can be either, words sentences or other meaningful elements of the syntactic structure. POS tagging can be broadly classified under the following categories:

1) Supervised Taggers: These taggers need data that is manually annotated. The annotated data is then used by the tagger to train itself and then the extracted features are given a machine learning classifier. The classifier then uses the tokens that are feature represented to their corresponding POS tags. Such models work best on test data that is very similar in nature to the training data [1].

2) Unsupervised Taggers: These taggers analyze large quantities of data and then group words by similarities in their context. The words that are grouped together are taken to implicitly belong to the same class. For unsupervised taggers the class name is not made explicit due to which they are added as features in a supervised technique [2].

3) Rule-based Taggers: In this case various tokens are mapped to POS tags based on a particular set of rules or patterns. This why they are called rule based taggers. The rules in such cases are explicitly specified and the model does not learn features on its own [1].

B. Named Entity Recognition (NER)

This process involves classifies the various named entities into predefined categories or classes. NER forms a very important part of information extraction. In any text named entities such as locations, countries, names of people and companies form an important part. Hence the extraction of such entities provides important information thus helping in knowledge extraction. The who, where and how much of a body of text serve as significant sources of information. Also in a processor chain, NER can be implemented as a primary step where the NER step can be followed the creation of semantic link between two named entities using a verb which would also help understand the what and how of a piece of text.
In this section, we review some of the existing works done in the field of POS tagging and NER by various people.

The following research paper discusses the various design challenges and misconceptions with respect to names entity recognition. The model proposed employs expressive features in order to achieve a good performance with NER tasks. It also stresses upon making use of prior knowledge and also non-local decisions in named entity recognition. The paper cites the following example for demonstrating its idea, a system based on the proposed model when passed a body of sports news text, recognizes the word BLINKER as the name of a person and the word Wednesday as referring to the club Sheffield Wednesday thus recognising it as an organization. Blinker was recognized as a person even though it was in a different case as the word was later identifiable in the in the text as Reggie Blinker. The club Udinese was also identified as a soccer club due to the existence of a related entry in Wikipedia. Thus it can be seen that the performance of a tagger is dependent on the presence of external resources. Text chunks representation, inference algorithms, use of non-local features and external knowledge were identified as the important design features for an NER task. Three different algorithms were compared for performance, namely, greedy left to-right decoding, Viterbi and Beam search. The Viterbi algorithm was found to prevent the incorporation of some of the non-local features. The comparison was performed for both the baseline features system and then thee end system. The result found the greedy algorithm to perform the best which corroborated with result achieved in the POS tagging experiment [3]. The study states a reason for the lesser performance displayed by Viterbi by demonstrating that the named entities are in fact separated by outside tokens and are short chunks. The Viterbi algorithm is broken by this independent separation into maximization of assignment over chunks due to which the greedy algorithm performs well. . However, dependencies among named entity chunks which are isolated have longer range dependencies and are not captured by second-order transition features. This requires other mechanisms. The BILOU encoding scheme was found to be better than BIO scheme. The experiments conducted demonstrate that word class models learned on unlabeled text could act as alternative choices to the traditional semi-supervised learning frameworks. The research also illustrates NER to be a knowledge sensitive task and that model built on such a model can also adapted across several frameworks [4]. One technique for extracting company names from text makes use of exception lists, heuristic combinations and extensive corpus analysis. The technique primarily spots a company suffix and then reads backwards from the suffix. For input that is mixed cased, the technique utilizes a combination of search methods for the first word that is not capitalized and heuristics for handling embedded conjunctions and other words that arent capitalized. For input that is in upper case, the method makes use of a stop list that is empirically derived, restrictions on company name word length, and lexical lookup to determine the start of a company name. Once a company name is recognised, the algorithm will generate the most likely variations for the company name so that it can be used in subsequent retrieval.

Another approach is the deep learning approach done by [5]. The authors have done NER for bio-medical texts.The NER in bio texts is a little different form the normal NER done for corpuses. The aim for BioNER is being able to successfully identify entities like diseases, genes, species and proteins. The authors consider BioNER to be more difficult than the usual NER process. The reasons they give are longer names of species, constant discovery of new diseases/species and different descriptions for the same biological entity. The previous approaches for doing NER in Bio texts have limitations which have also been listed in the paper. Deep learning approaches have given good performance in various domains like speech recognition, image classification and POS tagging but require a lot of labelled data for carrying out the supervised learning.Therefore, the authors have used an n-gram based CNN method for carrying our NER in Bio-medical domain. The main steps involved were word embedding and a Gram-CNN resolution. Embedding is a process where words are represented as vectors containing real numbers. The Gram-CNN model was then used to extract the information that is local to a word. The IOB2(Inside, outside, beginning) approach was then used to tag every word in the sentence. This developed tagger gave an F1 scores of 87.26% in 2 bio-text corpuses and a 72% in a third corpus.

An approach for extracting company names from financial texts has been explored by [6]. The survey conducted by the author states the presence of unknown words in a text as one of the most important problems in the analysis of texts. These unknowns constitute over 8% of the text. Over 4% of these unknowns are company names and serve as the most important information that is required to extract high level information like what the text is about. The author has therefore created a heuristic set of methods to identify the names of entities and their variations. The methods used are All-capital stop conditions, conjunctions, text segmentation, variation generator which aid in Named Entity Recognition. The all-stop method has a stop-list of the most common words found that directly precede the company names. This is done through extensive analysis of various corpuses. The conjunction method deals with company names that are embedded with commas and singular verbs. The word 'of' can also signal a conjoined company name. Such rules have been written in this method. Segmentation is done to interpret and understand the sentences. Predefined abbreviations of words have been defined to recognize variations in the company names and these are resolved at the tagging stage. The results showed an accuracy of 97.5% for extracting company names from a particular corpus. These were compared and verified by Human assigned tags for determining the accuracy of the tagged text.

