

INJECT

FUTURE
JOURNALISM,
NOW.

Project Number 732278

Deliverable N°: D1.3

Extended and Aggregated Digital Services and Tools (Second Version)

December 2017

Version N° 1

Main Author(s): Konstantinos Zachos, Neil Maiden, Stergios Ientsek, Fotis Paraskevopoulos, Jeremy Evans, Dimitris Apostolou

Institution(s): City, University of London, EXPLAAIN, ICCS

Project funded by the European Union from the EU

H2020 Programme under grant agreement number 732278



Horizon 2020
European Union Funding
for Research & Innovation

Project ref. no.	732278
Project title	Innovative Journalism: Enhanced Creativity Tools

Nature of Deliverable	R + D= Report and Demonstrator
Contractual date of delivery	Project Month 12 – December 2017
Actual date of delivery	3 rd January 2018
Deliverable number	D 1.3
Deliverable title	Extended and aggregated digital services and tools
Dissemination Level	PU = Public
Status & version	Final 1.0
Number of pages	48
WP relevant to deliverable	WP1
Lead Participant	City, University of London.
Author(s)	Konstantinos Zachos, Neil Maiden, Stergios Iantsek, Jeremy Evans, Fotis Paraskevopoulos, Dimitris Apostolou with contributions from INJECT partners
Project coordinator	Neil Maiden
EC Project Officer	Albert Gauthier
Keywords	Architecture, Journalism Service-Set

Executive Summary

The purpose of this deliverable is to document the extended and aggregated INJECT digital services and their status at M12 of the INJECT project. The deliverable reports the emerging stable version of the INJECT digital tool and services after most development resources have been committed to construct and test the tool and services. To meet reported journalist and newsroom requirements, INJECT is now available for use as plug-ins to different authoring tools including Google Docs, Wordpress and other content management systems that use the TinyMCE editor, to Adobe's InCopy editor, as a standalone web application for journalists who do not use any of these established authoring tools. INJECT also supports journalists fully who are work in 4 different languages – English, Norwegian, German and Dutch – by discovering news sources and generating creative content in these 4 languages, and there is also partial support for journalists who are working in French and Italian. INJECT services generate creative guidance from over 3 million different news articles indexed and retrieved from over 290 different news sources in different countries, as well as from Twitter and video channels, and bespoke archives that belong to the 3 newspapers in the Norwegian ecosystem. In response to journalist feedback, the interactive tool has new features to speed up the generation of creative content, and more explanation of INJECT's creative strategies and generated creative content.

Table of contents

EXECUTIVE SUMMARY3

TABLE OF CONTENTS4

LIST OF FIGURES6

LIST OF TABLES8

1. INTRODUCTION9

2. TECHNICAL ARCHITECTURE..... 10

2.1. USER INTERACTION LAYER 11

2.2. APPLICATION LAYER 11

2.3. CREATIVE SEARCH SERVICE 12

2.4. PRESSER: NEWS AGGREGATION AND CONTENT EXTRACTION SERVICE..... 14

2.5. CRAWLERS..... 14

2.6. NEWS CRAWLER..... 14

2.7. NEWS PROCESSOR..... 16

2.8. NEWS API..... 17

2.9. NEWS AUGMENTATION SERVICE..... 17

2.10. CREATIVE SPARKS..... 20

2.11. MULTI-LANGUAGE SUPPORT 21

3. DATA LAYER..... 22

3.1. PRIVATE NEWS ARCHIVES AND THE NEWS IMPORTER COMPONENT 23

3.2. PUBLIC SOCIAL MEDIA AND VIDEO SOURCES 23

3.3. THE VJ MOVEMENT’S DATABASE OF DIGITAL CARTOONS..... 26

3.4. INJECT’S PERSISTENCE COMPONENT 26

3.5. PRE-COMMERCIAL IMPLEMENTATION OF THE APPLICATION LAYER 27

- 4. INJECT USER INTERFACES 28**
 - 4.1. THE GOOGLE DOCS INTERFACE 28
 - 4.2. INJECT’S CMS & STANDALONE INTERFACES 40
 - 4.3. THE TINYMCE INJECT INTERFACE FOR CONTENT MANAGEMENT SYSTEMS 41
 - 4.4. THE ADOBE INCOPY INJECT INTERFACE 42
 - 4.5. THE STAND-ALONE INJECT INTERFACE FOR WEB BROWSERS 43
- 5. SUMMARY AND OUTLOOK..... 45**
- 6. REFERENCES 48**

List of figures

Figure 1: INJECT Technical Architecture, M12 status..... 10

Figure 2. The architecture of INJECT's Presser component..... 14

Figure 3: Examples of INJECT concept cards that build on the Explain technology and service 19

Figure 4. Example excerpt of a JSON file 24

Figure 5. Scoring Algorithm 25

Figure 6. Example of results for tweets posted by individuals 25

Figure 7. Querying Twitter by location..... 26

Figure 8. Different paid services that support the INJECT application layer 27

Figure 9 . The INJECT sidebar’s 4 functional areas and an example of the tutorial support available to journalists 30

Figure 10. The INJECT sidebar, displaying extracted keywords from highlighted text in the text editor..... 31

Figure 11. The INJECT toolbox and creative search parameters 32

Figure 12. The INJECT sidebar, showing news information retrieved using INJECT’s backing and evidence creative strategy, presented in interactive cards..... 33

Figure 13. INJECT’s interactive explanation text presented at the top of the sidebar. The journalist can click on the green text to change INJECT parameters and launch new requests for creative guidance 34

Figure 14. An example of INJECT’s more detailed explanation text, presenting the journalist rationale for the presentation of news information using the creative strategy selected by the journalist 35

Figure 15. INJECT’s creative guidance presenting individuals associated with the news story defined in the topic terms in the INJECT sidebar 35

Figure 16. INJECT’s mouse hover-over creativity spark feature, showing one spark generated from one entity extracted from a news story 36

Figure 17. INJECT features associated with each interactive card presented in the sidebar. The features, from left to right are the different reverse-side features, the list format of

the creativity sparks, and the visual word-cloud generated from the news article..... 38

Figure 18. The INJECT sidebar displaying digital cartoons associated with the topic terms, presenting using INJECT’s quirky and satire creative strategy 38

Figure 19. INJECT presenting original news information from source, in different web browser tabs..... 39

Figure 20. INJECT presenting the concept cards for clicked entities in the INJECT sidebar 40

Figure 21. INJECT plugin for TinyMCE 41

Figure 22. INJECT user interface for Wordpress..... 42

Figure 23. INJECT user interface for the Adobe InCopy text editor 43

Figure 24. INJECT user interface for standalone web browsers, on desktop computer and mobile device 44

List of tables

Table 1. Public news sources that contribute to INJECT's data layer, by country of news source 23

Table 2. Functional requirement satisfaction at month M12 46

Table 3. Non-functional requirement satisfaction at month M12 47

1. Introduction

The present deliverable of the INJECT project is prepared under WP1 and is the final result of Task 1.2 *Extend and aggregate digital services*. This task focuses on extending and aggregating the INJECT digital services to interoperate with the existing authoring tools such as WordPress, Google Docs and CMSs used by partner SMEs as well as bespoke digital content repositories of the partner SMEs. Given that the different SMEs use different combinations of authoring, content management and website tools, this task has developed an architecture that enables easy integration with different operational journalism ICT environments.

This deliverable is the second of the two versions of D1.2 which documents the technical architecture implemented in Task 1.2 and used for developing the INJECT services and tools, to meet the requirements of all project partners and future target customers in different markets. D1.3 documents the full extensions to the services implemented to month 12, in accordance to the development plan presented in D1.1. From months m7 to m12, different versions of the INJECT services and tool underwent iterative development and evaluation with newsrooms and journalists both within and outside of the INJECT consortium, and many of INJECT features and changes reflect feedback captured directly from journalists who used INJECT either in controlled feedback sessions and/or in-situ, to support their journalist work activities.

This deliverable is organised as follows: Section 2 describes the new version of the technical architecture that built on the technical work completed in the first 6 months of the project. Section 3 presents the new INJECT user interfaces and interactive capabilities that are now available to journalists, and the different forms of interface – through text editors, content management systems and a standalone web interface – that journalists can use INJECT through. Section 4 concludes this document with a summary and further possible developments of the INJECT services and product as part of the exploitation work packages and the planned social enterprise beyond the INJECT consortium.

2. Technical Architecture

In this section, we report the status of the technical architecture of INJECT, initially presented in D1.1 and revised in D1.2. An overview of the technical architecture and the implementation status at M12 are shown in Figure 1. The architecture is divided into three main layers:

- *User Interaction Layer.* Users will interact with the INJECT system through User Interfaces belonging to the User Interaction Layer;
- *Application Layer.* The application layer includes the key software components, their services and interdependencies that will support journalists to undertake creative story-writing activities;
- *Data Layer.* The data layer provides access to internal and external news sources and content repositories.

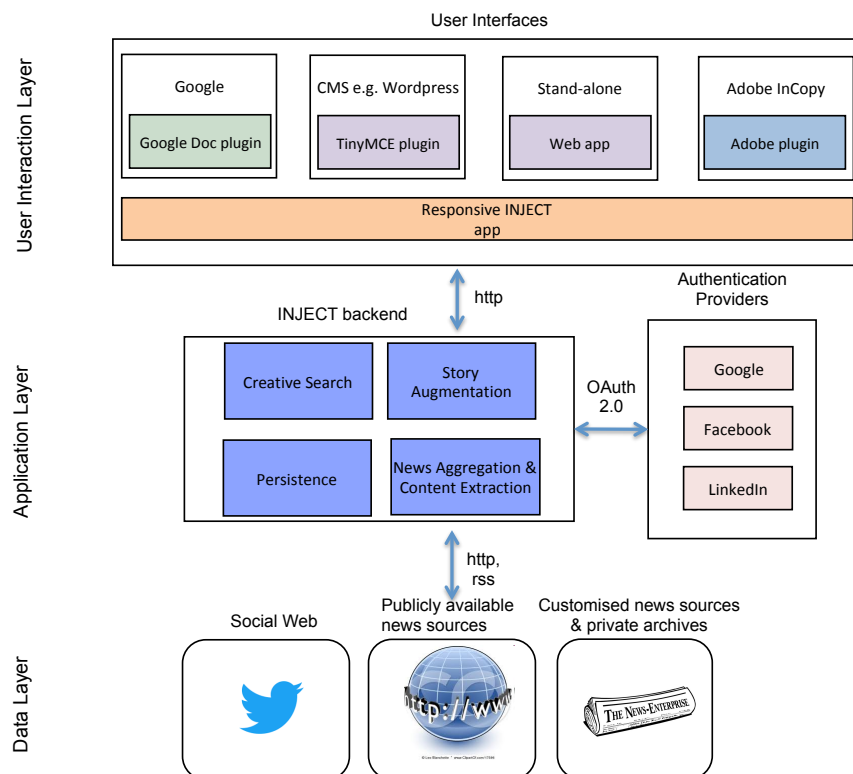


Figure 1: INJECT Technical Architecture, M12 status.

2.1. User Interaction Layer

The user interaction layer of the architecture comprises multiple interfaces through which the INJECT functionality is provided to journalists:

1. A Google Docs interface, which provides INJECT capabilities to journalists through an Add-on sidebar that uses Google Docs as the editor for creating, editing, storing and deleting news stories. The Google Docs interface has been developed for use by both freelance journalists, who are accustomed to the free on-line editing tools provided by Google and wish to have easy access to INJECT functionalities, and by newsrooms that increasingly use powerful editing tools such as Google Docs prior to uploading new news stories into their bespoke content management systems;
2. A content management system interface, which provides the INJECT functionalities through a TinyMCE plug-in and sidebar. TinyMCE is a popular free editor that can be used with all popular CMSs such as Wordpress, Joomla and Drupal, to create, edit, store and delete news stories. The INJECT interface is again a sidebar on the right-hand side of the text editor window. This interface is still expected to serve journalists who work in newsrooms and are accustomed to write their articles directly into the newsroom's content management system. TinyMCE was selected as the target text editor because it is among the most popular content management system editor;
3. An Adobe InCopy interface, which is part of the Adobe InDesign product and provides INJECT capabilities to journalists in a small but significant segment of the newsroom market, and used by the 3 original news organisations that are members of the project's Norwegian ecosystem;
4. A stand-alone web interface, which provides the INJECT functionalities via the TinyMCE editor to journalists using a web browser and to interact with these capabilities using other desktop applications using standard operating system capabilities. This interface is expected to be used by casual or first time INJECT users such as new members of the INJECT ecosystem.

The functionalities provided through the different INJECT user interfaces are described at length in in Section 3.

2.2. Application Layer

The application layer of INJECT is composed of four key software components & services: (i) the *creative search service* that manipulates text from the editor to generate search queries that form each input to 6 creative news angles; (ii) the *news aggregation and content extraction service*, called Presser, which searches 300 news sources using creative search queries generated by the creative search service; (iii) the *story augmentation service*, which parses and enriches search results with Explain cards; (iv) the *persistence component* which will deliver the necessary search session storage and retrieval capabilities, and provides user authorisation and authentication capabilities. Earlier versions of the components and services (i) – (iii) were reported in D1.2, on M6. In this deliverable we report updates to these

components and services, and describe the persistence component and service. Each is described in turn.

2.3. Creative Search Service

As reported in Deliverable D1.2, the creative search service was designed to manipulate input text automatically to generate search query terms that form the input to 6 creative news angles identified by experienced journalists and leading digital experts in journalism as effective for developing creative news angles. These original 6 creative news angles were:

- I. **Backing and evidence:** creatively explore numerical and quantified information about a news story in order to back it up with evidence in new and useful ways;
- II. **Individuals:** creatively explore different human angles in a news story based on the different people and roles associated with the story;
- III. **Causal:** creatively explore the background events that underpin a story in order to discover a new angle to the story from its background;
- IV. **Quirky:** creatively explore unusual or comical information about a story as a means of using wit to report serious news;
- V. **Ramifications:** creatively explore information about the consequences of events and actions related to a news story in order to generate new and useful stories about future consequences;
- VI. **Data visualizations:** creatively explore different data sets and visualisations in order to generate new and useful ways about how to display information in a news story to encourage human insight.

Each of these 6 angles was then designed in the form of a computerised algorithm to search for, retrieve and present news information to journalists. The original plan was to implement all 6 of these algorithms by month M12 of INJECT. However, by month M12, only the first 4 – backing and evidence, individuals, causal and quirky – were implemented. The primary reason for this was a project-wide decision to divert development resources to design and implement creative search algorithms that retrieve news archives of some of the project partners, in their native languages. Implementation of the algorithms for ramifications and data visualizations is planned as part of exploitation work in the remaining 6 months of the INJECT project.

All 6 creative search algorithms were based on the same manipulation of natural language text to make sense of then expand complex queries that implemented the different creative searches, as reported in Deliverable D1.2. All of the creative search services generate creative service queries in 3 steps. The first divides the inserted text into sentences that are then tokenized, part-of-speech tagged and modified to include each term's morphological root (e.g. committed to commit) using the Brill Tagger (Brill 1992). The second applies increasingly sophisticated procedures to disambiguate each term by discovering its correct sense and tagging it with that sense using context knowledge from other terms in the query (e.g. defining a parliament to be a legislative assembly in certain countries rather than a card game)

(McCarthy et al. 2004, Stevenson & Wilks 2001). The third implements different creative search strategies that expand each term with other terms that have similar meanings to the tagged sense to search for and retrieve news information (e.g. the term parliament is synonymous with the terms congress, senate and house are then also included in the query based on the creative strategy). These term senses are inferred automatically from WordNet, an on-line lexicon (Simpson 2005) that assigns senses to terms categorized as nouns, verbs, adjectives and adverbs. Each sense defines the meanings of a term, and WordNet organizes these senses into synonym sets that describe concepts with definitions or glosses, each of which contains a definition phrase composed of terms. The service uses this information about the disambiguated meanings of English language terms to implement 3 different creative search strategies:

- I. Synonym set expansion: each disambiguated term is replaced with its synonym set, for example the verb commit is extended with the synonym set for the disambiguated sense #1 [perpetrate, pull] and noun government extended with the set for the disambiguated sense #1 [authorities, regime]. The strategy expands the query to retrieve news information about each object or action described in the inserted text;
- II. Hypernym expansion: each disambiguated term is augmented by its direct hypernym. For example, the hypernym of the disambiguated term commit is act, so the revised query would include both terms. In our example, the service might retrieve news information about other forms of different government actions;
- III. Gloss words expansion: each term is augmented with all terms in its gloss specified in WordNet. For example, the sense #1 definition of the verb term commit is to perform an act, usually with a negative connotation. Hence the terms perform, act, negative and connotation are extracted and included in the extended query.

