NEWSREADER PUBLIC SUMMARY

Grant Agreement number: ICT 316404
Project acronym: NewsReader
Project title: Building structured event indexes of large volumes of financial and economic data for decision making
Funding Scheme: FP7-ICT-2011-8
Period covered: from 1/1/2013 to 1/1/2016
Name of the scientific representative of the project's co-ordinator, Title and Organisation:
Prof. dr. Piek Vossen, Faculty of Arts, VU University Amsterdam
Tel. + 31 (0) 20 5986466
Fax: Fax. + 31 (0) 20 5986500
E-mail: piek.vossen@vu.nl
Project website: http://www.newsreader-project.eu

Contents

1 Final publishable summary report

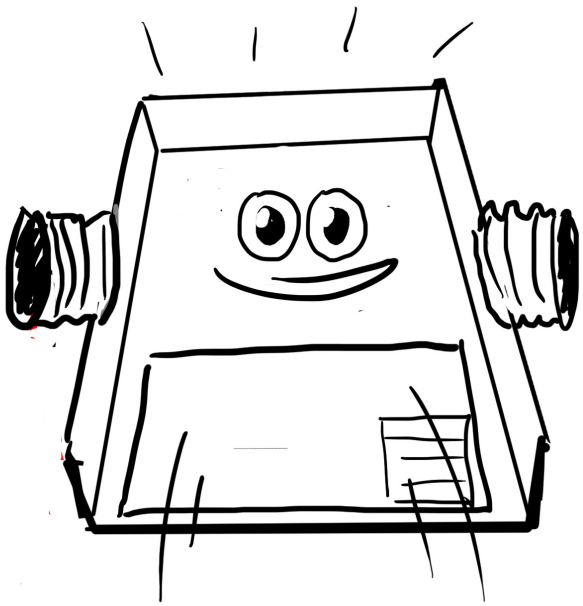
1.1 Executive summary

We developed a so-called Reading Machine that can read massive streams of news in 4 languages: English, Spanish, Italian and Dutch. The machine extracts *what* happened and *who* was involved, *where* and *when*. This information is represented in the form of billions of Semantic Web *triples* and stored in a KnowledgeStore that supports reasoning over the data. This allows us to detect trends, events with impact and social networks of people over time and regions. We can query long-term developments spanning decades for individuals or types of individuals to discover events that remained unnoticed. The project developed new and unique visualisations of the rich and complex data structures that provide efficient and intelligent access to the data. Currently, NewsReader technology is used in almost 40 follow-up projects.

1.2 Summary description

Sheila is a senior spokesperson of the Ministry of Internal Affairs. She monitors the daily stream of news: hundreds of documents per day! When Sheila reads an article that rumours about falsifications in CO2 emissions by a national automobile manufacturer she needs to advice her Minister on a response in the media within sixty minutes. To do this well, she needs to have an accurate picture of the full history, spanning decades, of the manufacturer, the management currently in charge and the connections with the government. Access to an overwhelming amount of data - millions of relevant documents - makes it almost impossible for Sheila to find the information she needs in the short time-span she has to give a well-informed advice.

Watch the story of Sheila and NewsReader's solution here on Youtube: https://www.youtube.com/watch?v=rYLaVN3oqLI&feature=youtu.be.



LexisNexis estimates the total volume of news that they archive per working day on 1.5 million articles. About 25% is about finance and economy: five-hundred-thousand documents per working day. A period of 10 years spanning the financial crisis will add up to an enormous volume of news and data. This accelerating growth of knowledge and information makes it nearly impossible to stay on top of developments. Making informed decisions and finding out about the consequences of your decisions becomes more difficult for businesses, governments and also citizens. Information and data not only grow exponentially because they become digitized. At the same time, our (online) activity and mobility accelerate, expanding our networks and intensifying the dynamics between them.

Professionals in any sector depend on access to accurate and complete knowledge to make well-informed decisions. Think about lawyers, politicians, heads of purchase in large firms, compliance directors and journalists. A missing piece of crucial information can be fatal to make the right decision. To find this crucial piece they need to search for a needle in a haystack, simply because there is more data than ever, it is highly interconnected through the Internet and quickly gets out of date in this rapidly changing world.

NewsReader developed the reading machine: a machine that can read millions of news items in four languages (English, Spanish, Italian and Dutch) day by day. This way, NewsReader helps these professionals allowing them to find that needle in the haystack by structuring information as stories: relating new events to past events. It stores all the details in the so-called KnowledgeStore and can intelligently reason over it by reconstructing histories over decades. It is thereby able to tell all the details about any person or organisation as told by thousands of different sources. NewsReader is also capable of measuring the impact of events on the people involved rather than the impact of news on the journalists and people that follow the news. Whereas the latter measures how much talk there is about topics or how trendy they are, NewsReader models the extent to which people are affected by the event: e.g. increase or decrease of ownership or sales, loss of jobs, etc.

How this works? In four simple steps: Identification, Deduplication, Aggregation and Perspectivation. IDAP.

Identification: NewsReader first identifies an event in text through similar components, by extracting *what* happened to *whom*, *when*, and *where*.

Deduplication: NewsReader makes sure similar information is represented only once, referencing every article across many sources in the haystack.

Aggregation: NewsReader aggregates complementary information across thousands of different sources in a single representation.

Perspectivation: NewsReader makes sure differences and different view points are traceable through their sources and mentions in text.

The result of this IDAP method is a complete, exact and rich record of the past, with access to original sources. The information from the news is stored as billions of 'factoid' statements, so- called **RDF triples**.

The overall architecture for this process is shown in Figure 1. In a first step, we use Natural Language Processing technology (NLP) to detect events, actors and time expressions in the news in 4 languages. The result is stored in XML files according to the Natural Language Processing Annotation Format (NAF), that we defined. NAF is interoperable across different languages. The result is stored in the KnowledgeStore, which is a scalable database platform for storing massive amounts of source data and interpretation layers of this data.

But this is not it! Typically, the same entities, dates and events are mentioned many times in news articles and especially across many different articles published around the

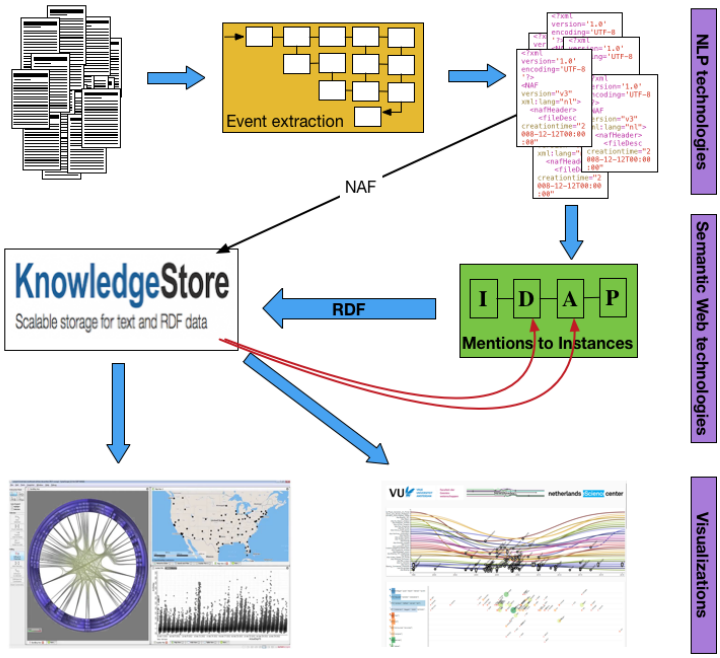


Figure 1: Global Architecture of the NewsReader reading machine

same day, which we can expect to report on similar events. We therefore make a distinction between mentions in (textual) sources and instances in the (assumed) world. We therefore *reinterpret* the Natural Language Processing output in NAF to an instance level, where each unique event and entity is represented only once using the IDAP method. The result is an RDF representation of the knowledge on the event, following the Simple Event Model (SEM) that is also stored in the KnowledgeStore.

