Following The News: Automatic Storyline Extraction*

David Jeswant Natarajan

1st March 2018

Abstract

As digital news media becomes an increasingly present and essential part of everyday life, it becomes equally necessary to organise and prepare it for consumption. Research in the field of natural language processing has produced many novel solutions in pursuit of a sufficiently powerful organisational model. Among others, topic detection and tracking is a popular solution to this problem and has powerful discriminatory properties. This approach, however, neglects the wealth of semantic information present in natural language and discourse by considering only the multiplicity of words in news articles. Moreover, it fails to provide sufficient insight into the nuanced narratives present in news. A more meaningful representation is therefore necessary to better capture the discourse of news. Building upon the power of topic detection and tracking, this research proposes the extraction of storylines as a means of interpreting cross-document narratives from news corpora.

Keywords: natural language processing, news storylines

1 Introduction

In a society that is increasingly dependent on knowing about the latest developments and having up-to-date information, it can be a daunting task to navigate through the constant torrent of news that is abundant in everyday life. It is therefore crucial to prepare this information for more convenient consumption by exploring new ways of automatically compiling, categorising, and summarising the data.

Considering textual data in particular, advances in the field of natural language processing (NLP) have spawned many novel solutions including the state-of-the-art topic detection and tracking (TDT) approach. TDT is a popular and well-studied approach to document categorisation. The goal of the application is to "detect topics [in the news] as they appear and track them through time" [4, 5]. While TDT is considered state-of-the-art, this research proposes that there may be a more effective model for summarising news data. However, to conceptualise a new or different model, the shortcomings of the previous model, TDT, should first be understood.

In order to feasibly process large corpora, these algorithms must sacrifice some level of semantic accuracy in order to reduce processing time. For instance, both of them attempt to keep a small memory footprint and achieve better performance by representing text data as a 'bag-of-words'. This means that each document is represented only by the multiplicity of words it contains, forgoing the semantic structure and linguistic subtleties of the original documents. Furthermore, each document is preprocessed by aggressively removing words that are deemed meaningless and unhelpful in identifying topics. On top of simply decreasing the amount of data to process, eliminating the vocabulary in this way allows these algorithms to reduce the effect of noise and avoid generating redundant topics.

For application in topic detection, this strategy is very effective, and topic detection remains a powerful tool that provides a nice overview of trends in the news over time. It quickly becomes clear, however, that topics are very limited in the amount of information they communicate. Herein lies the shortcoming of the topic relation, it does not provide contextual information on how these documents are related only that they are related by a short list of keywords. And while these keywords have an obvious semantic connection with the related documents, they do not provide any information on how real world events play out. It would be left up to the news consumer to read through the related documents to get a full picture. This lack of explanatory power stems from the approach of modelling documents as a bag-of-words and is a consequence of neglecting the grammatical structure of the documents as well as aggressively pruning words. In this sense, it becomes clear that TDT has very little explanatory power and is perhaps an inadequate solution to the task of identifying and summarising news stories.

Having understood this, it is evident that there is a lot of rich semantic information in each news article that is being left unused. To utilise this, a new model is sought that additionally allows the previewing of semantic relationships between documents. How can this information be incorporated into a replacement model? The solution may be to construct storylines with facts that have been extracted from relevant news items, facilitating coherent, chronological access to the data without the need of reading through each news item individually.

This is where NLP tools like Named Entity Recognition (NER) and Open Information Extraction (OpenIE) come in. They allow us to both preserve and harness the contextual information in the news articles. While NER is the process of identifying potential real world objects like people, places, and landmarks among other things, OpenIE is the process of extracting relation tuples from the text involving a subject, a relation, and an object. OpenIE can use named entities to improve results, leading to the extraction of tuples that relate entities to one another in the place of the subject and object. This is a useful

[*] This thesis was prepared in partial fulfillment of the requirements for the Degree of Bachelor of Science in Knowledge Engineering, University of Maastricht, supervisor: Gerasimos Spanakis, Evgueni Smirnov

feature to exploit in this use case.

With NER and OpenIE at our disposal, we have access to a new contextual dimension that provides the necessary leverage needed to power narrative deduction. How, then, can we utilise OpenIE triplets to infer a narrative from a corpus of news articles? An ideal solution can be found in the work of Hu, Huang, and Zhu [8], where they commandeer a graph structure called a *coherence graph.* The coherence graph embodies both the digital and visual representation of a narrative, representing news pieces as nodes positioned left to right in chronological fashion, with weighted edges that indicate the relevance or *coherence* between two news pieces. In this representation, each node in the graph would also store the OpenIE triplets that were extracted from the respective documents. Comparing the ratio of triplets that the documents share would then constitute the *coherence score* between those documents. Given this, it becomes clear that the coherence graph structure is in many ways superior to the topical relations provided by TDT. Advancing from here, the very visual nature of this graph structure, with *lines* connecting *stories*, led to the decision to name the completed coherence graph structures *story-lines* or *storylines.* In addition to this, the term storyline was chosen as it adequately encompasses the depth of document relations we hope to achieve with this research.

A storyline, in this sense, can be characterised as the linking of documents by factual relations in chronological fashion, highlighting events that cause diverging and converging news stories.

Having said this, processing a news corpus of hundreds of thousands of documents is hardly feasible with the aforementioned semantic tools as they are much more resource intensive than TDT for instance. Fortunately, as mentioned previously, TDT algorithms can give us a good idea of which documents belong to a similar topic as well as temporal information about the importance of that topic in the news. TDT can therefore be used as a starting point for the construction of storylines where the search space is limited to the documents that are related by a particular topic and time of publishing.

Problem Statement & Research Questions

Problem 1 Textual news is becoming simultaneously more abundant and more integrated in the daily functions of 21st century society. The sheer amount of data and the inconsistent formats in which it is published make it highly inefficient and inconvenient to navigate.

Problem 2 While TDT is a functional tool that organises news into general topics, it has weak summarising capabilities and is, in many cases, incapable of identifying a coherent news narrative. This is problematic, as it still requires the consumer to scrutinise the data to identify relevant news pieces.

Proposed Solution Moving forward, this research proposes the utilisation of storylines; a more expressive format for communicating the discourse of unfolding events, as a means to tackle the problem of information overload in textual news and to increase the resolution of information extraction beyond current standards (i.e. TDT). A composite solution is proposed that involves: exploiting the organisational powers of TDT; to put focus on relevant documents, applying OpenIE; to preserve the grammatical and contextual dependencies of the relevant news items, and employing the coherence graph structure of Hu et al. [8]; to utilise the factual information produced by OpenIE to identify meaningful inter-document narratives or storylines.