Each time that the service is invoked, it applies all 3 strategies to match sense-tagged query terms to a similar set of terms to return an unordered set of news articles or digital cartoons that achieved a threshold match score with the expanded search terms. Then, depending on the different creative search algorithm that is selected by the journalist, the service filters retrieved news articles and information using the terms specific to the strategy. For example, for the individual strategy, the service extracted from articles the name of each individual with a Wikipedia entry. For causal, it filters to retain matched articles with more than 500 words and a minimum threshold of keywords indicative of causal articles – terms such as cause, impact and studies – from sources such as the Economist and the New York Times. And for backing and evidence, it filters to retain articles with a minimum threshold of quantity, measure and value keywords, for example Sterling, population and actual numbers. The full set of terms and other constraints that are used to filter news articles according to each constraint are not reported in this public deliverable.

Furthermore, the service has two different search modes – Strict and Relaxed – that the user can set using an interactive feature of the Sidebar during creative explorations. The default search mode is Strict, which generates queries composed of logical ANDs between all query terms shown at the top of the Sidebar. Relaxed mode generates queries composed of logical ORs between the terms at the top of the Sidebar, and returns a broader set of generated angles.

The threshold-matched and filtered news articles are then passed to the Presser service that is described in the next section.

2.4. Presser: News Aggregation and Content Extraction Service

To enable effective performance INJECT has a pre-processing component, called Presser. Presser uses machine learning, NLP, and other advanced technologies to monitor, crawl, and index hundreds of news sources from across the Web that are available through RSS feeds. In particular, Presser leverages machine learning to process, normalise and analyse news content to make it easier for the Creative search service to use powerful filtering capabilities that will help journalists to source and analyse the most relevant content for their own specific needs. In contrast to the common crawling mechanisms our system is focused on fetching only news articles from the major and minor portals worldwide (multilingual), in order to deliver personalized content to users. Presser has three main components, as shown in Figure 2: Crawlers, News Processor, and News API. Next, we describe the three components in more detail.

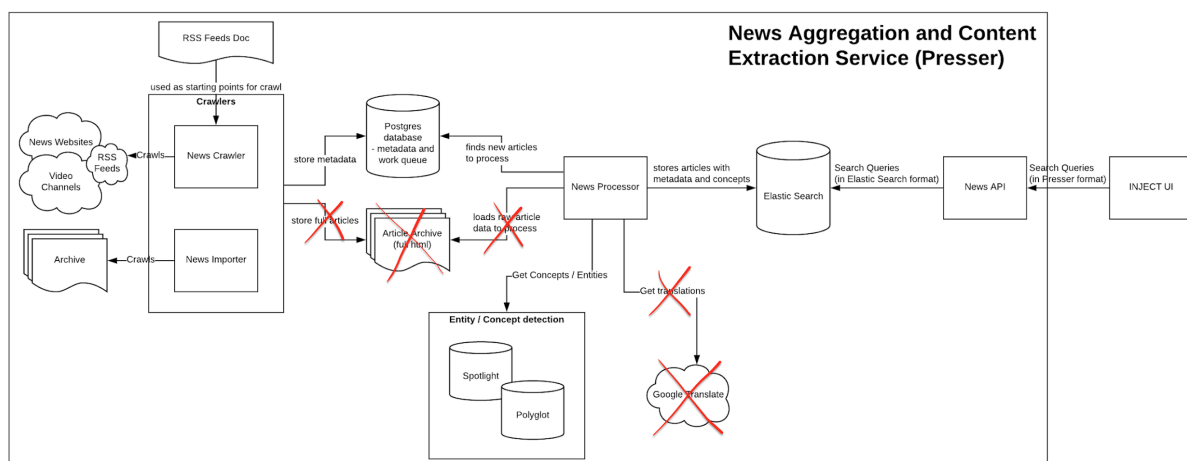


Figure 2. The architecture of INJECT's Presser component

2.5. Crawlers

Presser's crawling mechanism is divided in two sub-components: 1) News Crawler that fetches news articles from major and minor news sources worldwide (multilingual), in order to deliver personalised content to users; 2) News Importer that utilises local digital news archives to generate creativity support for discovering new angles on news stories stored in private archives. Next, we describe both crawlers in more detail.

2.6. News Crawler

Presser's News Crawler is responsible for accessing RSS feeds and downloading new news articles. The News Crawler component loads the RSS feeds document and scrapes all RSS feed URLs. To make this resource intensive task more efficient, Presser learns to predict the

rate that an RSS is publishing articles, based both on the static features and the complete posting history.

The architecture of the News Crawler consists of multiple sub-systems which are assigned with specific roles in order to achieve scalability. The basic parts of the system are (a) the centralised database, (b) the crawler's controller and (c) the terminals that execute the fetching and analysis.

The database is used for storing permanent information such as links to RSS and news articles. Each RSS stored in the database is followed by a number of indicators which change dynamically according to the rate that the RSS publishes new articles. Additionally, the database stores information concerning the articles that are fetched utilising the RSS feeds like the title, the URL, the language, the category and any other information that the crawler determines to be important.

The controller is used in order to organise the whole procedure and is assigned with two major tasks. The first is to load the RSS feeds document and the second is the job assignment and checking of the terminals. We maintain a document that contains details of all RSS feeds that are currently used in INJECT¹. The controller feeds the terminals with URLs from which to download information. At the same time, the controller examines the outputs of the terminals' analysis and stores any information to the database.

The News Crawler utilises a central Postgres database for permanent storage of data. The crawler runs every few minutes and the controller is responsible to specify which RSS feeds have to be revisited. This decision is made by estimating the number of new articles that are expected to be published from the last time that the RSS was retrieved. To achieve this, the controller utilises the hourly posting rates and the last retrieval time of each news source.

Finally, the terminals fetch and analyse the RSS feeds that were generated from the controller, and download every new article that is found. At this stage, any navigation links, ads and more undesired content are automatically removed from an article and the main body of an article including embedded media such as links, images, videos etc is extracted. Results are stored as metadata in a PostgreSQL database as well as on hard disk as raw article data including the complete article text as string as well as the URL that links to the original article. This data is kept there for the next processing phase. The distinction between controller and terminals which utilises a parallel architecture allows the News Crawler to broaden the overall bandwidth of the system. This means that the resource constraints are lowered and the performance of the News Crawler is increased.

As shown in Figure 2, the latest version of Presser's News Crawler also processes metadata from video feeds such as YouTube channels that are parsed and fetched in the same way as news articles. To achieve this, the News Crawler converts video channel URLs into RSS feeds that are fetched and scraped in order to extract the metadata as well as the video link.

¹ The remaining fields in the document are added to the scraping results so the original feed, region, etc. are stored for each article.

2.7. News Processor

Presser's News Processor is responsible for processing raw news articles in 2 steps - Named Entity Recognition and Natural Language Parsing.

The first detects and extracts named entities from a news article. Named Entity Recognition is a critical Information Extraction task, as it identifies which snippets in a text are mentions of entities in the real world. Strictly speaking, a named entity is a group of consecutive words found in a sentence, and representing concepts such as people, locations, organisations, objects, etc. For instance, in the sentence “Angela Merkel is a German politician serving as Chancellor of Germany since 2005 and leader of the centre-right Christian Democratic Union (CDU) since 2000”, “Angela Merkel”, “Germany” and “Christian Democratic Union (CDU)” are person, location and organisation entities, respectively. All of these entities can provide interesting and important information on a piece of text and improve the overall analysis of an article. The respective entity or tag doesn't need to appear in the exact wording in the text. An article may, for example, be tagged with Greece on the basis of containing the word Greece or Hellenic Republic. To achieve this, Presser's News Processor utilises two Named Entity Recognition services, DBpedia Spotlight (DBpedia Spotlight 2017) and Polyglot (Polyglot Documentation 2017). The former is a tool for annotating mentions of DBpedia resources in text, providing a solution for linking unstructured information sources to the Linking Open Data cloud through DBpedia. DBpedia Spotlight performs named-entity extraction, including entity detection and disambiguation with adjustable precision and recall. DBpedia Spotlight allows users to configure the annotations to their specific needs through the DBpedia Ontology (DBpedia 2017) and quality measures such as prominence, topical pertinence, contextual ambiguity, and disambiguation confidence. The latter is a natural language pipeline that supports multilingual applications and supports tokenisation in 165 languages, language detection in 196 languages, named entity recognition in 40 languages, part of speech tagging in 16 languages, sentiment analysis in 136 languages, word embeddings in 137 languages, morphological analysis in 135 languages, and transliteration in 69 languages. The reason we use both of these services is that DBpedia Spotlight is only available for major languages such as English, German, French and Dutch. Polyglot on the other hand applies its algorithm to many other “minor” languages such as Norwegian and Greek. In INJECT, we currently use the named entity recognition feature for the Norwegian language and intend to use Polyglot for other languages such as Greek.

The second uses advanced natural language parsing techniques to identify key noun and verb phrases that are used to improve Presser's search capability in the News API component as described. To achieve this, the Natural Language Parsing service parses the original article's text to extract (1) objects in the forms of noun phrases; and (2) activities in the form of verb phrases determining the active verbs. To do this, the service's algorithm splits the text into sentences and applies a part-of-speech tagging process to mark up words in each sentence as belonging to each lexical, part-of-speech category using information about both its definition and its context in the text. The algorithm then applies a natural language processing technique called shallow parsing that was developed to generate machine understanding of the structure of a sentence without parsing it fully into a parsed tree form. The output of this shallow-parsing process is a division of the text's sentences into a series of words that, together, constitute a grammatical unit. Finally, to select candidate objects and

actions from these grammatical units, the algorithm applies lexical extraction heuristics on a syntax structure rule-tagged sentence to extract content words relevant for the generation of one or more objects.

Once a news article has been processed, the service then uploads the processed results into an external Elasticsearch Cluster for post-search activities. From m6 to m12, this component was modified in response to legal guidance about the use of digital news information in the INJECT services. The original version of the news processor service stored the following information types persistently at the data layer about an individual article that it processes:

1. The set of entities, which our software extracts from the read article, e.g. names, place names, key events;
2. The set of the nouns, verbs and adjectives from the read article, with meanings attached, using our software, and;
3. The article text and URL, and string.

The information types A and B were stored persistently to enable INJECT's creative search algorithms to function are required – these types were deemed to be mandatory. Moreover, each of these information types were generated by INJECT's processing of each news article, and as a result could be viewed as new intellectual property that was generated by INJECT. In contrast, the original design decision to store the original article's text (C) was taken to improve INJECT's performance, i.e. to avoid the need for the engine to re-retrieve the news article from its source URL at run-time. Moreover, the original article's text can be used to re-index the complete database in order to perform any changes made to Presser's algorithm, which would not be possible to do without the original text. However, legal advice to the INJECT consortium was that such persistent storage of news articles represented a potential breach of the intellectual property of the author journalist and news outlet. Therefore, the News Processor algorithm was modified to store only A and B in INJECT's persistent data layer.

2.8. News API

Presser's News API allows easy access to the results in Elasticsearch by exposing key search capabilities that are used by the creative search service. Elasticsearch is a highly scalable open source search engine with a REST API. Elasticsearch can be used to search all kinds of documents. It provides scalable search, has near real-time search, and supports multi-tenancy. The News API allows users to retrieve JSON representations of the news articles that were parsed in previous stages.

2.9. News Augmentation Service

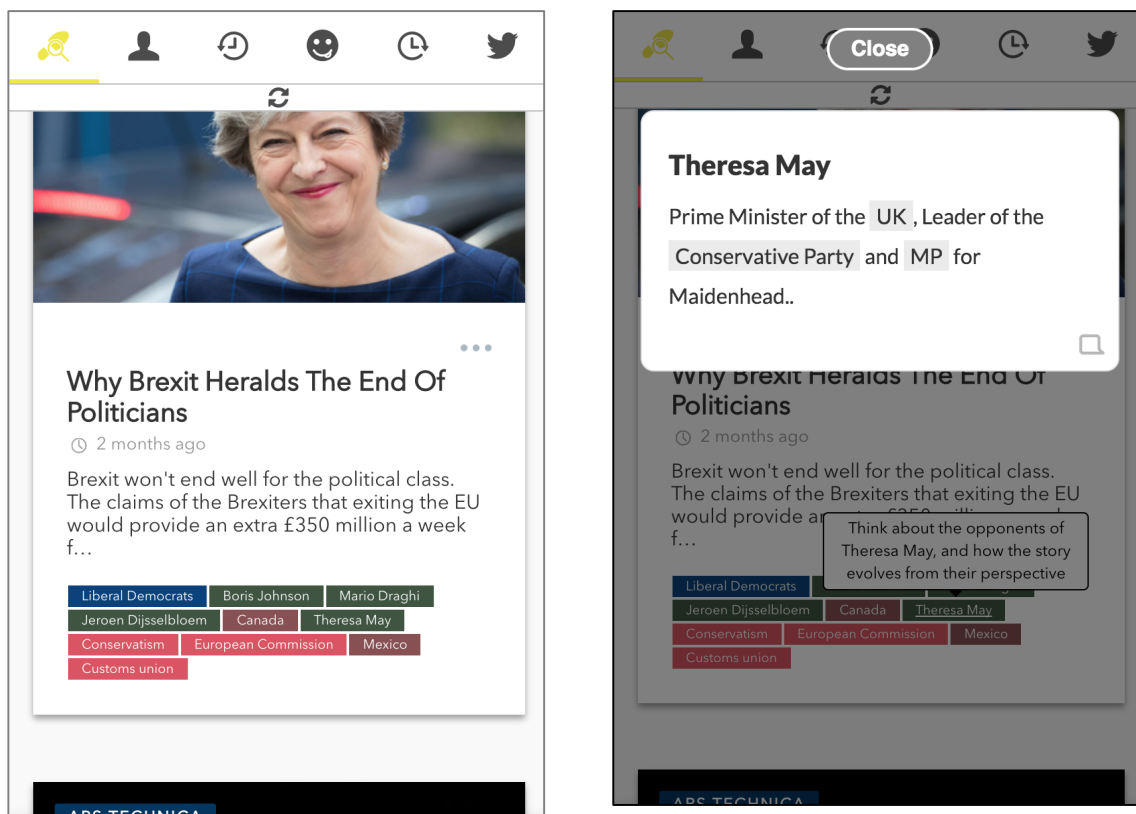
The News Augmentation Service takes advantage of the Explain framework and technology which are used to organise news information more productively than with standard journalism formats such as articles. Explain is designed to make the process of exploring and investigating information more efficient, easier to consume, and more organised.

Explain is a system of pop-up fact cards of content that give more information on a topic without the reader leaving the page. Each card is short, around the length of a tweet, and

focuses on answering a single question, such as defining a word or explaining a concept. However, cards are connected by related topic to other cards, and the interface allows for extremely quick navigation through cards.

Cards can be thought of as the ‘atoms’ of news. Card types include profiles, quotes, charts, facts, definitions, statistics, data, maps, timelines, lists, videos, audio, and other embeddable content. Words and phrases in a card are linked to other cards, in a manner similar to hyperlinks. Links are always to cards that go into more detail on that word or phrase. Therefore, the reader can ‘choose their own journey’ through cards as they explore exactly the information that they need to fully understand a topic.

Users of INJECT can interact with the News Augmentation Service by clicking on topics at the bottom of articles listed in the search results. This brings up a card, which, in turn, links to other cards, as depicted in Figure 3.



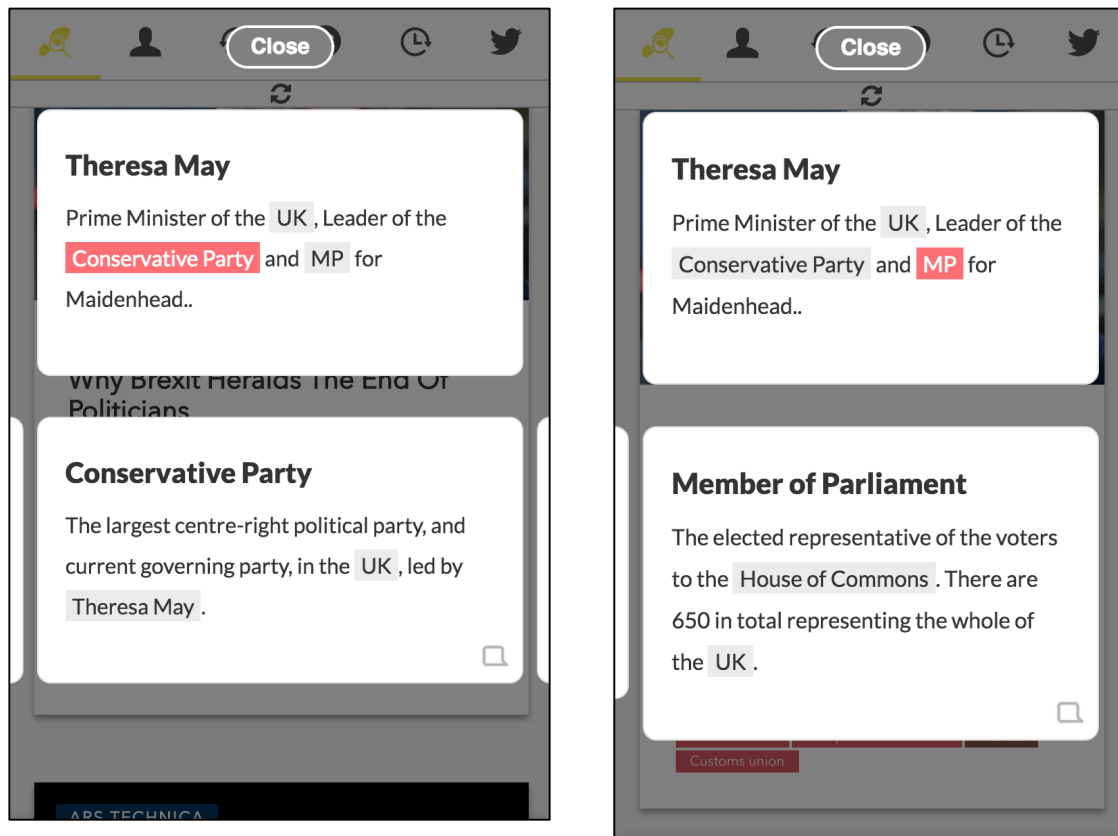


Figure 3: Examples of INJECT concept cards that build on the Explain technology and service

The News Augmentation Service includes an API to request cards from keywords, which enables the INJECT backbone to request cards from the News Augmentation Service plugin, based only on a key phrase provided.