This second processing step is illustrated in Figure 2, where show two descriptions of the same event from two different sources that use different words and expressions. The information is mapped to a unique representation of an event instance (Event_{12}, with labels *buy* and *sell*). The event has *buyer* and *seller* roles to entities that are identified through their DBpedia[1] identifiers. Since NewsReader interprets the events as instances of event types (represented here as *Commerce_money_transfer*), the system also understands what the transferred goods between these entities are. In this example, we see that similar information is deduplicated despite the different ways it was expressed. The event type predicts that there is also money involved but this is not expressed in the current source texts. Other (future) sources may tell us the amount of money paid and thus can provide this missing information to complete the picture. This will then lead to aggregation. We call these data structures Event-Centric Knowledge-Graphs or ECKGs, because all information is aggregated from different sources around the event rather than an entity.

[1]DBpedia is a Semantic Web database with content derived from Wikipedia, containing millions of entities and properties of these entities

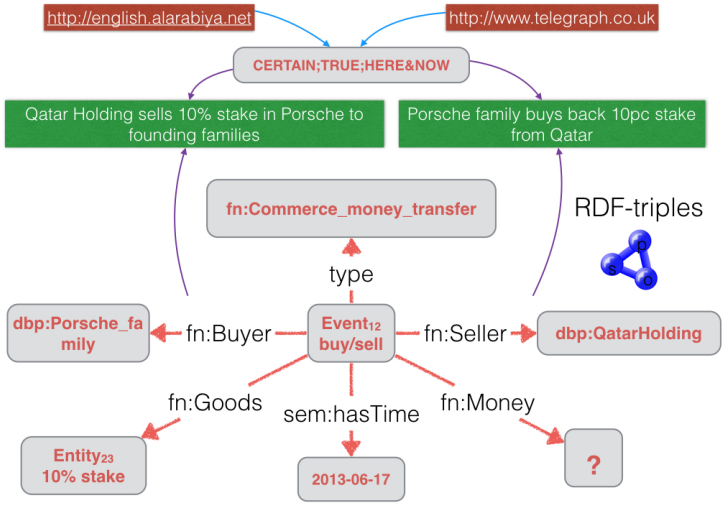


Figure 2: Representing instances of events and entities across sources

Aggregation and deduplication are important to make the correct inferences. If for example 10 articles report on a *sale* of *10% stake*, we need to know if we can add up these sales to 100%, which implies they report on different sales, or 10%, in which case all articles report on the same event, or anything in between.

Since we keep the link between the RDF representations of event and entity instances and their mentions in the original sources, we can go back to the original sources at any point to show where the knowledge and information is derived from. This allows us to model the perspective of the sources of the news on the events (both the authors and publishers of the news as well as the cited sources in the text). Our *reading machine* builds a data structure that also makes explicit *who* said what, how *certain* they are, whether they *deny or confirm* it and what *emotion* they have towards the events. This data layer is also stored as RDF triples in the KnowledgeStore and can be used to model the perspective on events across many different sources.

Processing millions of news articles over decades, as has been done in NewsReader, results in a KnowledgeStore filled with billions of RDF triples which are little 'factoid' statements on events and perspectives linked to the source texts. Each data element is bound to time and sometimes also to place, and is semantically typed according to ontologies. This allows for reasoning over the data (what are the implications of events) and allows for deep semantic search (using SPARQL). Querying for types of people (e.g. *management*) and companies (e.g. *car manufacturers*) and also for types of events (e.g. *financial transactions* or *crimes*) in which they are involved makes it possible to visualise trends over time and/or in regions, show biographies and social networks and event storylines of sequences of events with causal connectivity. At the bottom of Figure 1, we show two high-end visualisations developed in the project that can be used to efficiently

access the data, detect correlations and trends and discover hidden events that remained unnoticed so far. The project used these interfaces and the KnowledgeStore in end-user experiments and hackathons that study the effectiveness of our data processing and modelling for professionals.

1.3 Main results and foreground

NewsReader can rapidly read texts in four different languages and creates a single Semantic Web representation (RDF triples) to represent so-called event-centric data across different text sources and different languages. The information is stored in a scalable KnowledgeStore that can hold background knowledge and supports reasoning. The reading machine as a complete system is a major achievement that integrates many different components that are also important achievements in themselves. We summarise these components briefly below. In Section **??**, we provide a complete list of all the foreground results.

1.3.1 Architecture and design

We defined a unique system architecture that is open, flexible and extendable and that combines Natural Language and Semantic Web technology providing a technology bridge between unstructured and structured data. The data flow is shown in Figure 3. Textual sources are processed through pipelines of Natural Language Processing (NLP) modules that store the result in NAF-XML format. Next we interpret the mentions in NAF to instances in SEM and compare these across different articles. We store the final results as SEM-RDF triples.

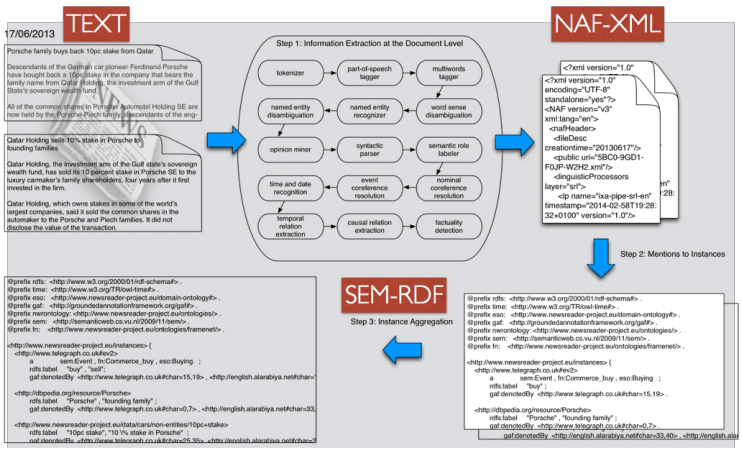


Figure 3: Overview of process and data flow

In Figure 4, we show the abstract SEM model that is used to capture the resulting RDF data. SEM allows modelling relations between event instances, actors, time and place.

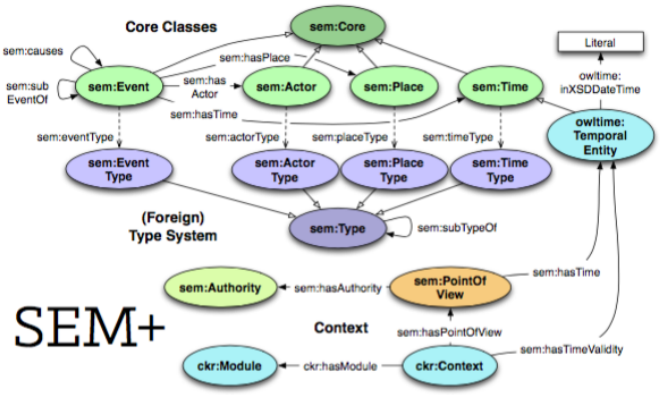


Figure 4: Simple Event Model

Whereas SEM is instance based, NAF representations allows for the annotation of mentions in text with interpretations. We defined the Grounded Annotation Format to link instance to mentions using gaf:denotedBy links. Each mention of information is also attributed to a source, which can either be the author or somebody cited. We model these attribution relations in our GRASP model (General Representations and Annotations of Sources and their Perspectives), which allows for the expression of perspectives of sources on events represented in SEM. Since each mention can represent a different perspective, we link the perspective to each mention. In Figure 5, we show a schematic overview how GAF combines all three models where different mentions in textual sources are mapped to the same SEM instance through denotedBy links and each mention is mapped to a perspective through hasAttribution and to a source through wasAttributedTo links.

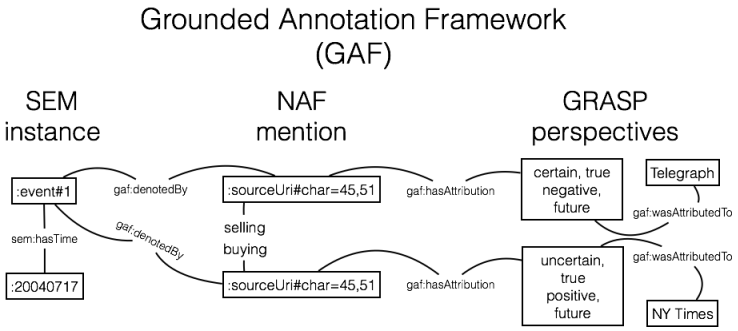


Figure 5: High level overview of the NewsReader models

Figure 6 then shows all formal triple relations for the semantics of the two example sentences presented before. We can see here that there is sharing of the SEM data, there are links to background ontologies indicating the type of event, there are links to their

mentions represented in NAF and to the perspective represented in GRASP.

