Semantic Web 0 (2099) 1
IOS Press

Building and Mining Knowledge Graphs for Newsroom Systems

Arne Berven [a], Ole A. Christensen [b], Sindre Moldeklev [b], Andreas L. Opdahl [b,*], and Kjetil J. Villanger [b]
[a] *Wolftech Broadcast Solutions, Nøstegaten 72, N-5011 Bergen, Norway*
E-mail: ab@wolftech.com
[b] *Dept. Information Sci. and Media Studies, University of Bergen, P.O.Box 7802, N-5020 Bergen, Norway*
E-mails: ole@infodoc.no, sndrem@gmail.com, Andreas.Opdahl@uib.no, kjetiljv@gmail.com

Abstract. Journalism is challenged by digitalisation and social media, resulting in lower subscription numbers and reduced advertising income. Information and communication techniques (ICT) offer new opportunities. The paper explores how social, open, and other data sources can be leveraged for journalistic purposes through a combination of knowledge graphs, natural-language processing (NLP), and machine learning (ML). Our focus is on how these and other heterogeneous data sources and techniques can be combined into a flexible architecture that can evolve and grow to support the needs of journalism in the future. The paper presents the state of our architecture and its instantiation as a prototype we have called News Hunter. Plans and possibilities for future work are also outlined.

Keywords: Computational journalism, News platforms, Newsroom systems, Knowledge graphs, Semantic technologies, RDF, OWL, Ontology, Natural-language processing, Machine learning

1. Introduction

Journalism is in crisis, but information and communication technologies (ICT) offer new opportunities [1, 2]. Journalists today have access to a wealth of digital information from news aggregators, social media, and open data providers in addition to traditional sources [3, 4]. Today, digital information can be automatically analysed, organised, prepared, and stored with increasing semantic precision and easily linked to related information [5–7]. Theories and techniques from artificial intelligence and machine learning can be leveraged to classify, label, cluster, detect events, and otherwise process streams of potentially news-relevant information in new and meaningful ways [8, 9].

Our university research group is therefore collaborating with a software developer of news production tools for the international market. Together we are developing News Hunter, an architecture and a series of proof-of-concept prototypes that *harvest* potentially news-related information items and social media messages from the net; *analyse and represent* them semantically in a knowledge graph; *classify, cluster, and label* them; *enrich* them with background information; and *present* them in real time to journalists who are working on related reports or as tips about new events.

Our collaboration has an overarching research goal as well as an industrial goal. Our research goal is to understand *whether and how information and communication techniques (ICTs) such as knowledge graphs, natural-language processing, and machine learning can be combined to make social, open, and other data sources more readily available for journalistic work.* Our industrial goal is *to develop and evaluate proof-of-concept prototypes of such as system.* Of course, the two goals mutually reinforce one another: work on the research goal supplies theories and ideas for development, whereas work on the industrial goal returns working prototypes and evaluations of the research.

A central research question has been: *can heterogeneous data sources and techniques be combined into a flexible architecture that supports the needs of modern*

*Corresponding author. E-mail: Andreas.Opdahl@uib.no.

journalism? The rest of the paper builds our current answer to this question. We first outline our research approach, before we explain the News Hunter architecture along with its prototype instantiation and components. We then review how we have evaluated them. Finally, we compare our results to previous work and proceed to discuss future plans and possibilities.

2. Background

Our work builds on knowledge graphs and other semantic technologies [10–12] and linked open data (LOD) [13] along with techniques from natural-language processing (NLP) [14, 15], machine learning [16, 17], and artificial intelligence in general [18]. A central idea behind semantic technologies and linked open data is to represent information as *knowledge graphs*, expressed using the Resource Description Framework (RDF) and standard IRIs for concepts, relations, and concrete objects [10].

Targetting news workers such as journalists, the NEWS project [19] uses semantic technologies and NLP to annotate news items precisely, using a domain-specific ontology [19–22]. Their IdentityRank algorithm is inspired by Google's PageRank algorithm and supports named-entity disambiguation and linking. NEWS offers a web API for newsrooms and news agencies to access selected services. The NASS 1 system [23] was developed in collaboration with Spanish newspapers in order to automatically classify news documents faster and more reliably than humans.

Targetting media workers of all kinds as well as the general public, the British Broadcasting Corporation (BBC) uses semantic technologies to interlink and interoperate their programme, music, and other resources across departments, systems, and websites. They use named-entity recognition, disambiguation, and linking to enrich media resources with metadata from DBpedia [24], a semantic version Wikipedia. BBC makes the ontologies they use [25] available on their linked-data platform[1]. The International Press Telecommunications Council (IPTC) offers an extensive family of standards for news production (NewsML-G2[2]), media taxonomy (the Media Topics[3]), and semantic mark-up of news (the rNews standard[4] for embedding semantic metadata in online news [26]).

Targetting the general public, news-aggregation platforms try to cope with ever-increasing streams of information by grouping news items by named entities (such as people, places, and organisations), topics, and time. The RELEVANT platform [27] used text similarity to automatically group news from different sources by topic, making them accessible through a web-feed reader. Google News[5] aggregates and attempts to prioritise news items by reports, which are also categorised. Yahoo News[6] offers similar services.

Targetting decision makers as well as the general public, the European Media Monitor (EMM[7]) [28] applies clustering and named-entity recognition on multilingual news streams, analysing and visualising news items from 2500 sources in 42 languages. For each named entity recognised, EMM maintains a resource page with additional information such as recent quotes (if the entity is a person), related keywords, and links to associated entities. webLyzard[8] is a web intelligence platform that harvests open information sources and combines semantic technologies and opinion mining techniques to process the collected data and extract actionable knowledge, such as brand reputations and emerging trends, which are presented in visually rich dashboards. The Organized Crime and Corruption Reporting Project (OCCRP) offers an investigative reporting platform "to give citizens and governments the information and tools they need to bring about a fair system in which criminality and injustice are fought with transparency, knowledge, and empowerment."[9]

Targetting news organisations and the general public, EventRegistry [5] is a news platform [4] that collects items from RSS feeds in many languages, groups them by event (defined as a significant happening that is reported several times), and uses named-entity recognition and wikification to link each group semantically to related information about locations, involved people, and organisations. EventRegistry also categorises news items according to the DMOZ taxonomy[10]. Reuters Tracer [6] "automates end-to-end news production using Twitter data [and] is capable of

[1] http://www.bbc.co.uk/ontologies
[2] https://iptc.org/standards/newsml-g2/
[3] https://iptc.org/standards/media-topics/
[4] http://iptc.org/standards/rnews/
[5] http://news.google.com
[6] http://www.yahoo.com/news
[7] http://emm.newsbrief.eu/
[8] http://weblyzard.com
[9] https://www.occrp.org/en
[10] http://dmoz-odp.org/Science/Biology/Taxonomy/

detecting, classifying, annotating, and disseminating news in real time [...] without manual intervention." Bloomberg's extensive NLP platform[11] also makes use of knowledge graphs and KG-based analyses for news purposes [7].