The automated card generation engine creates and displays cards from key phrases ‘on the fly’, upon request, even when they don’t already exist in the News Augmentation Service database. Upon request, the engine checks whether a card already exists, and if not it generates an appropriate card using external sources such as Wikipedia; the engine then returns the new or existing card to the INJECT backend components.

Asynchronous, dynamic formatting of links is supported. The technology leverages IBM’s Watson API to extract entities from the content. Watson provides Linked Data information on these entities, which is used to request the card content for those cards from sources such as Wikipedia.

For example, when the Theresa May card (shown above) is generated, the content is sent to IBM’s Watson API. Watson returns data confirming that it found the entities “UK”, “Conservative Party” and “MP” in the content. For each of these it includes a unique Linked Data URLs, such as a DBpedia URL. Explain uses these URLs to request content for each of those ‘subcards’, so that the Theresa May card can link to them, as shown.

In addition to generating cards automatically, the latest version of Explain allows cards to be created manually by journalists using the tool. Cards that have been generated can also be edited and updated by journalists.

This means that journalists can correct cards that are out of date due to breaking news, add extra information or notes, and create new, niche cards that their particular publication would find useful. This feature is particularly useful for local newspapers, who report on regional topics that may not have a Wikipedia page, and for newspapers in a language that Wikipedia has less coverage in.

Cards created, edited or indeed even just generated are available only within the organisation that produced them. This means that journalists can safely store their own research within the system without risk of losing an exclusive. A potential future use-case of the card system could be as a digital version of a reporter's notebook, building a smart web of all the interconnected elements in a story or investigation.

The latest version of Explaain includes a "Sources" field in each card. This gives information on how the content was created. For example, if it came from Wikipedia (via DBpedia), and was subsequently edited by a journalist at Der Spiegel, then at the bottom of the card it will say: "Sources: **Wikipedia**; Edited By: **Der Spiegel journalist**".

The Explaain technology has been upgraded significantly throughout the project to ensure that it is ready to cope with the demands of having many users and respond quickly to their interactions.

Having originally been hosted on the service Heroku, the cards now run from static files on the service Firebase, which is owned by Google. The technology has been written to run almost entirely client-side. Combined, these decisions mean that it is prepared to handle thousands of concurrent users without any noticeable drop in speed, because each user's device takes the load, rather than our server.

Explaain uses the software-as-a-service Algolia to provide powerful realtime search, and to store and index the card content. This means that it doesn't have to constantly rely on APIs such as DBpedia when providing cards that have been requested before, as once generated they are indexed in Algolia for future use.

Explaain is written in the modern framework Vue JS, which is well supported and has a fast-growing community surrounding it. In short, the technology is built to last.

2.10. Creative Sparks

A creative sparks service generated the pop-up sparks for each retrieved article and entity extracted from each article. An individual creative spark associated 1 extracted entity or news article to 1 creative instruction. The sets of instructions had been manually generated from websites and blogs that teach journalists to uncover new angles on stories, and 1 set of instructions had been generated for each of the 4 types of entity that were extracted – *people*, *events*, *places* and *organizations*. Examples included: *Unpick what the relevance of [Place], as opposed to somewhere else, might have on the story* and; *Explore the history and background of [Organization] to obtain a new perspective on your story*. A total of 34 creative instructions were implemented in INJECT. One set of instructions was also generated for news articles retrieved with each of the 6 creative strategies – *people*, *causal*, *quirky*, *quantifiable*, *ramifications* and *data visualizations*, and a total of 41 such instructions were implemented in INJECT. Examples included: *Use data types reported in this story, to generate a new angle*, and: *Make your angle more similar to the causal angle in this story*. When in-

voked, the service used a randomizing function to attribute one instruction string to one entity string of the same type, and concatenated the strings to generate the spark. So, for the extracted organization *Open Europe*, INJECT might have presented a spark such as: *Explore the history and background of Open Europe to obtain a new perspective on your story.*

2.11. Multi-language support

INJECT's target markets by m12 of the project were newsrooms and journalists working in the United Kingdom and Ireland, Norway, Netherlands, Germany, Austria and Switzerland. Unsurprisingly, most journalists in these countries worked in the native language of those countries – English, Norwegian, Dutch and German. Therefore, for the m12 version of the INJECT tool, new features were introduced to enable a user to toggle between and work in these 4 user languages. More specifically, this required new INJECT capabilities to:

- Present all user interactions and creativity support in the native language of the user. This use of native language was deemed to be particularly important for digital creativity support, which studies have shown seamless and impose no additional cognitive load or impediment on the journalist;
- Retrieve and explore digital sources in which the news was written in these native languages, e.g. new digital sources in the data layer of news in Norwegian, Dutch and German, as well as English, and;
- Capabilities to parameterise and extend some of INJECT's creative search algorithms to retrieve and filter news, and extract entities in all 4 of the native languages.

These new capabilities enabled a journalist to use INJECT to discover new angles on news stories, for example, from Dutch news sources with creativity support delivered in the Norwegian language. How a journalist could toggle between the different languages is shown as part of the user interaction layer in Section 3. Furthermore, in the development of this version of INJECT's services, the team experimented with use of Google translation services in the Presser tool, to generate automatically the translations of the indexed terms of each news article, so that a journalist writing and expressing topics of interest in, for example, English, would receive articles written in another language, for example German. INJECT experimentation with the Google translation services revealed the technical feasibility and potential for this capability, however the cost of using an online translation service proved inhibitive in the context of the current INJECT business models, and the capability was discontinued.

The remainder of this section describes how the INJECT tool was extended to provide multi-language support.

To generate the native language content in Norwegian, Dutch and German, native speakers in these languages from within and outside of the INJECT consortium were requested to translate all strings that were presented as part of the user interaction layer and all strings that were manipulated by INJECT's creative discovery algorithms. In addition, to enable INJECT to be demonstrated in other European markets, all user interactions were also translated into the French and Italian languages by native speakers.

3. Data Layer

INJECT's data layer is composed of news and news-related information from public and private news websites, from social media, and from a bespoke digital database. This news information is primarily in text format, but is also in image and video format. Each of these types of source in the data layer is reported in turn.

Public news sources

INJECT parses and semantically tags news stories from 295 different online digital news sources with public RSS feeds as part of the INJECT data layer. The data format used is PostgreSQL database while the processed/parsed results are stored into an external Elastic Search Cluster for later searching. The news sources currently indexed by the INJECT data layer are shown, by country source, in Table 1.

Germany: Aachener Zeitung; Apotheken Umschau; ARD; ARD.de Börse; Augsburgener Allgemeine; Bayerischer Rundfunk; Bento; Berliner Morgenpost; Brigitte; corrective; Darmstädter Echo; Der Freitag; Der Postillon; Der Spiegel; Der Spiegel International; Destatis - Statistisches Bundesamt; Deutsche Apothekerzeitung; Deutsche Ärztezeitung; Deutsche Verkehrszeitung (DVZ); Deutsche Welle; Deutscher Bundestag - Analysen des Wissenschaftlichen Dienstes; Deutscher Bundestag – Plenarprotokolle; Deutsches Ärzteblatt; Deutschlandfunk; Die Glocke; Die Welt; Die Zeit; Eltern; Evangelisch; Frankfurter Allgemeine Zeitung; Frankfurter Rundschau; Fränkischer Tag; Fudder; FUNK; Golem; Gründerszene; Hamburger Abendblatt; Handelsblatt; Hannoversche Allgemeine Zeitung; Heise News; Hessischer Rundfunk; Horizont; Impulse; Informationsdienst Wissenschaft; Junge Welt; Kieler Nachrichten; Kleine Anfragen – Parlamentsanfragen; Kölner Stadtanzeiger; Lebensmittelzeitung; Leipziger Volkszeitung; Lübecker Nachrichten; Mainpost; Manager Magazin; Mannheimer Morgen; Märkische Allgemeine; Meedia; Mitteldeutscher Rundfunk; Münchner Merkur; Münsterländische Volkszeitung; Münstersche Zeitung; n-tv; Netzpolitik; Neue Osnabrücker Zeitung; Norddeutscher Rundfunk; Nordwest Zeitung; Ostsee-Zeitung; Passauer Neue Presse; Presseportal Blaulicht; Rhein-Zeitung; Rheinische Post; Rundfunk Berlin-Brandenburg; Sächsische Zeitung; Schwäbische Zeitung; shz; Spektrum.de; Spiegel Online; stern; Stiftung Warentest; Stuttgarter Zeitung; Süddeutsche Zeitung; Südwestrundfunk; T-Online; t3n; tagesschau.de; Tagesspiegel Berlin; taz - die Tageszeitung; Telepolis; Thüringer Allgemeine; top agrar; Übermedien; VICE Germany; W&V; Weser Kurier; Westdeutscher Rundfunk; Westfälische Nachrichten; Wirtschaftswoche; Wolfsburger Allgemeine

United Kingdom: BBC; BBC Mundo; Belfast Telegraph; Birmingham Mail; Birmingham Post; Daily Express; Daily Mail; Daily Record; Daily Star; Farming Life; FT; Herald Scotland; Indy100; London Evening Standard; Manchester Evening News; New Statesman; Private Eye; Quartz; Reuters News; Sky News; South Wales Evening Post; Telegraph; The Economist; The Guardian; The Huffington Post; The Independent; The Metro; The Mirror; The Register Business; The Register Headlines; The Scotsman; The Sun; The Times; Times Education Supplement; Yorkshire Evening Post; Yorkshire Post. **US:** ABC News; Bloomberg; BuzzFeed; Chicago Tribune; CNN; CNN Money; Forbes; Fox News; Herald Tribune [International NY Times]; Los Angeles Times; Newsmax Media; Newsweek; The Daily Beast; The Huffington Post; The New York times; The Onion; The Wall Street Journal; The Washington Post; U.S. News and World Report; USA: 9 to 5 Google; 9 to 5 Mac; APPLE Press Releases; Ars Technica; CNET News; Daring Fireball; Hacker News; Mashable; NYTIMES Tech; Technology Review; The next Web; The Verge RSS; Venture Beat; WIRED Busiess; WIRED Culure; WIRED Science.

Netherlands: 360 magazine; Adformatie; Adformmatie; Algemeen Dagblad; Binnenlands bestuur; Bright TV; CBS; CNV; COC; De Groene Amsterdammer; De nieuwe reporter; De Telegraaf; de Volkskrant; Dutch News; Eerste Kamer; Elsevier; Europees Parlement Nieuws; FD; Follow the Money; Het Financieele Dagblad; Het Parool; Intermediair; Joop; Libelle; Metro; Nederlands Dagblad; Nederlandse Staatscourant; NOS; Novib; NRC Handelsblad; NU; Reformatorisch Dagblad; RTL Nieuws; SCP; TPO; TPO; Trouw; Vice; Villameida; WNL TV; Zondag met Lubach; Zorg + Welzij.

<p>Norway: Aftenposten; Bergens Avisen; Bergens Tidende; Dagbladet; Dagens Næringsliv; DIGI.NO; Drammens Tidende; E24; Folkehelseinstituttet; Forskning.no; Klassekampen; Lovdata; Nationen; NRK; Statistisk sentralbyrå; Stavanger Aftenblad; Stortinget; TV2; Verdens Gang (VG).</p>
<p>Australia: Herald Sun; The Brisbane Times; The Courier-Mail; The Daily Telegraph (Australia); The Sydney Morning Herald. Austria: Friedl News; FutureZone.at; ORF - Österreichischer Rundfunk. Belgium: New Europe. Bhutan: Bhutan Observer. Canada: The Globe and Mail; Toronto Star. Cyprus: North Cyprus News. Czech Republic: Prague Daily Monitor. Denmark: Copenhagen Post. Egypt: Daily News Egypt. EU: EuroFora. France: BFMTV; Connexion Newspaper; FranceInfo; L'OBS; Le Express; Le Figaro; Le Monde; Le Monde Diplomatique; Le Parisien; Le Point; Les Echos. Greece: Ekathimerini; Eleftheria; in.gr; Naftemporiki; News247; Newsbeast; Newsbomb; Parapolitika; Proto Thema; Real; Skai; To Pontiki; To Vima; Zougla. Hong Kong: South China Morning Post. Iceland: News of Iceland. India: Business Standard; Economic Times; Financial Express; Live Mint [INDIA]; The Indian Express; The Ken. Indonesia: Jakarta Globe. Ireland: Irish Daily Star; Irish Examiner; Irish Independent; The Irish Sun; The Irish Times. Kenya: Daily Nation. Namibia: The Namibian. Nigeria: The Punch; Vanguard. New Zealand: Dominion Post; New Zealand Herald. Poland: Krakow Post. Portugal: Portugal Resident. Qatar: Al Jazeera. Scotland: Herald Scotland. South Africa: Business Day Live; Independent Newspaper; Mail and Guardian; News24. Spain: Cinco Dias; El Mundo; El Pais. Sweden: The Local Newspaper. Switzerland: Neue Züricher Zeitung; SRF; Tagesanzeiger. Thailand: Bangkok Post.</p>

Table 1. Public news sources that contribute to INJECT's data layer, by country of news source

The current set of news sources was curated by an informal editorial team from INJECT, which used simple criteria to select news sources by country. These criteria included the inclusions of newspapers that represented a range political and social perspectives, to reduce the risk of echo chambers in the creative guidance that INJECT generates, and the exclusion of newspapers that encouraged or promoted nationalist agendas that are in conflict with the founding principles of the European Union.

3.1. Private News Archives and the News Importer Component

One requested INJECT capability not foreseen at the beginning of the project was to use newspapers' own digital archives to generate creativity support for discovering EU new angles on news stories. Whereas most of INJECT's digital resources at the data layer are extracted from national and international news outlets, the 3 regional Norwegian newspapers in the consortium highlighted that most of their news is local, and would best be supported with local digital news archives. Furthermore, all 3 local newspapers share a digital news archive, part of which was made available to the consortium for experimentation. A pilot version – 6 months of news reported by all 3 newspapers from January to June 2017 – was extracted from the original archive and made available as JSON files and integrated into Presser's Crawler component.

Unlike the News Crawler, the News Importer consists of only two parts: (a) the centralised archive database, and (b) the crawler's importer that processes the data for each archive.

3.2. Public Social media and video sources

INJECT uses Twitter in three ways. First, it searches for tweets posted by specific individuals, second it searches for tweets relevant to a user query and last it searches by location. Starting with the individuals, we start with the individual identified in the individuals view, we search for the corresponding Twitter account (handle) and we test if the account that we find is verified. If we find a verified account then we get the latest tweets that this person

wrote in the past four days. All tweets are saved in a JSON file (see an example JSON excerpt in Figure 4).

```
{
  "name": "George_Osborne",
  "tweet": ["Today's @EveningStandard - travel chaos as train guards call in sick & full TV guide & our #helpahungrychild Xmas campaign re:
  "And here's @AdamStooni @EveningStandard take on the Green resignation https://t.co/NiMofAaNva",
  "Here's our editorial @EveningStandard on the causes and consequences of Damian Green's resignation https://t.co/RT7e11ktka",
  "Today's @EveningStandard: the fallout from Damian Green's resignation as Tories turn their fire on the police https://t.co/SID0kSmpX2"],
  "date": ["Fri Dec 22 13:01:03 +0000 2017",
  "Thu Dec 21 12:31:39 +0000 2017",
  "Thu Dec 21 12:29:39 +0000 2017",
  "Thu Dec 21 11:55:43 +0000 2017"],
  "favorite_count": [23,
  122,
  10,
  15],
  "retweet_count": [9,
  52,
  11,
  15],
  "tweet_score": [38.95,
  253.65,
  30.4,
  46.55],
  "average_score": 92.3875,
  "sum_score": 369.55
},
{
  "name": "BorisJohnson",
  "tweet": ["Met brave campaigners for LGBT, media freedoms & human rights in Russia. A free, vibrant civil society is a vital component of any
  "Visited Bolshoy Moskvoretsky Bridge to lay flowers at the scene of Russian opposition leader Boris Nemtsov's murder. Justice must be served. #
  "Honoured to pay my respects at Tomb of the Unknown Soldier at the Kremlin Wall, Moscow, recognising Russia's military sacrifice in the Second Wo
  "Frank talks w/ Russian foreign minister Lavrov in Moscow. We recognised many significant differences but agreed shared interests & global de
  "Arrived in Russia for important talks with FM Lavrov. We may have many differences, but also much to work together on inc Syria, Iran & Nort
  "date": ["Fri Dec 22 16:21:53 +0000 2017",
  "Fri Dec 22 15:30:16 +0000 2017",
```

Figure 4. Example excerpt of a JSON file

Next, we score the retrieved tweets. Every tweet gets its own score. The score provides an indication of the impact of each tweet for the particular individual. The scoring algorithm is shown in Figure 5.