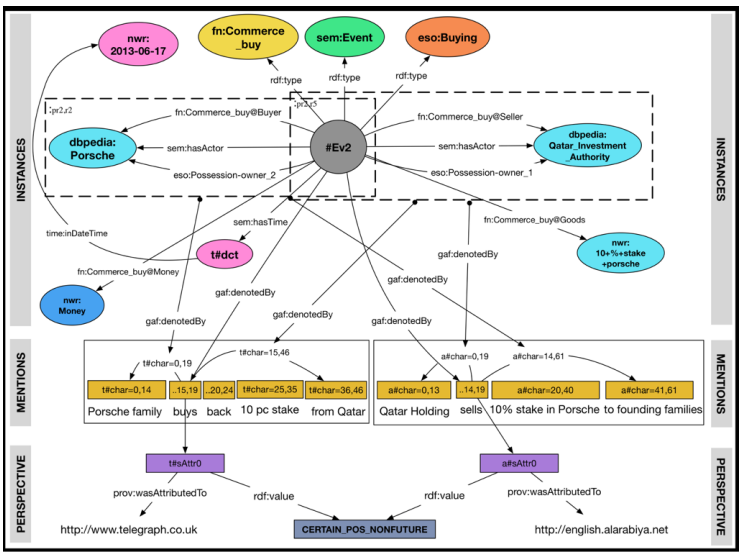


Figure 6: Example represented in NewsReader models: NAF, SEM, GRASP following GAF

Modelling the perspective allows us to find all statements of spokesmen over time and get an overview of their position and sentiment with respect to the events. In Figures 7 and 8, we show such lists for the Porsche CEO *Wiedeking* extracted from a large data set of news articles on the automotive industry processed by NewsReader.

The rich and complex modelling of data in NewsReader has a high potential for future research and technology development. Our models can deal with textual and non-textual sources and can be applied to any language in the world. We have been invited to participate in an ISO working group to investigate the standardisation of NAF. Our framework is already applied beyond the project's lifetime by external third parties.

The software design to implement the processing is shown in Figure 9. It shows a series of Natural Language modules that process incoming sources, such as news batches provided by LexisNexis, adding different interpretation layers. The central box represents the KnowledgeStore that contains all the source data, the interpretation layers and the final SEM-RDF triples with data. The KnowledgeStore can be queried by tools such as the visualisations developed by Synerscope.

We defined different architectures for parallel processing in batch mode (using Hadoop) and streaming mode (using Storm and a Mongo database). The Hadoop architecture performs best for large batches of data (millions), whereas the Storm architecture is optimal for continuous streams of data. We have demonstrated that our architecture can deal with massive amounts of news within operational limits: hundreds of thousands articles per day.

WENDELING WIEDEKING'S PERSPECTIVE

Event	Nr.	Perspective
achieve	10	CERTAIN_u_POS
decision	6	CERTAIN_NON_FUTURE_NEG
earnings	19	CERTAIN_FUTURE_POS
earnings	19	CERTAIN_NON_FUTURE_POS
earnings	17	u_u_u
focus	6	CERTAIN_FUTURE_POS
goal	7	CERTAIN_u_POS
increase	18	u_u_u
increase	15	CERTAIN_NON_FUTURE_POS
increase	11	CERTAIN_u_POS
increased	9	CERTAIN_NON_FUTURE_POS
increased	8	CERTAIN_u_POS

Event	Nr.	Perspective
interest	8	CERTAIN_NON_FUTURE_POS
interest	8	CERTAIN_u_POS
plans	9	CERTAIN_NON_FUTURE_POS
plans	9	CERTAIN_u_POS
predict	17	CERTAIN_FUTURE_POS
predict	17	CERTAIN_NON_FUTURE_POS
predict	17	u_u_u
stake	21	CERTAIN_u_POS
stake	19	CERTAIN_NON_FUTURE_POS
stake	10	u_u_u

Figure 7: Perspective on event expressed by Porsche CEO Wiedeking

WENDELING WIEDEKING'S SENTIMENT

Event	Nr.	Sentimen
interest	6	positive
voting	6	positive
location	3	negative
production	3	negative
waging	3	negative
war	3	negative
exercising	3	positive
gives	3	positive
have	3	positive
position	3	positive
restructuring	3	positive
build	2	positive
expertise	2	positive
hoping	2	positive
models	2	positive

Event	Nr.	Sentimen
retain	2	positive
wants	2	positive
avoid	1	negative
questioning	1	negative
team	1	negative
acquired	1	positive
influence	1	positive
maintain	1	positive
manufacturer	1	positive
order	1	positive
raise	1	positive
seeGermany	1	positive
stake	1	positive

Figure 8: Sentiment on event expressed by Porsche CEO Wiedeking

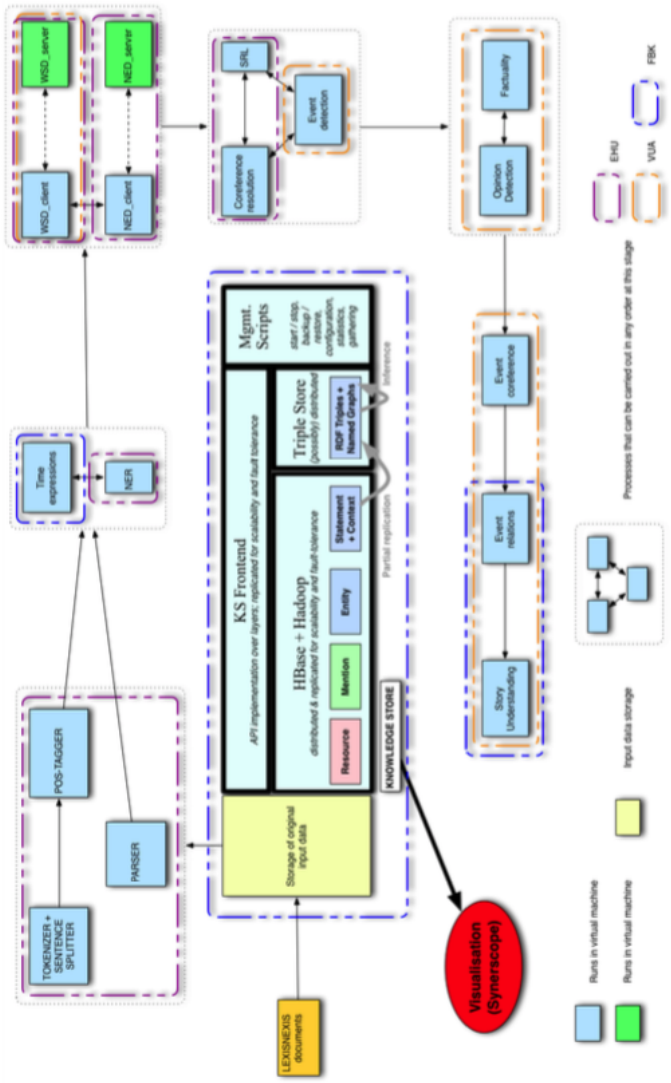


Figure 9: Overall software architecture

1.3.2 Interoperable Natural Language Pipelines in four languages

We created *reading machines* in four languages that can be downloaded as Virtual Machines and deployed for parallel processing. Each reading machine consists of a pipeline with many modules for processing textual sources, ranging from tokenisation and grammatical analysis up to detecting entities, linking these entities to databases, detecting time expressions and normalising them to ISO dates, detecting events and relations between actors, time and events.

Figure 10: Overview of the English pipeline

We tested the main modules on standard data sets comparing them to the state-of-the-art. For all four languages, our systems perform at or above state-of-the-art levels. This is remarkable since the NewsReader pipelines have not been trained specifically for these testing data sets. This also implies that NewsReader could perform relatively stable across general documents or news, while there is sufficient room for further improvement and adaptation to other domains. Even more remarkably, the performance for the Spanish, Dutch and Italian pipelines is similar to the English performance while having less appropriate linguistic resources and annotated datasets.