Against this backdrop, News Hunter stands out by the following combination of features: It targets journalists and newsrooms specifically. It uses knowledge graphs centrally to integrate, organise, and analyse information for journalistic and newsroom use. It harvests, lifts, and ingesting news items from multiple sources, including both pre-news (such as Twitter) and post-news (such as RSS) sources. And it enriches the knowledge graph with facts taken from the linked open data (LOD) cloud [13]. Although there are already tools that combine several of these features, to our knowledge, News Hunter is the only platform combining them all.

3. Research Approach

3.1. Collaborators

Wolftech Broadcast Solution[12] is a software company that develops integrated news systems for making live news production simpler and more efficient. Wolftech Production is a system that focusses on the technical production side, whereas Wolftech News is a newer system for effective news production for TV and eventually for newspapers.

Wolftech News aims to help journalists and other news workers to collaborate effectively and efficiently on creating, managing, and publishing media to a variety of publishing platforms. It supports and improves the workflows in a newsroom through mobile solutions for fieldwork that are integrated with central systems for news monitoring, resource management, news editing, and multi-platform publishing (on live and internet TV, web, social media, etc.).

The group for Semantic and Social Information Systems (SSIS)[13] at the University of Bergen (UiB) studies information systems in the interaction between semantic technologies and social media. The group sees the two as complementary [29], because social media tend to generate big datasets that can be enriched and made more useful with semantic technologies such as knowledge graphs and because social media is an important source of large-scale semantic annotations.

News Hunter [30, 31] is a collaboration between SSIS and Wolftech that aims to extend Wolftech News in order to better harvest, organise and leverage social media streams and other big-data sources for journalistic purposes, using techniques such as knowledge graphs, natural-language processing, and machine learning.

3.2. Research questions

The introduction has already presented our research and industrial goals and central research question for our investigation: *can heterogeneous data sources and techniques be combined into an architecture that supports the needs of modern journalism?*

3.3. Research method

Because our goals and research questions are explorative and involve technology development, we have used design science as our overall research method [32, 33]. The gist of design science research is to advance theory and improve practice by incrementally developing and evaluating artefacts. The two central artefacts in our research have so far been an *architecture* (a high-level structure of system components) along with a series of *instantiations* (a situated implementation in a specific environment) of that architecture in the form of prototypes [34].

In addition, design science research should be informed by theory from relevant fields and rigorously contribute to that theory. We have achieved this by focussing on the less-explored architectural and semantic challenges and opportunities of a platform like News Hunter, leaving research on components to their respective, focussed research fields. Design science research should also be relevant to practice in relevant fields and contribute to improving that practice. We have achieved this by relying on Wolftech's understanding of media organisations, newsrooms, and journalists developed through extensive and systematic discussions, experiences, and interviews with end users and their organisations.

3.4. Development method

We have followed an iterative development strategy with synchronised research and development iterations in continuous dialogue with Wolftech, in or-

[11] https://www.techatbloomberg.com/nlp/
[12] https://www.wolftech.no/
[13] https://www.uib.no/en/rg/ssis

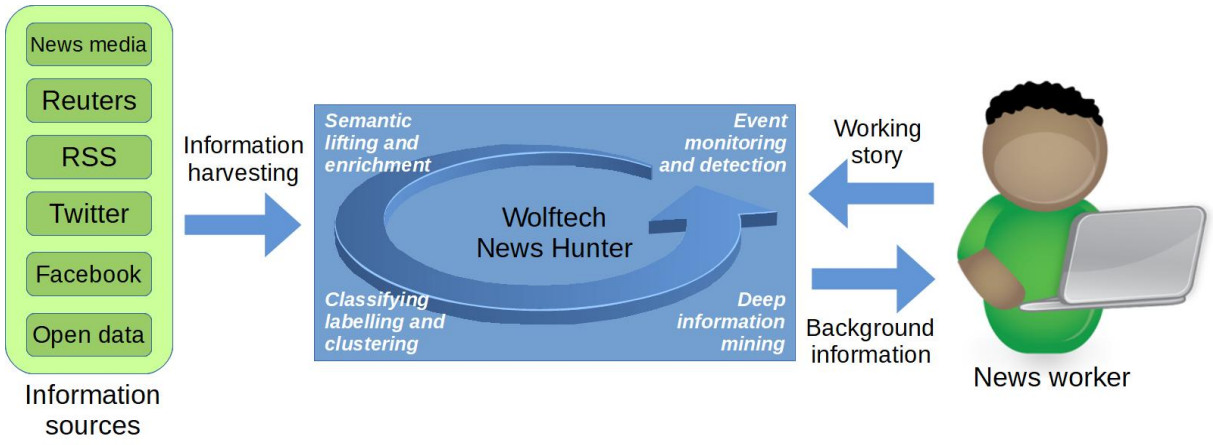

Fig. 1. News Hunter harvests information from social, open, and other sources and presents it as relevant background information to working journalists [31].

der to get feedback on results and ideas for new features. For each iteration, we attempted to define clear goals for both research and development, based on a clear development process that involved selected tools and technologies and where each step produced well-defined and validatable results. We sought short, XP-like [35] iterations, typically of 1-3 weeks' length, inspired by the minimum viable product (MVP) idea: build the minimum number of features that is required for the system to work as intended, and then evolve from there. As far as possible, we sought to align with international technology standards that Wolftech already used in their development and runtime environment, including: C#, Visual Studio 15 IDE (Integrated Development Environment), BrightstarDB (a graph database/triple store for the .NET platform), and Froala (a WYSIWYG web editor written in JavaScript). To this, we added Python, PyCharm IDE, Flask microservices, Visual Studio Code IDE, and Standard.js for back-end development; AngularJS, HTML and CSS, Sketch, and Marvel for the front end; and GitHub/Bitbucket and Trello for managing the project. While one central part of the application remains dependent on ASP.NET, more and more components are written in Python. They are tied to one another and with the ASP.NET component through the Flask microservices.

3.5. Evaluation methods

We have relied on two types of evaluation: most centrally, we have used (1) *proof-of-concept evaluations* after each major development iteration (typically every 1-3 weeks) to ensure that our prototype components worked both in isolation and together along with (2) *component evaluations*, to gauge the quality of the most central components in our architecture, such as the natural-language analysis and user-interface components, using established research methods for each type of component, both with and without human participants. A full end-user evaluation was not possible because the first of our two central artefacts, the News Hunter architecture, is abstract and the second artefact, the proof-concept prototype, would require extensive functionality completion, user-interface polishing, and data population before user testing became worthwhile.

4. Design Goals

In collaboration between SSIS and Wolftech, we established the following design goals for News Hunter, to which we will return in the discussion:

- *State-of-the-art:* News Hunter should leverage state-of-the-art techniques for semantic analysis of natural-language texts and for managing, enriching, and reasoning over knowledge graphs.
- *Embedded in newsroom environment:* News Hunter should operate as an integral part of the journalists usual work environment.
- *Live harvesting*: News Hunter should continually harvest potentially news-related information items from the net.
- *Real-time analysis*: News Hunter should analyse the reports that journalists are working on in real time to provide relevant background information.