The authors have discussed the various systems that took part in the CoNLL shared task of Named Entity Recognition [7]. The dataset consisted of the news corpus comlied by reuters. The data was in two languages, German and English. The data files contained pre tagged entries. Each line in the training set consisted of the word, its POS tag, the chunk tag and the named entity tag. There was a separate development set available to fine tune the parameters for the models developed. There were 16 different systems. 5 systems were based on the maximum entropy model, Hidden Markov Models were used by 4 systems, connectionist approach based models were used by another 4 systems, other systems had less frequently used models like CRF, AdaBoost.MH, memory based and SVM. The paper does not provide a detail of the individual framework. The Maximum Entropy based model gave the highest F score of 88.76 for the English texts and 72.41 for the German texts. This paper does not give the framework but suggests that maximum entropy model has given good accuracy and can therefore be explored further.

The authors in [8] have presented two types of taggers. The motivation for them to develop their own systems was their belief that the current named-entity recognition methods rely heavily on hand-crafted features and a lot of domain specific knowledge. All the present taggers work by learning from a small supervised subset of data and therefore are heavily reliant. The authors propose two architectures, one os based on a bi-directional long-short-term-memory (LSTM) and Conditional Random Fields, the other is based on logic which creates segments and labels them using a transition based approach which is inspired form shift-reduce parsers. The recurrent neural network (RNN) based models show evidence of being biased toward their more recent input. Thus, to combat this the authors use a LSTM model. For any given sentence, the LSTM creates a representation of the left hand side context of the given word. Another LSTM computes this for the right hand side of the word. This is done by giving the input in reverse. The left and right contexts are then concatenated to form one joint context representation. A CRF tagger is used to model tagging decisions. The other approach chunks and labels the input sequence.A stack based LSTM model is used which pushes word embedding into a stack and also permits their removal unlike the traditional LSTM model which uses left and right context. The resulting models have given a good accuracy. The LSTM-CRF tagger has an accuracy of 90.94% on the CoNLL-2003 test set and the segmentation LSTM has an accuracy of 90.33% on the same dataset.

A similar model based on the LSTM and CNN has been explored in [9].They use a hybrid bidirectional LSTM and a CNN architecture that helps eliminate the need to perform extensive feature engineering. Words and characters are transformed into a continuous vector for representations. All these are concatenated and fed into a bidirectional long-short-term-memory neural network. A convolution neural network is then used to reduce the word features. The system achieves a state of the art accuracy in the range of 85% to 90% on various tree banks. An HMM based chunk tagger approach has been explored in [10]. The model assumes that the information is mutually independent unlike the traditional conditional probability approach of the traditional HMM model. The model has different feature recognition logic embedded in it. The resulting model gives an F score of 96.6 on the MUC-6 corpus and 94.1 on the MUC-7 corpus.

A language independent Named Entity recognition system ahs been proposed by authors in [11]. The algorithm used is based on iterative learning and re-estimation of morphological patterns. The advantage of the algorithm is that it learns from text that is not annotated and gives a significant performance when trained on a short labelled data with no tokenization or language specific information. The algorithm considers that suffixes and prefixes to the word are good indicators. Examples given are for name suffixes which include ”son” for english, ”ovic” for polish, ”ivic” for serbian and many more. The bootstrapping algorithm learns such patterns in the words. Contextual pattern recognizes words like ”Mr”, ”mayor of” as important indicators of entities. The resulting system gives a very competitive F score of 75.4% for Romanian texts and a distribution between 73% to 79% for 5 other languages. The system achieves this for multiple languages and hence is language dependent.

Cononical Corelation Analysis to obtain embeddings in a lower dimension for determining candidate phrases for NER has been explored in [12]. Their system constructs automatic dictionaries for Named Entity Recognition by using unlabelled data. A Bio-med corpus was searched to determine various candidate phrases. Words like virus, diseases were searched and noun phrases were extracted from sentences that had such mentions. Then, context was determined was such words. This meant finding 3 words both to the left and right of the candidate phrase to generate 2 views that CCA algorithms can exploit. These lower embedded word representations were fed into a CRF tagger to determine the Named Entities. The resulting F scores were in the range of 75% to 80%.

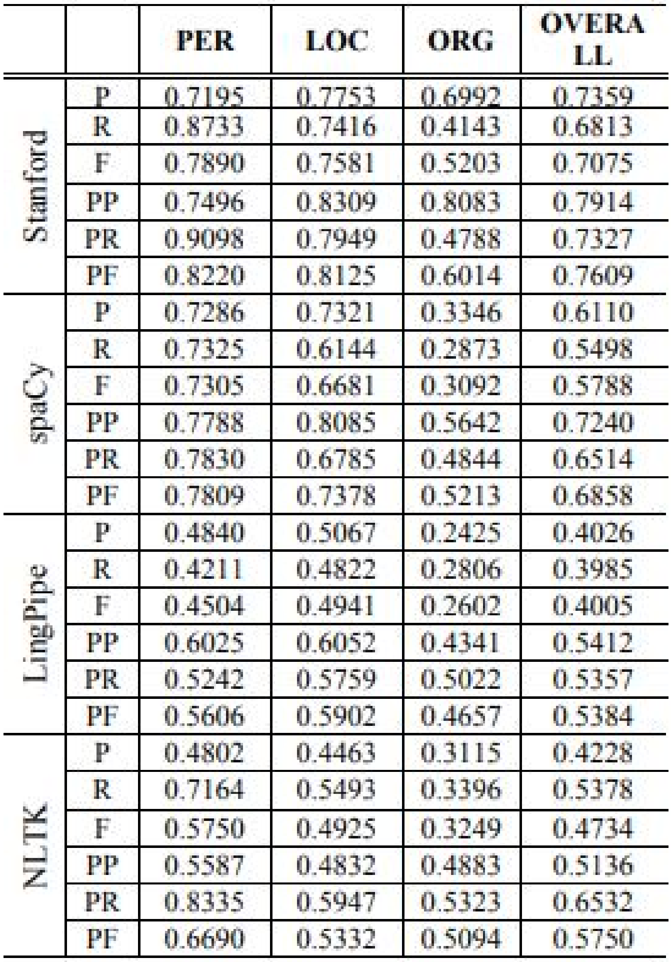
The following research compares various NER tools based on the following criteria:

- The tool is available freely and allows unlimited usage
- It can be downloaded and installed locally and works well with default setting
- The tool is not limited in the sense that it is trained for a particular domain
- The tool must be able to recognize the basic three entity types of person, location and organization

The tool must be able to recognize the basic three entity types of person, location and organization.Following taggers were then selected:

- Stanford NER [13]
- Spacy
- Alias-i LingPipe [14]
- Natural Language Toolkit (NLTK) [15]

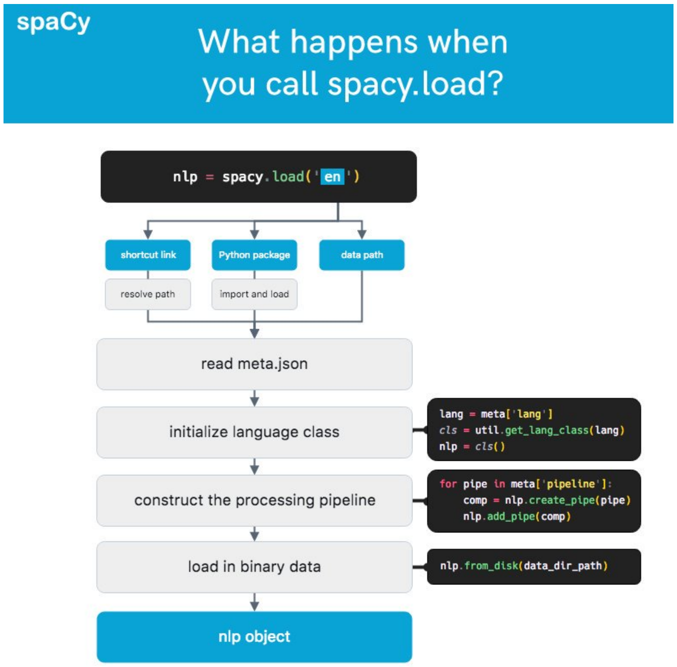
A comparative evaluation of the four NER tools, namely, Stanford NER, spaCy, Alias-i LingPipe and NLTK was performed. For validation purposes, a framework was designed in python, which can seamlessly work with different NER systems developed in a variety of programming languages. The output can be produced dynamically in both text documents or excel tables. The selected NER tools were then evaluated by using the gold standard corpus and manually annotated datasets developed by the team. The study showed that Stanford NER displayed the best performance across all datasets.The spacy tagger was the second best. However the spacy NER tools was found to have fastest processing speed. Stanford NER showed a general good performance on all tested datasets. However Spacy slightly outperforms Stanford NER with respect to the DATE entity type. On further assessing the false alarm and missing errors, it was observed that the Stanford NER faced difficulties in identifying the full date information from the body of text. For e.g., for the line on 1 February 1858, it can only identify February 1858, the date is always missing. The source of the problem could lie in the fact that Stanford NER is not trained for the date month year date format. An alternative solution of using the rule-based Temporal Tagger was suggested. A hybrid NER tool as then designed for the application domain by combining the best two performing NER tools [16].



		PER	LOC	ORG	OVERALL
Stanford	P	0.7195	0.7753	0.6992	0.7359
	R	0.8733	0.7416	0.4143	0.6813
	F	0.7890	0.7581	0.5203	0.7075
	PP	0.7496	0.8309	0.8083	0.7914
	PR	0.9098	0.7949	0.4788	0.7327
	PF	0.8220	0.8125	0.6014	0.7609
spaCy	P	0.7286	0.7321	0.3346	0.6110
	R	0.7325	0.6144	0.2873	0.5498
	F	0.7305	0.6681	0.3092	0.5788
	PP	0.7788	0.8085	0.5642	0.7240
	PR	0.7830	0.6785	0.4844	0.6514
	PF	0.7809	0.7378	0.5213	0.6858
LingPipe	P	0.4840	0.5067	0.2425	0.4026
	R	0.4211	0.4822	0.2806	0.3985
	F	0.4504	0.4941	0.2602	0.4005
	PP	0.6025	0.6052	0.4341	0.5412
	PR	0.5242	0.5759	0.5022	0.5357
	PF	0.5606	0.5902	0.4657	0.5384
NLTK	P	0.4802	0.4463	0.3115	0.4228
	R	0.7164	0.5493	0.3396	0.5378
	F	0.5750	0.4925	0.3249	0.4734
	PP	0.5587	0.4832	0.4883	0.5136
	PR	0.8335	0.5947	0.5323	0.6532
	PF	0.6690	0.5332	0.5094	0.5750

