NEWS HUNTER: BUILDING AND MINING KNOWLEDGE GRAPHS FOR NEWSROOM SYSTEMS

Arne Berven[2], Ole A. Christensen[1], Sindre Moldeklev[1], Andreas L. Opdahl[1], Kjetil J. Villanger[1]

[1] Dept of Info. Science and Media Studies, University of Bergen, P.O.Box 7802, 5020 Bergen
[2] Wolftech Broadcast Solutions, Nøstegaten 72, 5011 Bergen

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. We are exploring how social, open, and other data sources can be leveraged for journalistic purposes through techniques such as 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 future digital journalists. The paper presents the current state of our architecture and its instantiation as a prototype we have called News Hunter. Plans and possibilities for future work are also outlined.*

1. INTRODUCTION

Journalism is in crisis, but new information and communication technologies (ICT) offer new opportunities. Journalists today have access to a wealth of digital information from news aggregators, social media, and open data providers in addition to traditional sources. These sources can be automatically analysed, organised, prepared, and stored with increasing semantic precision and linked to related information. 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.

A university research group is therefore collaborating with a software developer of news production tools for the international market. Together we are developing *News Hunter*, a prototype tool and an architecture that harvests news items and social media messages from the net; analyses and represents them semantically in a knowledge graph; classifies, clusters, and labels them; enriches them with background information; and presents them in real time to journalists who are working on related stories 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 semantic technologies, 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 d*evelop and evaluate a prototype 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 proofs-of-concepts 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 digital journalism?* The rest of the paper presents our current and tentative 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 semantic technologies (Allemang and Hendler 2011, Berners-Lee et al. 2001, Shadbolt et al. 2006) and linked open data (LOD) (Bizer, Heath, et al. 2009) along with techniques from natural-language processing (NLP) (Castillo 2016, Sebastiani 2002), machine learning (Müller and Guido 2016, Sebastiani 2002), and artificial intelligence in general. A central idea behind semantic technologies and linked open data is to represent information as RDF graphs, called knowledge graphs in this paper.

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 (Bizer, Lehmann, et al. 2009), a semantic version Wikipedia. BBC makes the ontologies they use (Henden et al. 2014) available on their linked-data platform[1]. The Norwegian national broadcaster NRK also uses semantically linked data stored in a graph database (triple store) to make their archive of radio and TV programmes more findable, creating new possibilities for journalists in their daily work and enabling new services to journalists and to the public (Børdalen 2017). With similar aims, the International Press Telecommunications Council (IPTC) provides the rNews standard[2] for embedding semantic metadata in online news. Some other news organisations also define semantic-annotation standards.

Targeting the general public, semantic news aggregation tries to cope with ever increasing streams of information by aggregating news items based on named entities (such as people, places, and organisations), topics, and time. Google News[3] aggregates and attempts to prioritise news items by stories, which are also categorised. Yahoo News[4] offers similar services. Newsmap[5] uses tree maps to visualise aggregated Google News. The European Media Monitor (EMM) (Krstajić et al. 2010) applies clustering and named-entity recognition on multi-lingual 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. The RELEVANT tool (Bergamaschi et al. 2007) uses text similarity to automatically group news from different sources by topic, making them accessible through a web-feed reader. EventRegistry (Leban et al. 2014) collects news 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[6]. webLyzard[7] 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 rich visual dashboards.

Targeting news workers such as journalists, the NEWS project (Fernández et al. 2006) uses semantic technologies and NLP to annotate news items precisely, using a domain-specific ontology (Fernández et al. 2006, 2010, García et al. 2006). 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 (Garrido et al. 2011) was developed in collaboration with Spanish newspapers in order to automatically classify news documents faster and more reliably than humans.

1 http://www.bbc.co.uk/ontologies

2 http://iptc.org/standards/rnews/

3 http://news.google.com

4 http://www.yahoo.com/news

5 http://newsmap.jp

6 http://dmoz-odp.org/Science/Biology/Taxonomy/

7 http://weblyzard.com

3. RESEARCH APPROACH

3.1 Collaborators

The group for *Semantic and Social Information Systems (SSIS)* at the University of Bergen studies information systems in the interaction between *semantic technologies* and *social media*. The group sees the two as complementary (Gruber 2008), because social media tend to generate big datasets that can be enriched and made more useful with semantic technologies and because social media is an important source of large-scale semantic annotations.