Start: Given a twitter handle or a query

```

Items<- Initialise
for each tweet in items do
  NoTerms ← ∅
  NoJourn ← ∅
  if tweet ≠ ∅ then
    // Initialise score with # of likes
    score ← numberOfLikes(tweet)
    //Increase score based on # of retweets.
    score ← score + numberOfRetweet(tweet)
    if journalistExist(tweet) then
      score ← score + score*0.2
      NoJourn ← 1
    if queryTermExist(tweet) then
      score ← score + score*0.1
      NoTerms ← 1
    if NoJourn = ∅ and NoTerms = ∅ then
      score ← score - score*0.05
  end if
end for

```

Figure 5. Scoring Algorithm

In a nutshell, the scoring algorithm adds points for each like and retweet a tweet has. We add a bonus score if the tweet mentions any journalist site or term that we find important and we subtract if it does not. Once we compute the scores of all the tweets of an individual, we calculate an average score of his/her tweets. If a tweet is greater than the double of the average tweet score, we retain it and collect it in a new JSON file of impactful tweets. If an individual does not have any impactful tweets, we just retain the tweets with the greatest scores. We repeat the process for all the individuals. An example of results for tweets posted by individuals is shown in Figure 6.

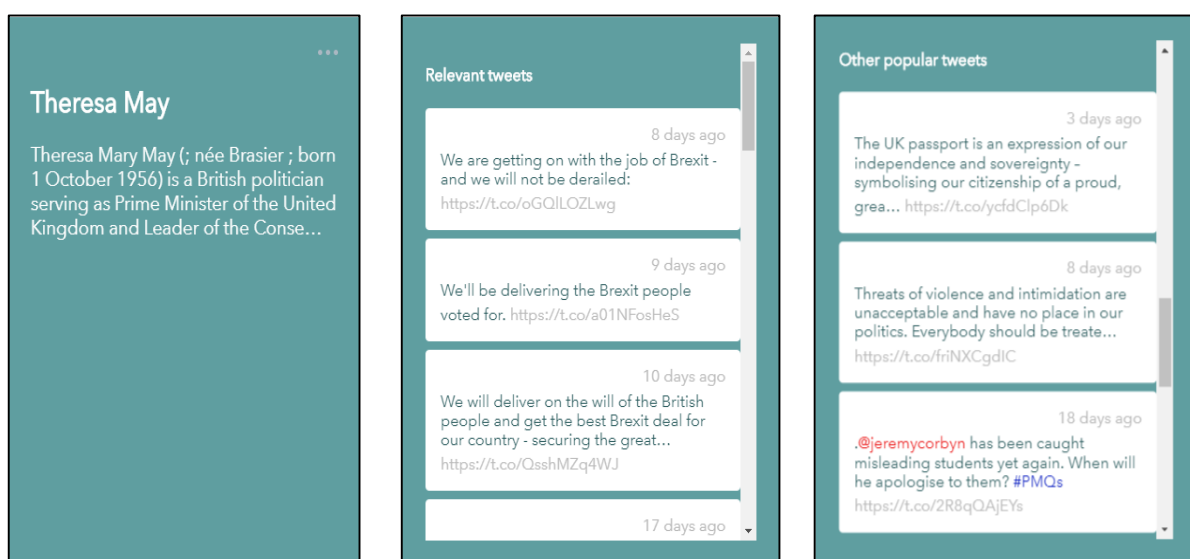


Figure 6. Example of results for tweets posted by individuals

When we query Twitter for tweets relevant to a user query, we follow a similar process as the one described above. We use a query of terms and we get the latest tweets relative to that query term by using the same API. Then we calculate the score of each tweet and create three JSON files. One JSON file contains with all the tweets returned by the Twitter API, a second JSON file contains the most impactful tweets and a third one contains tweets that have hashtag keywords that relate to the query terms.

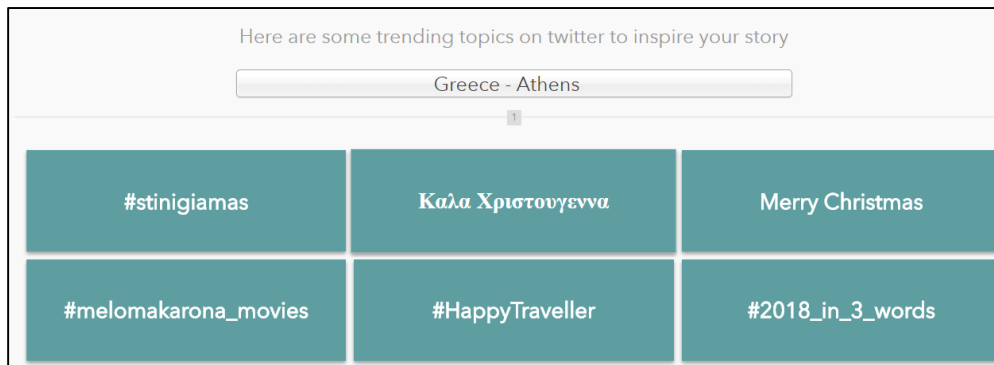


Figure 7. Querying Twitter by location

Last, we query Twitter by location. We identify the current location of the user and we use it to retrieve some of the trendiest hashtag or terms which are related with the chosen location, see Figure 7. By clicking on a hashtag, we re-direct the user to the twitter website that shows related tweets.

3.3. The VJ Movement's Database of Digital Cartoons

To support INJECT's *Quirky* strategy, INJECT also indexes and searches cartoons available via the Cartoon Movement, the Internet's #1 publishing platform for high quality political cartoons and comics journalism. The cartoons can be accessed manually at: <https://www.cartoonmovement.com/cartoon>. The database is contributed to, and therefore updated each day by a network of contributing artists. The database currently contains over 40,000 digital cartoons that form part of INJECT's data layer.

3.4. INJECT's Persistence Component

We developed a persistence component to enable personalisation of INJECT for specific users or groups of users. The first step to develop the persistence component has been the implementation of user authentication capabilities. INJECT takes advantage of three well known platforms and their authentication services, Google, Facebook and LinkedIn. INJECT users have three options, they can sign in by using either their Google, Facebook or LinkedIn account. Through the authentication part of the persistence component, INJECT users are able to sign in and reach their profile section.

The next step was to develop the so called access control list (ACL). ACL is a list of permis-

sions attached to an object. An ACL specifies which users or groups of users are granted access to objects, as well as what operations are allowed on given objects. Each entry in a typical ACL specifies a subject and an operation. Moreover, ACL enables different users to have access to different private archives. ACL ensures that only authorised users have access to newsrooms' private archives. ACL follows a group-based structure. Each group corresponds to a private archive, so members of each group are users that should have access to this archive. The archive administrator is able to authorize a group to have access to a private archive. After a user has been added to a group, he/she has the option to include the private archive as a source for his/her queries.

A further step was to enable journalists to “bookmark” articles that they find using INJECT. The bookmarking functionality allows journalists to keep records of articles that they found interesting and will help them to develop their story. With this functionality, users can select articles that they want to keep in their bookmarks by simply clicking at the specified field in the article view. They can also see in the results of their queries which articles are already bookmarked. Users are able to remove bookmark from an article, too. Last, we developed the display pane, in which users are able to see all their bookmarks and to organize them.

3.5. Pre-Commercial Implementation of the Application Layer

To deliver the levels of performance, reliability and robustness that newsrooms and journalists require from interactive tools such as INJECT, this work package also experimented with different technical set-ups and support services. At the start of month M12, after exploring different configurations and services, INJECT's application services were installed on commercial Heroku servers with monthly payments to ensure adequate performance levels. The current configuration uses the paid services reported in Figure 8, with associated monthly costs.

Commercial service and role	Approximate monthly cost
Hetzner Online (hetzner.com), which runs INJECT on a dedicated root server to perform the very CPU intensive data processing, both at a development as well as live environment. In addition to a live server, it was essential to have a server with the same setup that would 1) enable the technical team to test new functionality without affecting the live version and 2) roll over to the primary server in case of minor or major failure to the live server.	€50
Amazon AWS ElasticSearch Service (aws.amazon.com/ec2), which utilises Amazon Elasticsearch Service's capabilities to handle INJECT's availability, scalability, and security that INJECT production workloads require.	€180
Anturis (anturis.com), which minimises maintenance complexity and monitors INJECT's services and activities such as CPU usage, number of articles collected and (re)indexed, and number of API calls made in a specific timeframe.	€90

Figure 8. Different paid services that support the INJECT application layer

4. INJECT User Interfaces

Journalists interact with the INJECT tool through either a series of plug-ins to existing tools that journalists use or, to ensure wider accessibility, a standalone web application. As described in section 2.2, the existing user interfaces are the:

- Google Docs Add-on interface;
- Content management system interface, e.g. for WordPress and other commercial content management systems;
- Adobe InCopy interface, for use with this editor in use in some newsrooms;
- Stand-alone interface for web browsers.

The INJECT user interfaces were designed and developed to ensure qualities of effective creativity support tools were delivered – qualities such as pain-free discovery and exploration of information, simple ideas generation, and integration with existing work practices. Therefore, INJECT:

- Is integrated into the existing work tools of journalists;
- Switches on or off in these existing work tools with one interaction;
- Minimises the number of interactions to discover and generate ideas for new news stories;
- Plugs into bespoke information and news sources, as journalists require;
- Enables journalists to record the results of INJECT use directly into their regular work tools.

The INJECT user interfaces are also consistent with the wider INJECT product and service brand. The interfaces incorporate the brand logos, colours and style guidelines in order to offer a unitary experience to journalists who use the tool.

This section describes and demonstrates selected interactive capabilities of the INJECT user interfaces, and reports the rationale for these capabilities. As in the Deliverable D1.2 it focuses on the Google Docs Add-on version of the interface, but includes parallel examples from other tools, to demonstrate these other interfaces.

4.1. The Google Docs Interface

To support journalists to undertake the pain-free exploration of information and ideas that was associated with creative thinking, INJECT's digital creativity support was integrated into the Google Docs editor. Google supported a research sidebar of the editor window called the Google Docs Add-on that allowed a user to start searches and browse results. Therefore, the team decided to deliver INJECT's creativity support through this Add-on, referred to simply as the sidebar.

The sidebar was the pre-defined Google Docs component that appeared to the right of the editor. However, the sidebar had a fixed width (300px) and poor performance with server-side libraries, so design challenges were to embed usable and effective creativity support

that called different server-side services into the sidebar. INJECT was implemented with: (A) features to generate candidate search terms directly from text already written in the larger editor window; (B) small icons with mouse hover-box descriptions, to control the sidebar; (C) mouse hover-boxes to present additional information quickly in context; (D) overlays to present more information, and; (E) server-side services that preloaded news information to overcome the sidebar's performance limitations. The designs and implementations of these 5 features were improved incrementally during the journalist-led prototyping, for example by merging control icons and reducing the number of required interactions, so that, for example, the journalist received creativity support in as little as 2 clicks.

Figure 9 shows the final design of the INJECT sidebar. The left side shows the launch setup and 4 functional areas. The first area was the news topic space – a journalist could enter terms or import them directly from the text editor to describe topics of the current news story. The second area was the strategy space – a journalist could invoke different creative search strategies using the described topics. The third, which occupied over 80% of the sidebar, was the information space. A journalist could scroll, mouse hover-over and click retrieved news information and creative sparks to discover and generate new ideas for news angles. The fourth space, fixed at the bottom of the sidebar, was the control space. The journalist could access different tabs to manage tool settings, collaborate with online communities and follow a tutorial. An example of this tutorial is shown on the right side of Figure 9. A small number of INJECT features were also implemented in a separate dialog window that would appear over the editor window. However, due to the limited functionality of that dialog component, most features were implemented in the sidebar, so that journalists could simultaneously use it and write new articles.

When the INJECT Google Docs Add-on is installed, a journalist launches the INJECT tool by selecting the Start INJECT pull-down menu option in Google Docs. The INJECT start page is shown in Figure 9. INJECT opens as the branded sidebar of the text editor tool in Google Docs. From this single sidebar, the journalist can access all INJECT capabilities.

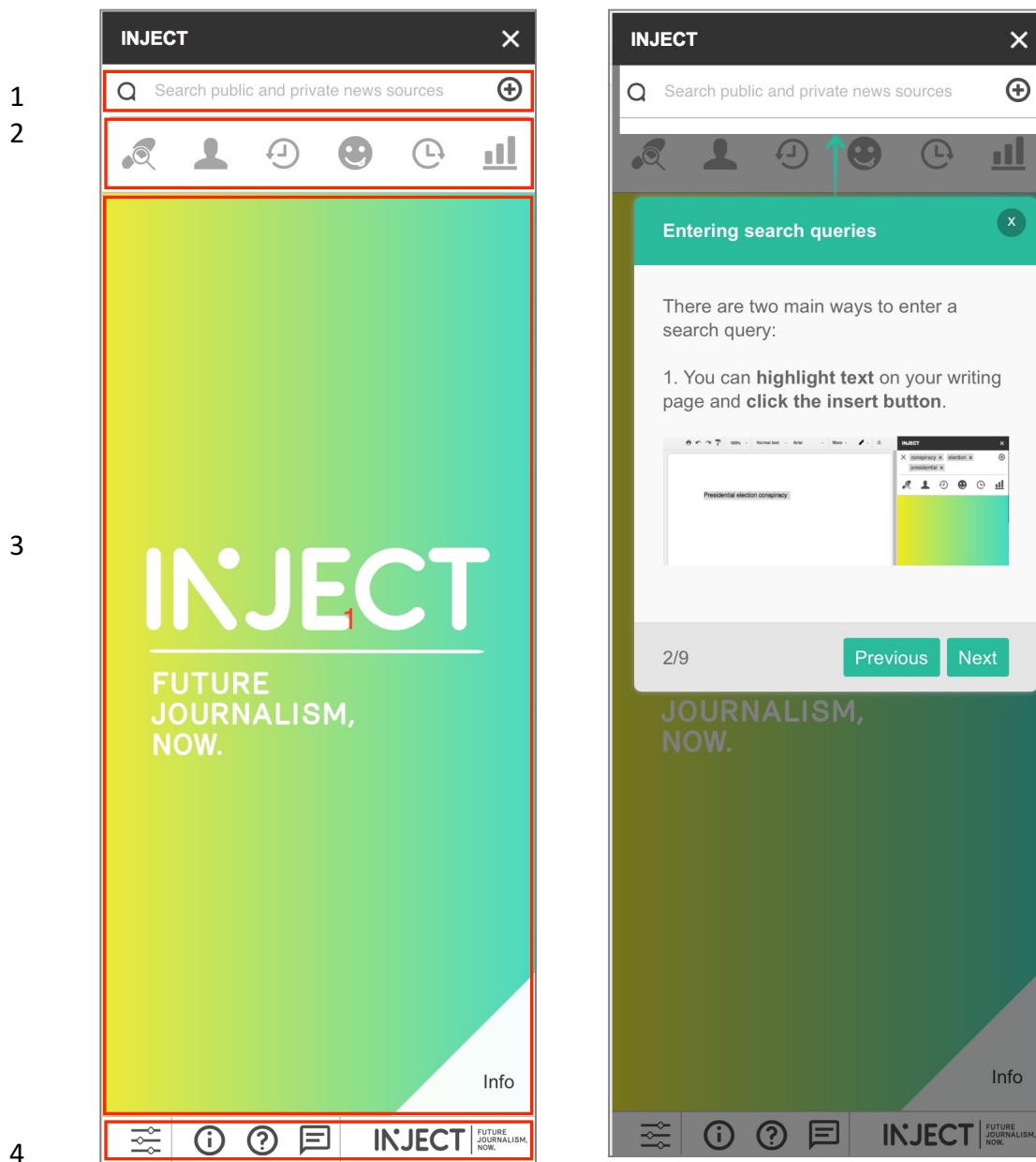


Figure 9 . The INJECT sidebar's 4 functional areas and an example of the tutorial support available to journalists

The journalist is free to work as normal with the text editor, and to invoke and use INJECT as required, at the time and in the work flow that the journalist chooses. Such flexibility was considered to be important for journalists from different backgrounds to take up and use INJECT. The journalist can start to write the new news story using the text editor, s/he can enter keywords directly into INJECT to initiate creative searches, or s/he can select text from her evolving story, to initiate creative searches. INJECT does not impose any restrictions on how and when the journalist can use it, so that it is open to the widest range of journalist work practices.

