Text Mining - Assessment 2

Anders Fischer-Nielsen
afin@itu.dk

Lauritz Baess-Lehmann
laba@itu.dk

***Abstract*— Using natural language processing algorithms to extract detailed and summarized information from news articles has been attempted for several decades. However, no single approach has been identified to outperform others, and the problem of summarizing news articles in meaningful ways is therefore still an open research question. In this paper a novel approach on how to identify organisations and summarize the reasons why they are mentioned in news articles is presented. Using a subset of the CCAT data set the proposed system has been tested and evaluated. In addition to that, the top 5 most frequent organisations that appeared in the news and why they were mentioned are presented. Results show that the system is able to achieve high accuracy in extracting the correct reasons for why organisations are mentioned in the news. However, a high amount of false negatives have been observed indicating that the proposed system is unable to extract all reasons, giving a lower Recall. This means that the system extracts a satisfying number of the correct reasons, but only a subset of those present in the news. Further work should therefore be performed to increase the amount of extracted reasons while ensuring that the accuracy remains high.**

Keywords: Machine Learning, NLP, Natural Language Processing, POS Tagging, Part-of-Speech, NER, Named Entity Recognition, Information Extraction, Semantic Frame Detection, Organisation Extraction, Organisation Occurrence, Reason For Appearance

I. Introduction

Every day thousands of news articles are posted on various news sites, resulting in an enormous amount of news data, too much data for any human to be able to digest and get an overview of. Advances has therefore been made in automatically extracting information from news articles using natural language processing (NLP) algorithms. In recent years, the NLP field of research has experienced an increased interest from researchers and companies, which have lead to multiple new advances within the field. Part-of-speech (POS) tagging and named entity recognition (NER) algorithms have, in particular, experienced increased accuracy and performance. These state-of-the-art algorithms can be used to extract meaningful data from text input, such as identifying people, locations and organisations in a text, and extract the reasons why these might have been mentioned in the texts. This functionality can be useful to navigate the stream of news generated each day, and enable people like journalists to get a quick and precise overview.

This is the motivation for this paper, as we will examine how organisations can be extracted from news articles and report the reasons why they appeared in these articles. Using a subset of the CCAT[1] data set consisting of 2500 text files containing news articles, a system is presented that can determine which organisations appear in each article and determine the reasons for why the organisations are mentioned. The result of the system is the five most frequently occurring organisations and the reason for their

[1] [Lewis et al., 2004]

occurrence.

Based on the above, the following research question was formulated:

"Given a large corpus of news articles represented as text, what is the achievable performance of a proposed system that is able to extract a list of organisations, the number of times a given organisation occurred in the articles and the reason for the appearance of each individual organisation, as compared to a human extracting the same information from the corpus?"

In this paper, we present a system for extracting organisations from news articles and the reasons why they are mentioned in said articles. We will present the design of our system, how it works and the choices that led to its final form. The approach to test the accuracy of the system is explained, and the top 5 organisations mentioned in the 2500 CCAT text files is presented. Lastly, ideas and suggestions for further work and changes to the system is proposed.

II. Literature Review

In this section previous work on NER and POS tagging is presented and relevant features of this work highlighted. The state-of-the-art within both of these is used as the bases for the system proposed in this paper. The research presented in this section focus on attempts to summarize news articles, extracting names of organisations using NER and POS tags, efficient sentence splitting and sentiment analysis. This is in order to identify the possible approaches that can be taken to achieve the goal presented in section I.

[McKeown and Radev, 1999] were one of the first to propose a system able to produce a summary of a series of news articles. Their system explores how such summaries can be created for news articles related to the same event, focusing on how the perception of an event changes over time in the news. They utilize the ARPA messaging system in an attempt to generate fluent text based on a set of templates. A template is a specific structured representation of a news article comprised of different activity types. Templates were presented in [Costantino et al., 1997], where they were used to extract information regarding financial activities to enable operators to understand the information of an article without having to read the entire article. [Costantino et al., 1997] describe templates as containing three different types of activities: company related events, company restructuring events and general macro-economic events.

An example of a template is seen below:

```
Template: TAKEOVER
 COMPANY_TARGET: Tele-Communications Inc.
 COMPANY_PREDATOR: BELL ATLANTIC
 TYPE_TAKEOVER: FRIENDLY
 VALUE: 18 billion dollars
 ATTRIBUTION: BELL ATLANTIC
```

The strength of templates is that new ones can be defined to extract additional information. The system presented in [Costantino et al., 1997] is implemented using a syntactic network containing 100.000 interconnected nodes. A *semantic analysis* module associates each word of the input text with an appropriate meaning, which is compatible with the syntactic network. How this association is performed and implemented is not explained in the paper, which makes the approach unusable in for project. However, the paper describes a *parser* module which recognizes *"the role of each of the words in the sentence for example subject, object, verb, adjective, etc."*. This module would most likely have been defined as a POS tagging module today. This indicates that POS tagging is necessary to extract and summarize

information from news articles.

In contrast to [Costantino et al., 1997], [McKeown and Radev, 1999] does describe how their system work. Templates are utilized in a more general manner than [Costantino et al., 1997] – to generate fluent text based on a defined set of templates and produces summaries from lexical cues, such as ”however”, ”exactly” and ”finally”. This approach requires changes to be made to templates by hand to fit incoming news articles accurately. Changes in prose will, for example, require template changes. This make the approach of [McKeown and Radev, 1999] infeasible for the purposes of this paper, as it is not possible to manually fit templates for all types of incoming news articles, with the amount of news generated today. Another more automated approach should be identified.

