

# INJECT

FUTURE  
JOURNALISM,  
NOW.

**Project Number 732278**

**Deliverable N°: D1.2**

## **Extended and aggregated digital services and tools**

**June 2017**

**Version N° 1**

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

Institution(s): CITY U., 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

<b>Nature of Deliverable</b>	<b>R + D= Report and Demonstrator</b>
Contractual date of delivery	Project Month 6 – June 2017
Actual date of delivery	30 June 2017
Deliverable number	D 1.2
Deliverable title	Extended and aggregated digital services and tools
Dissemination Level	<b>PU</b> = Public
Status & version	Final 1.0
Number of pages	30
WP relevant to deliverable	WP1
Lead Participant	CITY U.
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 M6 of the INJECT project. The deliverable presents the work done to interoperate with the existing authoring tools and CMSs used by partner SMEs and bespoke digital content repositories of the partner SMEs such as the digital archive used by the Norwegian SMEs. Given that the different SMEs use different combinations of authoring, content management and website tools, this deliverable describes the work performed to develop an abstraction layer on top of existing INJECT digital services which enables the deployment of INJECT in different operational ICT infrastructures without the need to make changes in the INJECT core software. Extensions to the tools include new interactive features to enhance journalist creativity and productivity in different SME work practices and journalism environments.

**Table of Contents**

**EXECUTIVE SUMMARY..... 3**

**1 INTRODUCTION ..... 6**

**2 TECHNICAL ARCHITECTURE ..... 7**

2.1 TECHNICAL CONSIDERATIONS & DESIGN DECISIONS ..... 8

2.2 USER INTERACTION LAYER..... 10

2.3 APPLICATION LAYER..... 11

    Creative Search Service ..... 11

    News Aggregation and Content Extraction Service ..... 12

        News Crawler ..... 13

        News Processor ..... 13

        News API ..... 13

    News Augmentation Service ..... 13

2.4 DATA LAYER..... 14

**3 INJECT USER INTERFACES..... 19**

3.1 GOOGLE DOC INTERFACE ..... 19

3.2 CMS & STANDALONE INTERFACES ..... 27

**4 SUMMARY AND OUTLOOK ..... 29**

**5 REFERENCES ..... 30**

**List of Figures**

Figure 1. INJECT Technical Architecture and M6 status (faded blocks indicate work in progress)..... 7

Figure 2. Refactoring of JUICE for developing INJECT ..... 9

Figure 3. INJECT adapters..... 10

Figure 4. The INJECT start sidebar, as it opens in Google Docs ..... 20

Figure 5. The INJECT sidebar, displaying extracted keywords from highlighted text in the text editor..... 21

Figure 6. The INJECT toolbox and creative search parameters ..... 21

Figure 7. The INJECT sidebar. On the left the sidebar present information about one individual and one news article related to the journalist’s request. On the right the sidebar presents political cartoons that were retrieved using the quirky and satire creative news angle..... 23

Figure 8. INJECT’s individual dialog box ..... 24

Figure 9. INJECT’s news article dialog box..... 25

Figure 10. INJECT sidebar features based on extracted entities associated with retrieved news articles – on the left, the recommendation feature, on the right, the concept definition feature ..... 26

Figure 11. INJECT plugin for TinyMCE ..... 27

Figure 12. INJECT user interface for Wordpress..... 28

## 1 Introduction

---

The present deliverable of the INJECT project is prepared under WP1 and is the preliminary 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 first 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. Moreover, D1.2 documents extensions to the services implemented up to month 6, in accordance to the development plan presented in D1.1. Extensions focus on supporting extended creative news angles for different journalism work tasks and algorithm refinements required due to the nature of different native languages and characteristics of available news stories archives. To design these extensions, the development team relied on the storyboards presented in D1.1, wireframe and rapidly prototype emerging interactive designs enabling user interaction and experimentation and the collection of rapid feedback.