For example, imagine that the journalist wishes to explore the on-going humanitarian crisis in Yemen. She has already typed some first thoughts into the text editor. She highlights, for

example, the header text in the editor, then clicks the + feature in the right-hand top corner of the sidebar. In response, INJECT parses and extracts keywords from the entered text, and displays these topic terms, as shown in Figure 10.

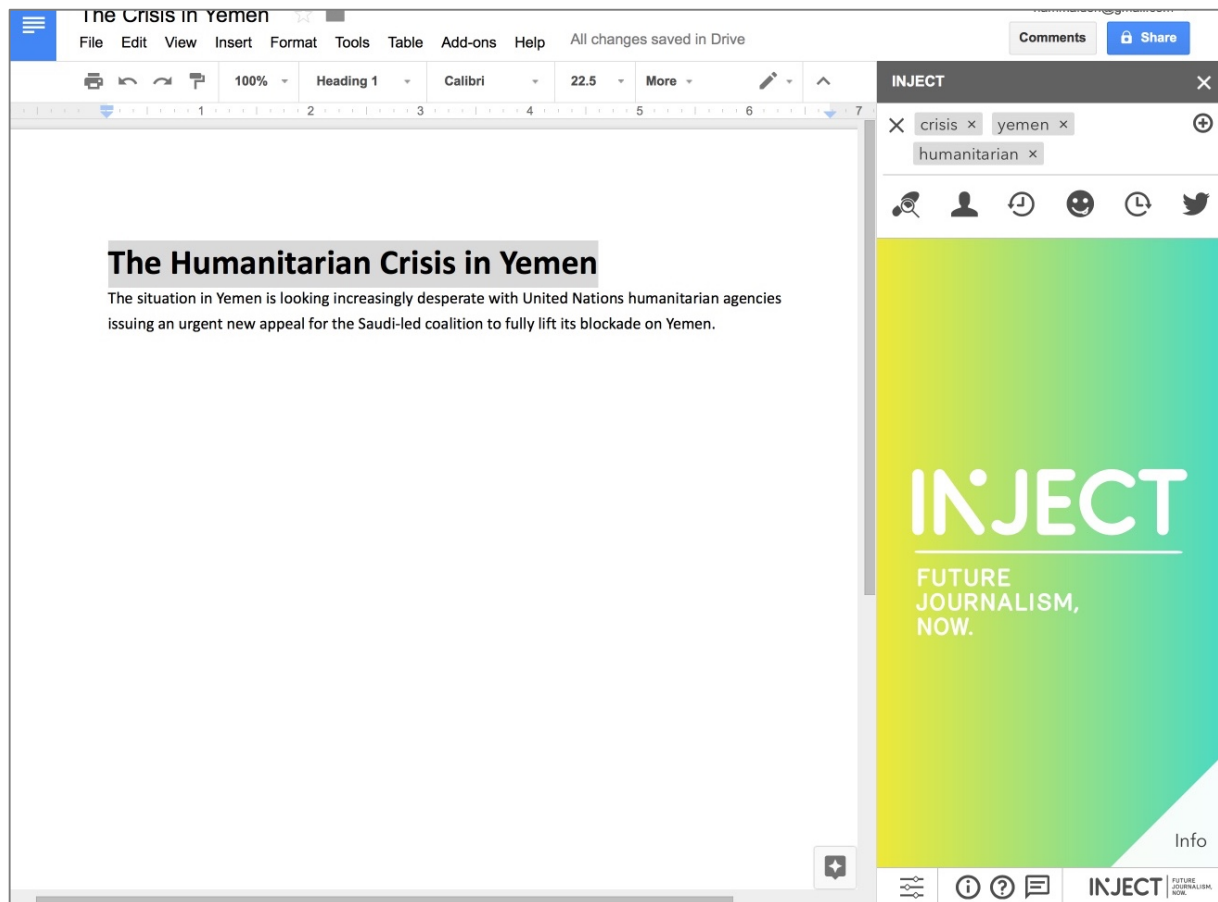


Figure 10. The INJECT sidebar, displaying extracted keywords from highlighted text in the text editor

INJECT allows the journalist to review edit, add and delete each topic term directly in the top of sidebar shown in Figure 10, before launching each creative discovery activity. Moreover, if needed, s/he can use the INJECT toolbox feature accessible via the footer of the INJECT sidebar, to parameterise and control other aspects of INJECT's creative support. The INJECT toolbox feature is shown on the left-hand side of Figure 11.

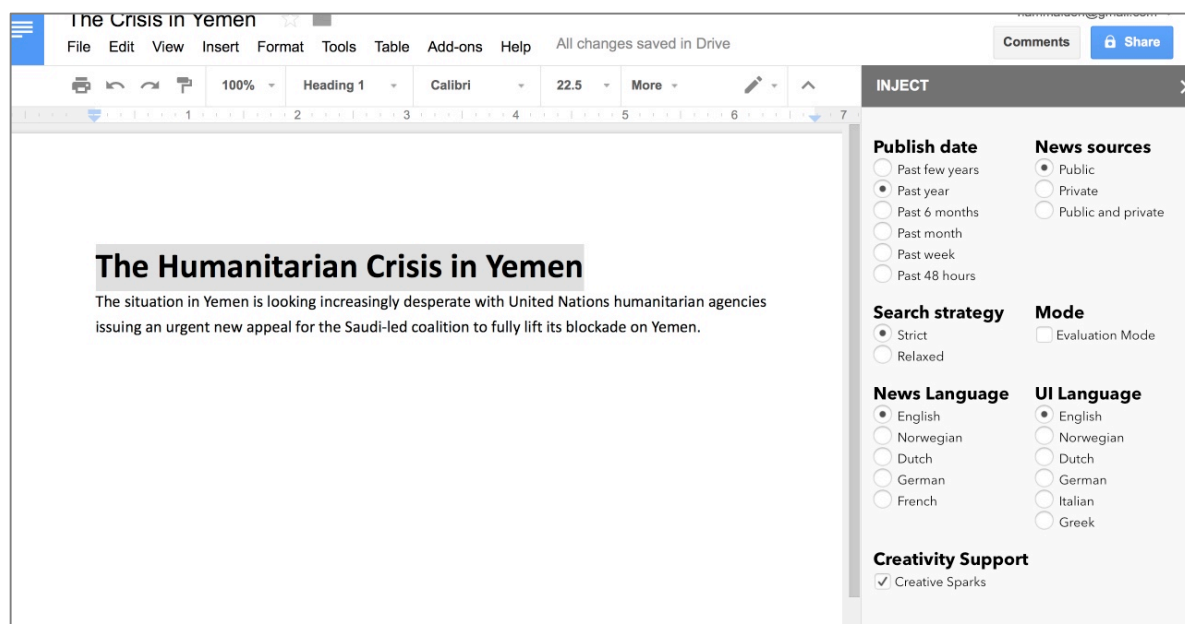


Figure 11. The INJECT toolbox and creative search parameters

The INJECT toolbox offers the journalist features to manage various parameters and hence control use of their use of the INJECT tool. These features and parameters are:

- The publication date(s) of the news information that INJECT uses to generate creative guidance, to provide the journalist some control over generation and hence the creative guidance that the journalist receives;
- The different types of news source that INJECT uses to generate creative guidance – either from the 300 public news sources in the data layer that are reported in Appendix A, or the news organisation’s local news archives, or both;
- The publication language of the of news information that INJECT users to generate creative guidance, to provide journalists with filters to control the language that the creative guidance is generated in. This language control emerged as an important requirement of journalists in different European countries;
- The language of INJECT’s user interface, so that INJECT offers journalists creativity support in the native languages of these journalists and reduces cognitive barriers to creative thinking with the news information and creative sparks. The changing of the language parameter changes the presented language of all interactive features, help support, explanations of the creative search results and the presented creative sparks;
- The use or otherwise of the creativity sparks capabilities, so that the journalist can work both with and without this interactive feature. Feedback on earlier versions of INJECT revealed that more experienced journalists will not use creativity sparks as much as inexperienced journalists, so a feature to remove this capability from journalist sessions was added;
- The use of INJECT in work or evaluation mode. Evaluation mode was added to enable effective in-situ testing of the INJECT tool and services with consortium partners, to collect important testing data about the tool’s use.

At any time, the journalist can return to the main screen, and the INJECT sidebar, to continue working.

To request creative guidance in INJECT, the journalist simply clicks on the 6 creative search icons underneath the sidebar's topic terms in INJECT's strategy space. Each icon was designed to communicate the specific creative news angle. A simple hover-over displays the text name of the creative news angle to the journalist. INJECT presents its creative guidance to the journalist in the sidebar, underneath the icons in INJECT's information space. Figure 12 shows a segment of the INJECT sidebar that presents news articles and individuals retrieved in response to the topic terms. The journalist can request more creative guidance at any time using the *refresh* icon that is presented below the icons. The sidebar presents news articles that INJECT computed have creative potential in chunks of 10, and the journalist can scroll down the sidebar to access the information that is not visible directly in the sidebar.

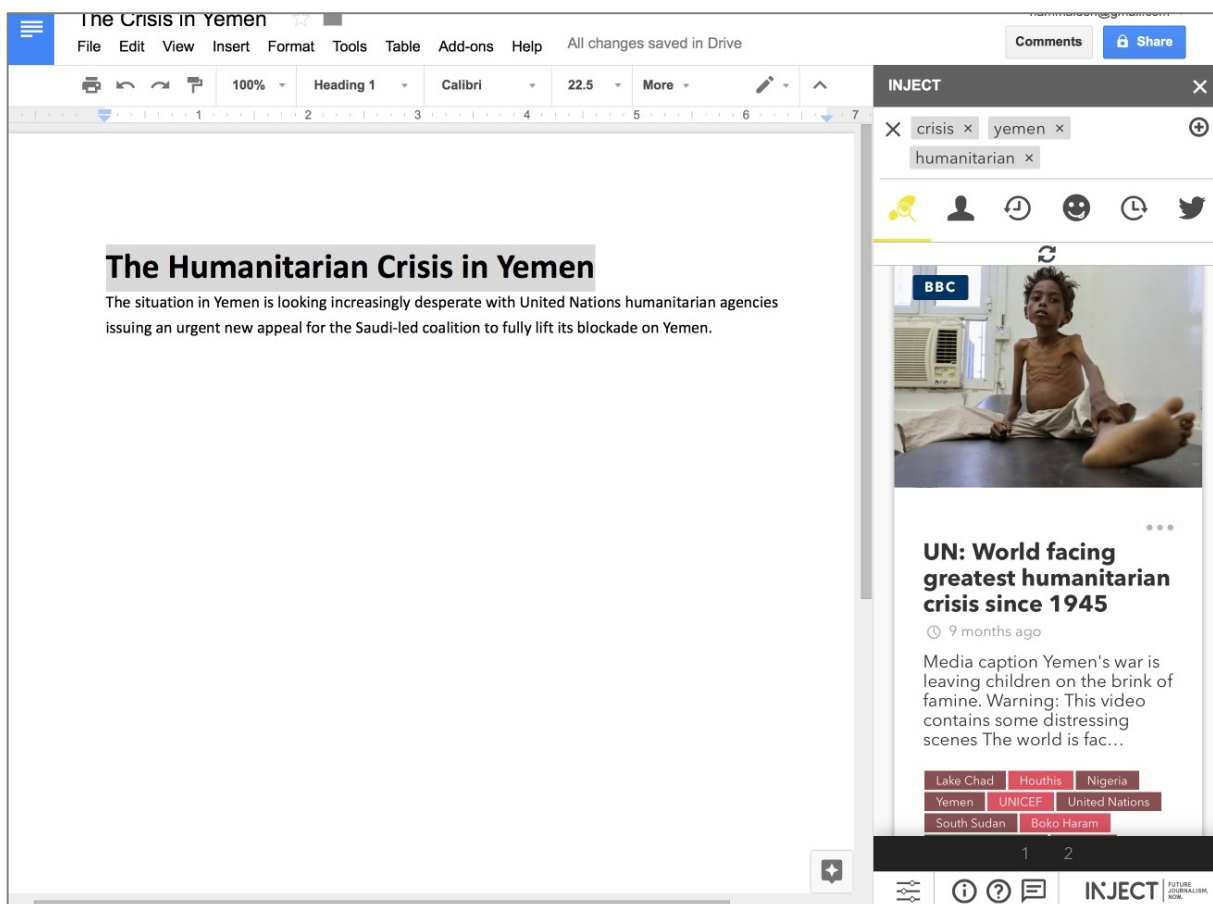


Figure 12. The INJECT sidebar, showing news information retrieved using INJECT's backing and evidence creative strategy, presented in interactive cards

In the sidebar INJECT presents a simple explanation for the news information that it presents to the journalist, as shown in Figure 13. Furthermore, some of the elements of the explanation, such as publication dates and public/private news information sources, are interactive, and can be clicked on to change the elements. This interactivity was added to the explanation to encourage the journalist to play and explore with different news sources and information, as part of the creative discovery process.

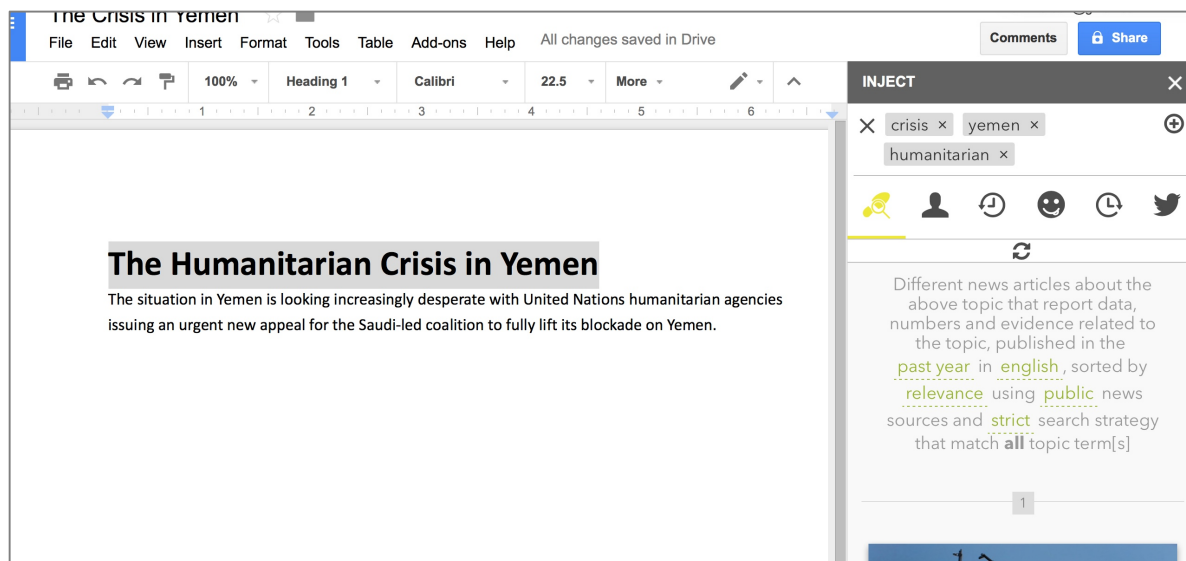


Figure 13. INJECT’s interactive explanation text presented at the top of the sidebar. The journalist can click on the green text to change INJECT parameters and launch new requests for creative guidance

Furthermore, at any time, the journalist can request INJECT to provide a more in-depth explanation why it is presenting the news information that it is. INJECT generates adapted explanation texts, which it presents to the journalist in the sidebar space, as shown in Figure 14.