[Marzinotto et al., 2018] present Semantic Frame Detection, which is the processing of text in order to detect an event or a scenario, called a *Frame*, while detecting text elements that can be mapped to this event in the sentence, called a *Frame Element*. *Full text parsing* and *partial text parsing* can be performed. *Full text parsing* will parse each individual word in a sentence in order to determine whether this word triggers a *frame*. *Partial text parsing* examines a subset of the frames in a sentence according to the given framework, and is cheaper for humans to frame tag. Frames can be used to extract actors in a sentence and who the performed action relates to. This provides information for organisation extraction and determines which organisations performed given actions in a sentence.

The approach described by [Krishnamoorthy, 2018] employs an association rule mining-based hierarchical sentiment classifier model in order to determine the sentiments of individual investors, institutions and markets in financial news articles. NLTK is used to extract investors, institutions and markets from texts, and sentiment analysis is then performed. NLTK has an accuracy of about 89.71% [2].

State-of-the-art results have been accomplished using alternate approaches than that of NTLK, however. Bi-LSTM neural networks are shown in [Marzinotto et al., 2018] to achieve a better recall than a multi-model CRF model, at the cost of achieving lower precision. Furthermore, the work described in [Wang et al., 2018] concludes that a Bi-LSTM model achieves state-of-the-art performance in intent detection on the ATIS benchmark. Using a similar neural network structure would therefore most likely deliver promising results for frame detection in news articles.

The Flair library consists of a Bi-LSTM-CRF model, containing a pre-trained Semantic Frame Detection model, achieving a 93.92% F1 accuracy on the Propbank 3.0 data set, initially defined in [Palmer et al., 2005] as an addition of *”a layer of predicate-argument information, or semantic role labels, to the syntactic structures of the Penn Treebank [Marcus et al., 1993]”*. Due to its high F1 score Flair has been chosen for performing Semantic Frame Detection in this project.

[Naughton et al., 2006] describes and evaluates methods for grouping sentences in news articles that refer to the same event. The articles concludes that removing non-event sentences using a trained classifier achieves significant improvements in the overall performance of the following clustering. As such, sentences in the news articles that do not contain organisations should therefore be removed to improve accuracy of a system. This approach is utilized in the system presented in this paper.

[2] [Dishmon, 2015]

III. Implementation

This section describes the preprocessing of the data set, what experiments were performed during implementation and, finally, the implementation of the proposed system.

A. Data Processing

The data set used in this project is a subset of the CCAT data set consisting of 2500 news articles. The data set is a folder containing all of the news articles as individual files. In order to work with the data set in the system it had to be preprocessed. The preprocessing was performed as follows: A folder containing the news articles gets passed to the proposed system, which then locates all files in the folder and reads each file individually. Each individual file is further processed and split up, resulting in a list of sentences. The sentences are analyzed, resulting in a sentence with accompanying POS, NER and semantic frame tagging. Each sentence, with its corresponding tags, is used to determine which organisations appeared in the specific sentence, and what the reason for the appearance of the organisation in the sentence was.

B. Experimentation

Experimentation has been performed in order to determine the performance of our approach. Experiments were performed in stages, giving us an idea of how well we were able to extract organisation names and afterwards how well we were able to extract reasons for appearance.

More experiments were performed as we began to see promising results, with more granularity of experiments as the weak points of our approach became apparent through previous experiments. We had a broad scope of what experiments could be performed initially and during experimentation narrowed our scope in order to determine what experiments were relevant for our approach. An illustration of this approach can be seen below.

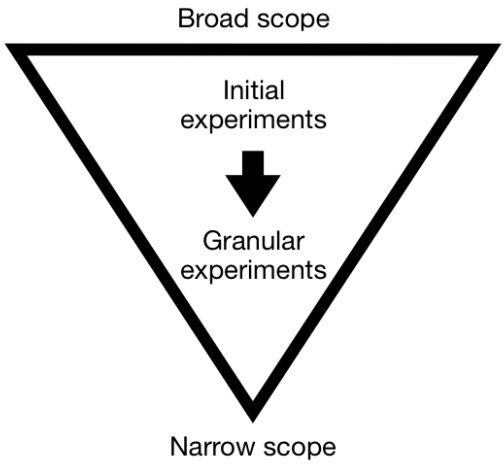

Fig. 1. An illustration of our experimentation approach.

Our first experiments dealt with splitting the paragraphs of a news article split into sentences. Initial experiments split on each full stop character (”.”), but this did not allow abbreviations in the news articles and was therefore deemed insufficient.

Utilizing the open-source the rule-based sentence segmenter *segtok*[3] allowed us to split the paragraph into sentences with an acceptable precision. Further experimentation showed that the accuracy of the sentence splits was higher using the successor to the *segtok* library, *syntok*[4], which has evolved significantly in terms of providing better segmentation and tokenization performance. Our next experiments revolved around extracting POS tags. We utilized the open-source Flair LSTM-based library allowed us to quickly see POS tagging results. With minimal additional work, NER tagging and Semantic Frame Detection was added to the sentences, which allowed us to move knowledge extraction based on the sentence tagging.

The initial extraction experiments dealt with getting the

[3] https://github.com/fnl/segtok
[4] https://github.com/fnl/syntok

'ORG' NER tags extracted and added to the list of occurring organisations. These were counted, and a rudimentary extraction of the reason for occurrence was performed by extracting all text present after the 'ORG' tag until the first comma or full stop was reached. This proved to be inaccurate, again due to abbreviations or lack of punctuation, and the earlier sentence splits were relied upon instead.