Wolftech Broadcast Solution is a software company that spun off from TV2 in 2011. The company 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.).

Wolftech News Hunter (Opdahl et al. 2016) is a collaboration between UiB and Wolftech that aims to extend Wolftech News with techniques such as knowledge graphs, natural-language processing, and machine learning to harvest, organise and leverage social media streams and other big-data sources for journalistic purposes.

3.2 Research questions

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

3.3 Research method

Because our goals and research questions were explorative and involved technology development, we chose design science as our overall research method (Hevner 2007, Von Alan et al. 2004). 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 an *instantiation* – a situated implementation in a specific environment – of that architecture (Vaishnavi and Kuechler 2004). Of course, our research also involves more specific artefacts, such as *design principles* and *theories behind* the architecture and instantiation. In addition, design science research should be informed by theory in relevant fields and rigorously contribute to that theory and be relevant to practice in relevant fields and contribute to improving that practice.

3.4 Development method

We followed an iterative development strategy with synchronised research and development iterations in continuous dialogue with Wolftech, in order 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 encouraged an XP strategy (Beck 1999) of short 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 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 later added Python, PyCharm IDE, Visual Studio Code IDE, and Standard.js for development, and GitHub/Bitbucket and Trello for

project management. While the core application is is still dependent on ASP.NET, more and more components are written in Python. They are tied to one another and with the ASP.NET core through Flask microservices. The front end is written mainly in AngularJS, HTML and CSS, with the aid of Sketch and Marvel.

3.5 Evaluation methods

Of our two central artefacts, the architecture is abstract and the prototype not ready for end-user testing. Instead of evaluations with end users, we have therefore used two types of evaluations: most centrally (1) proof-of-concept evaluations to ensure that our prototype components worked both in isolation and together and, in addition, (2) component evaluations to gauge the quality of specific components, e.g., of our natural-language analysis and user interfaces, using established research methods for each type of component.

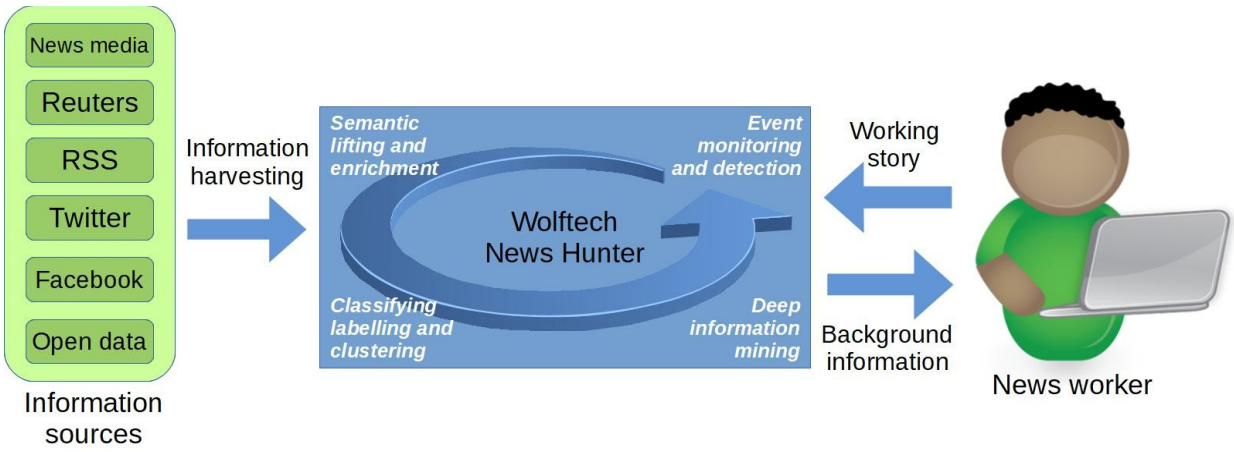

Figure 1. News Hunter harvests information from social, open, and other sources and presents it as relevant background information to working journalists.