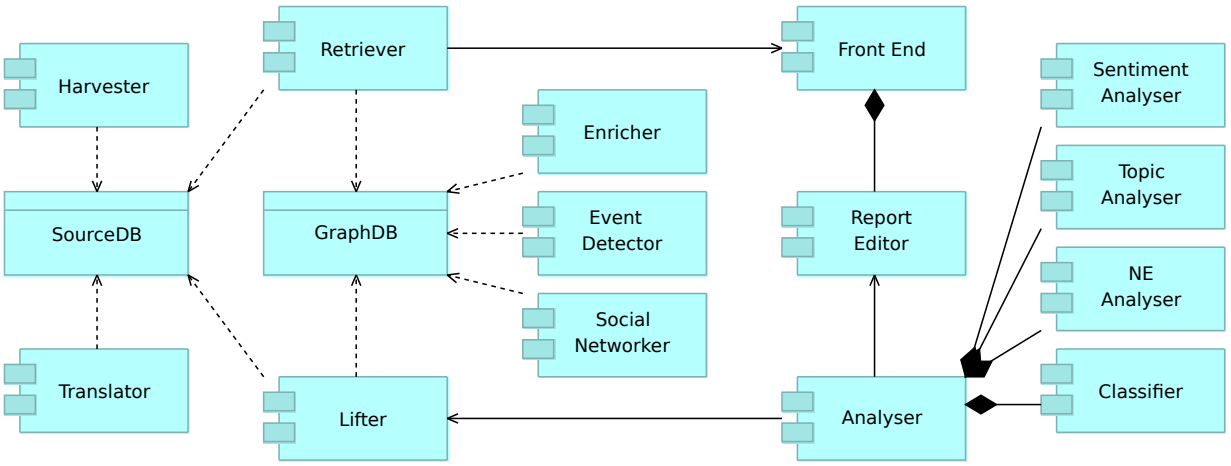

Fig. 2. The News Hunter architecture with access and serving relations between its components.

- *Push and pull:* News Hunter should support pushing information to journalists proactively as well as supporting on-demand information (pull) requests from users.
- *Multi-language input:* News Hunter should be able to harvest and lift information in different languages.
- *Language neutrality:* News Hunter should attempt to store information independently of language as much as possible, with no language being treated as primary by design.
- *Language-agnostic analysis:* News Hunter should support data analysis independently of the original language of the harvested information items.

5. Architecture

Figure 1 depicts News Hunter in its usage context, whereas Figure 2 illustrates the internal News Hunter architecture[14] with access and service relations between its application components:

- The *Harvesters* continuously download potentially news-related information items (articles, messages, posts, tweet...) from relevant sources such as Facebook and Twitter, RSS feeds, and news sites.
- The *Source DB* stores the harvested information items as JSON objects close to their original form.
- When necessary, the *Translator* creates a canonical-language translation of each text item in the Source DB (we have used English as canonical language).
- The *Lifter* runs the harvested (and translated) text items through a (natural-language) NLP pipeline to represent them semantically as small knowledge graphs that are inserted into a graph database/triple store.
- The *Analyser* implements the NLP pipeline and invokes sub-components for specific tasks such as *sentiment*, *topic*, and *named-entity (NE) analysis*, as well as *classification*.
- The *Graph DB* stores the resulting semantic representations and combines them into a contiguous journalistic knowledge graph — a *news graph* — and stored in a triple store to support semantic analyses and more precise retrieval.
- The *Enricher* augments the the news graph with background information from social, open, and other data sources on the web, such as DBpedia [24], to the news graph.
- The *Event Detector* clusters recent items by named entities, topics, and location to identify potentially newsworthy events.
- The *Social Networker* conducts affinity analysis to identify whether people in the news graph are on friendly terms or not, a useful feature for journalists when selecting informants and planning interviews.
- The *Report Editor* lets the journalist type in a report, which is sent to the analyser in real time to be lifted by the same NLP pipeline as the harvested text items.

[14] The figure has been drawn with the Archi tool (https://www.archimatetool.com/) using the ArchiMate notation [36].

- The *Retriever* uses the result of analysing reports-in-writing to retrieve similar reports and other related background information items from the graph DB, using pre-packaged SPARQL queries, and perhaps from the source DB.
- The *Front End* embeds the report editor in a working environment, collects the result of analysing the report-in-writing, and invokes the retriever to collect similar reports and other relevant background information to present to the journalist.

In addition to the application components shown in Figure 2, there is an *Ontology* that defines how information items are represented semantically in the knowledge graph, and there is a *Microservice Framework* that uses lightweight REST APIs to tie the components in the architecture together. The News Hunter prototype reported in this paper implements some central functionality of all these components, although a few of them only in rudimentary form. The next two sections will discuss the more evolved components and their evaluation in further detail.

6. Prototype

Work on the News Hunter prototypes started in the summer of 2015 [30]. The first version harvested posts from Facebook's public API and ran non-English posts through Google's Translate API. The English texts were then analysed using IBM's online Alchemy API (which is today part of the BlueMix platform) to extract metadata about topics, named entities, and sentiments. The Alchemy metadata, along with the message text itself and additional metadata from Facebook's API (date, title, location, etc.), were then loaded into a graph database/triple store, structured according to an early version of the News Hunter ontology (see Figure 3). The prototype was written in C# as an ASP.NET application with a BrightstarDB database for semantic storage of the graph.

The rest of this section will present the most central components of the second News Hunter prototype and their functionality. The following sections will review how we have evaluated the most central components and discuss future possibilities and plans.

6.1. Harvesters

Going beyond the Facebook API, the next prototype harvests posts from Twitter and RSS feeds. We have also collected articles from major English-, Norwegian- and Spanish-language news outlets. The harvesters are Python scripts that use the Tweepy library to harvest Tweets, the Feedparser library for RSS, and the Newspaper library to harvest news reports. The pre-processed messages are stored as JSON objects in a source database.

6.2. Source DB

Another Python script inserts the harvested text items along with selected metadata into an Elasticsearch database to make it available for later retrieval and further analysis. This script also takes care of duplicate data, which is important when running multiple harvesters simultaneously.

6.3. Translator

For language translation, the prototype uses Microsoft's online Translate API, because the standard libraries we have explored do not support small languages like Norwegian.

6.4. Analysers

To lift the harvested messages semantically, we run an analysis pipeline written in C# and Python.

Topic extraction. For unsupervised topic (or keyword) extraction, we are exploring several tools. For short messages like microtexts, we use the RAKE (Rapid Automatic Keyword Extraction) library, written in C#. For RSS items, we use Textacy, a wrapper library for Spacy (see below). For longer texts, we use the Python-implementation of the TextRank library, which supports automatic keyword extraction in addition to report summarisation. To offset the bias of longer texts generating more keywords, we weigh each keyword by its position in the text (assuming that news reports describe its most central aspects early in the text) and number of occurrences, so that frequent and late-occurring keywords weigh less.

Named-entity recognition. For part-of-speech (PoS) tagging and named-entity recognition (NER) we use the Python-library Spacy. For named-entity disambiguation, we use the DBpedia Spotlight tool [37], written in Scala. It tags named entities recognised in the text with DBpedia IRIs in order to provide more precise semantics and facilitate data enrichment with linked open data.

Sentiment analysis. The prototype uses the AFINN Python library for sentiment analysis.