Questions

1. How can storylines be extracted from a collection of news items?

2. Is this method justifiable, serviceable, and computationally affordable?

3. Are the resulting storylines a practical means of summarising news?

The preceding questions are discussed and answered in section 5: the conclusion.

2 Related Work

The concept of identifying and summarising multi-document narratives is not a novel one.

Perhaps most notably, in one of the most referenced articles on the subject, Chambers & Jurafsky [6] propose a novel method for unsupervised induction of "*narrative event chains* from raw news-wire text". In their work, Chambers & Jurafsky model their narrative event chains around a *protagonist.* This protagonist is an entity that is involved in a number of events, and together, those events form a narrative event chain. In conjunction with this research, they use coreference resolution to identify repeated mentions of the protagonist across text and, in a similar fashion, use verb-dependency pairs, where we use OpenIE triplets, to express an event or fact.

Another example is the research by Lin, Lin, Li, Wang, Chen, and Li [11], where instead of extracting facts from news articles, facts are represented as entire microblog posts (such as tweets). Using keywords from a user query, the relevant microblog posts are accumulated into a graph structure where an algorithm is run to determine the most meaningful storyline.

On a more conceptual level, Vossen, Caselli, & Kontzopoulou [16] attempt to characterise what a storyline is from a "narratology" perspective, introducing the concept of a "fabula" or a plot structure and it's components: a beginning, climax, and end.

Laparra, Aldabe, & Rigau [9] propose a method to construct "StoryLines" from "TimeLines" where the TimeLines are built around particular entities and are therefore also referred to as Entity TimeLines. By finding where these Entity TimeLines intersect, i.e. when different entities interact in an event, they are able to construct StoryLines involving certain subsets of these entities.

The work by Hu et al. [8], however, was what inspired the storyline extraction method detailed in this research. It provided crucial insight to developing definitions and understanding the problem at hand as well as outlining the concept of a coherence graph and how the structure could be utilised effectively.

A common theme in the research of storylines or narratives is the conclusion that TDT is not an effective means of summarising the events of a story, and while we maintain this, this research is nevertheless an attempt to build upon the discriminatory power of TDT as a means to reduce the search space of the storyline extractor. The TDT implementation by Brüggermann, Hermey, Orth, Schneider, Selzer and Spanakis [4, 5] serves as an intermediate step in the storyline extraction process.

3 Methodology

The following section is intended to give an overview of the storyline extraction process from start to finish as well as defend the choices made in this respect.

3.1 Corpus Preprocessing

As with most data of this nature, i.e. text data, it is inevitable that the structure and style of writing are influenced by the author. On top of this, certain news articles simply do not translate well into machine readable text. For example, stock value reports often take on a list like structure and are full of numbers that themselves vary in format. It is, therefore, crucial to understand the consistencies and inconsistencies of the corpus being used and to normalise them for the best results.

3.1.1 Reuters Corpus

For the purpose of this article the Reuters RCV1 corpus was used as it suited the needs of this research and was already at hand. On top of this, it is the same corpus used by Brüggermann et al. [4, 5] for their TDT experiments which are used as a platform in this research.

According to National Institute of Standards and Technology [13]:

> In 2000, Reuters Ltd made available a large collection of Reuters News stories for use in research and development of natural language processing, information retrieval, and machine learning systems. This corpus, known as “Reuters Corpus, Volume 1” or RCV1, is significantly larger than the older, well-known Reuters-21578 collection heavily used in the text classification community.

In addition to this, the corpus “contains about 810,000 Reuters, English Language News stories” labelled with many predefined topics including sport, finance, politics and a plethora of others [13]. The publishing date of these news stories range from “1996-08-20 to 1997-08-19” [13]. These factors are ideal and allow us to be selective with the documents used for processing.

3.1.2 XML Extraction

The documents in RCV1 are formatted in XML which means that the relevant text must first be extracted for further use. Exploiting the fact that each document is already labelled with news categories, the first step is to filter out the articles with unwanted categories.

In practice, we found that topics related to business, economics, finance, and markets were often problematic for our system as they included many numbers and very little context. While not all documents were problematic, on the scale of hundreds of thousands of documents it seemed to scale better to simply remove documents from problematic categories. The unwanted topics can be found in Table 4 in the appendices.

While this first step eliminated a lot of the problematic documents, there were many that were in fact not labelled with topics at all. Fortunately, the headlines of the news articles also often had repeating naming schemes. The next step was, therefore, to discriminate against articles with headlines that included phrases or terms that occurred in our curated stopword list. The stopwords can be found in Table 5 in the appendices.

At this point it is expected that the remaining documents are relatively useful for storyline extraction. Still, however, as detailed in [4, 5]:

> The text extraction is combined with a first cleaning step by identifying and removing editorial parts of the article, usually consisting of the location of the writer and a number connected to that location. The keyword “newsroom” is used to detect and eliminate the corresponding paragraph.

Once all of these steps are complete, the text from the headline and body of the document can be extracted.

3.1.3 Regular Expression Cleaning

Once the actual text has been extracted, it needs to be cleaned. For this we use regular expressions to pinpoint common text elements that are considered noise or are unhelpful in discerning factual information from the text.

The steps used are:

- Remove XML character entities.
- Replace “&” with “and”.
- Remove “.com”.
- Remove digit group separators from numbers.
- Remove unwanted special characters.

Each of these steps revolve around getting the best results from OpenIE portion of the application as this phase is the most sensitive to strange text formats and noise in general. Finally, the text can be written to individual text files for storage.

3.2 Topic Detection and Tracking

The TDT software detailed in Brüggermann et al.'s research [4, 5] offers two main approaches, namely an NMF approach and an LDA approach. Considering time constraints, it was only practical to use the NMF approach which was significantly faster than its LDA counterpart. According to Brüggermann et al. [4, 5], using an "i7 mobile CPU of the 4700 series ... the computation of the full corpus [using the LDA approach] can be estimated as taking 30 days". The NMF application, in comparison, completed the cleaned corpus within an afternoon. Moreover, The LDA implementation involved manually running separate, C, Python, and R scripts all of which required the user to either install dependencies, compile the script separately, or both. Needless to say, this is quite cumbersome, and given that "both algorithms show the powerful ability to develop comprehensive topics out of a large text data set" [4, 5], it was concluded that the NMF approach will suffice for the purpose of this research.

3.2.1 Non-Negative Matrix Factorisation

Non-negative matrix factorisation is an algorithm that can be used in the field of text mining to factorise and thereby decrease the dimension of a large matrix [10]. According to Brüggermann et al. [4, 5], there are two resulting matrices, "one representing a term-topic relation (how much a term is relevant to a topic), the other one representing a topic-document relation (how much a topic is relevant in a document). When multiplied with each other, these matrices form an approximation of the original matrix and fill in zero-values, acting as predictors for the cell values." It also has the property that all matrices have no negative elements making it easier to inspect.