4. ARCHITECTURE

Figure 1 depicts News Hunter in context. Each of the following components will be discussed further in the next section:

- *Harvesters:* Text messages (articles, items, posts, tweets…) are continuously pulled from relevant sources such as Facebook and Twitter, RSS feeds, and news sites.
- *Message store:* The harvested messages are stored in a text-oriented database.
- *Translator:* If necessary, an English-language translation is created for each message and stored in the text database.
- *Lifters:* The (translated) messages are run through an NL pipeline to identify central topics (described by keywords), named entities (such as people, organisations and places), and sentiments.
- *News graph:* The resulting metadata (topics, named entities, sentiments, etc.) about each message are stored in a graph database/triple store to support further analyses and more precise retrieval.
- *Ontology:* The news graph is structured according to a well-defined and evolving ontology.
- *Microservice framework:* The components in the architecture are tied together by lightweight REST APIs.
- *Organisers:* The harvested messages and their metadata are run through further NL and ML pipelines for single- and multi-label classification, clustering, event detection, and social network analysis.
- *News editor:* While journalists work on stories, they are continuously run through the same NL analysis pipeline as the harvested messages, in order to identify central topics, named entities and sentiments.

- *Front end:* The news editor is part of a front end that uses the identified topics and named entities to retrieve relevant background information from the news graph and the message store.
- *Enrichers:* Additional background information is retrieved from social, open, and other data sources on the web, such as DBpedia and Twitter.

5. INSTANTIATION

Work on the News Hunter prototype started in the summer of 2015 (Opdahl et al. 2016). 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 2). The prototype was written in C# as an ASP.NET application with a BrightstarDB database for semantic storage of the graph.

The work has since continued through NCE Media funding and several master theses at the University of Bergen (Christensen and Villanger 2017, Moldeklev 2018). The rest of this section will present the central components of the current 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.

5.1 Harvesters

Going beyond the Facebook API, we harvest posts from Twitter and RSS feeds and from major Norwegian news outlets. These Python scripts use the Tweepy library to harvest Tweets, the Feedparser library for RSS, and the Newspaper library to harvest news articles. To avoid downloading the same post twice, we use the XXhash algorithm (with salting disabled) on the URLs of news articles and posts. The pre-processed messages are stored as JSON.

5.2 Message store

Another Python script inserts the harvested message texts 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.

5.3 Translator

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

5.4 Lifters

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 also 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 summarisation of longer articles. To offset the effect of longer texts generating more keywords, we weight each keyword based on its position in the text (assuming that news articles describe its most central aspects early in the text) and number of occurrences, so that repeated or late-occurring keywords are given lower weight.

Named-entity recognition: For part-of-speech (PoS) tagging and named-entity recognition (NER) we use the Python-library Spacy (which also supports deep learning, see below). For named-entity disambiguation, we use the DBpedia Spotlight tool (Mendes et al. 2011), 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 AFINN Python library is used for sentiment analysis.

5.5 News graph

We represent the lifted messages in a knowledge graph managed persistently in the BrighstarDB triple store, a DBMS specifically for RDF graphs (Allemang and Hendler 2011). 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.