Experimentation also revealed that organisation names are not consistent. A simple example of this is the organisation "Microsoft Inc." appear both as "Microsoft", "Microsoft Inc." and "Microsoft Corp.". Through experimentation we found a solution to this by clearing up company names, removing "noise" and reducing the company name to the shortest form possible.

Concretely, the aforementioned *name cleaning* is performed by the algorithm seen in [1]. This *name cleaning* allowed us to map company names more accurately, improving the counting of occurrences of the individual companies.

Company names are shortened aggressively, possibly resulting in an incomplete company name, but given that the original longer version of the company name is shortened to the shortest common denominator of that name, this should not pose an issue, since company names do not overlap. In a use case where company names must not overlap, in e.g. a search for subdivisions of a company such as "Sony Europe Inc." and "Sony America Inc." is desired, this step would have to be re-implemented or even omitted.

Further experimentation was performed by parsing 10 news articles and looking at the results, showed that the extraction of the reason for the appearance of an organisation was not accurate enough, since too much of the text following the 'ORG' tag was extracted, resulting in most of the paragraph being extracted instead of a summary or overview of the reason for appearance. Experiments revolving around utilizing the Semantic Frame Detection tags were performed, attempting to find the "actioning" verb after the appearance of the 'ORG' tag. This proved to result in shorter more accurate reasons being extracted, with the only problem being missing Semantic Frame Detection tags. The Semantic Frame Detection in Flair has an accuracy of 93.92% in the experiments performed by the developers of Flair, which is not always accurate enough.

An experiment involving parsing the entire CCAT data set showed that the Flair POS tagger would confuse words such as *"I"* and *"We"* as organisations. These false positives from the POS tagging would then be extracted as the "organisation" that occurred the most. Removing these false positives from the pre-processing step of our approach eliminated the counting errors, and allowed us to present a correct count of the organisation that occur most often.

If semantic frame detection does not lead to being able to tie an 'ORG' tag together with a reason for appearance a loss of information might occur if this 'ORG' tag is followed by a reason for appearance. A fallback has therefore been devised in order to avoid this loss of information. Using POS tags of sentences, it is possible to extract the first verb occurring after the aforementioned 'ORG' NER tag and extract the remainder of the sentence if no semantic frame was available. This approach showed promise through experimentation on a bigger subset of the CCAT data set. This fallback blindly extracts the remainder of the sentence after the first occurring verb and might therefore extract irrelevant or superfluous sentences that do not represent the reason for the appearance of the organisation. We describe the effects of this in the next section where we evaluate and describe test results.

During experimentation it was discovered that processing the files took more time than expected, and a "caching" solution was devised to allow running tests in smaller batches, re-

suming work where the last processing took place by storing intermediate results to disk. This allowed further gathering of extraction results and fine tuning of the extraction.

C. Algorithms

This section describes the two main helper algorithms[5] used in the main algorithm[6] of our approach. The main algorithm executes pre-processing of texts, before calling the helper algorithms. All algorithms are utilized in our approach are shown as pseudo code below.

The aforementioned algorithm for cleaning organisation names in order to make them uniform is shown below.

Algorithm: CleanOrganisation
Input: F, the full organisation name
Result: C, the cleaned organisation name

$C \leftarrow F$.lowercase.removeCharacters(−−; "; $'s$; '; (;))
if , ***in*** C **then**
  $C \leftarrow C.split(",")[0]$
end
if . ***in*** C **then**
  $C \leftarrow C.split(".")[0]$
end
$C \leftarrow$ C.remove(,)
$S \leftarrow$ C.split()
$C \leftarrow S[0]$.capitalize()
for s *in* $S[1:]$ **do**
  $C \leftarrow$ **if** *s.length* $<$ *4* **then** C
  else $C + S.capitalize()$
end
return C

Algorithm 1: Our algorithm for cleaning organisation names.

The second helper algorithm for extracting reasons for the appearance of an organisation is shown below. The algorithm takes two input, an organisation name and the sentence it occurred in and return a reason for appearance.

The main algorithm for extracting organisation names and their reason for appearance can be seen below. The algorithm takes a set of paths to the news articles that should be processed, and return a mapping from each organisation name to the reasons for the appearance of the organisation.

Input: $Organisation$, the tagged segment of the sentence containing an organisation
Input: S, the tagged sentence the organisation appeared in
Result: R, the reason for appearance

end = organisation.end
$frameTags$ = S.taggedSpans("$frame$")
$posTags$ = S.taggedSpans("ner").filter($t \rightarrow$ "ORG" **in** t)
$frameTagsAfterOrg$ = $frameTags$.filter($t \rightarrow t.start > end$)
$posTagsAfterOrg$ = $posTags$.filter($t \rightarrow t.start > end$)
if ***not*** $(frameTagsAfterOrg$ ***or*** $posTagsAfterOrg)$ **then**
  return
end
$first \leftarrow$ **if** $frameTagsAfterOrg$ **then** $frameTagsAfterOrg[0]$ **else** $posTagsAfterOrg[0]$

return $R = S[first:]$

Algorithm 2: Our extracting the reasons for the appearance of an organisation in a sentence.

IV. Results

In this section the results of the approach described in this paper is presented in terms of accuracy, precision, recall and F_1 score. In addition to that, the 5 most frequent organisations, according to our system, that appeared in the subset of CCAT news articles is presented.

A. Test Results

As detailed in section III, we initially tested our approach on a subset of the given data set and finally on the full CCAT subset.

Results after extracting organisations and their reasons for appearance look promising at first glance, though errors still occur. The output for the *Sartid* company is shown below, after processing 110 of the 2500 news articles.

{

[5]The algorithms mentioned are shown as pseudo code in [1] and [2].
[6]The algorithm mentioned is shown as pseudo code in [3].