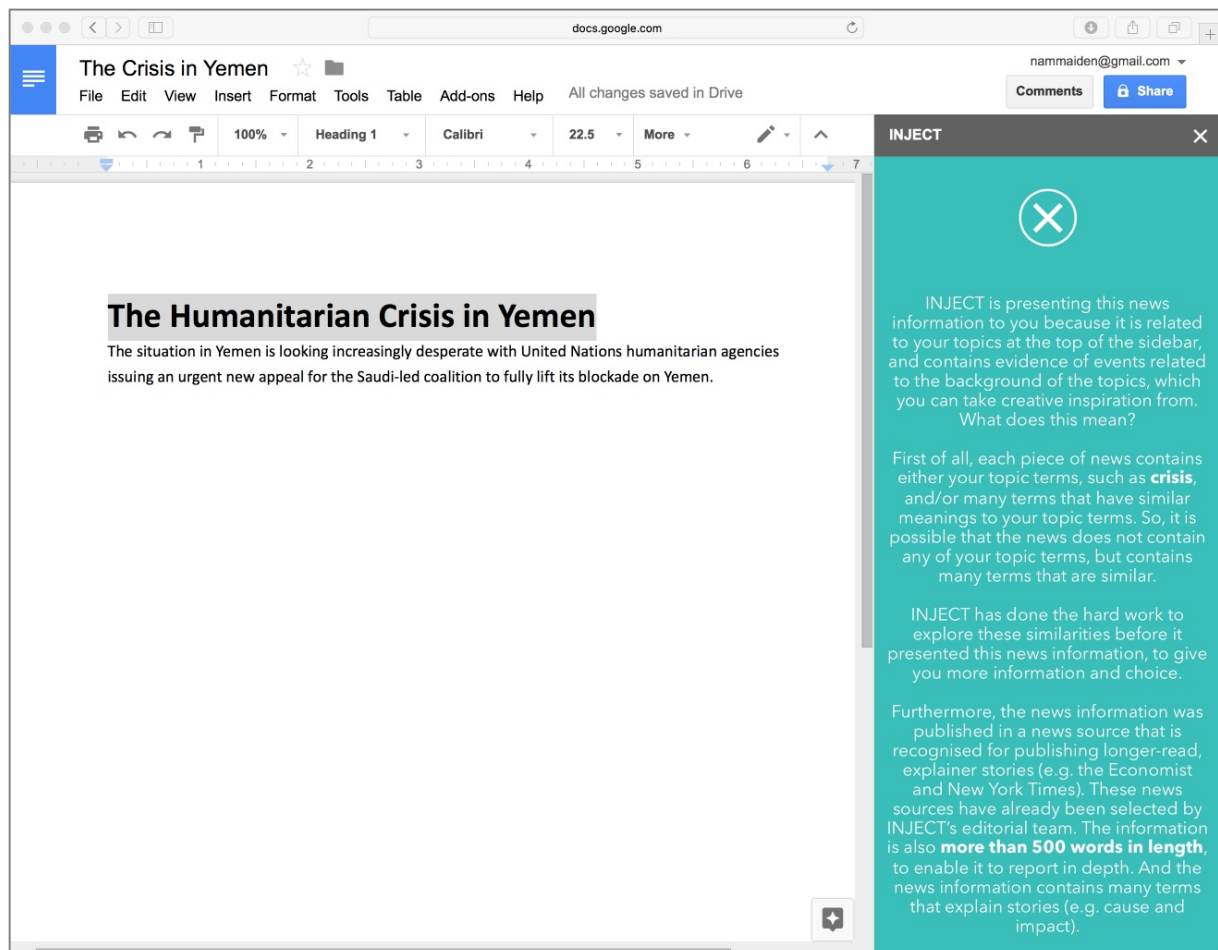


Figure 14. An example of INJECT's more detailed explanation text, presenting the journalist rationale for the presentation of news information using the creative strategy selected by the journalist

One INJECT challenge was to design complex interactions that inputted to and received outputs from sophisticated algorithms that produced unpredictable out-comes. Artificial intelligence was framed as a new design material that required interactions that were transparent, opaque and offer shared control [Holmquist 2017], and journalists both in the evaluations and elsewhere requested transparent creative algorithms to explain the presented news information. However, INJECT's designers also believed that these algorithms still needed to be opaque, in order to retrieve information that was surprising and hence creative to users. Therefore, future work will refine INJECT's interactions to offer different degrees of support related to the journalist's familiarity with the tool, for example to offer greater transparency and shared control to less experienced users, and to increase the opaqueness for more experienced ones.

One new feature of the interaction layer of this version of INJECT is the use of interactive cards. Unlike in the version submitted at month M6, each presented chunk of creative guidance for the journalist is presented an interactive digital card, which are ordered, one after the other, in the Google Docs and other sidebar representations of the creative guidance. These new interactive cards offer the journalist new interactive capabilities, depending on the source of the creative advice that is presented. For each individual presented as creative advice to the journalist, the card presents the following information, to inspire and guide the journalist, as shown in Figure 15:

- The name of the individual, which the journalist can click on;
- Basic information about the individual that is extracted from the person's Wikipedia description.

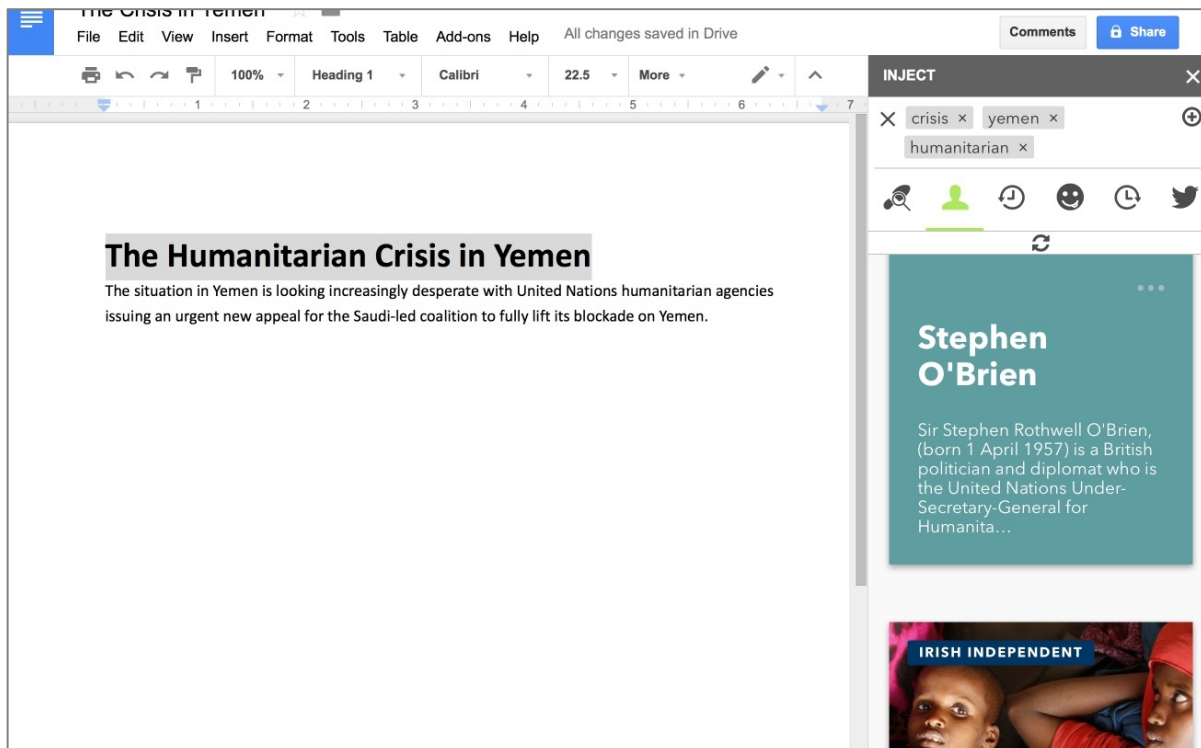


Figure 15. INJECT's creative guidance presenting individuals associated with the news story defined in the topic terms in the INJECT sidebar

This card is different to the creative guidance delivered in the month M6 version of INJECT. In contrast, the card for each news article presented as creative advice to the journalist presents the following, to inspire and guide the journalist, and is quite different to the m6 version of the user interface:

- The news article headline. If the journalist clicks on it, INJECT opens the original news article at source in a news web browser tab. If the journalist hovers the mouse over it, INJECT presents one creative spark that is related to headline, in hover-over text;
- The name of source of the article, and when it was published;
- If available, an image from the original news article;
- Part of the first sentence from the article, to provide the journalist with some context about it;
- 10 colour-coded entities that INJECT extracted from the article. These entities are of 4 types: people, places, events and organisations, and provide an overview of some of the important entities that are talked about in the news article. Moreover, each of the presented entities is interactive using a mouse hover-over feature. Whenever the journalist hovers the mouse over the entity, INJECT presents one creative spark that is related to the entity, in the hover-over text. An example of one creativity spark is shown in Figure 16. As such, the journalist can view quickly up to 10 contextualised creative sparks simply by moving the mouse over a small part of the sidebar. Although the sidebar only presents one recommendation at a time, the journalist can quickly explore a large number of these recommendations by simply moving the cursor over the entities, thereby supporting and directing the discovery of new news angles.

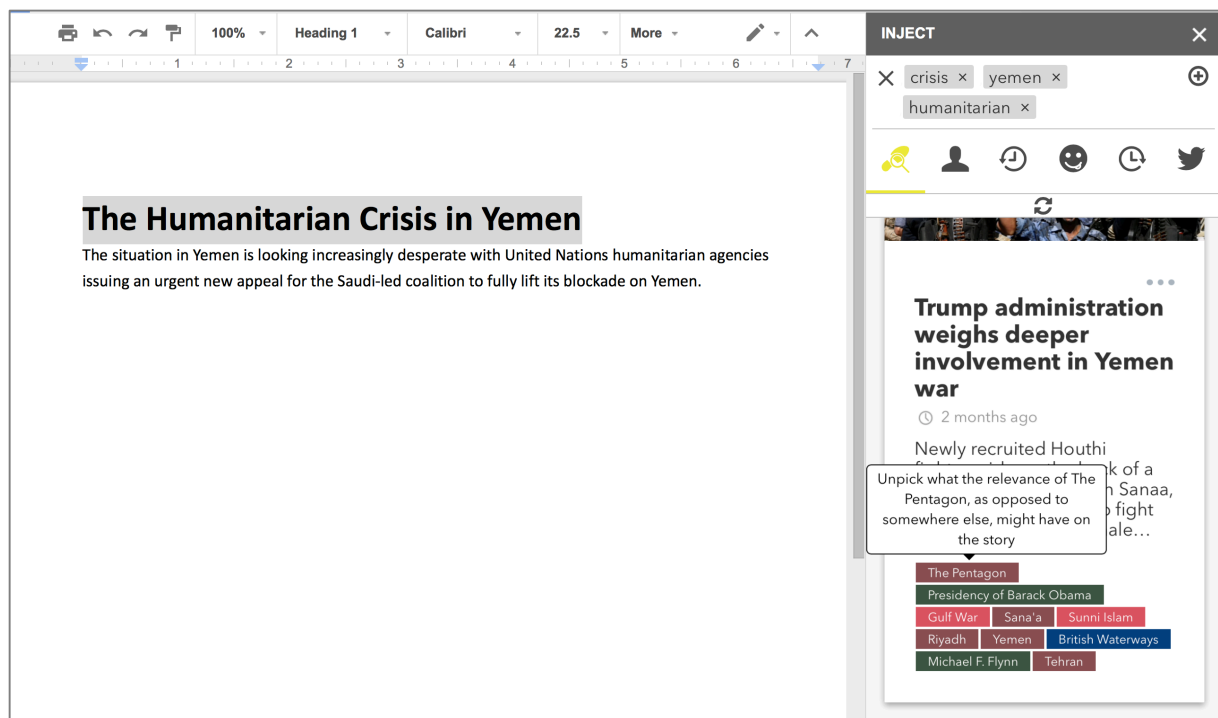


Figure 16. INJECT's mouse hover-over creativity spark feature, showing one spark generated from one entity extracted from a news story

Moreover, by clicking on the top-right of each card, the journalist can 'flip' the interactive

card and access further capabilities associated with this source of creative guidance. This clicking and the features are shown on the left side of Figure 17. These capabilities are:

- A feature to open the news article at source in a new browser tab. If the journalist clicks on it, INJECT opens the original news article in a news web browser tab;
- A feature to copy the citation reference for the news article into the device’s clipboard, so that it can be pasted directly into any document that the journalist is working on;
- A feature to view the 10 creative sparks that can be viewed about the 10 entities that INJECT extracted from the news article as a single list, on the flipside of the card. This feature is shown in the middle of Figure 17;
- A feature to view the news article as a visual word-cloud on the back of the card, so that the journalist can gain quick and new insights from the article without the need to open or read it. This feature is depicted on the right side of Figure 17;
- A feature to instruct INJECT to generate 10 new creative sparks about the 10 entities extracted from the news article and presented in the mouse hover-over feature;
- A feature to generate, in a new web browser tab, a new Google search based on the topic(s) of interest and creative discovery angle entered and selected by the journalist. Many of the evaluations of earlier versions of INJECT revealed that journalists combined the use of INJECT, to discover new creative angles on news stories, with Google searches to examine and collect data about the new news angle. Therefore, for each request, INJECT automatically generates a Google search composed of the terms used to describe the topic of interest in the INJECT sidebar and terms that describe some elements of the creative discovery angle that the news article was retrieved with.

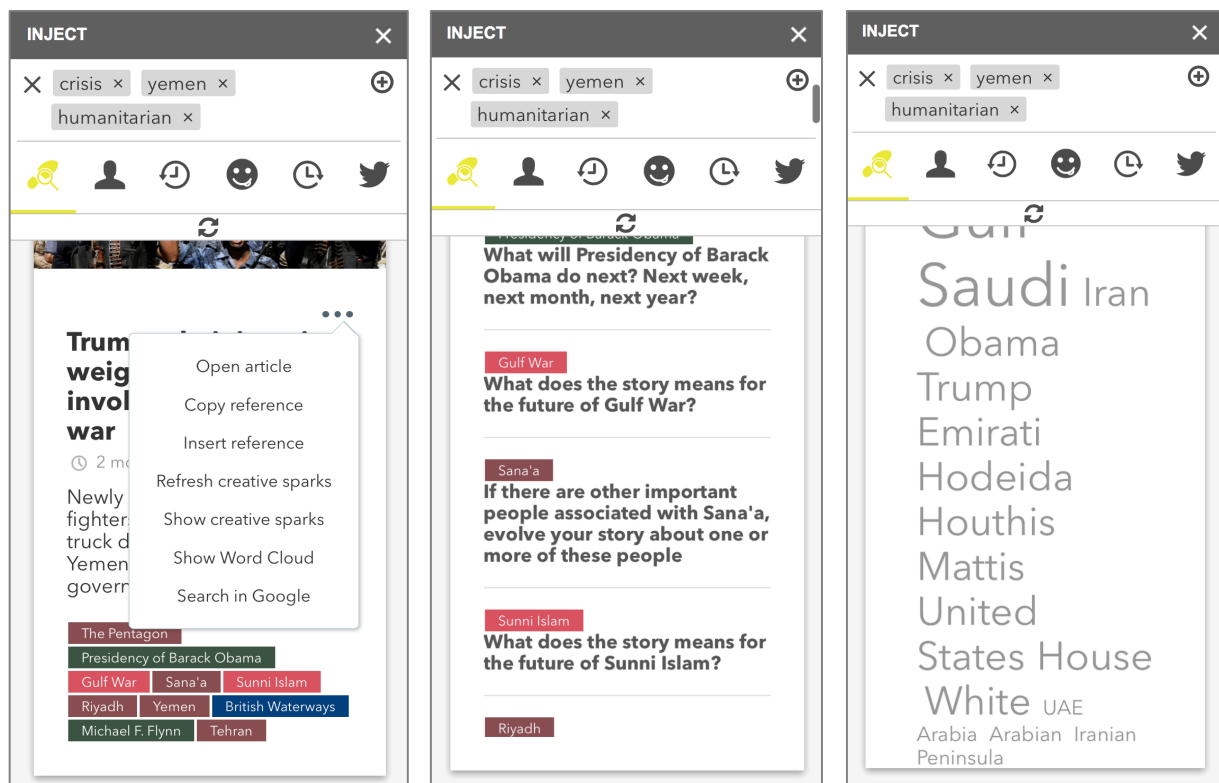


Figure 17. INJECT features associated with each interactive card presented in the sidebar. The features, from left to right are the different reverse-side features, the list format of the creativity sparks, and the visual word-cloud generated from the news article.

The card to present each retrieved political cartoon that provides creative guidance using INJECT's Quirky and satire angle presents:

- The cartoon caption and a thumbnail image of the cartoon;
- The capability to open an enlarged version of the cartoon in a separate web browser tab, if the journalist clicks on the thumbnail image;
- A feature to copy the citation reference for the news article into the device's clipboard, so that it can be pasted directly into any document that the journalist is working on;
- To generate, in a new web browser tab, a new Google search based on the topic(s) of interest and Quirky and satire angle entered and selected by the journalist.

Figure 18 shows an example of a cartoon's thumbnail image and caption presented in the INJECT sidebar.