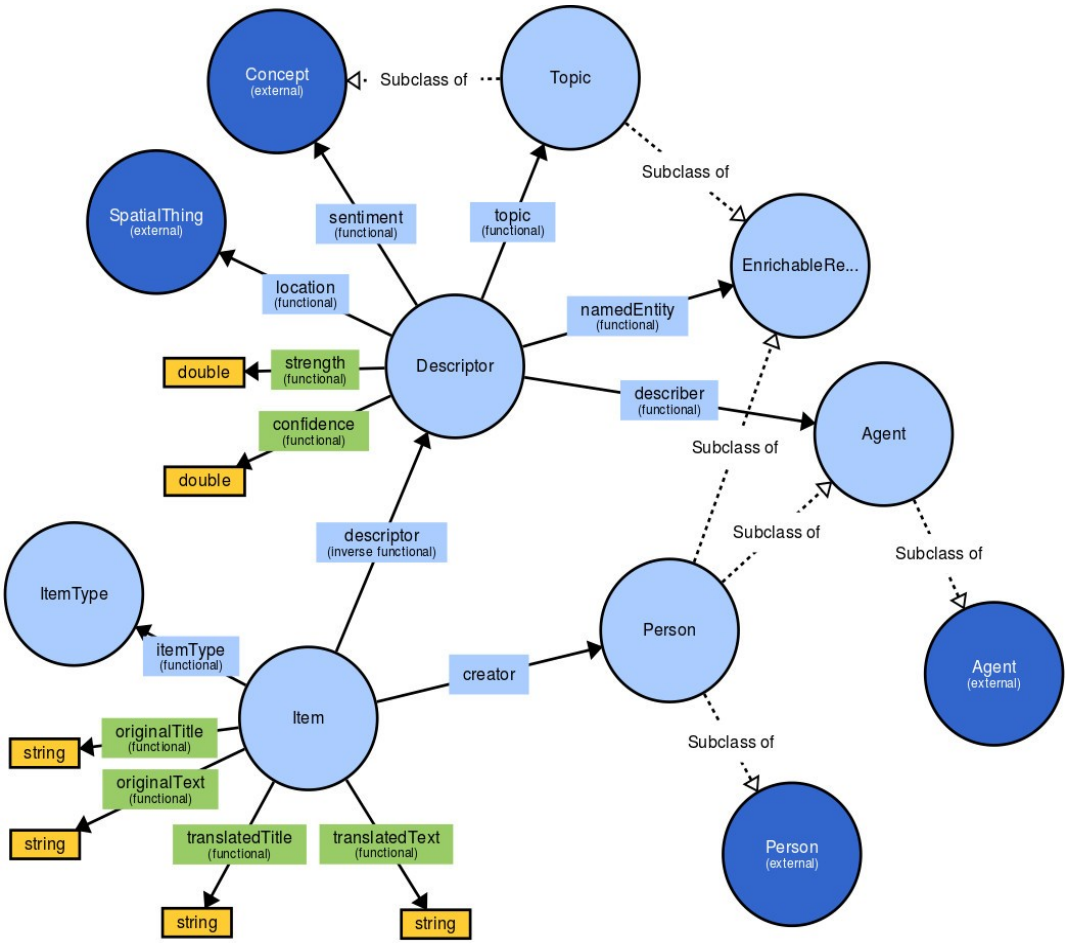

Figure 2. The revised News Hunter ontology.

5.6 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. The current version of the ontology is shown in Figure 2 using the WebVOWL visualiser[8] (Lohmann et al. 2014).

Items are potentially news-relevant items, for example news articles, stories-being-written, RSS feeds, blogs posts, and microtexts such as tweets. *Descriptors* represent item contents, for example topics, named entities, locations, and sentiments mentioned or expressed in the text. Topics and named

8 http://www.visualdataweb.de/webvowl

entities such as persons are *enrichable entities* 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 2 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.

5.7 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 micro framework. Flask is written for Python, but is also usable from C#.

5.8 Organisers

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 (Kaur and Bajaj 2016). As training and evaluation data (Sebastiani 2002), we use pre-labelled RSS feeds. Each message is pre-processed for cleaning 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 explore two much-used classifiers: a support-vector machine (SVM) and a multi-layer perceptron (MLP) neural network, both wrapped in a Flask API.

*Multi-label classification: M*ulti-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 IPTC newscodes (Troncy 2008). As training and evaluation data, we use a set of pre-labelled articles from The Guardian, available through a public API. We use string equivalence and Wordnet synsets 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.

Clustering and 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.

Social network analysis: We also explore 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.

5.9 News editor

Froala's WYSIWYG editor plugin is used to write up news stories. 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 article. Other journalists in the same organisation that are working on stories 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.