1.3.3 Cross-document and cross-language event modelling

Although each NewsReader pipeline is different, the output is interoperable across the different languages. Entities are mapped to English DBpedia entries (URIs), dates are normalised to ISO dates (e.g. *yesterday* will be interpreted as a date) and even events are

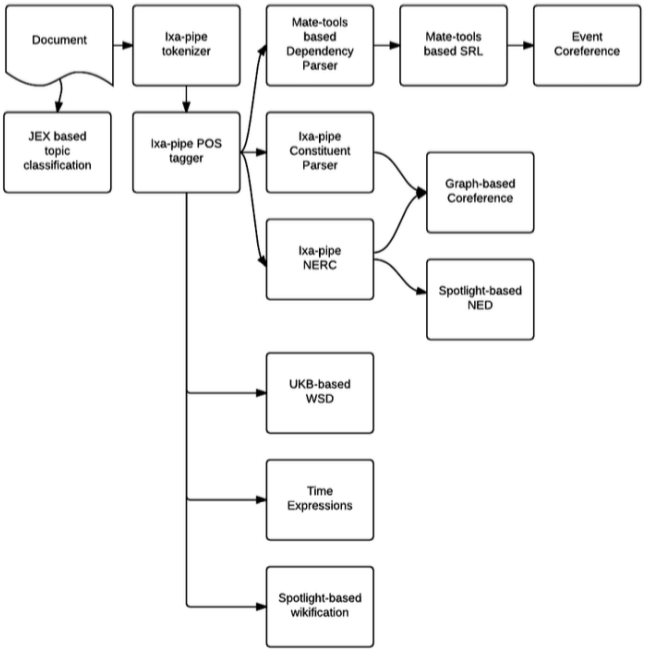


Figure 11: Overview of the Spanish pipeline

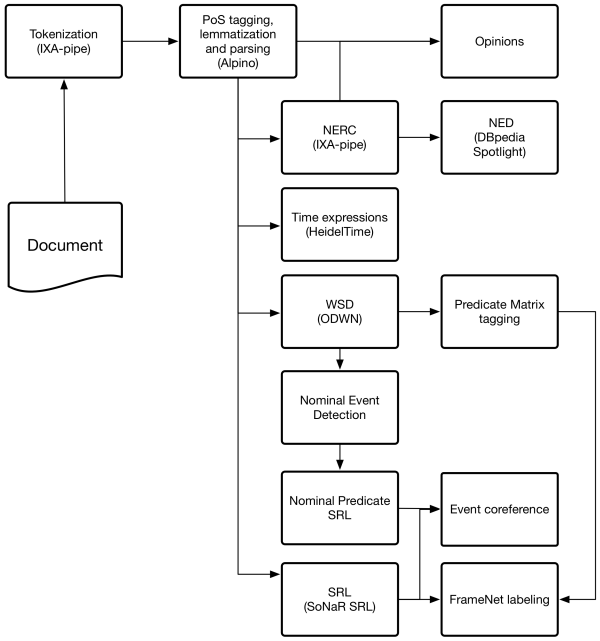


Figure 12: Overview of the Dutch pipeline

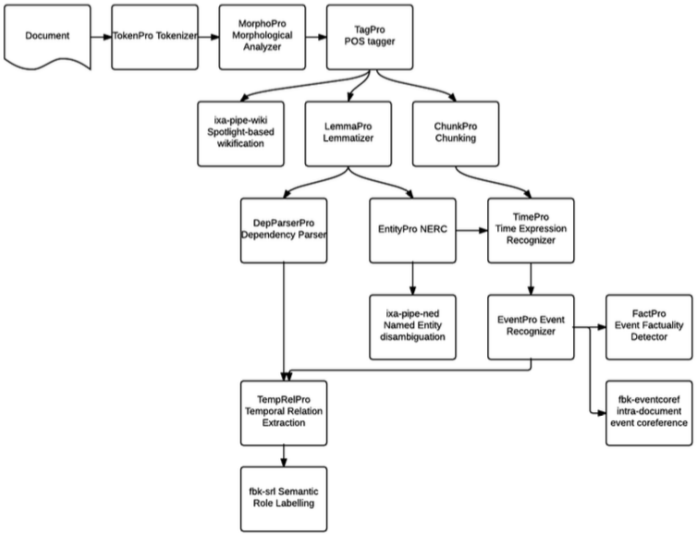


Figure 13: Overview of the Italian pipeline

mapped to a shared ontology across the languages (see Subsection 1.3.7 below). As a result of that, we are capable of representing the pipeline output of the four languages in the same SEM representation using the same language independent approach. We developed software that can translate the interoperable interpretations to these SEM representations following the IDAP method. The module first extracts event data from a single source document by detecting mentions of the same events and entities. Figure 14 illustrates how *coref* relations connect mentions referring to the same event in a single source thus gathering information on the event that is spread throughout the document. After aggregating the event data into a Composite Event representation, we then compare these representations across different sources. When comparing events, we abstract from the way the information is expressed, i.e. we do not base our comparison on the exact words that are used to describe the event, but rather on the kind of event they refer to as modelled by our language-independent ontology. Following this approach, we can merge information within and across sources and as well as across languages.

The granularity of identity can be adapted depending on the need of the users and the type of data that is processed.

1.3.4 Manually annotated evaluation data

We created two unique data sets (MEANTIME and ECB+) to test the semantic processing of text. Both are freely available and are already used by other researchers outside the consortium. In Figure 15, we show the MEANTIME corpus that was manually annotated for many of the semantic layers in NewsReader. We translated the English originals to other languages and annotated the translations in the same way as the English sources. The

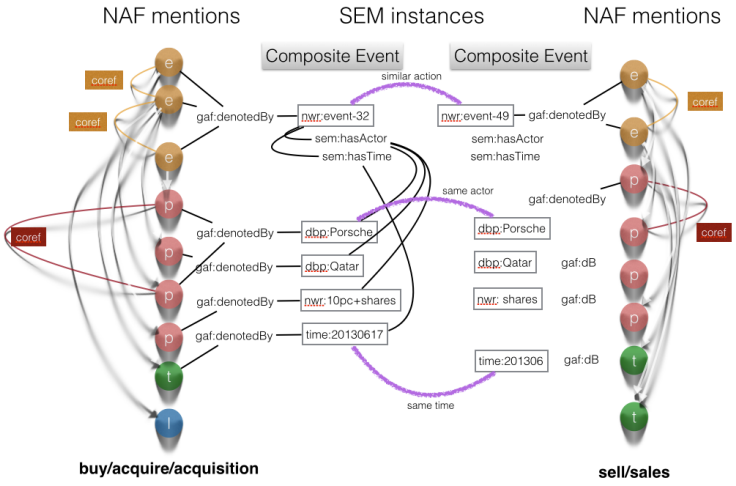


Figure 14: Interpretating mentions as instances, applying IDAP

data set is unique because it combines many semantic annotations, contains annotations across documents and annotations across languages.

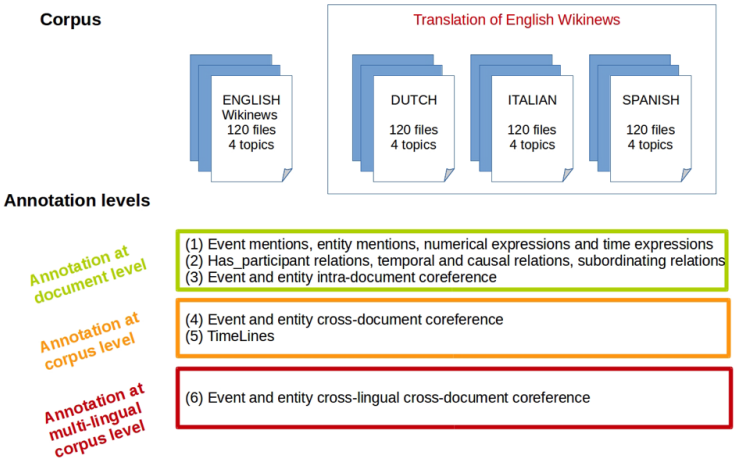


Figure 15: MEANTIME benchmarck dataset with annotations across 4 languages

We also created the ECB+ data set. This is an extension of the original Event Coreference Bank (ECB) developed at Stanford University for the purpose of cross-document event-coreference. The cross-document coreference task consists of determining that two different documents make reference to the same event in the world. We added more seminal events to the data set to make the task less trivial. This has already led to publications by other researchers testing on our data.

1.3.5 KnowledgeStore technology

The KnowledgeStore is at the heart of the NewsReader system. It is a scalable database platform that can handle a variety of data streams and the relation to the interpretation of these data streams in the form of RDF triples. It allows for reasoning and inferencing on the data, possibly combined with background data. Figure 16 shows an overview of the architecture.