Figure 1 shows a visual representation of NMF.

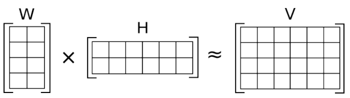

Figure 1: Illustration of approximate non-negative matrix factorization: the matrix V is represented by the two smaller matrices W and H, which, when multiplied, approximately reconstruct V. [14]

For NMF to work, the text documents must first be put into a term frequency representation. This is done via a matrix, where each row represents a word in the vocabulary and each column represents a document. The cell values, in this case, represent the number of times the respective word occurs in the respective document.

With the current matrix, however, an advantage would be given to terms that occur frequently across many documents. These terms are often, by extension, not very expressive or useful for topic detection. To avoid this problem a TF-IDF value can be used in place of the number of occurrences. TF-IDF stands for "Term Frequency - Inverse Document Frequency". Inverse document frequency is a measure of how much information a word provides, that is, whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient. As argued earlier, a term that appears in many documents many times is most likely inexpressive. If this is the case, then, conversely, a term that appears many times in only a few documents must be highly expressive. Hence, including this, we have the formula:

$$TF - IDF = TF \times log_{10}\Big(\frac{N}{DF}\Big)$$

Where TF-IDF is value that will replace the term frequency in the matrix, TF is the term frequency of the respective term in the respective document, N is the number of documents in the matrix, and DF is the document frequency or the number of documents in which the term appears.

Once all the TF-IDF scores have been calculated and inserted into the matrix, it can finally be factorised and the topics deduced. It is important to note that in the case of tracking over time, this process is repeated for documents published in the consecutive time steps.

The actual NMF application detailed in Brüggermann et al. [4, 5] comes in two parts. The first part is the topic extraction phase which performs the actual NMF calculations on text document input and returns an XML document per time step summarising the topics extracted. Running it requires seven parameters: interval size by number of days, number of intervals, date format, start date, corpus directory, output directory, and number of topics to be extracted. Here, an interval is equivalent to a time step and defines the search space or group of documents that were published within that time step. Combined with the results from other intervals, the topics can be tracked over time. It is also important to note that the text documents should be stored in directories according to date with the directories labelled with the specified date format.

The second phase of the NMF application involves combining the topic data extracted from each time step. This process then outputs a final list of topics as well as a mapping of the news articles to the topics. This will be the input for the storyline extraction application.

Figure 2 shows the steps of the NMF application visually.

3.2.2 Latent Dirichlet Allocation

Despite the decision to avoid the use of the LDA application, the steps involved are outlined here.

According to Blei, Ng, and Jordan [3],

> [LDA is] a generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document.

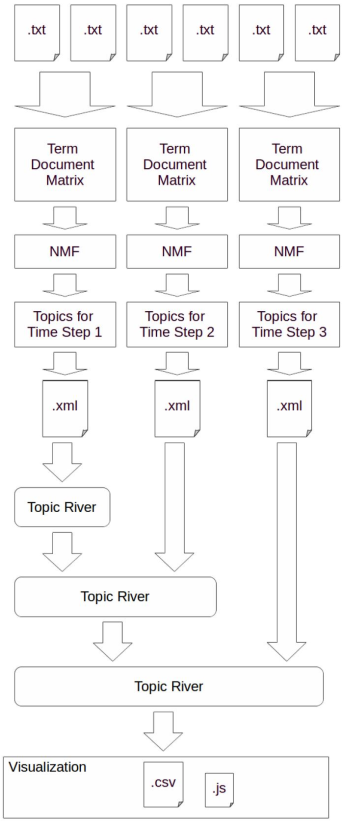

Figure 2: A diagram outlining the steps taken in Brüggermann et al.'s NMF application. [4, 5]

Figure 3 shows a visual representation of the standard LDA model.

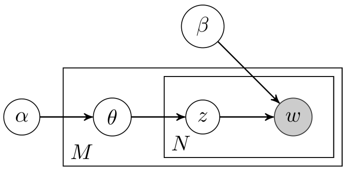

Figure 3: Plate diagram representing LDA as a Bayesian network. The outer box represents M documents, the inner box represents the N times repeated choice of topics and words within a document. θ represents the topic distribution per document. α and β represent the concentration parameters for the per-document and per-topic Dirichlet distributions. The variable w is observable while the other variables are latent. Edges denote dependencies among variables. [3]

According to Brüggermann et al. [4, 5], "Words define a vocabulary and topics are represented by a probabilistic distribution of words from this vocabulary." Similar to NMF, LDA represents documents by their term frequencies, foregoing all grammatical structure.

Advancing from here, the next step is to expand the model for topic tracking. In the case of Brüggermann et al. [4, 5],

> LDA is used on topics aggregated in time epochs and a state space model handles transitions of the topics from one epoch to another. A Gaussian probabilistic model to obtain the posterior probabilities on the evolving topics along the time line is added as additional dimension.

Figure 4 shows a graphical representation of the dynamic topic model.

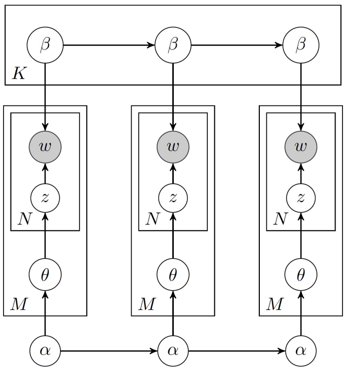

Figure 4: Plate diagram representing the dynamic topic model (for three time slices) as a Bayesian network. The model for each time slice corresponds to Figure 3. Additionally, each topics parameters α and β evolve over time. [2]

3.2.3 Topic River

In order to make use of the output data from TDT, there must be a suitable interpretation and representation. After analysing Brüggermann et al.'s work [4, 5], it becomes clear that topic waves and rivers are an ideal representation for the output. Apart from having nice visual properties, these structures provide an easy starting point for storyline extraction.

A topic wave in this sense is the accumulation of documents related to a particular topic. A topic river is then the combination of all the topic waves and, therefore, all of the documents in the corpus. Considering that each document is anchored in time by its publishing date, a topic river grows and shrinks depending on how many documents published in that time period are relevant to that topic.

The separation of topics, accompanied by the temporal aspect of the topic river, provide an excellent starting point for limiting the document search space of the storyline extraction process.

Figure 5 shows the a topic river being visualised by Brüggermann et al.'s application [4, 5].

3.3 Storyline Extraction

This section aims to explain the process of extracting storylines given a news corpus and topical mappings of its documents. This outlined system is the main contribution of this research and is an attempt at retrieving and summarising news information from a massive stream of news.