Fig. 1. Comparison of Various taggers

Identifying relation between words and phrases contained in documents hold significant implications not only for knowledge extraction but also for question answering systems and text summarisation. The prior efforts in this direction have require a large amount of effort and resources by making use of a large annotated corpora for identifying relations. The following study hence proposes an unsupervised method for relation identification. The technique makes use of similarity of intervening context words in order to cluster pairs of named entities. For instance, paired of named entities were considered such as PER-GPE where PER denotes the name of person and GPE, a geopolitical entity like USA or Canada. The occurence of such named entities in pairs was termed as a



```
nlp = spacy.load('en')
```

```
lang = meta['lang']
cls = util.get_lang_class(lang)
nlp = cls()
```

```
for pipe in meta['pipeline']:
    comp = nlp.create_pipe(pipe)
    nlp.add_pipe(comp)
```

```
nlp.from_disk(data_dir_path)
```

Fig. 2. What Happens When You Call spaCy

co-occurence and was defined as the occurrence of the two named entities in pairs with N number of intervening words. The technique proves that it can not only be used for the detection of relations between named entities with high recall and precision but also for assigning appropriate labels for the relations.The model showed a particularly good performance in the PER-GPE domain. This can be attributed to the fact that there were more Named entity pairs with high cosine similarity in the PER-GPE domain than in the COM-COM domain where COM denotes Company tag. The graphs in both the domains were similar particularly when the cosine similarity was less than 0.2 [17]. The following study proposes the design of a system which takes company announcements, categorizes these into a specific amount of report types and then performs information extraction in order to identify particular elements of information in every report types. The system consists of three processes:

- Text Categorization: The categorizer is trained on documents that are manually annotated using the Signal G corpus. This is then used to determine the report type of each document.
- Information Extraction: Given the documents type, we then apply a collection of IE routines that are suited to particular document types. Thus the key elements of information which are relevant for that document type are located.
- Information Rendering: The information once extracted can be represented in a variety of ways to different users.

The system performs particularly well but faces certain issues with respect to variations encountered in proper nouns and conjunctions of proper nouns. Effort is also being taken in order to make the system learn from its previous experience [18].

III. System Design and Implementation

The system architecture that was chosen was greatly dependant on the nature of data to be analyzed.

A. Data Description

The objective of the proposed system is to design and implement a system that allows the extraction of useful information which can be used by businesses for decision making. The task to be performed involves the extraction of the top five companies that have appeared in the news articles. Since the data to be mined is from news texts the nature of data is more or less structured. This is one of the reasons why the system implemented does not have to deal with a lot of noisy data as seen in the case of micro-blogging texts. Also the articles to be mined are stored as 2500 text files due which might necessitate certain techniques in order to access each file or aggregate all data as one. the following section details the overall architecture of the system

B. System Architecture

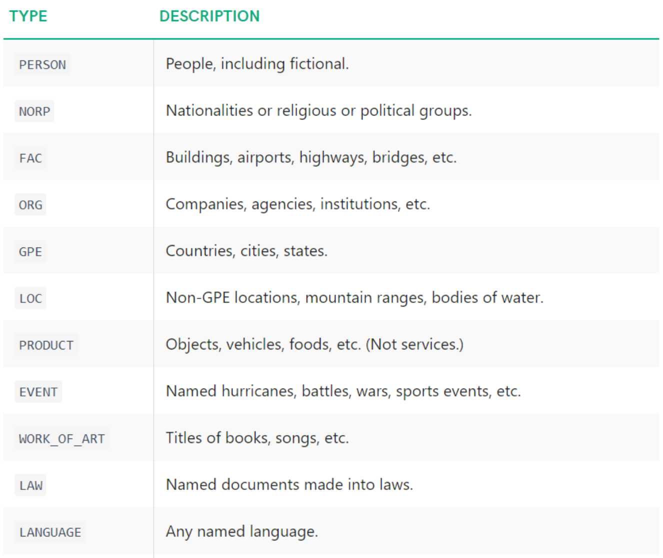
The overall system architecture consisted of 3 main components:

- POS tagging
- Named entity recognition
- Context extraction

1) POS Tagging: The POS tagger that was used for the text mining task was the spaCy tagger.The spaCy tagger was chosen by its decent comparative performance with other taggers and its fast and intuitive functionality. The spaCy tagger is trained on the OntoNotes5 corpus and supports various entity types thus providing a large number of functionality to extract named entities [19].

The python programming language was used in as a framework to design the system. The os library was used in order to access the repository of various files containing the news articles. The listdir() function of the os library was given the home path of all the news articles and was used to parse through each file, one at a time. Thus the POS tagging was thus performed for each news article, one at at a time. Thus the process of combining all the text from each file into a single text file was eliminated.

The system also makes use of the en_core_web_sm model developed for English which displays an accuracy of 96.78% for POS tagging and NER accuracy of about 86%. for the POS tagging process the text in the news article was first read into a variable which was then passed on the ’nlp’ function provided by spacy. The nlp function takes a string as an input, word tokenizes it and then assigns a part of speech tag to each word in the nlp text. the result of the nlp function used was assigned to a variable named ’docs’. The ents() function provided by spaCy was then used in order to extract only the



TYPE	DESCRIPTION
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language.

Fig. 3. NER Tags provided by spaCy

named entity types from the the ’docs’ variable. The further process is detailed in the next section:

2) Named entity Recognition: The ents() function was used in order to extract all the named entities. the entities that were extracted were a mix of various entity types that spacy provides. spaCy provides different entity types such as PER for persons, GPE for Geopolitical entities like countries and cities,ORG for companies and organisations, LOC for non GPE entities like mountains and continents, LANGUAGE for the spoken languages that exist, DATE for the dates in text and many more. For the purpose of the task at hand the ORG tag was of particular interest and would help in the extraction of the company names from the text. A condition for extracting all the entities with the ORG tag was specified thus enabling the extrication of the company names which were then appended to global list. The process was then repeated iteratively for all the other files. Therefore the company names extracted from each article,i.e., from each text file would be appended to global list. When all the files were accessed the global list contained all the company names that were mentioned in the news articles combined. It must be noticed that this list is not a set and contained multiple mentions of the same company name which enabled us to count the number of times a particular company came in the news.