Input: $\{Paths\}$, the set of file paths to news articles
Result: $Reasons \equiv \{O_0 \rightarrow \{R_{1_{O_0}}, ..., R_{N_{O_0}}\}, ..., O_N \rightarrow \{R_{1_{O_N}}, ..., R_{N_{O_N}}\}\}$,a mapping from each organisation name to the reasons for appearance, where O is an organisation and R is a reason

$\{NER, FRAME, POS\} \leftarrow loadFlairTaggers()$
$O_Counts \leftarrow \{O_0 \rightarrow \mathbb{N}, ..., O_N \rightarrow \mathbb{N}\}$

F, O_R, O_C = checkCache()

for $file \leftarrow$ *readContentsOf(p in Paths)* **do**
  $P \leftarrow$ extractSentences($file$)
  for S *in* P **do**
    $S \leftarrow NER.predict(S)$
    $S \leftarrow FRAME.predict(S)$
    $S \leftarrow POS.predict(S)$
    $Names \leftarrow S$.taggedSpans("ner").filter($s \rightarrow$ "ORG" **in** s)
    if $Names == \varnothing$ **then**
      continue
    end
    for $first$ ***in*** $Names[1:]$ **do**
      $name \leftarrow$cleanOrganisation[7]($first$)
      $reason \leftarrow$ getReason[8]($first$, S))
      O_R.add($name$, $reason$)
      $O_C[name] \leftarrow O_C[name] + 1$
    end
    for $remaining$ ***in*** $Names[:1]$ **do**
      $name \leftarrow$cleanOrganisation($remaining$)
      O_R.add($name$, *None*)
      $O_C[name] \leftarrow O_C[name] + 1$
    end
  end
  F.add(p)
  saveToCache(F, O_R, O_C)
end
return O_R

Algorithm 3: Our algorithm for extracting organisation names and their reasons for appearance.


```
"Sartid": [
  "based in the east-Serbian town of
      Smederevo, produced 63,000 tonnes of
      iron, 74,000 tonnes of steel and 80,0
      00 tonnes of finished products.",
  "fired its furnace in March, and from
      April to the end of July produced a
      total of 228,000 tonnes of iron, 222,
      000 tonnes of steel and 222,000
      tonnes of finished products.",
  "stopped producing iron and steel at the
      beginning of 1993," said Markovic.",
  "received about $122 million in credits
      in the past six months to kick-start
      the plant -- mostly from Swiss banks
      .",
  "said the plant's exports rose to $1
      million a day after almost four years
       of idleness under the international
      trade embargo.",
  "is still sluggish.",
  "plans to start its second blast furnace
      by the end of 1997."
]
}
```


This result would give a user, such as a journalist wanting to determine why a company appeared in a given news article, a satisfactory overview of why the organisation *Sartid* was in the article. However, it is unclear how the results of the proposed approach holds up to what a human would find. In order to compare the proposed system with a human, the same news articles were summarized by hand. The results of the manual summarizing can be seen below:


```
{
  "Sartid":
    "has raised production to pre-sanctions
        levels, producing over 200,000 tonnes
         of iron and steel in July, the Iron
        and Steel Industry Association said
        on Tuesday.",
    "produced 63,000 tonnes of iron, 74,000
        tonnes of steel and 80,000 tonnes of
        finished products.",
    "fired its furnace in March, and from
        April to the end of July produced a
        total of 228,000 tonnes of iron, 222,
        000 tonnes of steel and 222,000
        tonnes of finished products.",
    "received about $122 million in credits
        in the past six months to kick-start
        the plant -- mostly from Swiss banks
```

```
        .",
    "said the plant's exports rose to $1
        million a day after almost four years
         of idleness under the international
        trade embargo.",
    "raw materials are imported -- crippling
        the factory during sanctions.",
    "plans to start its second blast furnace
        by the end of 1997.",
}
```

We see that the sentence *"has raised production to pre-sanctions levels, producing over 200,000 tonnes of iron and steel in July, the Iron and Steel Industry Association said on Tuesday"* is missing in the result from the automated approach, with the rest of the reasons for appearance looking similar to the human summary. In addition to that, the proposed approach extracts the sentence *"is still sluggish"*, which is not relevant to the user of the proposed system, and does not provide any valuable information.

In other words, we observe a false negative and a false positive in the output of the proposed system, though the general reason for the appearance of *Sartid* is still apparent from the output.

We extracted organisation names and their following reason for appearance of 18 files by hand, in order to validate the precision of the proposed system. These files contain approximately 176 sentences, containing 3865 words in total and were all validated by hand. The ability of extracting organisation names correctly of the developed model and the ability of extracting reasons of the appearance of the organisation was then measured based on comparing the results of our approach against the same news articles processed by a human.

The results showed missing reasons when comparing the results of our approach and the human extraction, especially in reasons where the company name was omitted and replaced by "the company", "it" or "the President of the company". Our approach does not register these sentences as relating to the company name, giving a number of false negatives in the results. Reading the reasons for appearance of a given company name gives an overview of why the company is in the given news articles, but it does not give the reader the complete picture due to sentences missing. This includes details dealing with the share rise or drop of companies and quotes from company leaders.

The results of the proposed system have been compared to the human results through the use of the confusion matrix seen below. The confusion matrix above shows

Population: 176	Actual Positives	Actual Negatives
	63	113
Predicted Positives	23	10
Predicted Negatives	40	103

TABLE I
CONFUSION MATRIX.

a high false negative rate, meaning that the proposed system does not extract all reasons for the appearance of an organisation name. The confusion matrix furthermore shows that the system extracts a smaller, but still significant, number of false positives, consistently in the shape of small sentence fragments, which does not contribute to a better understanding of why the organisation appeared. In the *Sartid* example above, the fragment *"is still sluggish"* is one such false positive.