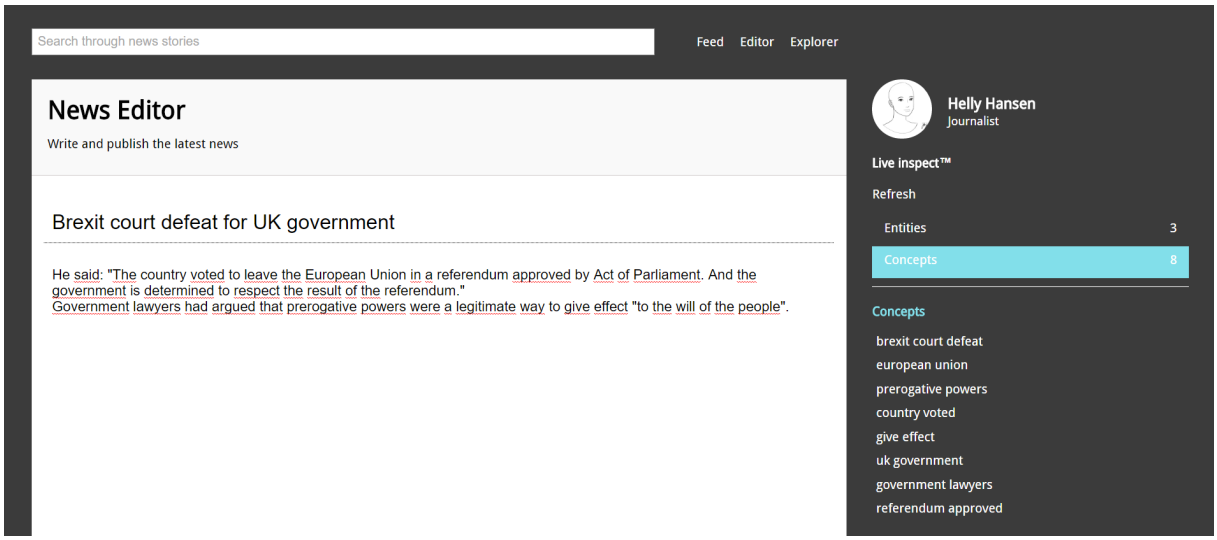

Figure 4. Prototype front-end. The left side shows an early version of the text editor. The right part shows the results of analysing the story-being-written.

5.10 Front end

The news 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.

5.11 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 (currently restricted to sports athletes) are used to retrieve recent tweets on demand from the official Twitter API.

6. EVALUATION

Because our research goal and artefact has been on the architecture level, our main mode of evaluation has been repeated proof-of-concept tests after each development iteration (typically every 1-3 weeks). In addition, we have conducted limited point evaluations of central components in our architecture.

6.1 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.

6.2 Evaluations with users

Developing or improving new specific theories or technologies for components has not been our central research goal so far, but we have conducted limited evaluations of central components to ensure that they perform acceptably, relying on established research methods for each type of component.

Front-end: We let six journalists and domain experts use the prototype and presented them with questions about the usability and usefulness of each of the following features: inspecting messages through the front end; displaying named entities for stories-being-written; displaying keywords for stories-being-written; showing related messages for stories-being-written; generating summaries of longer messages; and showing clusters of recent messages. In general, the respondents were positive to all the features, but pointed to many areas of improvement for further work. For details, see (Christensen and Villanger 2017).

Topic extraction: We let the users read through a sample of medium-length news articles and asked them to suggest suitable keywords for each. We then showed them two keyword lists generated by TextRank and RAKE respectively. The results showed many overlapping keywords, and the users tended to prefer the keywords generated by TextRank over those from RAKE. Performing a quantitative comparison with standard information retrieval measures (*P*, *R*, and *F1*) turned out to be difficult because the different users each produced different keywords; because one of the algorithms (RAKE), produced many keyword phrases as opposed to single words; and because it was not clear how to score overlapping phrases.

*Named entity recognition: F*rom a convenient sample of three news articles, we also extracted 53 named entities, along with their types, and presented them to the users for comments. They found between 73% and 89% of the entities appropriate, having smaller or larger issues with 11% to 27% of them. Some users commented that the types identified for named entities should have been more specific (i.e., *football player* instead of *person*).

Labelling: The users agreed that the (single-label) categories were correct, but too general. More precise and descriptive categories would be needed in a practically useful tool.