3.3.1 Open Information Extraction

As mentioned in the introduction, preserving the grammatical structure of the news articles and harnessing their semantic information is the key to finding meaningful storylines. This is where NER and OpenIE come into play.

Fortunately, for those working in the field of NLP, There is a plethora of modern, powerful NLP tools and libraries

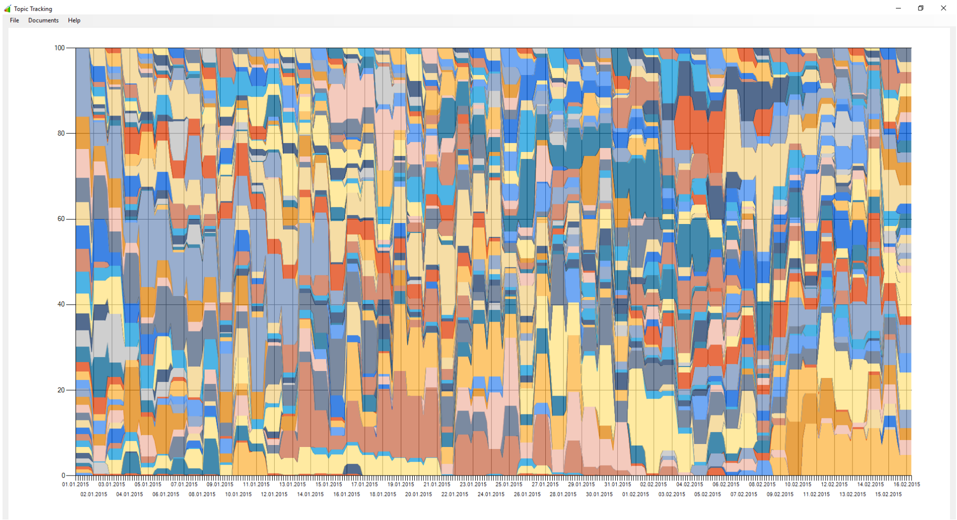

Figure 5: Visualisation of a year long topic river by Brüggermann et al. [4,5]. Here, each time step is clearly identifiable with a distribution of topics that are differentiated by colour. It can also be seen how topics are dynamic, disappearing and reappearing as time progresses.

available for commercial use and research. The Stanford NLP group, however, has undoubtedly curated one of the most comprehensive NLP development suites on the market today. It is because of their Stanford CoreNLP pipeline [12] that we are able to do this research. This pipeline incorporates most of their NLP tools into one streamlined annotator including all of the relevant semantic tools that this research is based on. In order to use this system, the pipeline must be initialised with the desired annotators and options and then one can simply pass their text documents to the pipeline to be annotated.

One of the annotators available with the Stanford CoreNLP Pipeline is the OpenIE annotator [1]. As stated on the OpenIE home page on the Stanford NLP website [7];

> Open information extraction (open IE) refers to the extraction of relation tuples, typically binary relations, from plain text. The central difference is that the schema for these relations does not need to be specified in advance; typically the relation name is just the text linking two arguments. For example, Barack Obama was born in Hawaii would create a triple (Barack Obama; was born in; Hawaii), corresponding to the open domain relation was-born-in(Barack-Obama, Hawaii).

Figure 6 shows an example of how OpenIE triplets are extracted from a sentence.

To annotate documents with OpenIE triplets, they must first be annotated with other features. Although not the default option, annotating the document with named entities encourages the OpenIE annotator to extract triplets involving said entities, leaving less room for redundancy. In addition to this, it is possible to run a coreference resolution annotation on top of the named entities, reducing "pronouns to their canonical antecedent." This means that not only will the OpenIE annotator prefer triplets with named entities, but the named entities across all the extracted triplets will be placed in their single most expressive form, effectively normalising the triplets. This makes it easier, later on, to compare triplets against each other.

Considering the nature of the task at hand, it seems only reasonable to trigger these options. The unfortunate consequence, however, is that the processing power and memory requirements of running this program increases significantly. Having said that, it nevertheless remains feasible to run the program as long as the document search space is sufficiently small.

In order to narrow the document search space sufficiently the user must select a topic from the list of topics, generated by TDT as output, as a starting point for OpenIE annotation. This is, in a way, a sort of user query that directs the search toward a certain topic, the only difference being that the user is limited to preset queries. Moreover, considering that a good portion of documents relating to a topic will likely be located near the conception of the topic, we narrow the search space temporally, dictating that if there is a gap in the data of more than a week, the remainder of the topic wave is cut off.

Moving on, it becomes clear that the produced OpenIE triplets resemble events and facts that are highly contextual and provide excellent insight into the happenings described in news articles, ideal for use as a basis for constructing informative news narratives. This, in combination with the tactical search space reduction, will allow for a relatively efficient process that maximises the chance of producing relevant and coherent storylines. For simplicity, OpenIE triplets will be referred to as *facts* for the remainder of the article.

With this in mind, we utilise the Stanford CoreNLP parser with coreferenced OpenIE annotation and write the discovered facts from each of the documents in the search space to a single CSV file for further use.

3.3.2 Storyline Construction

Now that there is a collection of facts at our disposal, how do we communicate their role in a story? Furthermore, how do we link these news pieces together in a coherent

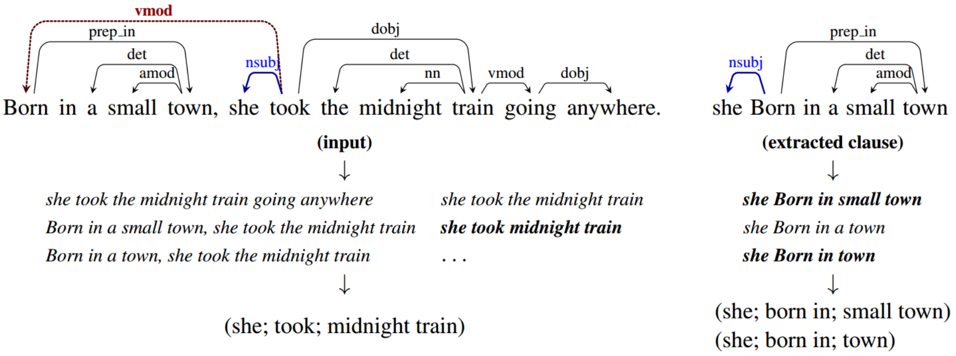
vmod
prep_in
det
amod
nsubj
dobj
det
nn
vmod
dobj
Born in a small town, she took the midnight train going anywhere.
(input)
she took the midnight train going anywhere
Born in a small town, she took the midnight train
Born in a town, she took the midnight train
she took the midnight train
she took midnight train
. . .
(she; took; midnight train)
nsubj
prep_in
det
amod
she Born in a small town
(extracted clause)
she Born in small town
she Born in a town
she Born in town
(she; born in; small town)
(she; born in; town)

Figure 6: To extract OpenIE triplets (without named entities) the system first splits each sentence into a set of entailed clauses. Each clause is then maximally shortened, producing a set of entailed shorter sentence fragments. These fragments are then segmented into OpenIE triples, and output by the system. [7]

fashion? Ideally we need a structure that can express all of this as well as the temporal aspect of the news. This research stipulates that a *coherence graph* structure is in fact the ideal representation that is sought.