6.5. Graph DB

We inject the lifted text items in a knowledge graph managed persistently in the BrighstarDB triple store, a DBMS specifically for RDF graphs [10]. We also use the Microsoft Entity Framework along with LINQ (Language Integrated Query), which is a .NET component for C#-native data querying of domain-specific models. It makes it easier to write datatype-specific queries that are automatically translated to SPARQL and processed by BrightstarDB.

6.6. Ontology

To accommodate the new data sources and additional analysis tools, we have revised the News Hunter ontology, which from the start was closely tied to Facebook and the Alchemy API. We have introduced better linked-open data practices, such as reusing and interlinking common terms from existing ontologies. Figure 3 illustrates the News Hunter ontology.[15]

Items are potentially news-relevant items, for example news articles, reports-in-writing, RSS feeds, blogs posts, and microtexts such as Facebook messages and tweets. *Descriptors* represent the semantic contents of items; for example named entities, topics, locations, and sentiments mentioned or expressed in the text. Topics and named entities, such as people, organisations, and places, are enrichable entities with a unique IRI, about which additional information can be retrieved from linked-open data sources such as DBpedia and added to the news graph.

The core ontology in Figure 3 can be extended with terms from other common ontologies, for example to represent the source IRIs and creation dates of items, social relations between persons, and semantic relations between topics.

6.7. Microservice framework

To make the architecture less coupled and more flexible, and because the different APIs we use are written in different languages, we tie them together with REST endpoints using the Flask microservice framework. Although Flask is written in and for Python, it is also usable from C#.

6.8. Classifiers

To organise the harvested and lifted messages further, we use several NL and ML analysis pipelines.

Single-label classification. To provide additional entry paths into the news graph, we use NL and ML techniques to classify (label) messages [40]. As training and evaluation data [15], we use pre-labelled RSS feeds. Each message is pre-processed and sent to Spacy for part-of-speech (PoS) tagging, stop-word removal, and lemmatisation. The following Scikit-learn pipeline comprises a CountVectorizer for tokenising and creating a term-document matrix. TF-IDF is used for feature selection to assign higher weight to potentially important words in a text. As final steps, we have explored a support-vector machine (SVM) and a multilayer perceptron (MLP) neural network, both wrapped in a Flask API.

Multi-label classification. Multi-label classification relaxes the single-label requirement and has the potential to represent message content even more precisely and findably, in particular when combined with standard news taxonomies such as the IPTC newscodes [26] and its Media Topics[16]. As training and evaluation data, we use a set of pre-labelled articles from The Guardian, available through their public API. We use string equivalence and WordNet synsets [41] to match Guardian labels to IPTC newscodes before inspecting the matches manually. We run the messages through the same pipeline we use for single-label classification, but instead using Scikit-learn with Keras, a Python library for high-level neural networks and deep learning.

6.9. Event detection

Multiple messages about the same event or topic may suggest that a newsworthy event is unfolding. Such events are detected by clustering recent messages in the news graph. The resulting clusters are also used to identify messages with different perspectives on the same or similar events. We select messages from recent days and pre-process them to calculate TF-IDFs. For clustering, we use Scikit-learn's DBSCAN algorithm, which offers scalability and focus on neighbourhood size at the expense of uneven cluster sizes.

[15] The figure has been drawn with the online WebVOWL visualiser [38] (http://www.visualdataweb.de/webvowl) using the LD-VOWL notation [39].

[16] https://iptc.org/standards/media-topics/

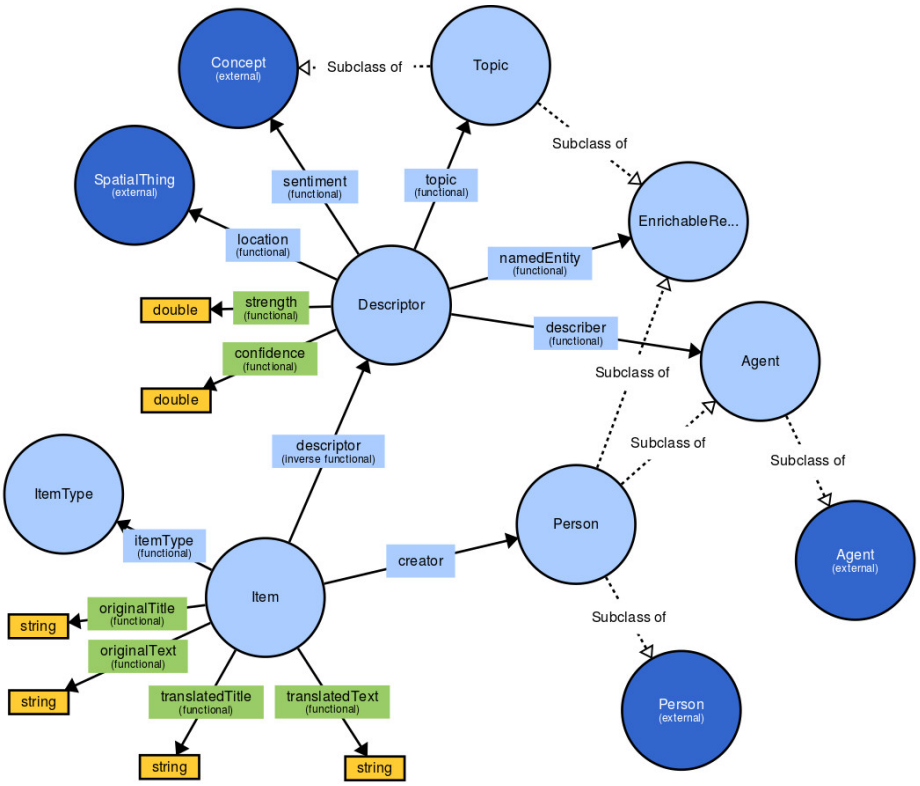

Fig. 3. The revised News Hunter ontology, adapted from [31].

6.10. Social networking

We also use simple *social network analysis* using who-knows-who graphs to help journalists prepare for interviews outside their usual areas, for example to avoid saying something wrong to the subjects. News Hunter also aids invitation of interview objects, giving journalists and other editorial workers a standard form for creating invitations and saving invitations in the news graph so they can be revised and reused in the future.

6.11. Report editor

Froala's WYSIWYG editor plugin is used to write up news reports. Whenever the journalist pauses writing, the text is sent asynchronously to the same pipelines that are used to analyse harvested messages semantically. This has several benefits. The journalist gets instant feedback on the sentiment of their writing and under which category they should save and publish their reports. Other journalists in the same organisation that are working on reports about the same topics and named entities can potentially be identified to foster collaboration and avoid duplicate work. And the news graph can be queried for relevant background information to present to the journalist. For the latter purpose, we explore simple algorithms based on semantic distance, so that the relevance of a harvested message is proportional to the number of related topics and named entities, weighted by the strength of each relation. Exact word-sense match is the strongest, followed by synonyms, hyper-/hyponyms, and then other semantic relations.

6.12. Front end

The report editor is part of a web front-end that shows analysis results and background information. The user can click on identified topics or named entities to show related messages, possibly as summaries. The front end also shows the most recent message cluster detected in the news graph. The News Hunter prototype thus provides a full application stack, from a web-based GUI, through a microservice architecture, down to persistent data storage.