6.3 Evaluations of algorithms

Translation: To evaluate automatic translation, we conveniently sampled two international news events: an iPhone release and an international football match. For each event, we chose news articles in English, Norwegian, and Spanish of comparable lengths and scopes. We then extracted keywords and named entities from the (translated English versions of) the articles. The results showed much overlap, but suggested that exact message matching across languages cannot reliably be based on automatic translation, keywords, and named entities 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 best for sports (*F1*=0.98..0.99) and worst for education (*F1*=0.68..0.69), perhaps because the training set contained a lot more sports than education articles.

Multi-labelling evaluations: 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 (Christensen and Villanger 2017).

Event-detection evaluation: 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.

7. DISCUSSION

To the best of our knowledge, the architecture and instantiation proposed in this paper go beyond the previous work we describe in the background section. Whereas *News Hunter* targets journalists specifically – and offers them an integrated front end for news editing – most existing tools and services are aimed at the general public or at other professions, such as archivists. Among the news aggregators, EventRegistry (Leban et al. 2014) is the one most similar to ours, using similar semantic message analysis, classification, clustering and event detection techniques. However, it is restricted to analysis of RSS feeds and uses only background information from Wikipedia. Among the journalistic tools, NEWS (Fernández et al. 2006) also uses semantic message analysis and classification. But it does not focus on event detection and is accessed through a web API instead of a front end. Another distinction is that both EventRegistry and NEWS are 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 webLyzard which is, however, a general web intelligence platform that does not target news workers specifically.

In summary, we consider our architecture and instantiation unique because they offer, at the same time:

- harvesting information from a variety of sources, including both established media, news aggregators, RSS feeds, and social media;
- harvesting both pre-news information and unfolding news items;
- handling multiple languages;
- analysing the harvested information semantically – and in a way that combines: named-entity recognition/disambiguation/linking, topic extraction, sentiment analysis, single- and multi-labelling, clustering, and event detection;
- semantic enrichment from a variety of sources; and
- a front end for working journalists – including a news editor that continuously analyses stories-being-written so that related news items and background information are suggested in the front end in real time.

The practical and theoretical contribution of this paper is to propose, instantiate, and partially evaluate an architecture and prototype that conjoins these features. We think the architecture and prototype answer the research question we posed in the introduction positively albeit tentatively. In may indeed be possible to combine heterogeneous data sources and techniques into a flexible architecture that supports the needs of digital journalism. Specifically, information and communication techniques (ICTs) such as semantic technologies, natural-language processing, and machine learning can most likely 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. Existing tools and services – such as webLyzard, EventRegistry, NEWS, Google/Yahoo News, NewsMap, and others – all suggest ways to extend our architecture and prototype tool to provide even richer support for news workers, and the research areas we build on most prominently – semantic technologies, natural-language processing, and machine learning – are all advancing rapidly.

8. CONCLUSION

The *News Hunter* and *News Angler* projects are positioned in a rich research area and emerging commercial market, which has ample potential for further research, development, innovation, and industrialisation. Much work remains, but we think the results so far are promising. Our most immediate focus is on streamlining and extending our prototype system with new types of analysis. A central challenge is to ensure that the prototype continues to work as the architecture evolves and more components are added. To achieve this, groups of components and pipelines that perform similar functions – such as the various harvesters, lifters, and analysers – need to be streamlined, sometimes parallelised, and provided with microservice-based standard interfaces. Building up an even larger-scale news graph and message base for further testing and research is a priority.

Many of the research and development challenges pointed out in (Opdahl et al. 2016) remain important, such as: (very) 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. Currently, News Hunter only uses the text of the story-being-written. 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.

These and other research ideas will be explored in the new research project *News Angler: discovering unexpected connections in the news*, which starts in 2018 and which will hopefully continue to build on the News Hunter prototype. Whereas News Hunter's focus has been on surface similarity, News Angler will attempt 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 angles on and provide surprising background information about unfolding news stories. Central sub-goals are: understanding journalistic work; preparing background information in a good way; supporting deep information mining; and empirically evaluating our systems and components.

We think that the theories and techniques that News Hunter/Angler develops potentially have importance beyond journalism, as an alternative to the surface similarity-based search and recommendation services that shape the information bubbles that increasingly surround us today.