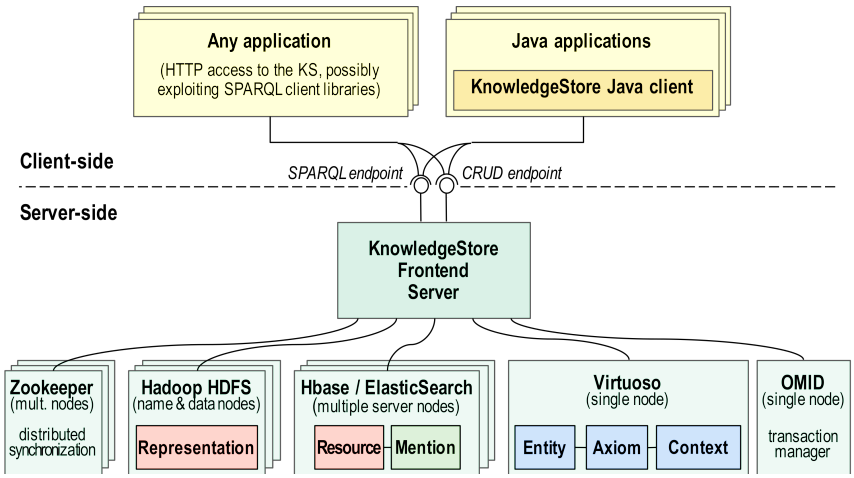


Figure 16: KnowledgeStore architecture

During the project, we populated the KnowledgeStore with massive data sets and its performance was thoroughly tested through a series of hackathons where hundreds of thousands of queries were fired by several teams of developers. We demonstrated that the KnowledgeStore performed well during these stress tests.

1.3.6 Processed data and Event Centric Knowledge Graphs

In addition to the software, NewsReader processed massive streams of news (millions of articles) thus generating large and rich data sets. In Table 1, we provide an overview of the data sets that were all loaded into the KnowledgeStore. We list the number of articles processed, the number of mentions of things (events and entities) and the actual number of instances, where the entities are subdivided into persons, organisations and locations. We also show how many have been mapped to DBpedia and how many have not. The latter make up a large proportion of the data making them an important object of research. We call *dark entities* as there is no background information about them available. The table also shows that we can apply the system to different domains without any adaptation. The final rows show the number of statements (Triples) for each data set. These statements are divided into background knowledge (from DBpedia) and those based on the mentions in the news (from Mentions). In the case of the largest data set, 2.3 million English articles on the Automotive Industry, we see that more than a billion statements are derived and

stored. This data set represents a massive history of the industry during the financial crisis over a period of more than 10 years.

1.3.7 Modelling events and their implications

NewsReader creates event-centric knowledge graphs or ECKGs. These ECKGs represent the changes in the world on which the news reports. It is very important to model these changes properly. We therefore developed the Event and Situation Ontology or ESO. The ontology captures the main types of events that occur in the automotive industry data set as shown in the hierarchy of Figure 17.

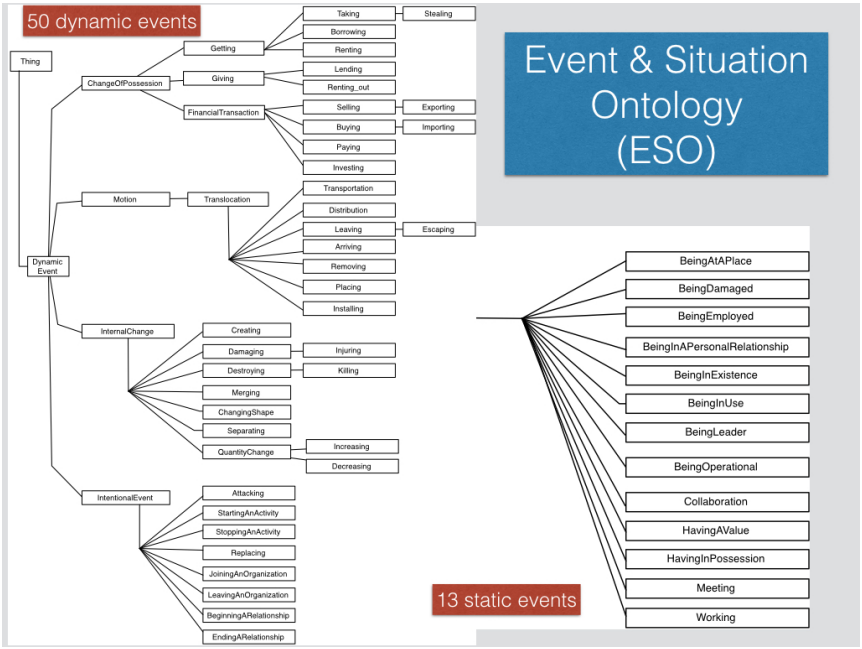


Figure 17: Event and Situation Ontology hierarchy

The hierarchy in ESO not only captures the most important events, it was specifically designed to formally model the implications of the events. An event that happens at a point of time, implies that something changed, e.g. *ownership* in case of *selling* or *buying*, or *employment* relations as a result of *hiring* or *firing*. In Figure 18, we show how ESO captures these implications for the involved actors. ESO makes explicit what entities are affected how and when by the events reported in the news. This allows us to reason over the changes and create timelines of changes for specific individuals.

The ESO hierarchy has been mapped to other well-known event ontologies as well as to wordnets in the different languages. Through the PredicateMatrix, another major resource developed in the project, we are capable of connecting ESO to the words in the 4 languages. The reading machines in NewsReader thus can detect what ESO event is

Table 1: Data processed during the NewsReader project. Numbers are obtained from the KnowledgeStore. Some data sets were processed several times.

	MEANTIME	**WikiNews (Ver. 2)**	**FIFA WorldCup**	**Dutch Parliament**	**Cars (Ver. 3)**
Topic	General News	General News	Sport, Football	Financial crisis	Automotive Industry
Period		2003-2015	2004-2014	around 2008/2009	2003-2015
News Providers	wikinews.org	wikinews.org	LexisNexis BBC, The Guardian,	Dutch House of Parliament	LexisNexis LexisNexis
Language	English	English	English	Dutch	English
Populated in	October 2015	October 2015	May 2014	June 2015	October 2015
Pipeline Version	3.0	3.0	1.0	1.0-dutch	3.0
News Articles	120	19,755	212,258	597,530	2,316,158
Mentions	35,237	5,206,202	76,165,114	9,231,113	842,639,827
Event instances	3,333	632,704	9,387,356	5,383,498	42,296,287
Entity instances	339	40,314	858,982	111,579	2,263,156
Persons	82	17,617	403,021	43,546	895,541
in DBpedia	46	10,784	40,511	13,942	126,140
Organizations	172	14,358	431,232	44,139	1,139,170
in DBpedia	115	4,940	15,984	12,907	44,458
Locations	85	8,339	24,729	23,894	228,445
in DBpedia	81	7,369	16,372	11,167	76,341
Triples	95,219,534	110,861,823	240,731,408	188,296,316	1,240,774,944
from Mentions	1,046,544	16,688,833	136,135,841	65,631,222	1,146,601,954
from DBpedia	94,172,990	94,172,990	104,595,567	122,665,094	94,172,990
distilled from	DBpedia 2015	DBpedia 2015	DBpedia 3.9	DBpedia 2014	DBpedia 2015

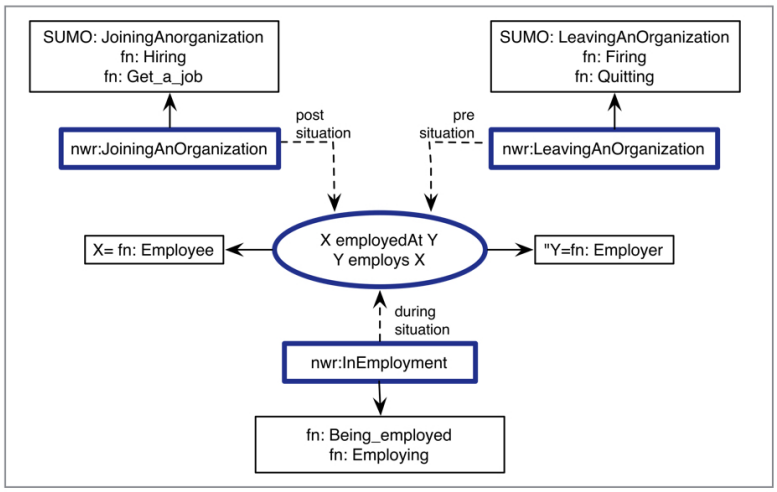


Figure 18: Pre and post-situations modeled in ESO

mentioned and what the ESO roles are of the actors in the event. Through the ontology, the KnowledgeStore can then infer the implications of the ESO situations for individuals.