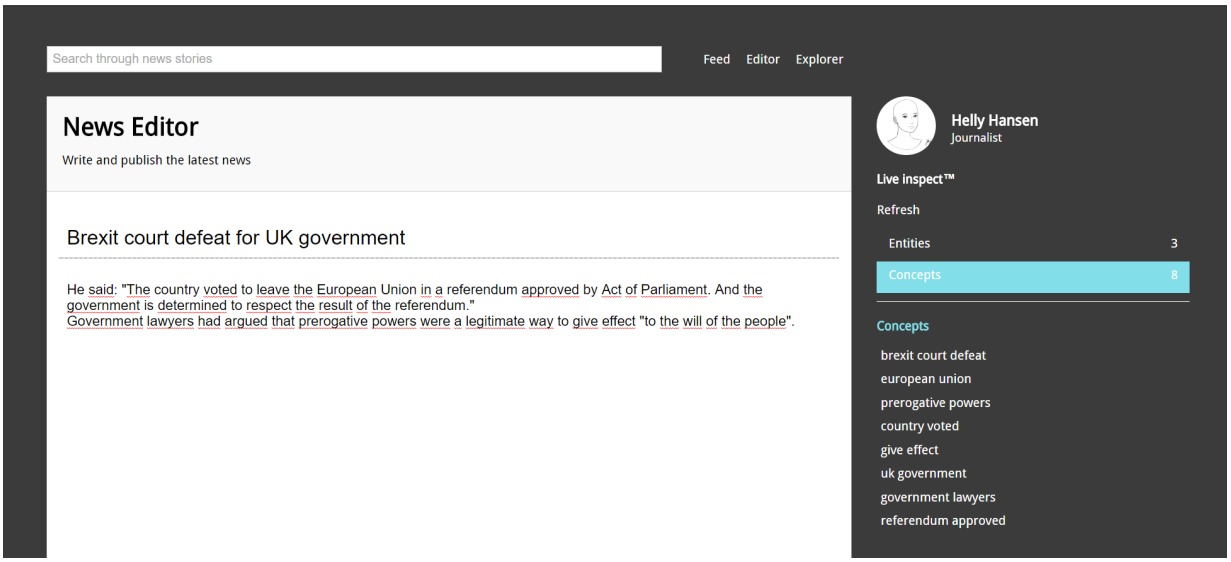

Fig. 4. The web front-end of the prototype reported in the paper. The left side shows an early version of the report editor. The right part shows the results of analysing the report-in-writing.

6.13. Enrichers

The font-end can also retrieve additional background information on demand from social, open, and other data sources on the web. Currently, the IRIs for named entities provided by DBpedia Spotlight are used to enrich the news graph with background information on demand from the live DBpedia endpoint using pre-built SPARQL queries. The names and Twitter handles of persons (in the prototype restricted to sports athletes) are used to retrieve recent tweets on demand from the official Twitter API.

7. Evaluation

This section reviews our proof-of-concepts evaluations, our component evaluations with human participants, and our evaluations of central algorithms.

7.1. Proof-of-concept evaluations

We ran functional tests at the end of each iteration to verify that the realised or revised components were indeed able to perform their expected tasks and that they were sufficiently integrated with other components. For example: when implementing harvesters, we verified that they indeed stored messages in JSON files; when implementing the message store, we verified that the JSON files were properly inserted into the store; when implementing the lifters, we verified that the JSON files were correctly translated and well represented in the news graph; and so on for all the components and features in the system. In this way, we ensured that the components in our evolving architecture were able to interact appropriately.

7.2. Evaluations with human participants

Developing or improving new specific theories or technologies for components is not our central research goal. But to ensure that they perform at least acceptably, we have conducted limited evaluations of the central prototype components, relying on established research methods for each type of component.

Front end. We let two journalists and four domain experts use the prototype after a brief introduction and demonstration. They were instructed to view different news reports and thus explore the following features of News Hunter: inspecting existing news reports through the front end; displaying keywords for reports-in-writing; generating summaries of longer messages; retrieving clusters of recent messages (“top stories”); retrieving related reports to the report-in-writing. We then presented them with questions about the usability and usefulness of each feature. In general, the respondents were positive to all the features, but pointed to many areas of improvement for further work.

Inspecting existing news reports: All participants found this feature useful for getting an overview of ex-

isting news reports. They suggested several possible improvements: an option to skip the inspect menu to go straight to the text of the report; a flag indicating when a report has been updated; access to the report's update history; more prominent display of report texts compared to entities and keywords; and displaying the most relevant named entities as a word-cloud.

Named entities: Most participants found the named entities associated with reports useful for retrieving background information without having to leave the tool to search the web. The named entities also provided helpful context for understanding a report. The respondents noted that this feature could benefit from more filtering options, from triangulating information from multiple sources, and from using verifiable sources. Also, not all the named entities present in the reports were identified. Filtering could be done by entity relevance, by type (people, organisations, places...), and by map. The term "Named Entities" in the user interface was confusing. Instead, the names of the main types of entities ("People", "Organisations", "Places"...) should be displayed.

Keywords: Most participants found the automatic report keywords useful for finding related content and for saving as metadata. Some of them were also seen as a more efficient way of figuring out what a report is about. Both journalists found the feature to be useful, but the domain experts were divided. A possible reason is that the quality of the keywords still needed improvement, for example through sense disambiguation. It should also be possible to remove bad keyword suggestions. In addition to keywords used in the text, higher-level topics could also be suggested.

Report summaries: Two of the domain experts said the summaries were an effective way of gaining a quick overview of a report. One of the journalists added that the summaries could be published already while the main report was being written. However, the quality of the summaries generated by the prototype needed improvement, and several of the respondents were therefore negative to the presented version of this feature.

Top stories: The top-story feature shows clusters of recent messages that have been detected as reporting the same event. The respondents found this useful for identifying multiple reports about the same event and for identifying unfolding events that need to be covered. However, only named entities and keywords that apply to all the reports in a group should be displayed, and the numbers of named entities and keywords should be restricted. Respondents also wanted categories and a slider to limit the time frame. Ability to search through groups, control which sources to include, and explain group membership criteria were also called for. Two participants saw it as problematic to make the top-story feature too prominent, because it might drown smaller but nevertheless important events.

Related reports: All the respondents appreciated the possibility to retrieve related previous news articles. One of them pointed out that the usefulness of this feature would be increased by better sense disambiguation. Some respondents would like more metadata about the previous reports, such as their date and source. Being able to see if other journalists were working on the same report was also suggested as a useful feature.

Report editor: The respondents found the automatic identification of named entities and keywords in the text useful, because they could be used both as metadata and to retrieve relevant background information. In-editor access to related previous news reports was also considered beneficial. The respondents missed the ability to remove unwanted named entities and keywords. Also, the prototype editor was considered too simple and should be extended with more features.

Topic extraction. We let the participants read through a convenient sample of three medium-length news reports: the Reuters article "Trump gives nod to Republican tax-credit proposal on Obamacare"; the BBC article "UKIP burka ban policy 'misguided' says party's MEP"; and the CNN article "Paul Pogba: Is he worth $120 million?". We then asked them to suggest suitable keywords for each article before we let them assess two keyword lists generated automatically by TextRank and RAKE. The participants tended to prefer the keywords generated by TextRank over those from RAKE, which tended to suggest many multi-word key phrases, whereas TextRank tended to propose single keywords that were more comparable to those suggested by the participants.