From the confusion matrix we are able to determine that the accuracy is 71.59%, with precision at 69.70%, recall at 35.38% and specificity at 91.15%. This leads to an F_1 score of 46.94. We see that the model does not find a

large number of true positives, but on the other hand does not find a large number of false positives either, as seen on the false positive rate of 0.088%. The model simply does not extract as many reasons as a human would, but the reasons it *does* extracts are in most cases correct. The false negatives are rather high, again due to the model not extracting as many reasons as a human would, with a false negative rate of 63.5%.

B. CCAT Results

Applying the system and approach described in section III on the subset of 2500 CCAT articles yields the top 5 most frequently mentioned organisations and the reasons why they were mentioned. These are presented in the following table.

In table II we observe that even though data was extracted from 2500 articles the most frequent organisation was only mentioned 18 times. Looking at all the reasons why these 5 organisations were mentioned, which can be seen in appendix I, it is possible to get a pretty good understanding as to why the organisations were in the news. In addition to that, the way the reasons are formatted – where you first read the name of the organisation and then the reason – supports journalists in using the extracted information in their work.

	Organization	**Mentions**
1.	Pioneer	18
2.	Omw	14
3.	Halifax	13
4.	Basf	12
5.	Maybank	12

TABLE II
TOP 5 MOST FREQUENTLY MENTIONED ORGANISATIONS IN THE 2500 CCAT NEWS ARTICLES.

As a result of analyzing the 2500 CCAT news articles 5669 unique organisations were identified in total. However, only 2915 of these organisations had any reasons for why they were mentioned tied to them. This results in reasons not being found for 48.58% of the total number of identified organisations. This is quite a large percentage of organisations without reasons – higher than expected.

Naturally, some of the organisations in the CCAT articles do not have any reason, as they could be utilized within a reason for the mention of another organisation. This could be the case of a sentence, such as ”Berkshire Hathaway sold 2.9 million of shares in Apple during the last three months.”, where ”Apple” would not have a reason, as it is used within the reason for why ”Berkshire Hathaway” was mentioned. However, examples like this would only explain a small amount of organisations without reasons at best. A more likely explanation for the result is that the proposed system is insufficient in its abilities to extract reasons for why organisations are mentioned. This would also reflect the observations of the results reported in the previous subsection, where it was shown that the system achieves a small amount of false positives, but a large amount of false negatives. In general terms, this means that the system is unable to find all of the reasons that exist in the news articles, but that the reasons it *does* find are correct.

V. FURTHER WORK

As seen in the evaluation of the proposed system in section IV, a number of false negatives were present in the final result. This is mainly due to a lack of co-reference resolution in the sentence pre-processing of the news articles, consequently leading the following search for organisations to miss sentences where the company name is implicitly referenced. The addition of a co-reference resolution pre-processing step and following marking of words such as *”it”* or *”the company”* as corresponding to the company name would most likely greatly improve

accuracy, we assume. The *neuralcoref*[9] Python library allows extracting co-reference resolution through the use of the NLP Python library, *spaCy*[10], and could possibly be used as an addition to the Flair library used in the system, in order to accomplish this.

Further use of the POS tags of sentences as a result of the pre-processing step in the proposed system could be used in order to determine if the assets of a company should be included in the reason for appearance. In the sentence *"Sony Inc.'s shares fell 22% yesterday, following a data leak."*, the remainder of the sentence following *"Sony Inc."* will not be extracted as a reason for appearance, even though it could be argued that this is should be done. Extracting *"shares"* as belonging to *"Sony Inc.'s"* from the sentence from POS tags could allow this.

VI. Conclusion

The following research question was formulated at the beginning of the paper:

"Given a large corpus of news articles represented as text, what is the achievable performance of a proposed system that is able to extract a list of organisations, the number of times a given organisation occurred in the articles and the reason for the appearance of each individual organisation, as compared to a human extracting the same information from the corpus?"

A novel approach to identify organisations and extract reasons for why they are mentioned in news articles has been presented. Applying the presented system on the CCAT subset of articles has lead to a satisfying performance being observed, with a 71.59% accuracy when compared to a human. However, a high number of false negatives were reported, which mean that a human actor is able to extract more reasons for organisations appearing in news articles, than the proposed system. The proposed system has a low false positive rate of 0.088%, but due to the relatively high number of false negatives, the system does not extract as many reasons as a human, resulting in a lower recall score of 35.38%.

The proposed system is able to give a user, such as a journalist, the ability to give the proposed system the location of a folder of news articles represented as text files and be presented with a list of reasons for the appearance of each organisation in the news articles. Though the system does not extract as many reasons for the appearance of a company as a human would, we conclude that, despite this, the user is provided with a high-level understanding of the why the company was mentioned in the articles.

Further work to improve the extraction of reasons have been identified, with the addition of co-reference resolution as the most promising addition to the system. Co-reference resolution would possibly allow the system to extract more reasons for appearance, leading to a higher recall score and a better understanding of the given news articles. In addition to that, utilizing additional POS tags could also lead to more reasons being identified.

If we were to carry out the project again, we would put more effort into co-reference resolution, than the previously mentioned semantic frame detection. The use of both would most likely be beneficial. In addition to that, we would also put more effort into mapping variations of organisation names into their base representation, such that the same organisation would not be present in multiple,

[9] https://github.com/huggingface/neuralcoref

[10] Initially described in [Honnibal and Johnson, 2015] and available at https://spacy.io.

almost identical, versions.