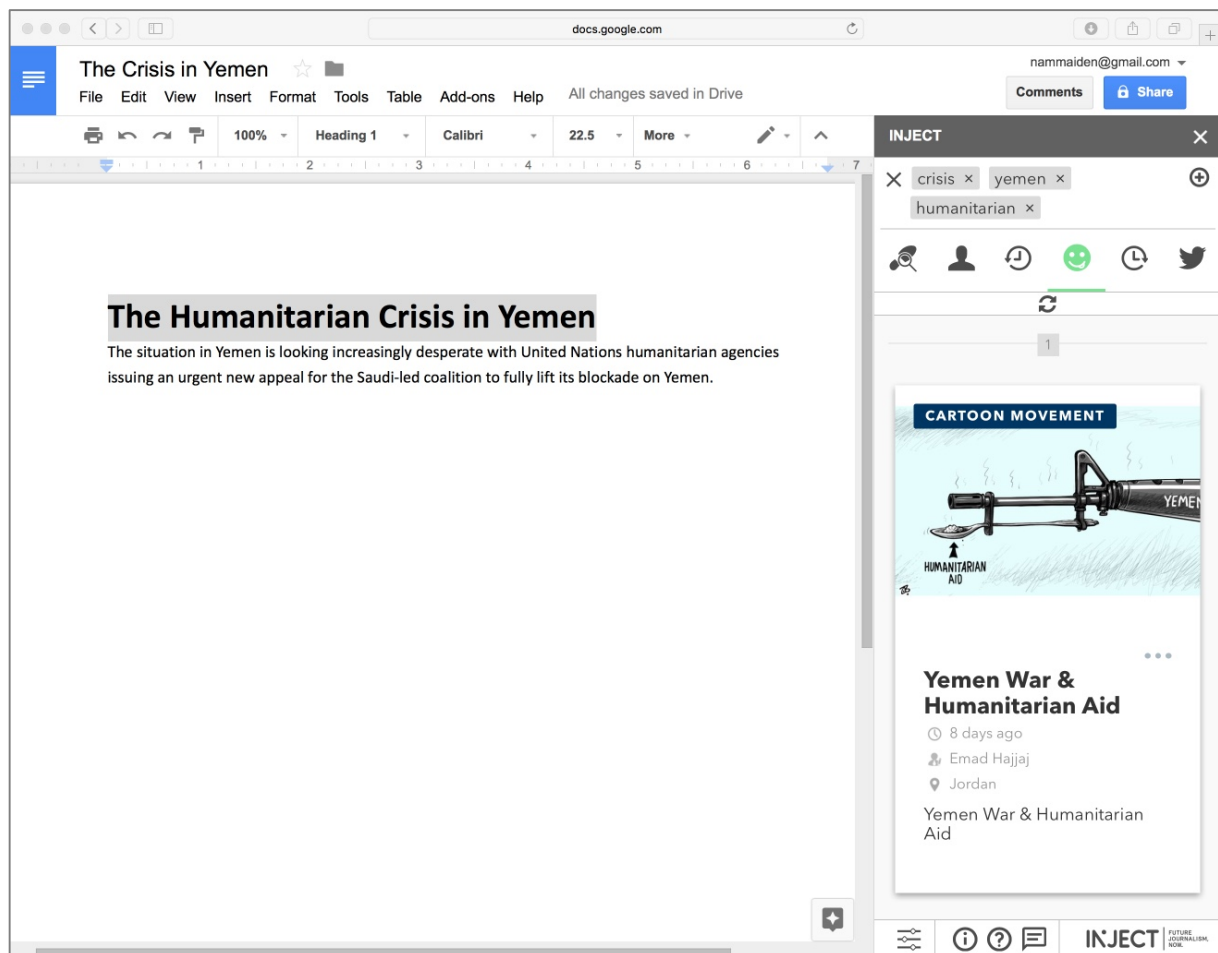


Figure 18. The INJECT sidebar displaying digital cartoons associated with the topic terms, presenting using INJECT's quirky and satire creative strategy

Newer versions of INJECT implement access to news information differently to previous versions. In the earlier versions, such as the one reported at month M6 in Deliverable D1.2, the

journalist was able to explore information an individual or news article in an INJECT dialog box over the editor tool. The dialogue box offered the journalist different capabilities, such as incorporating highlighted text from the news article directly into the text open in the text editor. In the newer versions, the dialog box feature was replaced with direct access to the original text from the newspaper or Wikipedia websites in new web browser tabs, to reduce risks associated with ownership of newspaper content. After consideration of legal advice, the re-representation of news information in the frame page was deemed to pose an unnecessary legal risk, so it was removed. INJECT now presents the original news information articles from source, in separate web browser tabs, as shown in Figure 19.

The screenshot displays the INJECT interface with two browser tabs open. The active tab shows a news article titled "Trump administration weighs deeper involvement in Yemen war" under the "National Security" category. The article features a photo of newly recruited Houthi fighters and is attributed to Karen DeYoung and Missy Ryan, dated March 26. The article text discusses Defense Secretary Jim Mattis's request for the White House to lift Obama-era restrictions on U.S. military support for Persian Gulf states engaged in a protracted civil war against Iranian-backed Houthi rebels in Yemen.

The interface includes a sidebar on the left with social media sharing icons (Facebook, Twitter, Google+, Email, LinkedIn, Pinterest, Tumblr) and a "552" notification. A sidebar on the right displays a UniCredit advertisement for a "Nuovo Mutuo UniCredit" with a 1% fixed rate (TASSO FISSO) and a TAEG of 1.62%. The advertisement lists terms such as a TAEG of 1.62% (example with TAN 1% for a 100,000€ loan over 10 years), a maximum loan amount of 50% of the property value (minimum 30,000€), and a maximum duration of 10 years. It also mentions services like "Taglia, Riduci e Sposta Rate" and a "SCOPRI DI PIÙ" button.

Figure 19. INJECT presenting original news information from source, in different web browser tabs

The INJECT sidebar also supports the journalist to explore each entity in more detail. By clicking on an entity, rather than hovering the cursor over it, the sidebar displays a concept card definition of the entity, which the journalist can use and manage as his/her own private concepts set. Figure 20 shows the sidebar when the journalist clicks on one entity – in this case Archibald Cox. The sidebar presents a concise definition of the person, which the journalist can then use to discover and examine the new news story further.

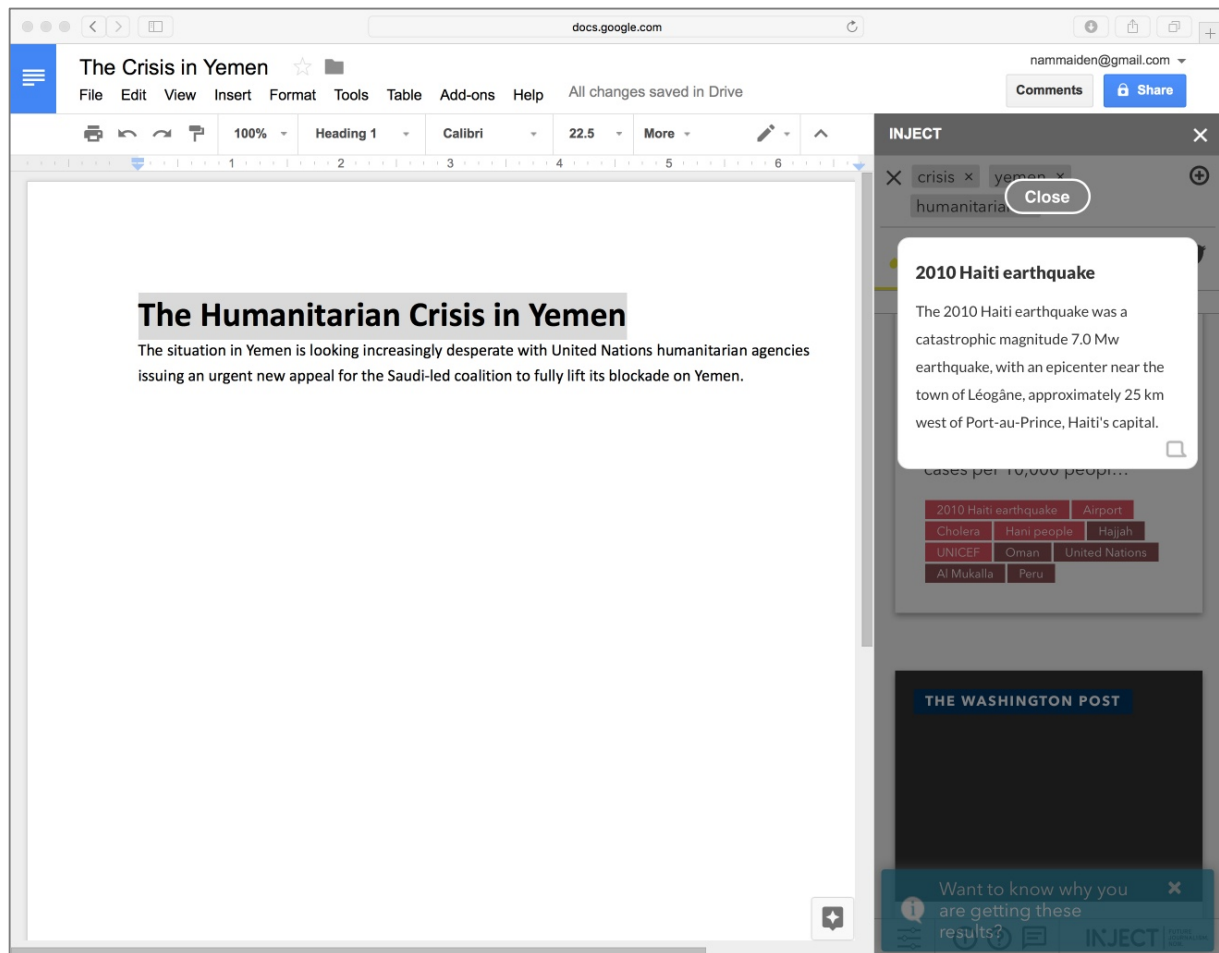


Figure 20. INJECT presenting the concept cards for clicked entities in the INJECT sidebar

4.2. INJECT's CMS & Standalone Interfaces

As of month m12, INJECT has been implemented for the following existing user interfaces in addition to the Google Docs Add-on:

- The CMS interface, e.g. for WordPress and other, often custom-made, content management systems using TinyMCE. TinyMCE is a popular HTML WYSIWYG editor compatible with most content management systems;
- The Adobe InCopy interface, for use with this editor in use in some newsrooms;
- The stand-alone interface for web browsers.

The CMS interface provides all of the capabilities described in the previous section for the Google Docs user interface. The Adobe InCopy and stand-alone web browser interfaces do not offer journalists some of the features due to their technical implementations. For example, in the web browser interface, the journalist cannot highlight text to parse and extract topic terms to auto-populate the side bar with these terms.

Each of these other user interfaces is outlined briefly in turn.

4.3. The TinyMCE INJECT Interface for Content Management Systems

After installing and configuring properly the INJECT plugin for TinyMCE, the user will get an “INJECT” button inside the editor as shown in Figure 21; by clicking it, the INJECT sidebar will be launched. From this point on, the user can have the same experience as described above for the Google Docs Add-on, s/he can retrieve the same articles, perform the same queries, take advantage of the same creative news angles and interact with the editor in the same way.

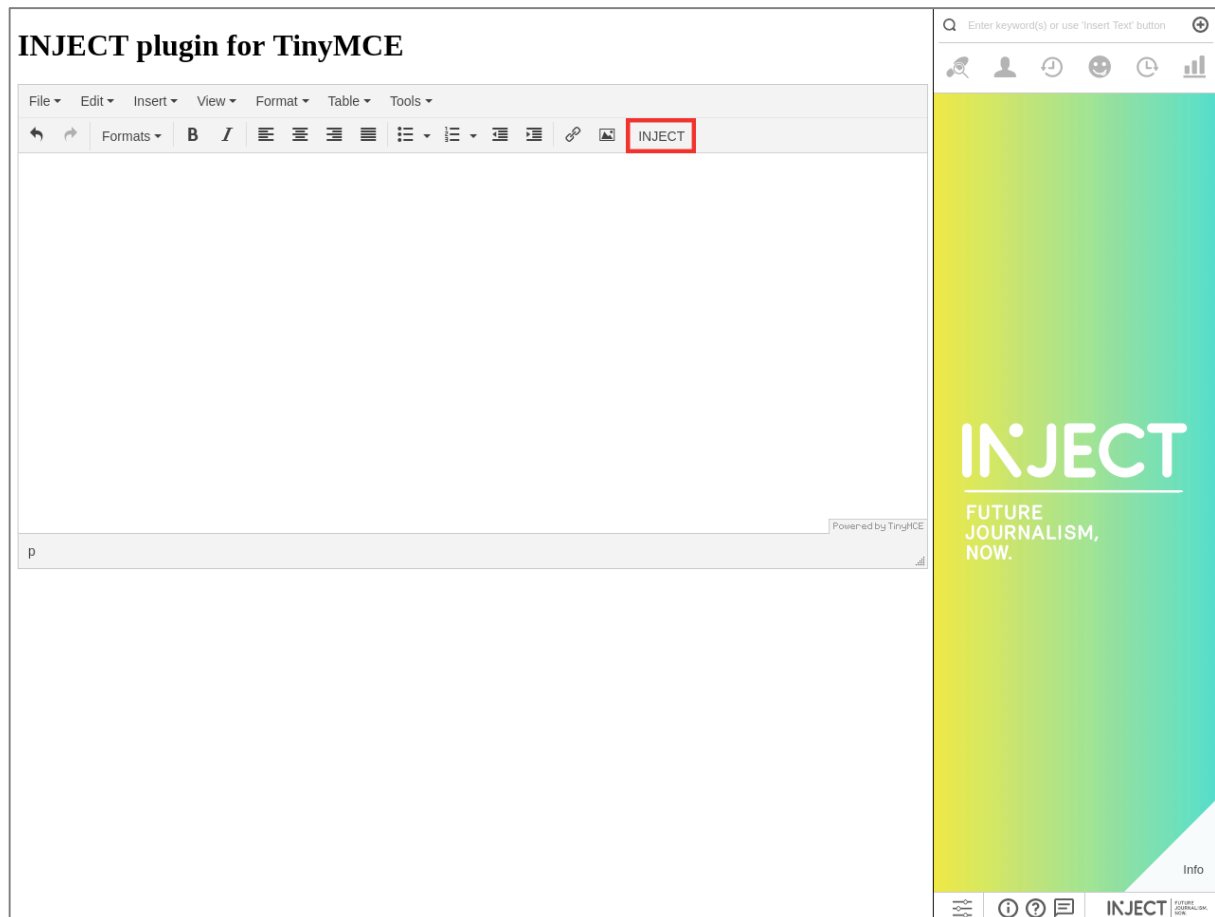


Figure 21. INJECT plugin for TinyMCE

Wordpress is one of the most popular Content Management Systems (CMSs), our users will be able to access the INJECT sidebar inside their WordPress environment without any limitations. By installing the INJECT WordPress plugin, WordPress will enable a new button inside the editor which launches INJECT sidebar. As shown in Figure 22, INJECT sidebar is launched inside WordPress and users can use INJECT exactly as described above.

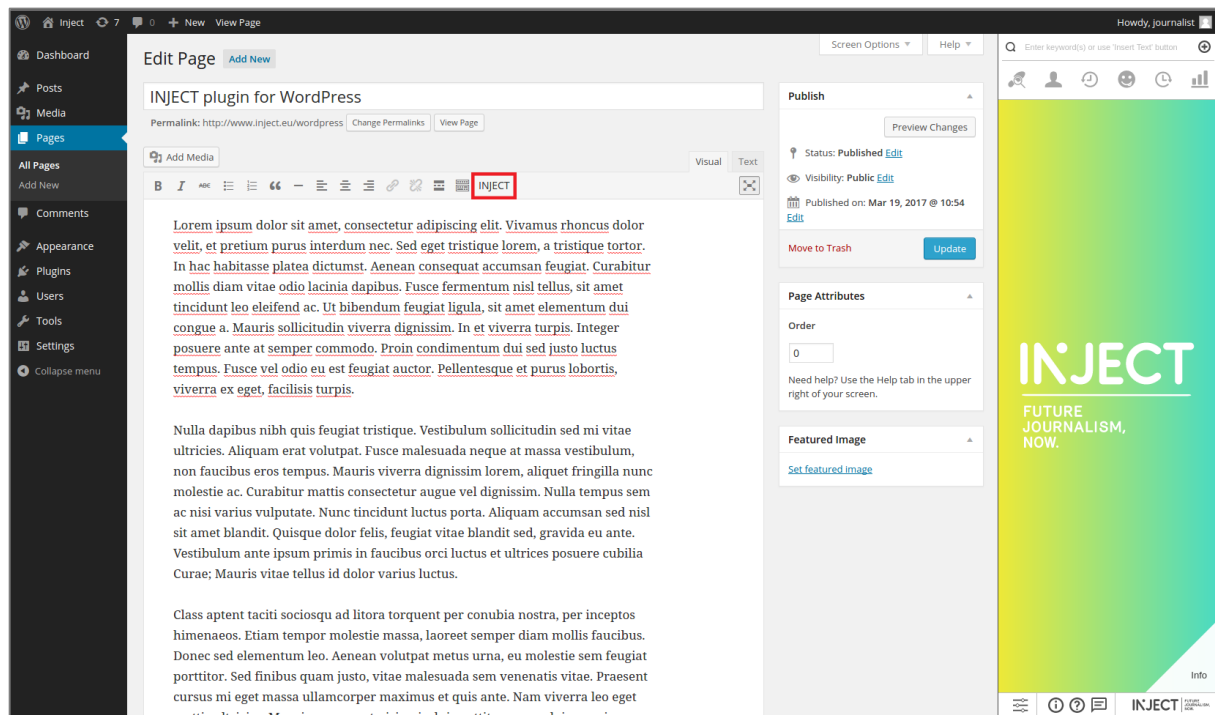


Figure 22. INJECT user interface for Wordpress

4.4. The Adobe InCopy INJECT Interface

In response to demands from newsrooms, a version of INJECT was also implemented to operate with Adobe InCopy text editor also used in some newsrooms. A tailored package was implemented, to enable INJECT to be implemented with the editor. This package can be accessed at: <https://injectproject.eu/downloads/plugins/adobe/Inject.zxp>. Due to technical limitations, not all interactive INJECT features could be implemented in this user interface. An example of the InCopy user interface is shown in Figure 23.