The technical work undertaken so far and documented in D1.2 has taken into consideration the scope of the first INJECTION ecosystem which will be created in Norway, consisting of the three newspapers Sunnhordland, Hallingdølen, Hordaland plus the Commercial Partner M'Labs. In Norway there are five big media organizations, Schibsted, Egmont Fonden, Amedia, Gyldendal, and Aller. Amedia alone owns 62 local newspapers across Norway, Schibsted owns national newspapers and local newspapers. In an effort to assist journalists of primarily smaller local newspapers develop news stories efficiently, the INJECT demonstrator presented with D1.2 aims to support the journalists of the first INJECTION ecosystem in easily accessing relevant content and finding creative angles on news stories.

The INJECT software demonstrator is available at the following URL:

<https://INJECTproject.eu/toolkit/>

This deliverable is organised as follows: Section 2 describes the elaborated version of the technical architecture and outlined the technical work completed in the first six months of the project. Section 3 presents the INJECT user interfaces and the functionalities they make available to journalists. Finally, section 4 concludes this document with a summary and outlook.

## 2 Technical Architecture

In this section, we report the status of the elaborated version of the technical architecture of INJECT, initially presented in D1.1, and we describe the technical decisions made to properly address the design specifications, functional and non-functional requirements.

An overview of the technical architecture and the implementation status at M6 is shown in Figure 1. Note that faded blocks, e.g. Persistence, indicate work in progress, which is not reported in this deliverable. 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.