Considering only keywords suggested by multiple respondents and counting $> 50\%$ overlap between multi-word phrases as a match, keyword extraction with TextRank had an F1 score from 0.43 to 0.47 for the three articles, with higher recall (R=0.57...0.75) than precision (P=0.3...0.4). This suggests that standard keyword extraction methods must be improved or at least better tuned before they can support journalistic work. We did not calculate precision and recall of keyword extraction with RANK, because its

many multi-word key phrases were too dissimilar to the mostly single-word phrases suggested by the respondents.

Named-entity recognition. From the same sample of reports, we let the prototype extract named entities along with their types, and selected all the entities with background information available in DBpedia. We then present the named entities to the respondents along with the corresponding report. For the Reuters article "Trump gives nod to Republican tax-credit proposal on Obamacare", 8 out of 11 entities were considered fine, but named entities such as the "White House" and "health insurance" were criticised. For the BBC article "UKIP burka ban policy 'misguided' says party's MEP", 12 out of 15 identified entities were ok, but "Commonwealth" and "EU" were types as countries and the wrong election was linked. For the CNN article "Paul Pogba: Is he worth $120 million?", 24 out of 27 were fine. In summary, the respondents found between 73% and 89% of the entities appropriate for each report, having smaller or larger issues with 11% to 27% of them. Some of them commented that the identified entity types should have been more specific, for example listing the type "football player" instead of "person".

Labelling. The respondents agreed that the (single-label) categories suggested for the three reports were correct, but too general. More precise and descriptive categories would be needed to make the platform useful in practice.

7.3. Evaluations of algorithms

Translation. To evaluate automatic translation, we conveniently sampled two international news events: the release of a new iPhone and an international football match. For each event, we chose news reports in three languages (English, Norwegian, and Spanish) of comparable lengths and scopes. We translated the (Norwegian, and Spanish) reports into English using Microsoft's Translate API. We then extracted keywords and named entities from the English versions. The results showed considerable overlap, but suggested that exact message matching across languages cannot reliably be based on automatic translation and keyword/named-entity extraction alone.

Single labelling. To evaluate single labelling, we used BBC's Insight dataset, which contains 2225 documents from BBC News categorised into: sport, business, entertainment, politics, and technology; to which we added health, science, environment, and crime to capture a wider variety of streams and news content. We evaluated several configurations of two classifiers: a linear-kernel support-vector machine (SVM) implemented in Scikit-learn and a multi-layer perceptron (MLP) built using Keras. Both classifiers reached an F1 score of 0.89. They performed excellently for sports (F1=0.98...0.99) and worst for education (F1=0.68...0.69), most likely because the training set contained a lot more sports than education articles.

Multi labelling. To evaluate multi labelling, we retrieved 544 820 news articles from after January 1st 2010 directly from The Guardian API. The best F1 scores were 0.84 for the Scikit-learn SVM and 0.72 for the Keras MLP.

Event detection. To evaluate event detection, we analysed 1292 articles from a variety of newspapers collected with our own harvester. We compared the resulting clusters with the top stories listed in Google News. Two out of the six identified clusters were deemed correct after manual inspection, and the four remaining ones were also among the top events in Google News.

8. Discussion

8.1. Comparison to related work

To the best of our knowledge, the architecture and instantiation proposed in this paper go beyond the previous work reported in the literature. Whereas many existing tools and services are aimed at the general public or at other professions, such as archivists, News Hunter targets journalists specifically and offers them a front end for news reporting that is embedded in their newsroom environment. Among the tools and platforms that targets journalists and newsrooms, EventRegistry [5] uses similar semantic message analysis, classification, clustering and event detection techniques to ours. However, it is restricted to post-news analysis of RSS feeds and uses non-semantic background information from Wikipedia. Reuters Tracer [6] is restricted to pre-news Twitter messages and is not centrally based on a knowledge graph. Bloomberg's in-house NLP platform have similar goals to ours and may use knowledge graphs [7], but to what extent and how is not reported in the literature. Among the journalistic tools, NEWS [19]also uses semantic message analysis and classification. But it does not focus on event detec-

tion and is accessed through a web API instead of a front end. And like EventRegistry, NEWS is limited to analysing items that are already in the news, whereas News Hunter also aims to analyse pre-news information originating, e.g., from social media. In this sense, News Hunter is more similar to EMM [28] and webLyzard which are, however, general web intelligence platforms that do not target news workers specifically.

In summary, we consider the News Hunter architecture unique because it offers, in combination:

- harvesting information from a variety of sources, including both established media, news aggregators, RSS feeds, and social media;
- harvesting both pre-news information and (post-)news items about unfolding events;
- handling multiple languages;
- analysing the harvested information semantically, combining: named-entity recognition/disambiguation/linking, topic extraction, sentiment analysis, single- and multi-labelling, clustering, and event detection;
- enriching the information with semantic data from the LOD cloud [13]; and
- offering a front end for working journalists — including a report editor that continuously analyses reports-in-writing so that related news items and background information can be provided through the front end in real time.

8.2. Research question

The practical and theoretical contribution of this paper was to propose, instantiate, and partially evaluate an architecture and prototype that combines the features listed above. We think the architecture and prototype tentatively answer the research question we posed in the introduction positively: *it may indeed be possible to combine heterogeneous data sources and techniques into a flexible architecture that supports the needs of modern journalism.* Specifically, *it is likely that information and communication techniques (ICTs) such as knowledge graphs, natural-language processing, and machine learning can be combined to make social, open, and other data sources more readily available for journalistic work.* At the same time, we are fully aware that our architecture is in no way final nor complete. An obvious challenge for further work is how to make our news-graph architecture work in real time on web scale. Existing tools and services — such as EventRegistry, Reuters Tracer, Bloomberg's NLP platform, EMM, webLyzard, NEWS, Google/Yahoo News, and others — all suggest different ways in which our architecture and prototype tool can be extended to provide even richer support for news workers, and the research areas we build on most prominently — knowledge graphs, natural-language processing, and machine learning — are all advancing rapidly.

8.3. Design goals

We turn to revisit the design goals for News Hunter that we presented in Section 4.

State-of-the-art. The previous sections have shown that News Hunter has indeed built on state-of-the-art techniques for semantic analysis of natural-language texts and for managing and enriching knowledge graphs. However, the technology in this area is advancing rapidly. In parallel with our work on the architecture and prototype, neural-network based techniques for NLP and graph analysis have matured, making them increasingly likely paths for further work.

Embedded in newsroom environment. Through the provision of a front-end and integrated report editor, News Hunter is able to operate as an integral part of the journalists usual work environment. Our Froala-based report editor is designed to be compatible with the editor used by Wolftech News. Indeed, Wolftech already offers a reimplementation of basic News Hunter functionality to customers as part of their News system.

Live harvesting. Although News Hunter has been designed with components that can support live harvesting, they have not run in production in parallel with development. To support live harvesting on a large scale, the architecture must be parallellised and performance improved.

Real-time analysis. For the same reason, News Hunter does not support live analysis. But the prototype is able to analyse news reports the journalist is working on in real time and propose related concepts, named entities, related reports, and other information proactively.