In the coherence graph, news articles are to be represented as nodes, placed from left to right in accordance with document publishing dates, with ranking of importance dictating the size of the node. The edges between them are to represent a factual relation between the documents and shall be weighted to indicate the level of relatedness or coherence.

The idea of using a coherence graph structure was inspired by the work of Hu et al. [8] who described a system of evaluating inter-document coherence with a "combination of coherence factors". On top of this, they outlined a method of identifying "informative events" through a PageRank algorithm. Unfortunately, we do not have access to the same kind of information that they used to put together their coherence metrics and, in any case, this information is out of the scope of this research. In this case, we must define our own coherence metric and a method for ranking the importance of documents.

Making use of the extracted fact data, the implementation produced for this research uses the ratio of facts shared between two documents to the collective number of facts in the two documents as a means to quantify coherence. While there are more sophisticated metrics, like those detailed in the work of Hu et al [8], the normalisation effect that coreference resolution has on the facts hopefully still allows for the generation of representative coherence scores.

Now that a coherence metric has been defined, we need a way to identify important documents in the graph. To communicate different levels of importance in documents, similar to a PageRank algorithm as suggested by Hu et al. [8], we use the number of coherent documents, i.e. the number of documents with a coherence score higher than 0, as the measure of importance. In this way we can identify not only which documents are in fact related, but also how important or pivotal they are in the general narrative. Like the coherence metric, the importance scores produced are representative of reality and truly help in identifying documents that report on important news stories.

As mentioned in the introduction, the graph-like structure produced, and its chronological format, draws a keen similarity to the concept of projecting stories onto intersecting lines, hence the term *story-lines* or *storylines* is used to characterise the distinct sub-graphs in the complete coherence graph.

3.4 Visualisation

Using the GraphStream library [15], a rudimentary visualisation can be generated. Unfortunately, due to time constraints, the graph visualisation is not ideal as certain elements overlap or are difficult to make out. Nevertheless, the visualisation does indeed display the news articles as nodes, labelled with respective headlines, positioned from left to right in chronological fashion and with the size of each node indicating its importance. It also displays the edges between coherent documents labelled with highlighted coherence scores.

An example of the visualisation can be seen in Figure 8.

At this point, the user can close the visualisation and terminate the program. This leads us to the end of the methodology section. A visual summary of the entire process can be seen in Figure 7.

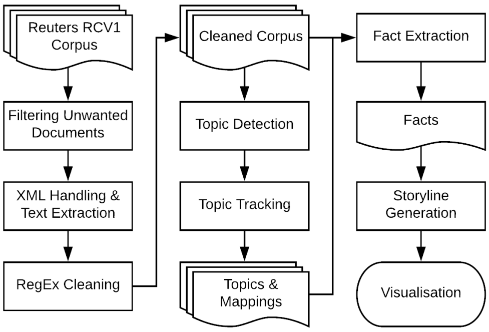

Figure 7: A flow chart highlighting the steps involved in the storyline extraction process.

4 Experiment

Following the procedure outlined in the methodology, this experiment attempted to extract storylines from the Reuters RCV1 corpus.

The initial step of handling and preprocessing the original XML files was easily the most tedious of the entire process. Effectively going on the basis of trial and error, an unwanted topics list and stopword list was constructed and the RegEx cleaning portion was tuned to get a relatively unproblematic corpus. The exact lists are available in the appendices and the resulting RegEx cleaning steps are listed in the methodology.

The unwanted topic list contains the labels from Reuters' predefined topics where many of news articles were unfavourable for fact extraction, i.e. had messy formatting, consisted of large lists of numbers, and had indiscernible sentences. The stopword list has roughly the same idea and contains words or phrases that were found to repeat in the headlines of unfavourable documents. The results can be seen in Table 1.

Filtering Options	Documents
None	806,791
Topic Filtering	140,028
Headline Filtering	614,368
Topic & Headline Filtering	105,971

Table 1: Number of documents in the RCV1 corpus at different stages of filtering.

Around 106,000 of the original, roughly, 810,000 news articles were kept. This is a huge reduction in articles, and while we can expect relatively clean and useful articles to have been kept, we can also expect that that the information lost from those articles might be damaging to the final storylines. This can be circumvented with more comprehensive and selective document filtering.

Again, the RegEx cleaning was centred around making sentences easily identifiable by the fact extractor.

Next, we needed to pass our cleaned corpus into the TDT application. Choosing the NMF implementation over the LDA implementation was a result of actually running the LDA application and realising that it would not terminate in a reasonable time. On top of this, the advent of rerunning LDA in case of poor results was intimidating because, as mentioned earlier, the plethora of manually triggered options and side scripts were cumbersome, to say the least.

Moving on, recall that there are seven parameters for running the topic detection step of NMF: interval size by number of days, number of intervals, date format, start date, corpus directory, output directory, and number of topics to be extracted. Obviously, only 3 of the parameters will have an effect on the outcome, namely interval size by number of days, number of intervals, and number of topics to be extracted.

For the sake of convenience, it was decided that intervals would be 7 days or a week long and there would be 52 intervals, effectively capturing a year of data. This, thankfully, matches up perfectly with the publishing dates of the corpus and does not leave out any data. The only other main decision regards the number of topics to extract for each interval. The decision was made to extract 30 topics per time step following the advice of Brüggermann et al., stating that "for weekly processing, 30 topics result in a decent separation of the content without many irrelevant topics." In practice, however, we find that not much can be said for the quality of the output as the resulting topic mappings are difficult to inspect. The topics, however, being represented as a group of topical keywords, seemed cohesive and sensible as far as we could tell. The parameters used are summarised in Table 2.

TDT Parameters	Inputs
Interval Size	7
No. of Intervals	52
Date Format	yyyyMMdd
Start Date	20-08-1996
Corpus Directory	`out\corpus_cleaning\RCV1`
Output Directory	`out\topic_detection_and_tracking\nmf\topics_and_mappings`
No. of Topics	30

Table 2: Summary of parameter inputs used for TDT.

In total, under these conditions, the NMF application gave 195 topics as output.

After reading the topic data into the application, it was time to pass our cleaned corpus and the topic river into the fact extraction portion of the application. But first, of course, as outlined in the methodology, the user must first choose a topic from the topic list. This allows the application to focus on documents that are relevant to the chosen topic.