1.3.8 NewsReader intelligence

The semantic data structures and ECKGs produced by NewsReader are very powerful if it comes to finding information, detecting trends, receiving notifications on unexpected developments or high-impact events and observing connections between people, organisations and events. The data sets contain millions of people and organisations as well as millions of events. Obviously, you can search for a specific person such as the Porsche CEO *Wiedeking* involved in a specific event such as being *sued*. You may find out that this indeed happened and was described in certain news articles at a specific point in time. But if you do not know what happened you also do not know what questions to ask.

However thanks to the ontologies in NewsReader, you can also ask more general questions such as all key persons or CEOs being involved in any court examination at any moment in time. This will give you the complete set of events and their reports in the news, including the case in which *Wiedeking* was *sued*. This is possible because NewsReader interprets the data using its background ontologies such as ESO for events and DBpedia for entities, as is shown in Figure 19 by the red dotted lines at different abstraction levels.[2]

[2]Since a large proportion of the entities is not in DBpedia, we also show a DBpedia version here with the dark entities and knowledge that NewsReader can derive for the entities from the news.

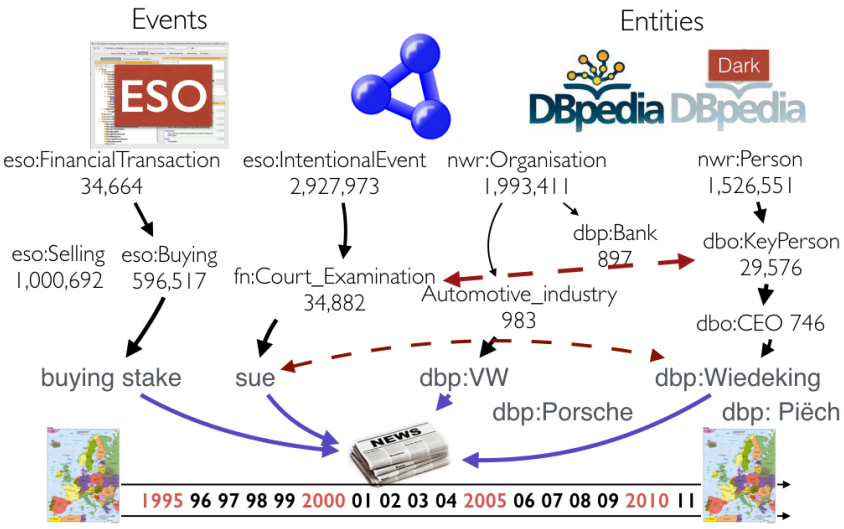


Figure 19: Semantic search in NewsReader using ontological classifications of event and entity instances

Although our data sets contains millions of events, people and organisations, the ontologies define intermediate levels to generalise over these individuals in many different ways. We can use this to observe more general trends (e.g. increase of lawsuits involving CEOs over time) and get complete overviews but we can also use this to find single events that are 'hidden' in the massive data. Hidden events are events that have once been reported in the news but that we are not or no longer aware of. These events can be still very critical for professional decision makers but are difficult to find in the data, e.g. a decision to acquire a company may depend on knowing with whom they did business in the past. On the basis of the semantic search, we designed the so-called *the triple haystack method* to discover such hidden events in the NewsReader data. The method is based on the assumption that the *need for news* can be roughly divided into three specific questions:

Relevant impact event of today: Everyday there is a massive stream of news but not all of it is interesting and there is too much to follow. The first problem people have is to monitor this stream of news and find the bits that matter to them.

Anybody involved in an impact event: Assume you happen to hear about an impact event, the second question could be to find everybody that is involved so that you can trace their history and role in relation to the impact event.

Hidden events that explain the impact event: Assume you know the names of the people and organisations that are involved you need to get an overview of all the

past events and see how they connect. This history may contain information that was published a long time before but nobody realised its value and importance for the impact event of today.

The answers to these three questions correspond to three needles that need to be found in three haystacks. Figure 20 represents this situation. We see a haystack to the right that represents 1 million events in today's news, one of which can be crucial. We see another haystack in the middle with 1 million people in the database of which one may be important and involved in today's news. Finally, the haystack at the left represents all the 42 million events from the past. One or some of these events may be crucial in connection to the person in the second haystack and today's event in the first haystack.

How to find these 3 needles? You can start the search process with the middle haystack and list all the people or organisations you care about, in which case you simply pick your first needle and then look for events with impact that they are involved in from the daily news: the right-most haystack. Alternatively, you may first look for impact events in general in the daily news regardless of who is involved and then look who is involved. In either case, it is important to know what are the events with impact.

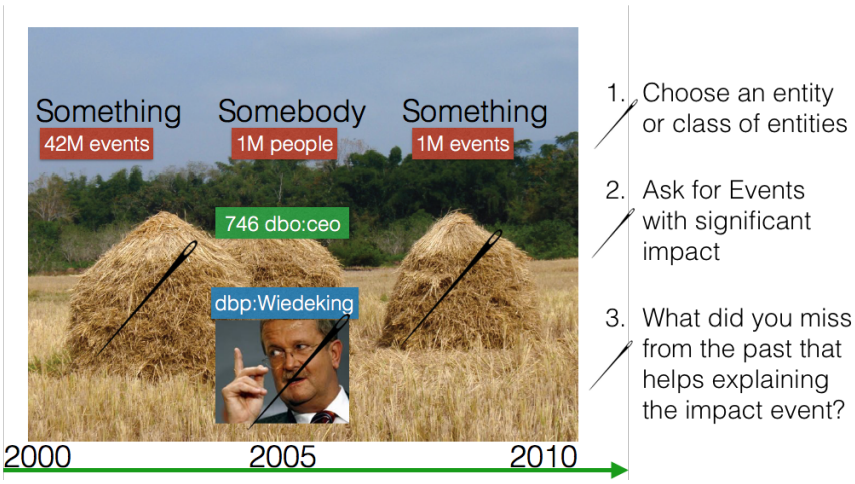


Figure 20: Triple hay stack method to find hidden events with impact

There are various 'classical' retrieval and alert solutions to help finding the first two needles. A traditional method is to trust the newspaper editor who decides to put certain news on the front page and other news not. A more modern method is to make a profile of your interest, e.g. *CEOs*, *Wiedeking*, *legal events*. Whenever there is impact news that matches your profile, you will get an alert.

The impact of an event can then be measured by considering the volume of news, microblogs or queries (compare Google trend) or the strength of the sentiment in social media.[3] An example of a news tracking solution is the European Media Monitor. Figure

[3]We can also check structured data such as stock exchange values or financial business news to see

21 shows a screen dump of their Newsbrief that tracks trending topics over time, based on volume and clustering. The more trending they are, the more they will stand out.