In order to count the number of mentions the Counter function from the python ’collections’ module was used. The Counter function allowed to count the total number of mention of each company name in the list. The most_common function from spaCy was then used in order to get the top n number of companies from the list. Thus a list of the top 5 companies was obtained. However there was certain issue that had to be dealt with. There were certain entities in the in the text that were erroneously tagged as organisations. For instance the list of the top 5 companies that was initially received was Reuters, EPS, ING, Microsoft, PCT. On verifying the results it was found out that EPS and PCT were not companies and that the tagger had erroneously tagged them as organisations probably because of their case. Reuters though was correctly tagged as a company could not be considered as one for this task. Reuters is a news organisation and all the mentions that were found in the news articles consisted of news about other companies reported to Reuters. Therefore the articles had many lines which had the phrase ’told Reuters’ at the end. Hence an updated list for the most mentioned companies was generated which consisted of ING at the first position followed by Microsoft,Chrysler, Ofgas and British Gas.

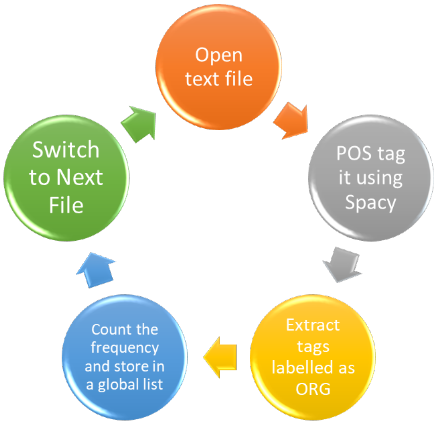

Open text file
POS tag it using Spacy
Extract tags labelled as ORG
Count the frequency and store in a global list
Switch to Next File

Fig. 4. Process for extracting top 5 Organizations

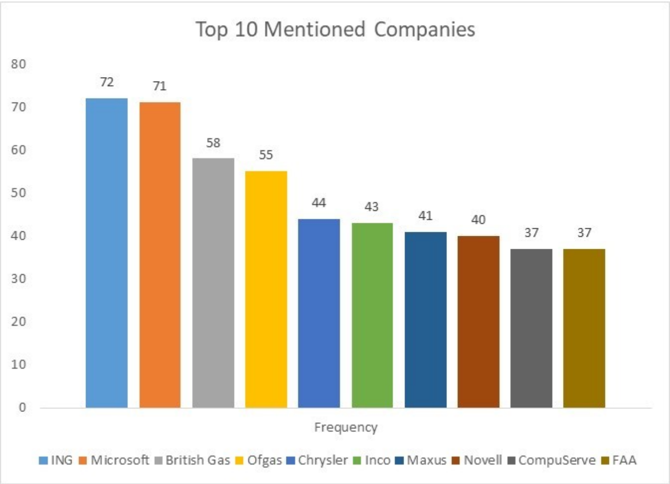


Fig. 5. Top 10 Organizations in the News Texts

C. *Context-Extraction*

The next step involved the extraction of context from the news articles. For the purpose of the task it was required to find the reason for which the top companies had appeared in the news articles. The first step taken in the direction of extraction context was the context representation. The context to be extracted was required to be readable by a journalist. For this purpose the context representation was to be performed in two ways. The first way would be identifying the context for a company name and then writing it to a file with the name of the company. Any new context found would be appended to the same file. The other way was the representation of words as some visual pattern and representation that would link a particular word to its context which are again words or groups of words.

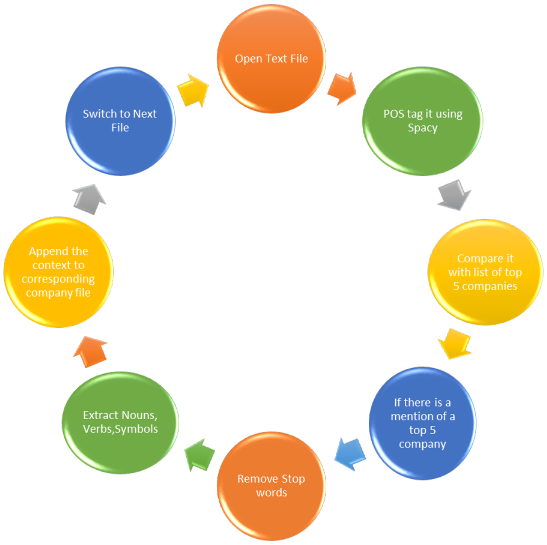

Open Text File
POS tag it using Spacy
Compare it with list of top 5 companies
If there is a mention of a top 5 company
Remove Stop words
Extract Nouns, Verbs,Symbols
Append the context to corresponding company file
Switch to Next File

Fig. 6. Process for extracting context of the top 5 Organizations

The first step involved storing the names of the top 5 companies in a list. The technique involved an iterative process similar to the one done for POS tagging.Let us consider the first article. The text in the file was converted to a an nlp object and stored in variable 'doc'. Then all the entities were extracted from the article and compared to the company list in order to find which company was mentioned in the list. When the match for a company was found the article was interpreted to be about that particular company and was recorded as one mention. Now in order to link the company to a context the article was subjected to a summarisation technique and then appended to file containing the company name. If such a file did not exist a new file was created. The summary of the article thus presented would act as the context for the particular mention of the company.