Push and pull. News Hunter is also able to retrieve further information about these concepts, named entities, and reports on demand, thus offering both push and pull information.

Multi-language input. News Hunter is integrated with a translator component and thus able to harvest and lift messages in many languages. However, the prototype reported in this paper uses an online transla-

tor that is slow and that imposes strict limits on translated items per time period. Further work is needed to provide either a locally running item translator or a multi-language message lifter. Of these two alternatives, a locally running translator, perhaps leveraging multilingual word embeddings such as Facebook's MUSE[17], has the advantage that it produces a *canonical-language* version of every input item, which can be stored and used for further text analyses. A multi-language lifter can potentially provide more precise semantic representations, although this remains an empirical question.

Language neutrality. The News Hunter architecture has been designed to be language neutral. However, it uses the notion of *canonical language* in which all text items are stored after translation — in addition to the *original-language* version. The purpose is to enable single-language text indexing and searching of all stored items and to offer canonical-language lifting of items expressed in weakly-resourced languages for which lifting services are wanting. In the prototype, the canonical language is English but, in principle, it can be any language for which sufficient item translators are available.

Language-agnostic analysis. News Hunter thus supports cross-language analyses in two ways: most importantly through its news graph, which represents concepts, relations, named entitites, and other phenomena using language-neutral IRIs and language-tagged labels and other strings, but also through its canonical-language items.

In summary, the News Hunter architecture and prototype instantiation reported in this paper satisfy most of our initial design goals. Remaining goals to address in further work include: live harvesting, parallel architecture, built-in translation, and full language independence.

8.4. Further components

In the future, additional News Hunter components will therefore be needed:

- *Filter* components would screen *Harvester* outputs to discard less news-relevant information items, which abound on social media.
- *Aggregators* would make life easier for the *Lifters* by concatenating semi-relevant shorter text items, for example tweets, that are highly likely to report the same event. Aggregation thus gives the lifters fewer and longer texts to work on, using coarser and less reliable aggregation techniques than, for example, event detection.
- *Relation Extractors*, often combined with concept or named-entity extractors, determine which annotations of an item are related and how. Currently, text items are represented semantically as small knowledge graphs with a star structure, leaving out central parts of their meaning. For example, when a text item is annotated with dbpedia:Barack_Obama, dbpedia:Peace_process, and dbpedia:North_Korea, does it mean that North Korea is the originator of, subject of, the location of, or excluded from the process?
- *Feeders* would, in contrast to the retriever, *push* presumably relevant information onto working journalists, either related to or independently of the reports they are currently working on. One example is new information items published by notable people already involved in a report-in-writing. Another example is information that another journalist in the same organisation has started working on a similar report.

Also, several of News Hunter's existing components are currently only rudimentary and will need to be extended: The *Enricher* is limited to harvesting personal data on demand from a single data source (DBpedia). Future versions should harvest more types of data continuously from a broader variety of sources, in order to provide richer data as input to the organisers and other analyses. The *Social Networking* subcomponent is limited to simple affinity analysis. Future versions should offer a broader variety of analysis based on enriched social data. The *Event Detector* should also be strengthened to identify composition, temporal, causal and other relations between events as they unfold and are reported through information items. The *Retriever* currently uses only simple prepackaged SPARQL queries to satisfy recurring information needs from journalists.

9. Conclusion

The News Hunter platform is positioned in a rich research area and emerging software market that of-

[17] Multilingual Unsupervised and Supervised Embeddings (MUSE): https://code.fb.com/ml-applications/under-the-hood-multilingual-embeddings/

fer ample potential for further research, development, and innovation. Much work remains, but we think the results so far are promising. Our most immediate focus is on redesigning the prototype into a big-data ready platform built on top of state-of-art technologies; improving the precision of its semantic-analysis components; and extending it with new types of analysis components. To achieve this, the architecture needs to be parallelised, streamlined, and provided with well-defined microservice-based interfaces. Building an even larger-scale news graph and message base for further testing and research is a priority. Many other research and development challenges remain important, such as: large-scale analysis of message feeds; language neutrality; advanced semantic searches; automated enrichment with background information; and taking into account journalists' and editors' preferences and work styles. The News Hunter version reported in this paper only uses the text of the report-in-writing. Other cues, such as the journalist's background and interests, along with their contacts, location, and assigned tasks, are not considered. We have so far only harvested texts, and plan to continue doing so at least in the short term, acknowledging that audio and video are also important information sources in the longer run.

These and other research ideas will be followed up in a new research project, *News Angler: discovering unexpected connections in the news*, which started in the autumn of 2018 and which will continue work on the News Hunter prototype. Whereas News Hunter's focus has been on surface similarity, a central aim of News Angler is to support deeper information mining that adapts, combines, and extends theories and techniques from analogical and other types of computational reasoning. We want to suggest unexpected *news angles* on and provide surprising background information about newsworthy events as they unfold. Indeed, our work on News Angler has already produced early examples of how News Hunter can be extended with support for news angles [42–44]. Central sub-goals of the project are: understanding journalistic ICT needs; selecting the right background information and presenting it in the most suitable ways; supporting deep information mining; and empirically evaluating our systems and components. We think that the theories and techniques that News Hunter and News Angler develop potentially have importance beyond journalism, as alternatives to the surface similarity-based search and recommendation services that shape the information bubbles that increasingly surround us today.

Acknowledgements

Early development of News Hunter was supported by NCE (Norwegian Centre of Expertise) Media. News Angler is funded by the Norwegian Research Council's IKTPLUSS programme as project 275872.

References

[1] M. Machill and M. Beiler, The importance of the Internet for journalistic research: A multi-method study of the research performed by journalists working for daily newspapers, radio, television and online, *Journalism Studies* **10**(2) (2009), 178–203.

[2] B. Ekdale, J.B. Singer, M. Tully and S. Harmsen, Making change: Diffusion of technological, relational, and cultural innovation in the newsroom, *Journalism & Mass Communication Quarterly* **92**(4) (2015), 938–958.

[3] B.R. Heravi and J. McGinnis, Introducing Social Semantic Journalism, *The Journal of Media Innovations* **2**(1) (2015), 131–140.

[4] N. Diakopoulos, Computational journalism and the emergence of news platforms, *The Routledge Companion to Digital Journalism Studies, London: Routledge, Taylor and Francis group* (2016).

[5] G. Leban, B. Fortuna, J. Brank and M. Grobelnik, Event Registry: Learning about world events from news, in: *Proceedings of the 23rd International Conference on World Wide Web*, ACM, 2014, pp. 107–110.

[6] X. Liu, A. Nourbakhsh, Q. Li, S. Shah, R. Martin and J. Duprey, Reuters Tracer: Toward automated news production using large scale social media data, in: *2017 IEEE International Conference on Big Data (Big Data)*, IEEE, 2017, pp. 1483–1493.

[7] N. Voskarides, E. Meij, R. Reinanda, A. Khaitan, M. Osborne, G. Stefanoni, P. Kambadur and M. de Rijke, Weakly-supervised contextualization of knowledge graph facts, in: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, ACM, 2018, pp. 765–774.

[8] N.L. Latar, The robot journalist in the age of social physics: The end of human journalism?, in: *The new world of transitioned media*, Springer, 2015, pp. 65–80.