ACKNOWLEDGEMENT

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

Allemang D, Hendler J (2011) *Semantic web for the working ontologist: effective modeling in RDFS and OWL* (Elsevier).

Beck K (1999) Embracing change with extreme programming. *Computer* 32(10):70–77.

Bergamaschi S, Guerra F, Orsini M, Sartori C, Vincini M (2007) Relevant News: a semantic news feed aggregator. (Giovanni Semeraro, Eugenio Di Sciascio, Christian Morbidoni, Heiko Stoemer), 150–159.

Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. *Sci. Am.* 284(5):34–43.

Bizer C, Heath T, Berners-Lee T (2009) Linked data-the story so far. *Int. J. Semantic Web Inf. Syst.* 5(3):1–22.

Bizer C, Lehmann J, Kobilarov G, Auer S, Becker C, Cyganiak R, Hellmann S (2009) DBpedia-A crystallization point for the Web of Data. *Web Semant. Sci. Serv. Agents World Wide Web* 7(3):154–165.

Børdalen G (2017) NRK bygger infrastruktur for å finne igjen programmene ("NRK builds infrastructure to find programmes"). (March 1) https://nrkbeta.no/2017/03/01/nrk-bygger-infrastruktur-for-a-finne-igjen-programmene/.

Castillo C (2016) *Big crisis data: Social media in disasters and time-critical situations* (Cambridge University Press).

Christensen OA, Villanger KJ (2017) News Hunter: a semantic news aggregator.

Fernández N, Blázquez JM, Fisteus JA, Sánchez L, Sintek M, Bernardi A, Fuentes M, Marrara A, Ben-Asher Z (2006) News: Bringing semantic web technologies into news agencies. (Springer), 778–791.

Fernández N, Fuentes D, Sánchez L, Fisteus JA (2010) The NEWS ontology: Design and applications. *Expert Syst. Appl.* 37(12):8694–8704.

García R, Perdrix F, Gil R (2006) Ontological infrastructure for a semantic newspaper.

Garrido AL, Gomez O, Ilarri S, Mena E (2011) NASS: news annotation semantic system. (IEEE), 904–905.

Gruber T (2008) Collective knowledge systems: Where the social web meets the semantic web. *Web Semant. Sci. Serv. Agents World Wide Web* 6(1):4–13.

Henden C, Rissen P, Angeletou S (2014) Linked geospatial data, and the BBC.

Hevner AR (2007) A three cycle view of design science research. *Scand. J. Inf. Syst.* 19(2):4.

Kaur G, Bajaj K (2016) News Classification using Neural Networks. *Commun. Appl. Electron.* 5(1).

Krstajić M, Mansmann F, Stoffel A, Atkinson M, Keim DA (2010) Processing online news streams for large-scale semantic analysis. (IEEE), 215–220.

Leban G, Fortuna B, Brank J, Grobelnik M (2014) Event registry: learning about world events from news. (ACM), 107–110.

Lohmann S, Link V, Marbach E, Negru S (2014) WebVOWL: Web-based visualization of ontologies. *Proc Int. Conf. Knowl. Eng. Knowl. Manag.* (Springer), 154–158.

Mendes PN, Jakob M, García-Silva A, Bizer C (2011) DBpedia spotlight: shedding light on the web of documents. (ACM), 1–8.

Moldeklev S (2018) *Semantic Technologies and Sports News (working title)*. Master. (University of Bergen, Bergen).

Müller AC, Guido S (2016) *Introduction to machine learning with Python: a guide for data scientists* (O'Reilly Media, Inc.).

Opdahl AL, Berven A, Alipour K, Christensen OA, Villanger KJ (2016) KNOWLEDGE GRAPHS FOR NEWSROOM SYSTEMS.

Sebastiani F (2002) Machine learning in automated text categorization. *ACM Comput. Surv. CSUR* 34(1):1–47.

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

Troncy R (2008) Bringing the IPTC news architecture into the semantic web. (Springer), 483–498.

Vaishnavi V, Kuechler W (2004) Design research in information systems.

Von Alan RH, March ST, Park J, Ram S (2004) Design science in information systems research. *MIS Q.* 28(1):75–105.