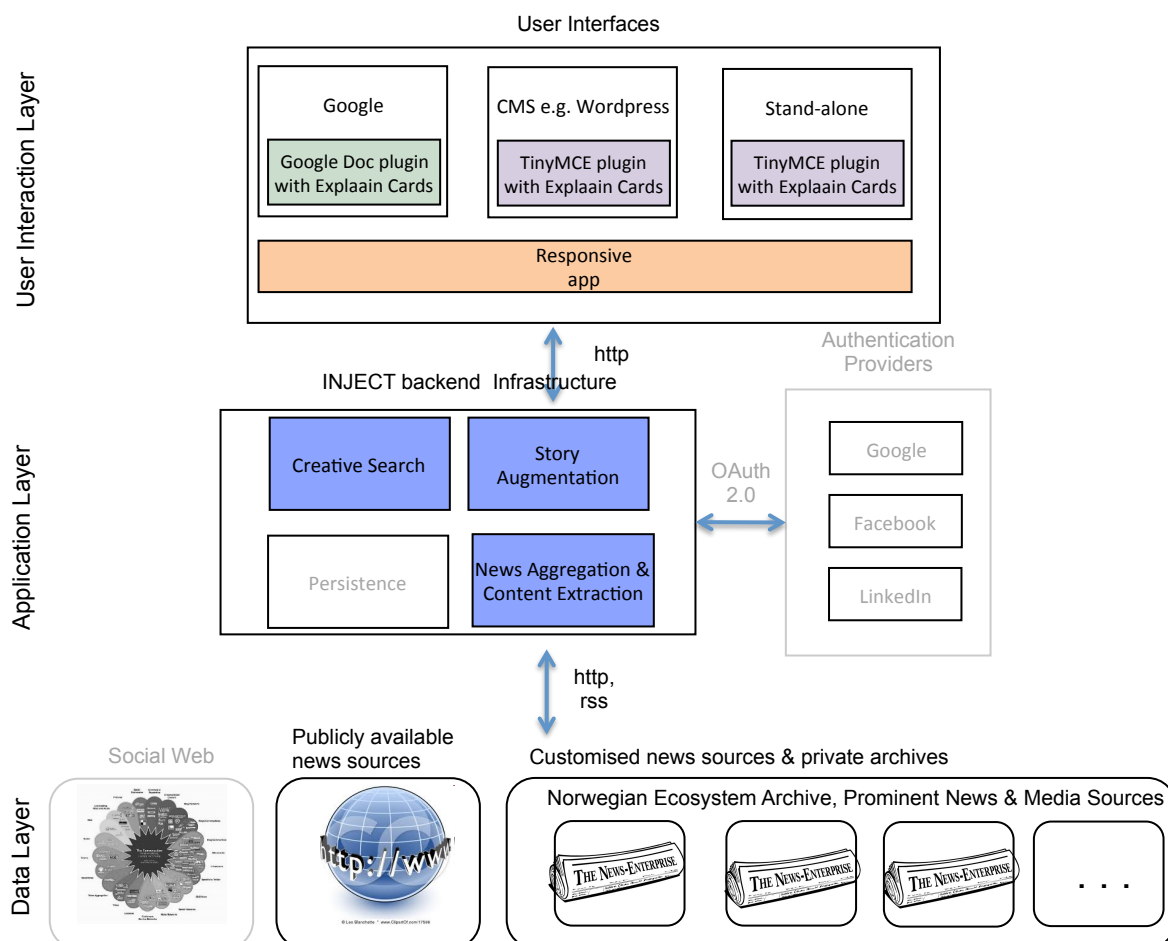


Figure 1. INJECT Technical Architecture and M6 status (faded blocks indicate work in progress)

---

## 2.1 Technical Considerations & Design Decisions

INJECT is a tool which has been derived from a set of existing tools. Among them, JUICE is the prominent news search tool, based on which INJECT has been developed. Initially the entire logic of JUICE was hosted on the Google App Engine, which provided a Google Docs Add-on plugin, developed in *Google Apps Script*. This meant that both the client side and the backend application were on a single repository and deployed as one. Furthermore, the JavaScript implementation which interacted with the Google Docs Add-on was very case-specific, including JavaScript libraries and namespaces only available during runtime within the Google App Engine.

We needed to extract the core functionality of JUICE and implement a more flexible architecture, which would allow the use of the tool with multiple word processors, in multiple environments. We have made use of different application design patterns, such as the Adapter Pattern and the Observer Pattern, to achieve this.

The first step was to refactor calls made to Google App Engine specific namespaces into generic code, and the result was an object with several empty method stubs. This object acted as the INJECT-adapter, and it was the reference implementation to which other implementations should adhere by. The Google Apps Script code was moved to an implementation of the INJECT-adapter into the INJECT-google-adapter. This modification allowed us to create further implementations of the INJECT-adapter such as the INJECT-TinyMCE-adapter which provided the exact same functionality as that of the Google Docs Add-on plugin, but with the popular web editor called TinyMCE.

Since the INJECT tool was running on its own, we needed to provide a communication layer between the INJECT tool and the different environments in which it can run. The environments which we initially aimed to support are:

- Standalone - INJECT runs on its own without any word processor;
- Google Docs Add-on Plugin;
- TinyMCE Plugin – This is a simple editor view using the TinyMCE web editor, allowing incorporation of INJECT in CMSs compatible with TinyMCE such as Wordpress.

*For INJECT to be used in these environments, and to be able to extend this list, we opted to load INJECT much like the Google Maps load INJECT much like the Google Maps JavaScript API does*