Making the assumption that any topic in the list would yield an equally valid storyline, we simply chose the first topic in the list. However, if needed, this step makes it possible for a user to select a topic which they find most relevant to their search. The keywords of the topic were: whales, massachusetts, shipping, watched, and scientists. As explained earlier in the methodology, we narrow the document search space accoring to topic and time. In relation to time, we take the documents from the conception of the topic in the topic river to the first disappearance of the topic in the topic river, i.e. when there is a gap of a week in the publishing dates of the related documents. Taking this into account, the relevant search space in our case was determined to have 105 documents.

The relevant documents were then passed to the Stanford CoreNLP pipeline to be annotated with OpenIE triplets or facts. In order to get the OpenIE annotations there are some precursor annotations that need to be specified. According to the OpenIE home page on the Stanford NLP website [7], The annotators we need to add to the pipeline in order to get OpenIE triplets are "tokenize, ssplit, pos, lemma, depparse, natlog, openie". In addition to these, in order to enable coreference resolution we need to add the "parse, ner, mention, and coref" annotators. Using coreference resolution allows us to effectively normalise our facts for easier inspection later on.

An example of this can be seen in Table 3.

OpenIE Triples		
Without Coreference Resolution		
Obama	was born in	Hawaii
Obama	was	born
He	is	our president
With Coreference Resolution		
Obama	be bear in	Hawaii
Obama	be	bear
Obama	be	we president

Table 3: OpenIE triplets with and without coreference resolution for the string: "Obama was born in Hawaii. He is our president."

In the example, the term "He" was resolved to be in reference to the named entity "Obama" and so was represented in the triple as such. The same effect will take place with other named entities in the text, like people and places. Clearly, using coreference resolution is a good idea if we want to compare these facts later on. However, as can also be seen in the example, the annotator will "resolve pronouns to their canonical antecedent" [7] which effects the legibility and coherence of the facts produced. This is the compromise we have to make, as choosing machine readability over human readability will improve the overall coherence of the storylines assembled.

Finally, passing the fact filled CSV file to the coherence graph portion of the application yielded the fact-ratios seen in Table 6 and the storylines seen in Table 7.

As can be seen in Table 6, the system has compared the facts of each document and found the number of facts that are shared between them. While there are some documents which share a lot of factual information, the majority of documents which do indeed share facts share only 1 fact. This is suggestive of a number of things;

- That our fact-ratio metric is substandard and does not effectively identify factual links between documents.
- That the documents labelled with the chosen TDT topic were not in fact very related to begin with.
- That the TDT produced bad topic mappings because our preprocessing steps removed too many valuable and relevant documents from the corpus.

These points are problematic as they will inevitably decrease the overall coherence of our storylines. The first point, however, is the most problematic as it is the final metric used to decide which document actually belong to a particular storyline. If this metric is comprehensive, it should always produce coherent storylines and if there are no actual storylines in the data, the metric should indicate that any storylines produced are in fact incoherent or have a low coherence.

Despite this, our system has nevertheless extracted some storylines as can be seen in Table 7. The good part is that on inspection of the headlines, most of these storylines are meaningful and we have a decent correlation between the phrases. Unfortunately, however, this is not the case for all of them and, moreover, most of our storylines include only 2 documents. It can only be speculated at this point, but in practice it seems that these results are quite poor as out of 105 documents in the search space, only 22 were found to be involved in a continuing story. Furthermore, on inspection of the actual facts, it turns out that some documents were linked by redundant facts, for example Nodes 60, 67, and 68 were all related by a fact stating "newspaper.report on.Saturday", which, if compared with the headlines, clearly is irrelevant to the content of the articles and links documents which from a user standpoint do not belong in the same story. Having said this, the mixed results were just that, mixed. There were some prime examples of documents being justifiably linked and that made sense in a story, such as the story involving nodes 9, 70, and 79, and there were examples of documents being linked which were not justifiably part of the same story, such as the story involving nodes 22, 67, 60, 62, and 68.

In any case, the final step was to visualise the coherence graph. A screen-shot of the visualisation can be seen in Figure 8.

On inspection of the admittedly messy visualisation, it is possible to identify separate storylines. For example there is a storyline regarding beached whales in Australia, or a storyline involving Prince Charles and the British royal family. It is also possible to identify the nodes which are supposedly more important as they are larger in size than others. The visualisation tries to place the nodes in chronological order along the x-axis and in random order along the y-axis unfortunately failing to place them in a readable way. The edges are labelled with the coherence scores between the respective documents; however, many edges overlap rendering them indiscernible from the other.

5 Conclusion

Going back to the introduction, we recall that the research questions were;

1. How can storylines be extracted from a collection of news items?
2. Is this method justifiable, serviceable, and computationally affordable?
3. Are the resulting storylines a practical means of summarising news?

How can storylines be extracted from a collection of news items? storylines can be and have been extracted from the Reuters RCV1 corpus via the application produced as a result of this research and that has been outlined in the methodology. The produced storylines can be seen in Table 7.

Is this method justifiable, serviceable, and computationally affordable? Throughout the methodology and experiments sections, many arguments were put forward in defence of the decisions made in the production