The summarisation technique consists of eliminating words that do not convey much meaning and retaining words that convey more meaning than the others. the words that convey most meaning were decided as named entities, verbs,Number and symbols such as the ones denoting currency. Thus once a match was found the doc variable containing the POS tagged words were parsed one by one and compared to see if they were stop words or punctuation marks. Such words were eliminated as the relative meaning conveyed by such words was decided to be less than others. The token types provided by spaCY, namely,'VERB','NOUN','PROPN','NUMBER','SYM' were only considered. Here 'PROPN' stands fro proper nouns, 'NUMBER' stands for an numeric occurrences such as financial statistics and 'SYM' stands for symbols to denote currencies such as dollars and pounds. A new file for each of the top 5 names in the company list was created. Once a match for a company name was found in an article, summarized form of that article would be appended to the text file with the company name. This step would be similarly continued for all the other files. Hence every file would be compared against each of the top 5 companies and any file that is found to have a mention of that company would have its content summarised and appended to the file having the name of that company. Thus for a journalist that wants to know the results of the top 5 companies would open the text files for the respective companies in order to know why the companies were mentioned in the news articles.

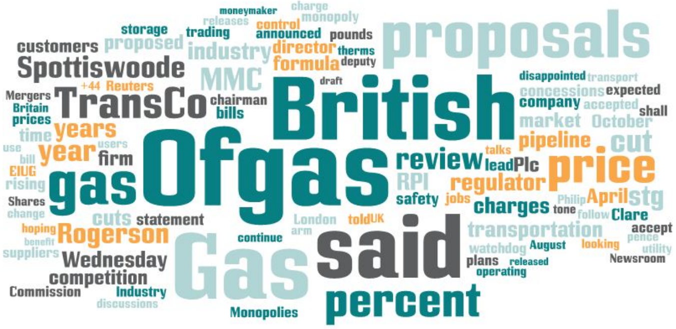


Fig. 7. Word Cloud for Ofgas

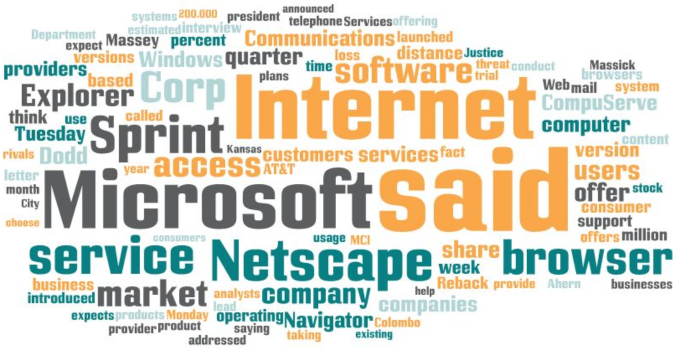


Fig. 8. Word Cloud for Microsoft

The visual representation of the words to show context so that it can be deciphered immediately was achieved by using Word Cloud. It is a novel of representing keywords that occur in a particular text. It is most commonly used for citing keywords in a website. The size of the font of a particular word represented in the cloud corresponds to its frequency in

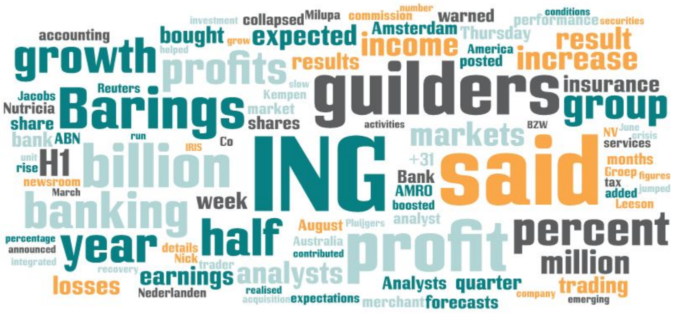


Fig. 9. Word Cloud for ING

the text. The more frequent the word, the bigger its size in the cloud. We have taken the top 100 words of context for each company and represented it using word cloud for better understanding.

IV. Analysis and Discussion

Thus it can be observed that the system extracts the top 5 mentions of the companies and provides their context through summarised text files with company names.This would be of particular interest news agencies that have to sift through large quantities of data in order to do their preliminary research. A contextual summary as offered by the system can be used to extract certain keywords or phrases that would further aid their investigation. May a times journalists are looking for leads in order to keep their investigation running and find themselves searching for keywords or links between different words. Having the context of a particular company summarized from a large amount of data and presented in an appropriate form has significant implications for their research. For instance, context for a company that is presented in the form of a word cloud would enable the journalist to look for mentions and words in a large body of articles that relate to his company which could give him an idea about the various entities that the company that he is studying could be linked to. this could also help the journalist in using the context words as keywords to be typed in an online search engine. the same goes for researchers, analysts and entrepreneurs that can use representations such as word clouds to help them facilitate their research about an entity. A slightly more detailed version of the extracted context was also made available in the form of the text files each having the company names. In this case the person reading the text can view the context in the form of key phrases. which are not completely structured sentences but more of compressed versions of the orthodox syntactical representation.