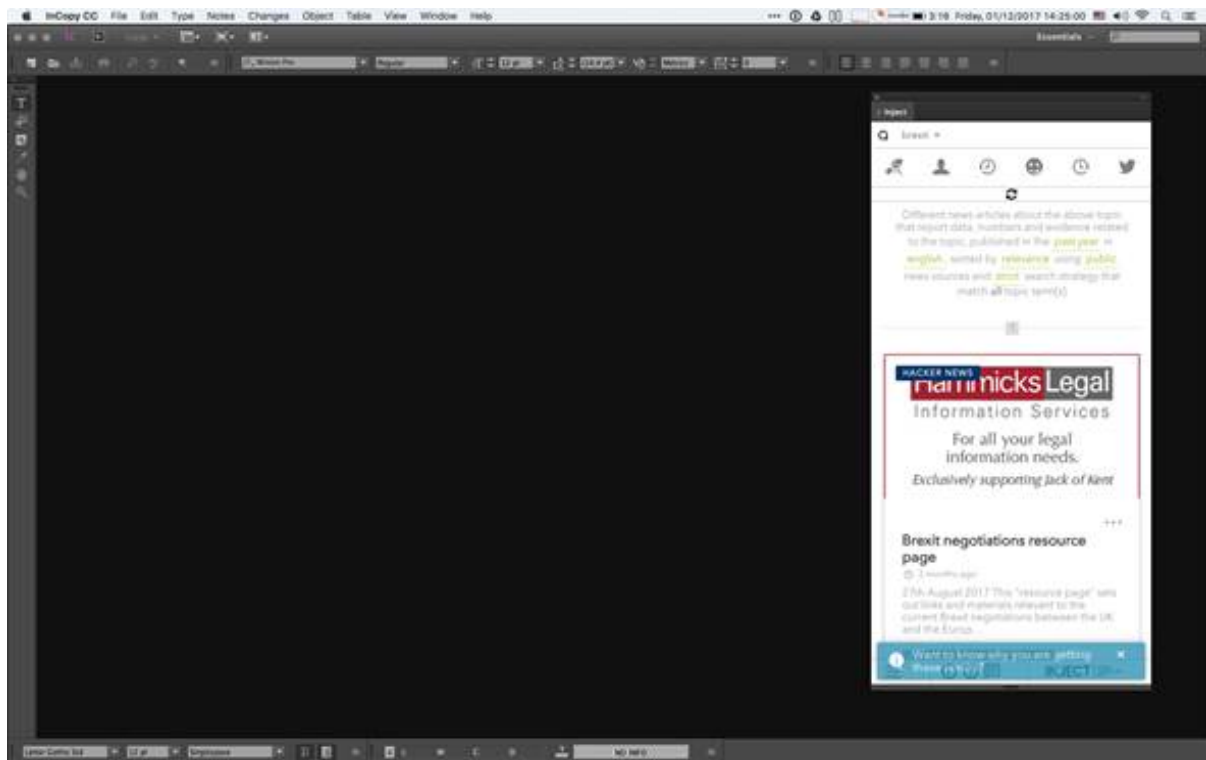


Figure 23. INJECT user interface for the Adobe InCopy text editor

4.5. The stand-alone INJECT interface for web browsers

Investigations revealed that many independent journalists and bloggers do not work with the text editors and content management systems that are used in more traditional newsrooms. Moreover, an increasing number of journalists think about angles for and write news stories in different contexts with different types of device, so there was a need to be able to implement INJECT on mobile devices using a responsive web page. Therefore, INJECT was implemented as a responsive web page accessible on different types of device.

One restriction of the original sidebar design for Google Docs and other content management systems was the fixed width of the sidebar. This fixed width resulted in a single-column format that the user scrolled, to access news information and creative guidance. This restriction did not apply to the web browser version. Moreover, the single-column format on a standard-width web browser page on a computer or laptop device resulted in a poor readability and inefficient use of screen real estate. Therefore, the web browser version was redesigned to allow a multi-column format that responded dynamically to the resizing of the web browser window. An example of this format, showing 3 columns of news information retrieved from the Backing and evidence creative strategy, is shown on the left side of Figure 24. Likewise, the responsive version of the app also displays INJECT on a mobile device, as shown on the right side of Figure 24. On most devices, INJECT's creative guidance is presented single-column on the device.

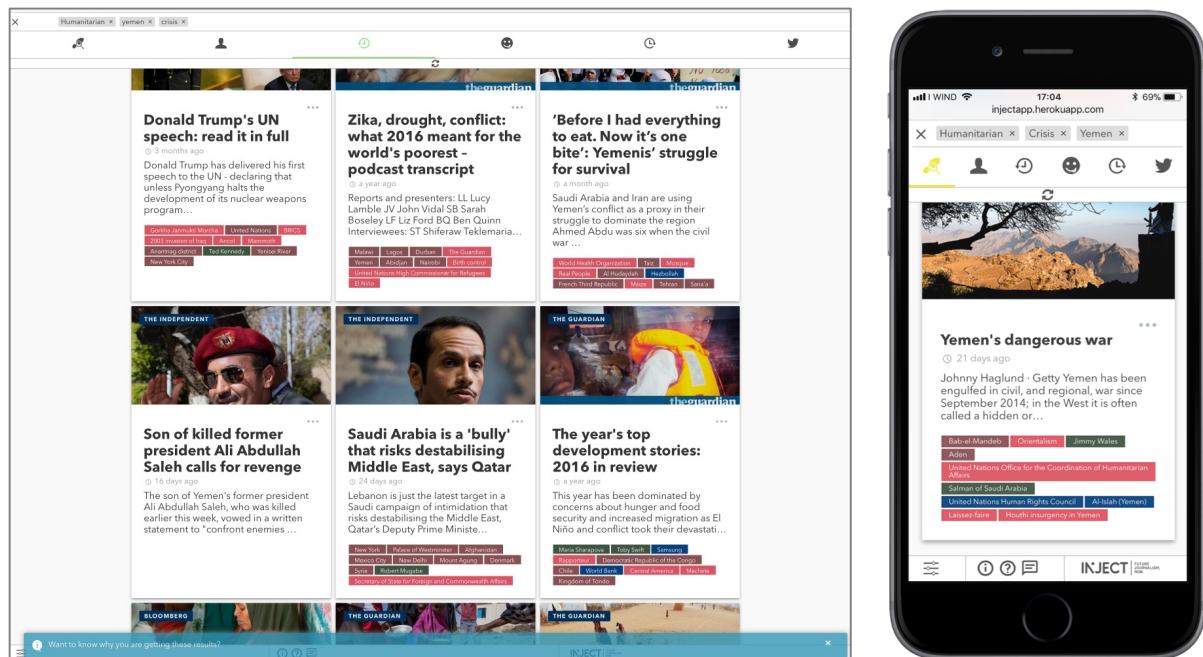


Figure 24. INJECT user interface for standalone web browsers, on desktop computer and mobile device

Furthermore, to allow journalists to collaborate and share the results of different creative guidance that is generated by INJECT, a new interactive feature is available in month M12 to copy and share a bespoke tinyURL, which enables the receiving journalist to view the same creative guidance as the sending journalist within their own web browser.

5. Summary and Outlook

Technical progress in WP1 over the first 12 months of INJECT has been good. The project has delivered a sophisticated new product for use by journalists in newsrooms and other professional environments that is based on a set of advanced services and components. The product is in use in newsrooms within the consortium. Moreover, the product's use of the 3-tier architecture has enabled cost-effective adaptation of the product for use in different settings:

- The distinct interface layer has enabled new versions of INJECT to be delivered quickly for different text editor and web application environments;
- The separate application layer has been enabled these new versions of INJECT to have most if not all of the reasoning and interaction capabilities of INJECT at launch;
- And the separate data layer has enabled configurations of the INJECT product to manipulate news and social media information from different private and public sources and written in different languages.

Table 2 reviews the project's delivery on the 15 key functional requirements that were specified in Deliverable D1.1, at month M3 of the project, by the consortium's news partners. Of these 15 key functional requirements, a total of 7 have been satisfied fully (including the 5 highest-ranked requirements) and 3 remain unsatisfied.

Functional Requirement	Ranking by journalists	Satisfaction or otherwise at month M12
1. Individuals: creatively explore different human angles in a news story based on the different people and roles associated with the story.	1 st place	Requirement satisfied fully: Journalists are able to explore human angles on stories using the fully implemented <i>People</i> strategy.
2. For each article extract and present different concept categories such as places, things, people and organisations.	1 st place (tied)	Requirement satisfied fully: Journalists are able to explore creative angles related to places, things, people and organisations based on INJECT's implemented entity-extraction algorithms and creativity sparks associated with these entities.
3. Search private news archives.	2 nd place	Requirement satisfied fully: Journalists at the 3 Norwegian newspapers are able to retrieve news information from their private news archives, and the VJMO is able to access its library of digital cartoons via the <i>Quirky</i> strategy.
4. Quantifiable elements: creatively explore numerical and quantified information about a news story in order to back it up with evidence in new and useful ways.	2 nd place (tied)	Requirement satisfied fully: Journalists are able to explore quantifiable elements related to stories using the fully implemented <i>Quantifiable</i> strategy.
5. Data visualizations: creatively explore different data sets and visualisations in order to generate new and useful ways about how to display information in a news	2 nd place (tied)	Requirement not satisfied: The data visualization is unimplemented at month M12, due to other emerging requirements that were prioritized to be satisfied for this deliverable. The data visualization creative search algorithms have been designed, and can be implemented before month M18.

story to encourage human insight.		
6. Search additional web news sources, upon request.	3 rd place	Requirement satisfied fully: The INJECT data layer is fully extensible, and new news information sources can be added to it with minimum designer effort, as long as the news source provides a public RSS feed and is not behind a pay-wall.
7. Bookmark search results and categorise bookmarks using tags.	3 rd place (tied)	Requirement partially satisfied: Completion satisfaction of this requirement will be implemented once full user authentication and persistence features are in place.
8. Causal: creatively explore the background events that underpin a story in order to discover a new angle to the story from its background.	3 rd place (tied)	Requirement satisfied fully: Journalists are able to explore causal elements related to stories using the fully implemented <i>Causal</i> strategy.
9. Search for topic timeline.	3 rd place (tied)	Requirement partially satisfied: This requirement was not fully satisfied by month M12. Although capabilities to show the timeline of news articles have been developed, these capabilities have yet to be integrated into the product at the end of month M12.
10. Generate search queries for news information to stimulate journalists to create new story angles.	4 th place	Requirements satisfied: Journalists are able to automatically generate search queries from news information using features implemented in the target Google Docs and TinyMCE text editors. However, these capabilities do not exist for other INJECT-supported interfaces.
11. Rank articles in provided sources by social media impact.	5 th place (tied)	Requirement not satisfied: The requirement referred to the capability to rank articles in provided sources by social media impact, where social media impact is related to the number of Likes and/or Shares. This is scope to implement this requirement in the remaining 6 months of INJECT.
12. Find social media entries that relate to topics and have links to articles published in specific sources.	5 th place (tied)	Requirement partially satisfied: This requirement was refined to specify that INJECT should use news information from Facebook and Twitter. The requirement has been implemented for Twitter, but not for Facebook.
13. Search well-established and known news sources.	6 th place	Requirement fully satisfied: Journalists can use INJECT to discover new angles on stories using almost 300 selected news sources, selected by the INJECT editorial group. Moreover, these news sources are reported in the Norwegian, German, Dutch and French languages, as well as English.
14. Quirky: creatively explore unusual or comical information about a story as a means of using wit to report serious news.	6 th place (tied)	Requirement satisfied: Journalists are able to explore unusual or comic information related to stories using the implemented <i>Quirky</i> strategy, which retrieves information from the VJMO's digital library of cartoons and predefined video sources.
15. Rank "tipping points" based on the number of retweets and favourites.	7 th place	Requirement not satisfied: Other emerging requirements and features were implemented due to greater importance.

Table 2. Functional requirement satisfaction at month M12

There are 2 main reasons for non-satisfaction of some of the requirements – both associated with optimising the use of the project's development resources. The first was the emergence of new requirements after month M3, which were then prioritised by consortium members, which delayed implementations to meet all of the requirements. The second reason was the

need to dedicate more development resources early to refactor some aspects of INJECT's architecture, services and components, to increase the resulting product's flexibility. It is hoped that some resources available in work packages WP1 and WP4 from month m13 onwards can be committed to implement the remaining functional requirements.

Likewise, Table 3 reviews the project's delivery on the 15 key non-functional requirements that were also specified in Deliverable D1.1, at month M3 of the project, by the consortium's news partners. All of these requirements have been satisfied to some degree, with 3 of the 6 requirements fully satisfied.

Non-Functional Requirement	Satisfaction or otherwise at month M12
1. Accessible from desktops, tablets and smartphones.	Requirement fully satisfied: INJECT is accessible and usable through a responsive web application for desktop, tablet and smartphone devices, as well through web browsers to access the other text editor applications that are supported.
2. Integration with popular editors.	Requirement fully satisfied: INJECT can be used as an integrated sidebar within Google Docs, Adobe InCopy, Wordpress, Joomla ,Drupal, and other TinyMCE text editors.
3. Support for different languages.	Requirement fully satisfied: Journalists can interact with INJECT in the English, German, French, Dutch, Norwegian and Italian languages. Journalists working in any of these languages can manipulate news information written in the English, German, Dutch and Norwegian languages.
4. Enhanced usability and gentle learning curve.	Requirement satisfied: Journalists are able to interact with INJECT through a positive user experience that was co-designed with and validated by working journalists. At month M12, the number of reported usability problems is small. Journalist learning of INJECT is supported with an embedded tutorial system and extensive help features.
5. No user perceived performance penalty when compared to general-purpose search engines.	Requirement partially satisfied: Recent implementation of the enhanced Hetzner Online, Amazon AWS ElasticSearch and Anturis services was intended to reduce perceived performance lags with respect to Google Search. INJECT performance has improved since these upgrades, but ongoing service monitoring will be needed to ensure satisfaction of this requirement.
6. User authentication with support for institutional accounts.	Requirement partially satisfied: Institutional accounts have yet to be implemented for INJECT news partners.

Table 3. Non-functional requirement satisfaction at month M12

Of the 2 requirements that are partially satisfied at month M12:

- INJECT's performance will be determined in part by the level of monthly expenditure on commercial servers and services. This performance-expenditure trade-off will be explored during the T1.3 Summative Evaluation work from months M13 to M15;
- Completion of INJECT's user authentication features will ensure that the requirement is satisfied.

6. References

Brill E. (1992) A simple rule-based part of speech tagger. Proc., 3rd Conf. on Applied Natural Language Processing, ACL, Trento, Italy.

DBpedia (2017), <http://wiki.dbpedia.org>, accessed December 21, 2017.

DBpedia Spotlight (2017), <https://github.com/dbpedia-spotlight/>, accessed December 21, 2017.

Holmquist L.E., 2017, 'Intelligence on Tap: Artificial Intelligence as a New Design Material', ACM Interactions 24(4), 29-33.

McCarthy D. Koeling R. Weeds J. Carroll J. (2004) Using Automatically Acquired Predominant Senses for Word Sense Disambiguation, Proc. ACL 2004 Senseval-3 Workshop Barcelona, Spain.

Polyglot's documentation (2017), <http://polyglot.readthedocs.io>, accessed December 21, 2017.

Stevenson M. Wilks Y (2001) The Interaction of Knowledge Sources in Word Sense Disambiguation. Computational Linguistics, 27(3), 321-349.

Simpson, T. (2005) Wordnet.net. open- source.ebswift. com/WordNet.Net.