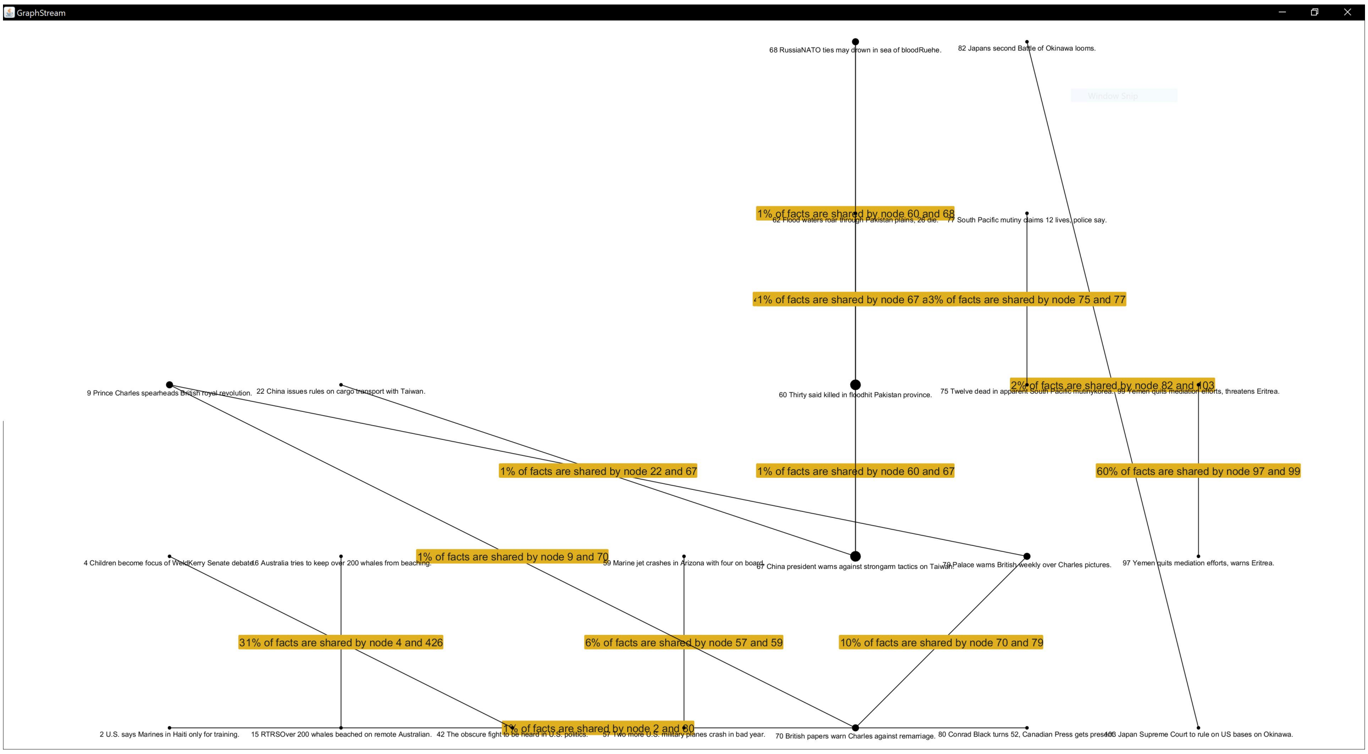
GraphStream
68 RussiaNATO ties may drown in sea of bloodRuehe.
82 Japans second Battle of Okinawa looms.
1% of facts are shared by node 60 and 68
62 Flood waters roar through Pakistan plains, 26 die.
77 South Pacific mutiny claims 12 lives, police say.
‹1% of facts are shared by node 67 a3% of facts are shared by node 75 and 77
9 Prince Charles spearheads British royal revolution.
22 China issues rules on cargo transport with Taiwan.
60 Thirty said killed in floodhit Pakistan province.
75 Twelve dead in apparent South Pacific mutinykorea.
2% of facts are shared by node 82 and 103
99 Yemen quits mediation efforts, threatens Eritrea.
1% of facts are shared by node 22 and 67
1% of facts are shared by node 60 and 67
60% of facts are shared by node 97 and 99
4 Children become focus of WeldKerry Senate debate
16 Australia tries to keep over 200 whales from beaching.
1% of facts are shared by node 9 and 70
59 Marine jet crashes in Arizona with four on board.
67 China president warns against strongarm tactics on Taiwan.
79 Palace warns British weekly over Charles pictures.
97 Yemen quits mediation efforts, warns Eritrea.
31% of facts are shared by node 4 and 426
6% of facts are shared by node 57 and 59
10% of facts are shared by node 70 and 79
2 U.S. says Marines in Haiti only for training.
15 RTRSOver 200 whales beached on remote Australian.
42 The obscure fight to be heard in U.S. politics.
1% of facts are shared by node 2 and 80
57 Two more U.S. military planes crash in bad year.
70 British papers warn Charles against remarriage.
80 Conrad Black turns 52, Canadian Press gets president
103 Japan Supreme Court to rule on US bases on Okinawa.

Figure 8: Storyline visualisation using the GraphStream library. Nodes are organised left to right in chronological fashion, i.e. documents published on the same day stack vertically on top of each other. Nodes are labelled with document headlines and are sized according to importance. The highlighted edge labels express the ratio of shared facts between two documents.

of the storyline extraction application. However, as can be observed in the results, There are issues with the application which resulted from making assumptions about the data and the performance of intermediary steps. The problems encountered during experimentation were: that the preprocessing phase is not comprehensive enough and potentially removes many important or relevant articles, that the TDT phase produced poor topic mappings and placed highly unrelated documents together in a topic, and that the coherence metric used is not effective at linking related documents. These problems reduce the quality of the storylines produced in turn decreasing the eligibility this method as a candidate for further research. However, with the consideration that this application was produced as an initial prototype, further iterations have a lot of room for improvement, especially considering that the aforementioned issues have already been identified. Furthermore, the highlighted issues are preventable.

In the case of the preprocessing phase, a more comprehensive deconstruction of the consistencies and inconsistencies of the corpus will reveal better ways of cleaning it. In regards to the cleaning methods used, perhaps the filtering process could be discarded entirely, leaving the bulk of the cleaning process to be done by regex cleaning. This would ensure that no important documents are filtered out, and at the same time, with sufficiently thorough regex cleaning, would remove troublesome in-text structures like tables and lists of numbers that inhibit the functioning of the OpenIE program.

In the case of the TDT phase, it might be favourable to use different parameters. Increasing the amount of topics generated may improve the relatedness of the documents within each topic.

And finally in the case of the coherence metric, A more profound combination of identifiers like the appearance and frequency of named entities in documents could help to find more meaningful inter-document interactions. Furthermore, after being understood, the "combination of coherence factors" employed by Hu et al. [8] can be appropriated and incorporated into further iterations of the software.

Moving on from the problems faced during experimentation, it is worth considering the usability of the software included with this research. For the most part, the software is relatively easy to use, only requiring the user to indicate the relevant computer directories for the respective files. While the application requires the user to run the 4 phases (preprocessing, TDT, fact extraction, storyline extraction) manually one after the other, this can be easily streamlined in following versions.

In addition to this, with regards to being accessible to the general public, this software, along with the dependencies, are written in java which is a highly portable and cross-platform language. In terms of computational efficiency, the most demanding portions of the program were TDT and OpenIE. With reference to the experiment detailed above, the entire application could run to completion within a day on a machine with a quad-core i7 Intel CPU and 8GB of memory. This is a reasonable amount of time considering the size of the corpus was around 810 thousand documents.

Returning back to the question, the application outlined in this research has ultimately achieved its goal in producing discernible storylines and has brought forward a more meaningful representation of news data through these storylines. In addition to this, it performs decently within reasonable computational constraints and provides an uncomplicated interface for user control. In this sense, we believe that, with some refinement, the storyline extraction method proposed in this article can be established as a proof of concept, and can justifiably be used as a template for further research in this regard.

Are the resulting storylines a practical means of summarising news? With the advent and abundance of digital news media, the condensed view of news in chronological storylines makes it much more palatable to search for information. Promoted by user-submitted search terms, the system highlighted in this article can be adapted to find news articles that are increasingly relevant to the users search query. With an improved visualisation, perhaps including document summaries or condensed statements of occurring events, storylines can be pushed to the mainstream of how we consume news. In this way, it can be seen that storylines are indeed a practical way of summarising news and have a great potential to be an industry standard.