VII. Contributions Made by Team Members

A. Contribution Made by Anders Fischer Nielsen

Human processing of the 18 articles used for verification was performed by Anders in order to be able to verify the results of our approach.

The majority of the implementation and fine tuning the implementation of our solution was also performed by Anders, with feedback during implementation from Lauritz.

Early tests using other NLP tagging libraries was also performed by Anders, leading to the choice of *Flair* after reading through the previous work and research described earlier in this report.

B. Contribution Made by Lauritz Baess-Lehmann

The need to clean organisation names was quickly discovered and the solution to this problem was implemented by Lauritz early on in the project. Lauritz contributed heavily to this report, having a good eye for finding consistency issues between sections and seeing where sections needed to be elaborated on. In addition to that, Lauritz was responsible for applying the proposed system on the 2500 CCAT news articles, analyze the results and report the findings.

References

[Costantino et al., 1997] Costantino, M., Morgan, R. G., Collingham, R. J., and Carigliano, R. (1997). Natural language processing and information extraction: qualitative analysis of financial news articles. In *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*, pages 116–122.

[Dishmon, 2015] Dishmon, C. (2015). Testing nltk and stanford ner taggers for accuracy. blogpost.

[Honnibal and Johnson, 2015] Honnibal, M. and Johnson, M. (2015). An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

[Krishnamoorthy, 2018] Krishnamoorthy, S. (2018). Sentiment analysis of financial news articles using performance indicators. *Knowledge and Information Systems*, 56(2):373–394.

[Lewis et al., 2004] Lewis, D., Yang, Y., Russell-Rose, T., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

[Marcus et al., 1993] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

[Marzinotto et al., 2018] Marzinotto, G., Auguste, J., Béchet, F., Damnati, G., and Nasr, A. (2018). Semantic frame parsing for information extraction : the CALOR corpus. *CoRR*, abs/1812.08039.

[McKeown and Radev, 1999] McKeown, K. and Radev, D. R. (1999). Generating summaries of multiple news articles. *Advances in automatic text summarization*, pages 381–389.

[Naughton et al., 2006] Naughton, M., Kushmerick, N., and Carthy, J. (2006). Event extraction from heterogeneous news sources.

[Palmer et al., 2005] Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106.

[Wang et al., 2018] Wang, Y., Shen, Y., and Jin, H. (2018). A bi-model based RNN semantic frame parsing model for intent detection and slot filling. *CoRR*, abs/1812.10235.

Appendix I
CCAT Results With Reasons

In this appendix the top 5 most frequently mentioned organisations found by the proposed system on the subset of 2500 CCAT news articles are presented with their corresponding reasons. It should be noted that the number of times mentioned does not correspond to the number of reasons, as multiple mentions of an organisation can be related to one reason.

- **Pioneer** (Mentioned 18 times)
    1) said the substantial increase in profitability was mainly due to asset rationalisation and lower operating costs following the merger.
    2) said it expected petrol refiner margins to remain tight during 1996/97 due to strong competition

from Asian refineries but saw margin improvements for jet and diesel fuel.

3) earned $2.16 a share.
4) said preliminary estimates show its share of the North American hybrid seed corn market in 1996 appears to be about 44 percent.
5) said it expects to have a ”better idea” of hybrid seed corn market share next month when more information is available.
6) said in May it expected its share of the hybrid seed corn market to be stable to slightly lower.
7) has said previously it could make an purchase of up to A$1.0 billion.
8) building materials businesses worldwide had experienced a good start to the 1996/97 financial year in contrast to its petroleum refining and marketing joint venture, Ampol which was feeling the negative effects of low refiner margins for petroleum.
9) said it was assessing a wide range of acquisition opportunities, including add-on acquisitions, new country entries and larger acquisitions.
10) was commenting after reporting a nine percent rise in operating profit pre-tax of A$332.5 million for the year to June 30.
11) said that most of the European economies in which it operated were expected to be under pressure and that volumes could fall further.
12) said it expected the group’s U.S. operations to further improve their results in 1996/97.
13) refining and marketing operations were expected to remain competitive during 1996/97, with refining margins forecast to remain tight.
14) price was down one cent at A$3.72 at 12.30 p.m. (0230 GMT).
15) operations in Europe and South America have been particularly strong in fiscal 1996, which ends in August.
16) said Thursday that preliminary indications are that its North American seed corn market share will be about 44 percent, down from about 45 percent a year ago. ”We went on record (in January 1996) as saying they would lose market share,” Dahlman said.
17) said its North American seed field yields are expected to meet expectations, although crop maturity is running behind normal because of cooler than normal weather in the Midwest.

- **OMV** (Mentioned 14 times)

1) profits of over two billion schillings,” said one trader.
2) reported a pre-tax profit of 1.56 billion schillings for the first half.
3) reported steady first half earnings on Tuesday and said weakness in ”two key sectors would not prevent it from matching last year’s ” record profit.
4) ended 13.9 schillings down at 1,025.
5) achieved.
6) had reported a refining EBIT of 0.26 billion schillings in the first quarter.
7) operations also struggled to maintain a profit as polyolefin prices hit the skids.
8) continue to reduce its cost base over the next few years and aimed to cut its workforce by four percent annually.
9) employed 8,663 staff on June 30, down from 10,028 at the same time last year.
10) traced the upswing to higher crude oil prices and

restructuring.

11) said it planned to reduce the number of gasoline stations it operates in Austria over the next three years while expanding its network in neighbouring countries.
12) said on Tuesday its forecast for flat full-year earnings was based on a conservative outlook for polyolefins in the fourth quarter.
13) posted a record pre-tax profit of 2.09 billion schillings and earnings before interest and tax of 2.19 billion.