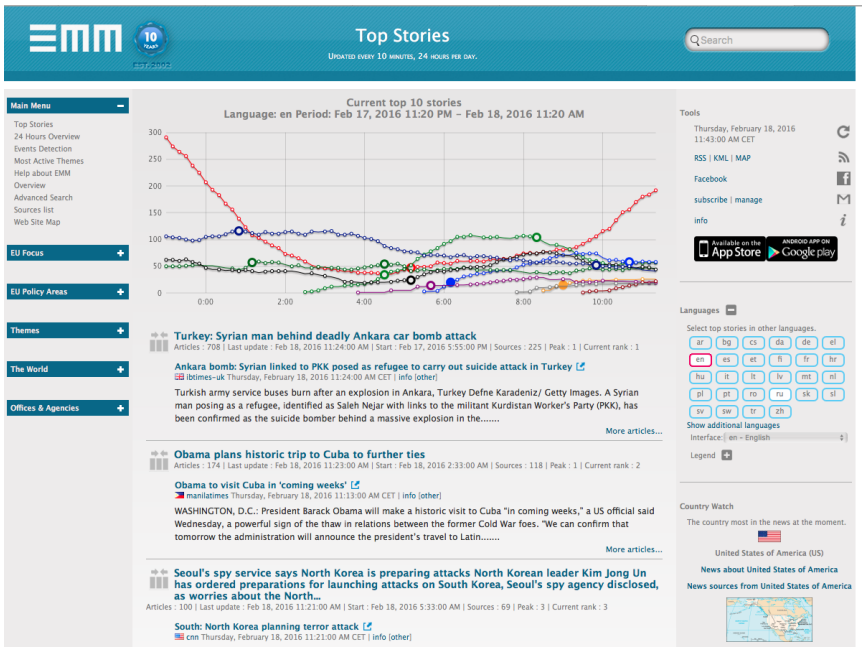


Figure 21: European Media Monitor Newsbrief tracking trending topics in the news. Every coloured line is a separate topic for which the volume of news within a topic cluster is measured by hour. Topics go to sleep around midnight and tend to wake up in the morning.

These classical solutions can only work if the news is somehow spread and there is some measurable activity as a response (e.g. tweets). Typically, the topics in Newsbrief *go to sleep* with the people whose activity is measured (i.e. news providers) and topics *wake up* again with these people in the morning. This shows that these solutions measure the impact of the news on the wider crowd and not the impact of the event on the people directly involved. As such they measure the talk about the event and not the change implied by the event. NewsReader's ESO can also deal with the latter. In Figure 22, we show how pre- and post-situations of many events can be interpreted as positive and negative changes with respect to the condition of the participant of the event. In this way, we model for example that companies or markets get better or go down over time.

In Figure 23, we show that we can use the ESO model to trace the volume of events reported with negative and positive impact on Volkswagen and Porsche over time. A concentration of such events in time may point to a *needle*. This can then be used to find who else was involved (a *second needle*). Likewise, our software can trace all negative-impact events involving CEOs to find the fact that *Wiedeking* is being *sued*.

If we assume that the first and second needle are detected, NewsReader can reconstruct

alarming changes that are not expected. Once observed, we can start digging into news for explanations

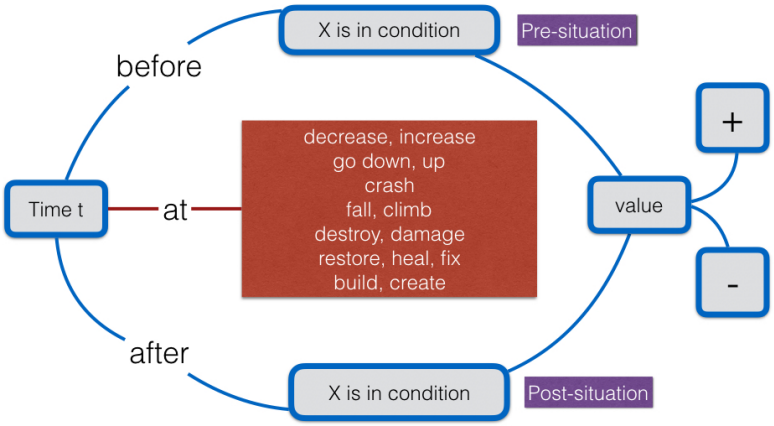


Figure 22: Positive and Negative impact of events modeled in ESO

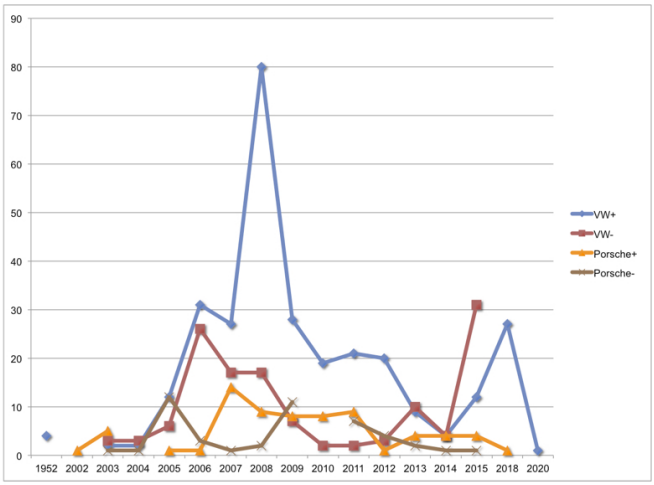


Figure 23: Positive and Negative impact events of VW and Porsche measured through in ESO

the past for the involved entity or entities leading up to the impact event, revealing what happened before (a *third needle*). For this we defined a computational storyline model, in which the impact event is considered the climax in the story and other events are connected to this event through so-called bridging relations. Bridging is achieved when events on a timeline share participants (e.g. all involve *Wiedeking*) and have some causal or coherence relation. Our model allows us to use thresholds and types of bridging relations to create different stories: short, long, tightly or loosly connected, big or small. In the next subsection, we show a visualisation of these storylines that helps finding 'hidden events' connected to the high-impact climax event. Note that NewsReader can also create these storylines on top of classical solutions of detecting relevant impact events in the news such as Newsbrief.

Concluding, we defined NewsReader's intelligence in terms of the semantic capabilities as follows:

- NewsReader can find *associations* between events and entities as needles in haystacks.
- NewsReader cannot always find *correlations*, but can support data scientists in finding them
- NewsReader cannot decide on *legal liability* nor provide any *scientific proof*
- NewsReader does not know what is *true or false* but can show what sources claim
- NewsReader is good at helping to *find a story* that may reveal so-called 'hidden events'
- NewsReader can *tell many different stories* from the data that is extracted

1.3.9 Data visualisations and interactions

Synerscope, member of the NewsReader consortium, is a start-up company that is specialised in visual tools for interacting with large and complex data sets. Their key asset is a series of views on data that are fully connected. Figure 24 shows an overview of different views supported. Any selection or filtering in a single view is projected on all the other views. This makes it possible to simultaneously analyse complex data represented in separated visualisations, each specialised in analysing a different aspect.

Synerscope adapted their tool to the event structures of NewsReader. Figures 25 and 26 show screen dumps of the tool showing social network graphs based on actors in thousands of events, event data plotted on maps and timelines and event data as word clouds. The Synerscope tool was used in a series of end-user-evaluations by professionals. The results of this are reported in user-studies but also raised a lot of interest from the participants for the use of NewsReader.

Whereas the primary data unit of NewsReader is the event, the ultimate goal is to derive narrative sequences of events that exhibit some storylines. Stories are explanatory structures that help us to understand the changes in the world. Whereas individual news