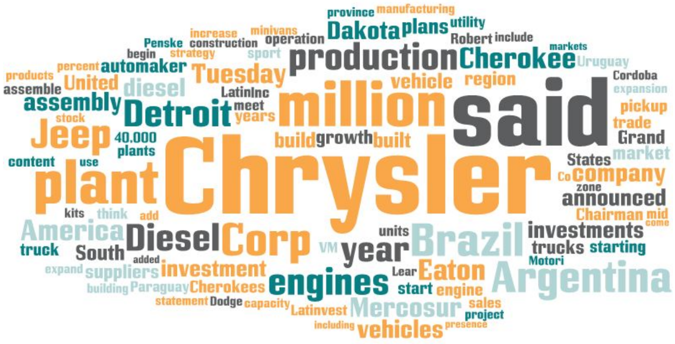


Fig. 10. Word Cloud for Chrysler

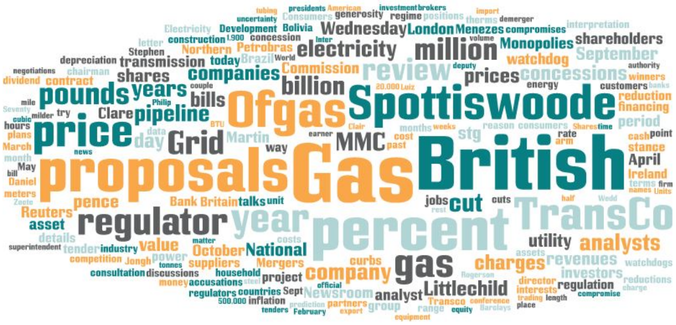
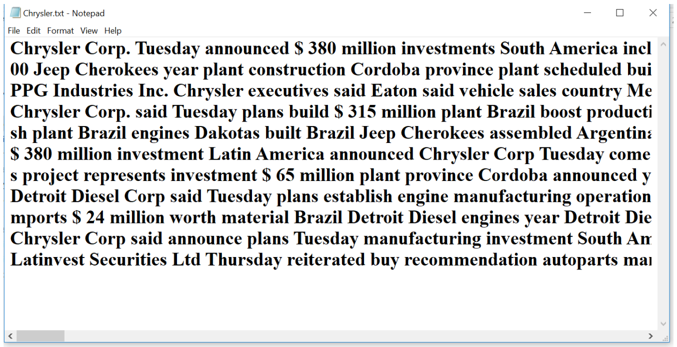


Fig. 11. Word Cloud for British Gas

However the system faces certain issues which are detailed below:

- The first issue that the system faces is the erroneous tags that are assigned to certain words. For instance, it was observed during the NER phase that the system had initially given the wrong list for the top companies by wrongfully tagging the words EPS and PCT as the top companies.
- Reuters which was correctly tagged, was not related to the context and was in fact a reporting agency that was mentioned for a lot of articles, due to which the large number of mentions, and thus does not necessarily make it a top trending company. This particular problems also needs to be addressed through techniques that filter out the the correctly tagged entities that are irrelevant to the task at hand. For the purpose of the use case, an implementation filtering out unwanted mentions of news agencies must be included.
- The other issue that was faced was the variation in spelling of the same company. For instance the company Chrysler was also mentioned as Chrysler Corp. in some cases. These two instances were taken as two separate

instances which was one of the actors affecting the system performance. Thus novel methods need to be devised in order to handle the variations in spellings. Methods such as the creation of mappings from a particular company name to its spelling variations need to be explored.



```
Chrysler Corp. Tuesday announced $ 380 million investments South America incl
00 Jeep Cherokees year plant construction Cordoba province plant scheduled bui
PPG Industries Inc. Chrysler executives said Eaton said vehicle sales country Me
Chrysler Corp. said Tuesday plans build $ 315 million plant Brazil boost producti
sh plant Brazil engines Dakotas built Brazil Jeep Cherokees assembled Argentin
$ 380 million investment Latin America announced Chrysler Corp Tuesday come
s project represents investment $ 65 million plant province Cordoba announced y
Detroit Diesel Corp said Tuesday plans establish engine manufacturing operation
mports $ 24 million worth material Brazil Detroit Diesel engines year Detroit Die
Chrysler Corp said announce plans Tuesday manufacturing investment South Am
Latinvest Securities Ltd Thursday reiterated buy recommendation autoparts ma
```

Fig. 12. Text file generated for context

V. Conclusion

Thus a system for extracting the top 5 trending companies from news articles along with the context has been designed and implemented. The model provides a good framework for the analysis and summarisation of large news texts.The context that is extracted is visually represented in two formats. the first format includes key phrases that are stored in text files while the other format is in the from of word clouds.The word clouds manage to present the context in more vivid and graphical from which enables the user to quickly spot the related words that have appeared in the news articles which can prove particularly useful for search engine querying and other research. The text files on the other hand present context in a detailed manner.The system in the future will aim to refine its existing tagging process in order to increase the tagging accuracy fro company names extraction and also work on handling the spelling variations for a particular entity through new techniques.

References

[1] T. Horsmann, N. Erbs, and T. Zesch, "Fast or accurate?-a comparative evaluation of pos tagging models." in *GSCL*, 2015, pp. 22–30.

[2] A. Ritter, S. Clark, O. Etzioni *et al.*, "Named entity recognition in tweets: an experimental study," in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2011, pp. 1524–1534.

[3] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1*. Association for computational Linguistics, 2003, pp. 173–180.

[4] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the thirteenth conference on computational natural language learning*. Association for Computational Linguistics, 2009, pp. 147–155.

[5] Q. Zhu, X. Li, A. Conesa, and C. Pereira, "Gram-cnn: a deep learning approach with local context for named entity recognition in biomedical text," *Bioinformatics*, vol. 34, no. 9, pp. 1547–1554, 2017.

[6] L. F. Rau, "Extracting company names from text," in *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, vol. 1. IEEE, 1991, pp. 29–32.