- **Halifax** (Mentioned 13 times)

1) is expected to bring the biggest single boost to share ownership in Britain.
2) said its financial services arm achieved 23 million stg of profits in the first half of the year, compared with seven million stg in the six months to January 1996.
3) said it had issued 775 million stg of subordinated debt in recent months.
4) told Reuters in an interview these levels were below its normal market share of around 20 percent because it had shied away from re-mortgaging which accounted for 30 percent of the total mortgage market.
5) said its 649 million pounds of half year profits represents a rise of six percent rise over the six month period to January 31.
6) plans a stock market flotation next year, took a 7.2 percent share of net mortgage lending in the first half of the year and 15.2 percent of gross mortgage lending.
7) told Reuters in an interview, because the society has shied away from re-mortgaging which accounts for 30 percent of the total mortgage market.
8) price index for the 12 months to July showed a 5.3 percent rise.
9) profits in the first half of the year would have been 64 million stg higher if the society had not written off its mortgage incentives – discounted rates and cashbacks – straight away.
10) said it made pre-tax profits of 649 million stg in the six months to June 30.
11) said it had changed its year end from January 31 to December 31 and its results for January 1996 were included in two of the reported periods, including a 88.2 million stg charged for merger and integration costs.
12) expected several hundreds of thousands of shares to change hands on the first day of trading in its shares, likely to be in June 1997.
13) are due to vote on the plans for flotation in February 1997.

- **BASF** (Mentioned 12 times)

1) reported a 14.7 percent rise in its first-half profit and conservatively predicted a slight increase in full year earnings following more orders last month in most businesses.
2) told a news conference here.
3) profit rose to 2.32 billion marks ($1.55 billion) in the first half of 1996 from 2.02 billion, coming in right at the middle of ”analysts forecasts.”
4) said.
5) said he believed this process had stopped as economies were picking up.
6) sales rose marginally to 24.3 billion marks from 24.1 billion as favourable exchange rates more than offsett an average two percent drop in selling

prices.

7) said it was still keen to make acquistions in the pharmaceutical sector, but added it would move cautiously because of the current high prices for any takeover prey.
8) expects total 1996 capital expenditure in tangible fixed assets to be 3.2 billion marks, of which 50 percent will be invested outside Germany.
9) said it saw no regulatory obstacles that could block its planned sale of a 51 percent stake in German potash group Kali und Salz AG to a Canadian company.
10) said.
11) closed 89 pfennigs higher at 43.13 marks.

- **Maybank** (Mentioned 12 times)

1) was confident Amcol and Perwaja would turn around.
2) reported a group net profit for the year ended June 30, 1996 of 1.07 billion ringgit, against 863.53 million ringgit in the previous year.
3) reported a net profit of 1.07 billion ringgit against 863.53 million ringgit in the prior year.
4) dispelled recent concerns about its exposure to loans from troubled companies Perwaja Steel and Amcol Holdings Ltd, saying the provisions are ”insignificant”.
5) had raised its loan loss provision for both companies had prompted recent speculation that its full-year results would fall under estimates.
6) said on Wednesday, adding that it expects the bank to ”easily” achieve an average growth rate of 15 percent per annum over the next two financial years.
7) managing director Amirsham Abdul Aziz said.
8) was also expected to continue.
9) were suspended on Tuesday and closed at 22.50 ringgit on Monday.
10) dispelled recent concerns about its exposure to loans from troubled companies Perwaja Steel and Amcol Holdings Ltd, saying the provisions are ”insignificant”.
11) said one dealer with a local brokerage firm.