[9] A. Miroshnichenko, AI to bypass creativity. Will robots replace journalists? (The answer is "yes"), *Information* **9**(7) (2018). doi:10.3390/info9070183. http://www.mdpi.com/2078-2489/9/7/183.

[10] D. Allemang and J. Hendler, *Semantic web for the working ontologist: Effective modeling in RDFS and OWL*, Elsevier, 2011.

[11] T. Berners-Lee, J. Hendler, O. Lassila et al., The semantic web, *Scientific american* **284**(5) (2001), 28–37.

[12] N. Shadbolt, T. Berners-Lee and W. Hall, The semantic web revisited, *IEEE Intell. Syst.* **21**(3) (2006), 96–101–.

[13] C. Bizer, T. Heath and T. Berners-Lee, Linked data: The story so far, in: *Semantic services, interoperability and web applications: emerging concepts*, IGI Global, 2011, pp. 205–227.

[14] C. Castillo, *Big crisis data: Social media in disasters and time-critical situations*, Cambridge University Press, 2016.

[15] F. Sebastiani, Machine learning in automated text categorization, *ACM computing surveys (CSUR)* **34**(1) (2002), 1–47.
[16] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*, MIT press, 2016.
[17] A.C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*, O'Reilly Media, Inc., 2016.
[18] S.J. Russell and P. Norvig, *Artificial intelligence: a modern approach*, Malaysia; Pearson Education Limited,, 2016.
[19] N. Fernández, J.M. Blázquez, J.A. Fisteus, L. Sánchez, M. Sintek, A. Bernardi, M. Fuentes, A. Marrara and Z. Ben-Asher, News: Bringing semantic web technologies into news agencies, in: *International Semantic Web Conference*, Springer, Berlin, Heidelberg, 2006, pp. 778–791.
[20] N. Fernández, D. Fuentes, L. Sánchez and J.A. Fisteus, The NEWS ontology: Design and applications, *Expert Systems with Applications* **37**(12) (2010), 8694–8704.
[21] R. García, F. Perdrix and R. Gil, Ontological infrastructure for a semantic newspaper, in: *Semantic Web Annotations for Multimedia Workshop, SWAMM*, 2006.
[22] R. García, F. Perdrix, R. Gil and M. Oliva, The semantic web as a newspaper media convergence facilitator, *Web Semantics: Science, Services and Agents on the World Wide Web* **6**(2) (2008), 151–161.
[23] A.L. Garrido, O. Gomez, S. Ilarri and E. Mena, NASS: news annotation semantic system, in: *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, IEEE, 2011, pp. 904–905.
[24] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak and S. Hellmann, DBpedia-A crystallization point for the Web of Data, *Web Semantics: science, services and agents on the world wide web* **7**(3) (2009), 154–165.
[25] C. Henden, P. Rissen and S. Angeletou, Linked geospatial data, and the BBC.
[26] R. Troncy, Bringing the IPTC news architecture into the semantic web, in: *International Semantic Web Conference*, Springer, 2008, pp. 483–498.
[27] S. Bergamaschi, F. Guerra, M. Orsini, C. Sartori and M. Vincini, Relevant News: a semantic news feed aggregator, in: *Semantic Web Applications and Perspectives*, Vol. 314, Giovanni Semeraro, Eugenio Di Sciascio, Christian Morbidoni, Heiko Stoemer, 2007, pp. 150–159.
[28] M. Krstajić, F. Mansmann, A. Stoffel, M. Atkinson and D.A. Keim, Processing online news streams for large-scale semantic analysis, in: *2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010)*, IEEE, 2010, pp. 215–220.
[29] T. Gruber, Collective knowledge systems: Where the social web meets the semantic web, *Web semantics: science, services and agents on the World Wide Web* **6**(1) (2008), 4–13.
[30] A.L. Opdahl, A. Berven, K. Alipour, O.A. Christensen and K.J. Villanger, Knowledge graphs for newsroom systems, *NOKOBIT—Norsk konferanse for organisasjoners bruk av informasjonsteknologi* **24** (2016).
[31] A. Berven, O.A. Christensen, S. Moldeklev, A.L. Opdahl and K.J. Villanger, News Hunter: Building and mining knowledge graphs for newsroom systems, *NOKOBIT — Norsk konferanse for organisasjoners bruk av informasjonsteknologi* **26** (2018).
[32] A.R. Hevner, A three cycle view of design science research, *Scandinavian journal of information systems* **19**(2) (2007), 4.
[33] R.H.V. Alan, S.T. March, J. Park and S. Ram, Design science in information systems research, *MISQ* **28**(1) (2004), 75–105.
[34] V. Vaishnavi and W. Kuechler, Design research in information systems, 2004.
[35] K. Beck, Embracing change with extreme programming, *Computer* (1999), 70–77.
[36] M.M. Lankhorst, H.A. Proper and H. Jonkers, The architecture of the archimate language, in: *Enterprise, business-process and information systems modeling*, Springer, 2009, pp. 367–380.
[37] P.N. Mendes, M. Jakob, A. García-Silva and C. Bizer, DBpedia Spotlight: shedding light on the web of documents, in: *Proceedings of the 7th international conference on semantic systems*, ACM, 2011, pp. 1–8.
[38] S. Lohmann, V. Link, E. Marbach and S. Negru, WebVOWL: Web-based visualization of ontologies, in: *International Conference on Knowledge Engineering and Knowledge Management*, Springer, 2014, pp. 154–158.
[39] M. Weise, S. Lohmann and F. Haag, Ld-vowl: Extracting and visualizing schema information for linked data, in: *2nd International Workshop on Visualization and Interaction for Ontologies and Linked Data*, 2016, pp. 120–127.
[40] G. Kaur and K. Bajaj, News classification using neural networks, *Commun. Appl. Electron* **5**(1) (2016).
[41] G.A. Miller, WordNet: a lexical database for English, *Communications of the ACM* **38**(11) (1995), 39–41.
[42] M. Gallofré, L. Nyre, A.L. Opdahl, B. Tessem, C. Trattner and C. Veres, Towards a Big Data Platform for News Angles, in: *4th Norwegian Big Data Symposium — NOBIDS 2018.*, 2018.
[43] B. Tessem and A.L. Opdahl, Supporting Journalistic News Angles with Models and Analogies, in: *Proceedings of IEEE RCIS'19, Brussels, Belgium*, IEEE, 2019.
[44] A.L. Opdahl and B. Tessem, Towards Ontological Support for Journalistic Angles, in: *Proceedings of EMMSAD'19, Rome, Italy*, Springer, 2019.
[45] R. Kitchin, *The data revolution: Big data, open data, data infrastructures and their consequences*, Sage, 2014.
[46] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, in: *Advances in neural information processing systems*, 2013, pp. 3111–3119.
[47] F. Hogenboom, F. Frasincar, U. Kaymak and F. De Jong, An overview of event extraction from text, in: *Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011)*, Vol. 779, Citeseer, 2011, pp. 48–57.
[48] A. Gangemi, V. Presutti, D. Reforgiato Recupero, A.G. Nuzzolese, F. Draicchio and M. Mongiovì, Semantic web machine reading with FRED, *Semantic Web* **8**(6) (2017), 873–893.