[7] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *arXiv preprint cs/0306050*, 2003.

[8] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[9] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[10] G. Zhou and J. Su, "Named entity recognition using an hmm-based chunk tagger," in *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 473–480.

[11] S. Cucerzan and D. Yarowsky, "Language independent named entity recognition combining morphological and contextual evidence," in *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[12] A. Neelakantan and M. Collins, "Learning dictionaries for named entity recognition using minimal supervision," *arXiv preprint arXiv:1504.06650*, 2015.

[13] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005, pp. 363–370.

[14] L. Alias-i, "4.1. 0," *URL http://alias-i. com/lingpipe*, 2008.

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

[16] R. Jiang, R. E. Banchs, and H. Li, "Evaluating and combining name entity recognition systems," in *Proceedings of the Sixth Named Entity Workshop*, 2016, pp. 21–27.

[17] T. Hasegawa, S. Sekine, and R. Grishman, "Discovering relations among named entities from large corpora," in *Proceedings of the 42nd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2004, p. 415.

[18] R. Dale, R. Calvo, and M. Tilbrook, "Key element summarisation: Extracting information from company announcements," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2004, pp. 438–449.

[19] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "Ontonotes: The 90\% solution," in *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, 2006.

Appendix

```
#Code to generate top5 companies
import spacy
from collections import Counter
import os
import en_core_web_sm

# nlp is the object used for POS tagging using spaCy
nlp = en_core_web_sm.load()

def load_file_names(fpath):
    '''Loads file names in the path'''
    return [filename for filename in os.listdir(fpath)]

#Gives the parent directory path
mypath='D:/Aditya/AUT/Text␣mining/Assignment␣2␣data/CCAT/'

#Creates a list of all the files in the parent directory
filelist=load_file_names(mypath)
print(filelist)

# Creating an empty List
plist =[]

# Creating a compare list to eliminate non-needed organizations. This was done after the first
comparelist= ['Reuters','EPS','PCT','ACTUAL']

# A loop which incrementally loops through each file path created by concatenating the homepat
for z in range(len(filelist)):
    mytext = open(mypath + filelist[z], ”r”) #This reads each file
    mytext = mytext.read()
    # POS tagging each file using the object NLP of spaCy
    doc=nlp(mytext)

    #This checks for the POS tags with the label ”ORG” and writes then into a list
    for X in doc.ents:
        if(X.label_=='ORG'):
            plist.append(X.text)
# print(plist)

#Finding out the top 10 frequent Organizations as we need to eliminate a few that are incorree
plist1 = (Counter(plist).most_common(10))

#REmoving Reuters, EPS, PCT, Actual from the list
plist1.remove(plist1[0])
plist1.remove(plist1[0])
plist1.remove(plist1[2])
plist1.remove(plist1[2])
plist1.remove(plist1[4])
```

```
#Execute this to see the frequency of the top mentioned Organization
# print(plist[0])


#Since the resulting list gives the string and number of times it has appeared, we extract jus
# Execute this to check the required output
# print(plist1[0][0])

#Writing just the names of the Top5 entities in a file
for a in range(len(plist1)):
    file2 = open(”D:/Aditya/AUT/Text mining/Assignment 2 data/top5.txt”,”a+”)
    file2.write(str(plist1[a]))


import spacy
from spacy import displacy
from collections import Counter
import os
import en_core_web_sm
nlp = en_co

def load_file_names(fpath):
    '''Loads  file data for a specified subset'''
    return [filename for filename in os.listdir(fpath)]

#Gives the parent directory path
mypath=’D:/Aditya/AUT/Text mining/Assignment 2 data/CCAT/’

#Creates a list of all the files in the parent directory
filelist=load_file_names(mypath)
plist =[]
comparelist= [’Reuters’,’EPS’,’PCT’,’ACTUAL’]
companies=[’ING’,’Microsoft’,’Chrysler’,’Ofgas’,’British Gas’]

# Dictionary to store all the context extracted in the form of key value pairs. The context fo
# value list with key ING in the dictionary top5
top5={
    ’ING’:[],’Microsoft’:[],’Chrysler’:[],’Ofgas’:[],’British Gas’:[]}
print(companies)

# A loop which incrementally loops through each file path created by concatenating the homepatl
for z in range(len(filelist)):
    mytext = open(mypath + filelist[z], ”r”) # This reads each file
    mytext = mytext.read()

    # POS tagging each file using the object NLP of spaCy
    docs=nlp(mytext)
    for x in docs.ents:
       #  Comparing the text of each article fetched with the list created of the top5 compani
       for i in range(5):
         if(x.text==companies[i]):
            for a in docs:
                # In each article, comparing each word and ensuring it is not a punctuation or
                if(a.is_stop != True and a.is_punct != True):

                    # To determine context, if company name is matched in the file, this code
                    # propernouns, common nouns, verbs, numbers or symbols which can help deci
                    if( a.pos_ == ’PROPN’ or a.pos_ == ’NOUN’ or a.pos==’VERB’
or a.pos_ == ’NUM’ or a.pos_ == ’SYM’):

                        #print(a, end=’ ’)

                        # Once context is extracted, it will again search the company name to
                        # the dictionary # which has the same key
                        if (x.text == companies[i]):
                            top5[companies[i]].append(a)
                            file=open(’D:/Aditya/AUT/Text mining/Assignment 2 data/’+companies
                            file.write(str(a)+’ ’)
                            file.close()

         #Breaing out of the loop once context extrated
         if (x.text == companies[i]):
            break

       if (x.text == companies[i]):
           break

# Appending the dictionary with the context
for j in range(5):
    print(companies[j],’:’,top5[companies[j]])
```