Appendix

Implementation

```
#!/usr/bin/env python3

import traceback
import glob
import pprint
import sys
import json
import os
from syntok.segmenter import process
from flair.models import SequenceTagger
from flair.data import Sentence, Span


def find_organisations_reasons(folder: str):
    """ Go through files in the given folder, extract organisation names
        and their reason for appearance in file. """
    org_reasons, org_counts = {}, {}
    try:
        # Get flair models.
        ner_tagger, frame_tagger, pos_tagger = get_flair_taggers()
        # Fetch results from cache, if present.
        files_processed, org_reasons, org_counts = check_cache()
        file_count = 1 if len(files_processed) == 0 \
            else len(files_processed) + 1
        # Find files to process from path.
        files = glob.glob(f"{folder}/*.txt")
        print(f"Processing {len(files)} files in '{folder}'.")
        # Remove previously processed file names.
        to_process = [f for f in files if f not in files_processed]
        for path in to_process:
            print(f"[{file_count}/{len(files)}] Processing {path}...")
            file = open(path, "r")
            # Go through paragraphs sentence by sentence and extract information.
            paragraphs = process(file.read())
            for sentences_tokenized in paragraphs:
                for tokens in sentences_tokenized:
                    sentence = ""
                    for token in tokens:
                        sentence += f"{token.spacing}{token.value}"
                    sentence = Sentence(sentence.strip())
                    # Add NER, POS and Semantic Frame Detection tags to sentence.
                    ner_tagger.predict(sentence)
                    frame_tagger.predict(sentence)
                    pos_tagger.predict(sentence)
                    # Extract all organisations.
                    organisations = get_organisations(sentence)
                    if not organisations:
                        continue

                    # Find the first organisation occurence and its reason for appearance.
                    for first in organisations[:1]:
                        name = clean_organization(first.text)
                        reason = get_reason_for_appearance(first, sentence)
                        add_to_organisation(
                            name, reason, org_counts, org_reasons)

                    # Count remaining organisations, but don't find its reason for appearance,
                    # since the other organisations following the first one don't have meaningful reasons,
                    # leading to broken sentences.
                    for remaining in organisations[1:]:
```

```
                        name = clean_organization(remaining.text)
                        add_to_organisation(
                            name, None, org_counts, org_reasons)

            files_processed.append(path)
            # Store in cache after processing.
            dump_to_cache(files_processed, org_reasons, org_counts)
            file_count += 1

        if (org_reasons['I']):
            org_reasons.pop('I', None), org_counts.pop('I', None)
        if (org_reasons['We']):
            org_reasons.pop('We', None), org_counts.pop('We', None)

        print(f"\nFinished processing {file_count} files.")
        return org_reasons, org_counts
    except Exception as e:
        # Handle early exit by user (CTRL+C).
        print(e)
        print("\n\nExiting...")
        print(f"Finished processing {file_count} files.")
        return org_reasons, org_counts


def check_cache():
    """ Fetch previously processed results, if present. """
    try:
        processed_files = json.load(open("cache/files.json", "r"))
        org_reasons = json.load(open("cache/org_reasons.json", "r"))
        org_counts = json.load(open("cache/org_counts.json", "r"))
        return processed_files, org_reasons, org_counts
    except:
        return [], {}, {}


def dump_to_cache(processed_files, org_reasons, org_counts):
    """ Dump processed results to cache. """
    try:
        if not os.path.exists("cache"):
            os.makedirs("cache")
        json.dump(processed_files, open("cache/files.json", "w"))
        json.dump(org_reasons, open("cache/org_reasons.json", "w"))
        json.dump(org_counts, open("cache/org_counts.json", "w"))
    except:
        return


def get_flair_taggers():
    """ Get the Flair tagger and load their respective models."""
    print("Loading flair models...")
    frame_tagger = SequenceTagger.load("frame-fast")
    ner_tagger = SequenceTagger.load("ner-fast")
    pos_tagger = SequenceTagger.load("pos")
    return ner_tagger, frame_tagger, pos_tagger


def get_organisations(sentence: Sentence):
    """ Extract 'ORG' NER tags in a sentence """
    org_tags = list(filter(lambda span: "ORG" in span.tag,
                           sentence.get_spans("ner")))
    return org_tags


def get_reason_for_appearance(organisation: Span, sentence: Sentence):
    """ Extract the reason for the appearance of an 'ORG' NER tag in a sentence. """
```

```
    # Find ORG placement in sentence.
    org_end = organisation.end_pos
    frame_tags = sentence.get_spans("frame")
    # Extract frame and POS tags after organisation occurence.
    pos_tags = list(filter(lambda span: "VBD" in span.tag,
                           sentence.get_spans("pos")))
    frame_tags_after_org = list(
        filter(lambda span: span.start_pos > org_end, frame_tags)
    )
    pos_tags_after_org = list(
        filter(lambda span: span.start_pos > org_end, pos_tags))
    # If no frame tags are usable, fall back to POS tags.
    if not frame_tags_after_org and not pos_tags_after_org:
        return None

    first_after_org = (
        frame_tags_after_org[0] if frame_tags_after_org else pos_tags_after_org[0]
    )
    original = sentence.to_original_text()
    # Extract reason following ORG occurence.
    reason = original[first_after_org.start_pos:]
    return reason


def clean_organization(full_text: str):
    """ Clean an organisation name (e.g. 'Microsoft Inc.' -> 'Microsoft'). """
    cleaned = full_text.strip().lower() \
        .replace("--", "").replace('"', "").replace("'s", "") \
        .replace("'", "").replace("(", "").replace(")", "")
    if "," in cleaned:
        cleaned = cleaned.split(",")[0]
    if "." in cleaned:
        cleaned = cleaned.split(".")[0]
    cleaned = cleaned.replace(",", "")
    split = cleaned.split(" ")
    cleaned = split[0].capitalize()
    for s in split[1:]:
        cleaned = cleaned if len(s) < 4 else f"{cleaned} {s.capitalize()}"
    return cleaned


def add_to_organisation(name, reason, counts, reasons):
    """ Add a possible reason to the organisation dictionary. If no reason is present,
        count up the organisation appearance count anyways. """
    if name in reasons and reason:
        reasons[name].append(reason)
        counts[name] = counts[name] + 1
    elif reason:
        reasons[name] = [reason]
        counts[name] = 1
    else:
        reasons[name] = []
        counts[name] = 1


def pretty_print(*args):
    pp = pprint.PrettyPrinter()
    for to_print in args:
        pp.pprint(to_print)


def find_top_five(counts, reasons):
    """ Find the top occuring organisations and the reason for their appearance(s). """
    c_top = list(
        sorted(counts.items(), key=lambda item: item[1], reverse=True)
```

```
    )
    c_top_five = c_top[:5]
    r_top_five = dict((item[0], reasons[item[0]])
                      for item in c_top_five)
    return r_top_five, c_top_five


def main():
    try:
        if len(sys.argv) < 2:
            print()
            sys.exit(
                "Please supply a path for text processing (e.g. 'CCAT') as an argument for this script."
            )

        reasons, counts = find_organisations_reasons(sys.argv[1])
        no_reason = sum(v for _, v in reasons.items() if len(v) == 0)
        print(f"Number of found organisations: {len(reasons)}")
        print(f"Number of organisations with no reasons found: {no_reason}")

        top_five_reasons, counts = find_top_five(counts, reasons)
        pretty_print(top_five_reasons, counts)
    except Exception:
        traceback.print_exc(file=sys.stdout)
    sys.exit(0)


if __name__ == "__main__":
    main()
```