References

[1] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 344–354, 2015.

[2] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] Daniel Brüggermann, Yannik Hermey, Carsten Orth, Darius Schneider, Stefan Selzer, and Gerasimos Spanakis. Storyline detection and tracking using dynamic latent dirichlet allocation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 9–19, 2016.

[5] Daniel Brüggermann, Yannik Hermey, Carsten Orth, Darius Schneider, Stefan Selzer, and Gerasimos Spanakis. Towards a topic discovery and tracking system with application to news items. In *International Workshop on Future and Emerging Trends in Language Technology*, pages 183–197. Springer, 2016.

[6] Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, 2008.

[7] The Stanford Natural Language Processing Group. Stanford open information extraction. `https://nlp.stanford.edu/software/openie.html`. Accessed: 27-02-2018.

[8] Po Hu, Min-Lie Huang, and Xiao-Yan Zhu. Exploring the interactions of storylines from informative news events. *Journal of Computer Science and Technology*, 29(3):502–518, 2014.

[9] Egoitz Laparra, Itziar Aldabe, and German Rigau. From timelines to storylines: A preliminary proposal for evaluating narratives. In *Proceedings of the First Workshop on Computing News Storylines*, pages 50–55, 2015.

[10] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[11] Chen Lin, Chun Lin, Jingxuan Li, Dingding Wang, Yang Chen, and Tao Li. Generating event storylines from microblogs. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 175–184, 2012.

[12] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[13] NIST National Institute of Standards and Technology. Reuters corpora @ nist. `http://trec.nist.gov/data/reuters/reuters.html`. Accessed: 27-02-2018.

[14] Qwertyus. Non-negative matrix factorization (nmf). `https://commons.wikimedia.org/wiki/File:NMF.png`. Accessed: 27-02-2018.

[15] Graph Stream Team. Graph stream a dynamic graph library. `http://graphstream-project.org/`. Accessed: 27-02-2018.

[16] Piek Vossen, Tommaso Caselli, and Yiota Kontzopoulou. Storylines for structuring massive streams of news. In *Proceedings of the First Workshop on Computing News Storylines*, pages 40–49, 2015.

Appendices

A Code, Dependencies, and Resources

All of the code, its dependencies, and additional resources (including the output files of the experiment outlined in this article) can be found on Github at `https://github.com/davidjnatarajan/storyline-extraction`.

B Tables & Lists

Code	Description
2ECO	CURRENT NEWS - ECONOMICS
3SPO	CURRENT NEWS - SPORT
6INS	CURRENT NEWS - INSURANCE
CCAT	CORPORATE/INDUSTRIAL
ECAT	ECONOMICS
G151	EC INTERNAL MARKET
G154	EC MONETARY/ECONOMIC
GSPO	SPORTS
MCAT	MARKETS
MEUR	EURO CURRENCY
GWEA	WEATHER

Table 4: Unwanted Topic List. List of topics from Reuters predefined topics that contained documents deemed problematic for the Stanford OpenIE annotator.

stock
stocks
index
silver
gold
currency
revenue
revenues
profit
profits
earnings
shares
finance
takeover
net
trade
blue chips
bskyb
pct
bid
press digest
reuter ec report longterm diary
official journal contents
emergency weather conditions
reuters historical calendar
government list
radio
headlines
diary
cinema
timelines

Table 5: Stopword List. List of terms and phrases that consistently appear in the headline of documents deemed unfavourable for the Stanford OpenIE annotator.

N	2 U.S. says Marines in Haiti only for training.	4 Children become focus of WeldKerry Senate debate.	9 Prince Charles spearheads British royal revolution.	15 RTRSOver 200 whales beached on remote Australian.	16 Australia tries to keep over 200 whales from beaching.	22 China issues rules on cargo transport with Taiwan.	42 The obscure fight to be heard in U.S. politics.	57 Two more U.S. military planes crash in bad year.	59 Marine jet crashes in Arizona with four on board.	60 Thirty said killed in floodhit Pakistan province.	62 Flood waters roar through Pakistan plains, 26 die.	67 China president warns against strongarm tactics on Taiwan.	68 RussiaNATO ties may drown in sea of bloodRuehe.	70 British papers warn Charles against remarriage.	75 Twelve dead in apparent South Pacific mutinykorea.	77 South Pacific mutiny claims 12 lives, police say.	79 Palace warns British weekly over Charles pictures.	80 Conrad Black turns 52, Canadian Press gets present.	82 Japans second Battle of Okinawa looms.	97 Yemen quits mediation efforts, warns Eritrea.	99 Yemen quits mediation efforts, threatens Eritrea.	103 Japan Supreme Court to rule on US bases on Okinawa.
2	23																	1				
4		36					1															
9			40											1			1					
15				39	11																	
16				11	27																	
22						54						1										
42		1					77															
57								50	2													
59								2	16													
60										49	22	1	1									
62										22	51											
67						1				1		92	1									
68										1		1	60									
70			1											64			7					
75															101	3						
77															3	54						
79			1											7			72					
80	1																	164				
82																			174			3
97																				74	36	
99																				36	46	
103																			3			76

Table 6: Coherence scores or number of facts shared between nodes in the coherence graph, where N is the ID of a node.

Node IDs	Article Headlines
2	U.S. says Marines in Haiti only for training.
80	Conrad Black turns 52, Canadian Press gets present.
4	Children become focus of WeldKerry Senate debate.
42	The obscure fight to be heard in U.S. politics.
9	Prince Charles spearheads British royal revolution.
70	British papers warn Charles against remarriage.
79	Palace warns British weekly over Charles pictures.
15	RTRSOver 200 whales beached on remote Australian.
16	Australia tries to keep over 200 whales from beaching.
22	China issues rules on cargo transport with Taiwan.
67	China president warns against strongarm tactics on Taiwan.
60	Thirty said killed in floodhit Pakistan province.
62	Flood waters roar through Pakistan plains, 26 die.
68	RussiaNATO ties may drown in sea of bloodRuehe.
57	Two more U.S. military planes crash in bad year.
59	Marine jet crashes in Arizona with four on board.
75	Twelve dead in apparent South Pacific mutinykorea.
77	South Pacific mutiny claims 12 lives, police say.
82	Japans second Battle of Okinawa looms.
103	Japan Supreme Court to rule on US bases on Okinawa.
97	Yemen quits mediation efforts, warns Eritrea.
99	Yemen quits mediation efforts, threatens Eritrea.

Table 7: Storylines represented as article headlines divided by story. Inter-document coherence scores can be found in Table 6.