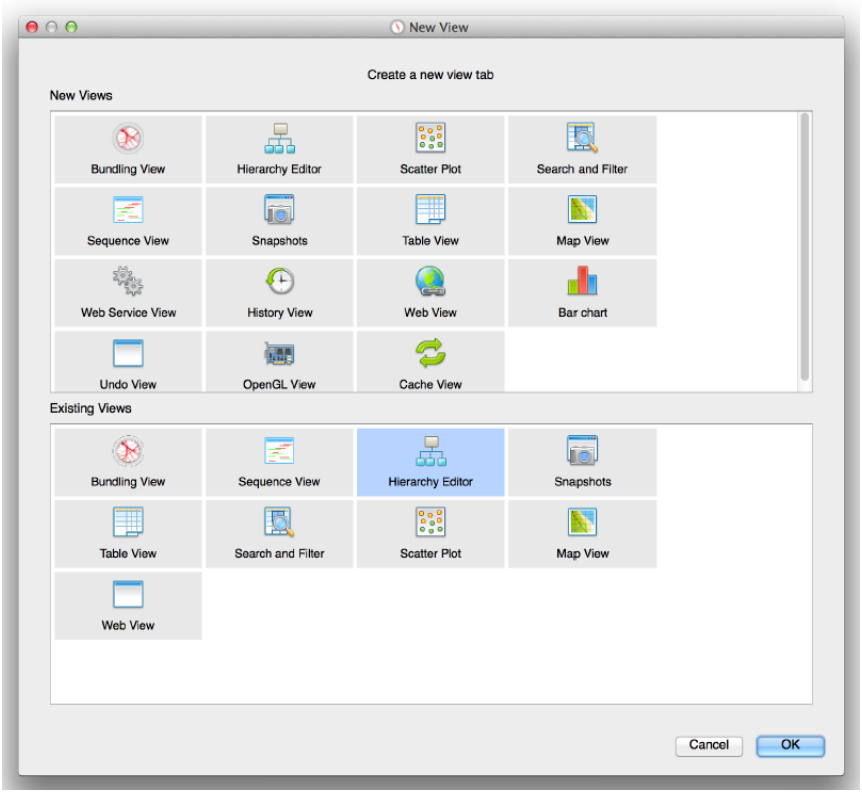


Figure 24: Multiple data views in Synerscope

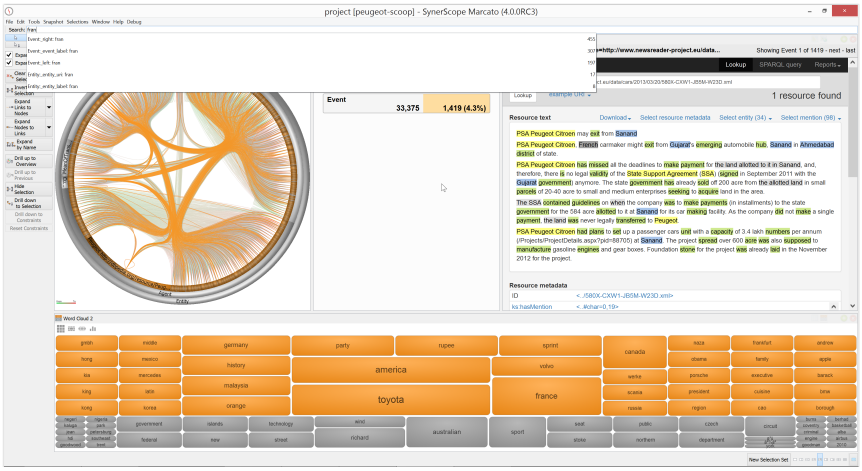


Figure 25: Social network of actors in the automative industry visualised in Synerscope

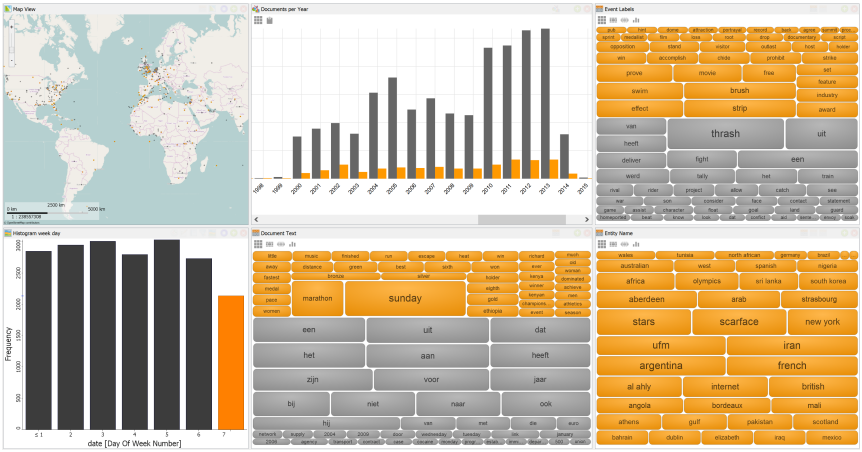


Figure 26: Other views in Synerscope

articles tell only part of the story, the NewsReader database may contain many stories that have not yet been discovered. We thus developed the Storyteller tool to visualise sequences of events around storylines, where we assume that a story contains at least one climax event with impact, preceded by events that lead up to it and followed by events that are the consequence.[4] In Figure 27, we show a screen dump of the Storyteller. It loads ECKGs generated by NewsReader and structures sequences of events to approximate stories using actor and topic relations. The upper part of Figure 27 is actor-centric. It lists all the actors or entities that participate in events and for all each events. Each line represents the events in which the actor is involved and shared events result in intersecting lines. The most 'intersecting' actor is at the bottom of the figure. The middle part is event-centric: each row represents a group of events that approximate a story. The largest circle is the climax event and others are grouped in the same row as preceding and following events based on shared actors and topics. The bottom part allows you to select parts of the data, based on a period or the degree of climax.

1.3.10 Hackathons and end-user-evaluations

The project organised four hackathons in which many different types of users participated, ranging from individuals, to start-ups, to big firms and organisations. ScraperWiki developed the SimpleAPI using the REST protocol and JSON formats to lower the threshold for developers to use the technology.[5] Participating groups could either directly query the KnowledgeStore or use the SimpleAPI to query the data. Most participants managed to understand the basics of the project, design an application and build a mock-up system within less than a day. We received many very enthusiastic responses from the participants.

[4]The tool can be tested online: http://www.newsreader-project.eu/results/demos/json-timeline-structure/

[5]www.newsreader-project.eu/tag/simple-api

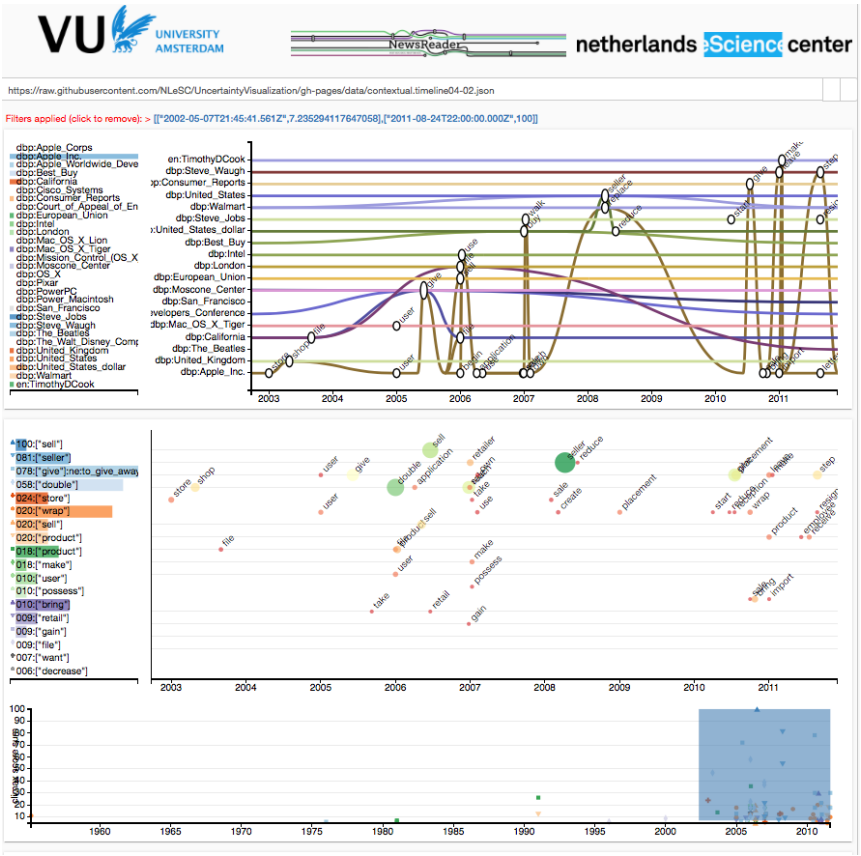


Figure 27: Storyline visualisation of event sequences with actor-centric and event-centric connections.

All the project results are available as open source, with source code and resources downloadable from Github. Results are also available as Virtual Machines, either binary or installable from source code through scripts.

1.4 Consortium details and contact

Project website: www.newsreader-project.eu

Project github: https://github.com/newsreader

Project coordinator: Prof. Dr. Piek Vossen, piek.vossen@vu.nl

Partner	Country	Contact	Email
Faculteit of Arts, Vrije University Amsterdam	Netherlands	Piek Vossen	piek.vossen@vu.nl
Euskal Herriko Unibertsitatea San Sebastian	Spain	German Rigau	german.rigau@ehu.es
Fondazione Bruno Kessler Trento	Italy	Luciano Serafini	serafini@fbk.eu
LexisNexis, Amsterdam	Netherlands	Pim Stouten	pim.stouten@lexisnexis.com
ScraperWiki, London	United Kingdom	Aidan Mcguire	aidan@scraperwiki.com
SynerScope, Helvoirt	Netherlands	Thomas Ploeger	thomas.ploeger@synerscope.com

Table 2: Consortium members and contacts