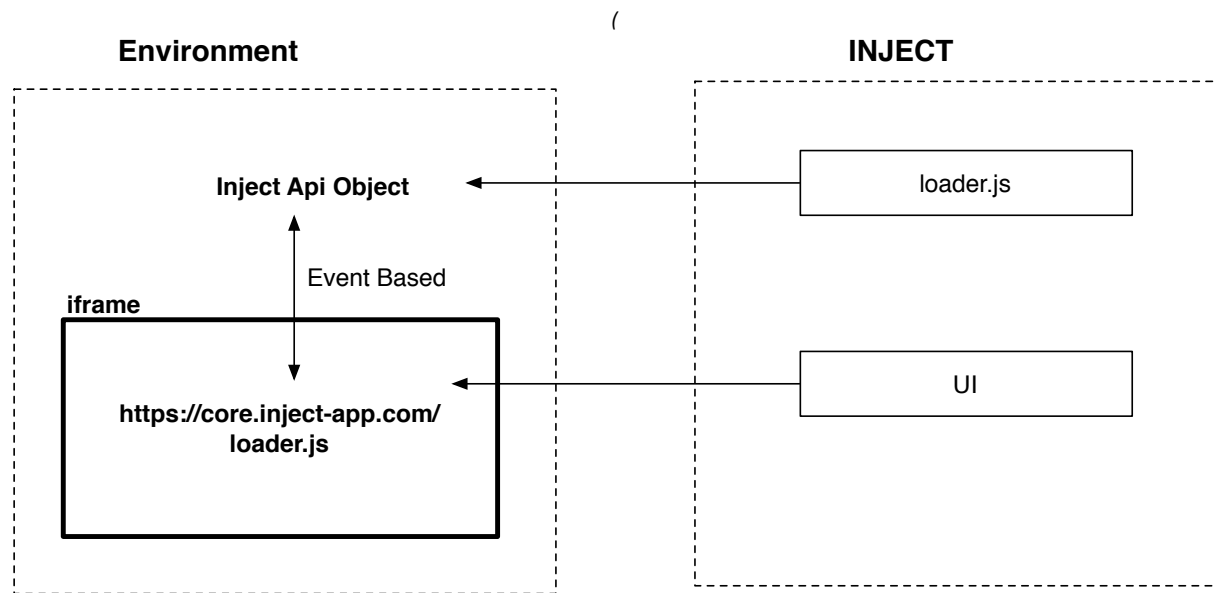


Figure 2). The developer includes a single JavaScript file which takes over the instantiation of the tool and includes the standalone version of the tools using an `<iframe>` on any location of the HTML documents. Hence, INJECT was built as a responsive application, allowing clients to decide where the INJECT tool will be displayed on a page using CSS and HTML.

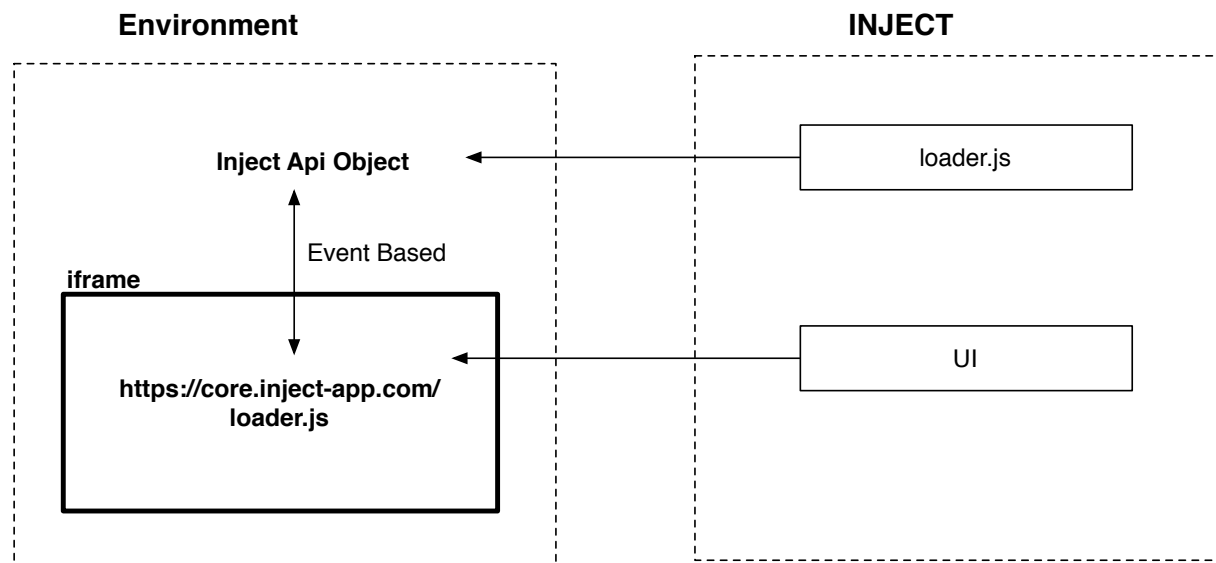


Figure 2. Refactoring of JUICE for developing INJECT

When a page is loaded within an `iframe` there are security challenges which are imposed by browsers that block `iframes` to alleviate the possibility of malicious activity taking place. A workaround is using the Observer Pattern, to implement an Event Based communication protocol to achieve bi-directional communication with the INJECT core. Finally having everything in place we were able, using the Inversion of Control (IoC) Pattern, to determine which INJECT-adapter implementation to use during bootstrapping, and application loading (Figure 3).

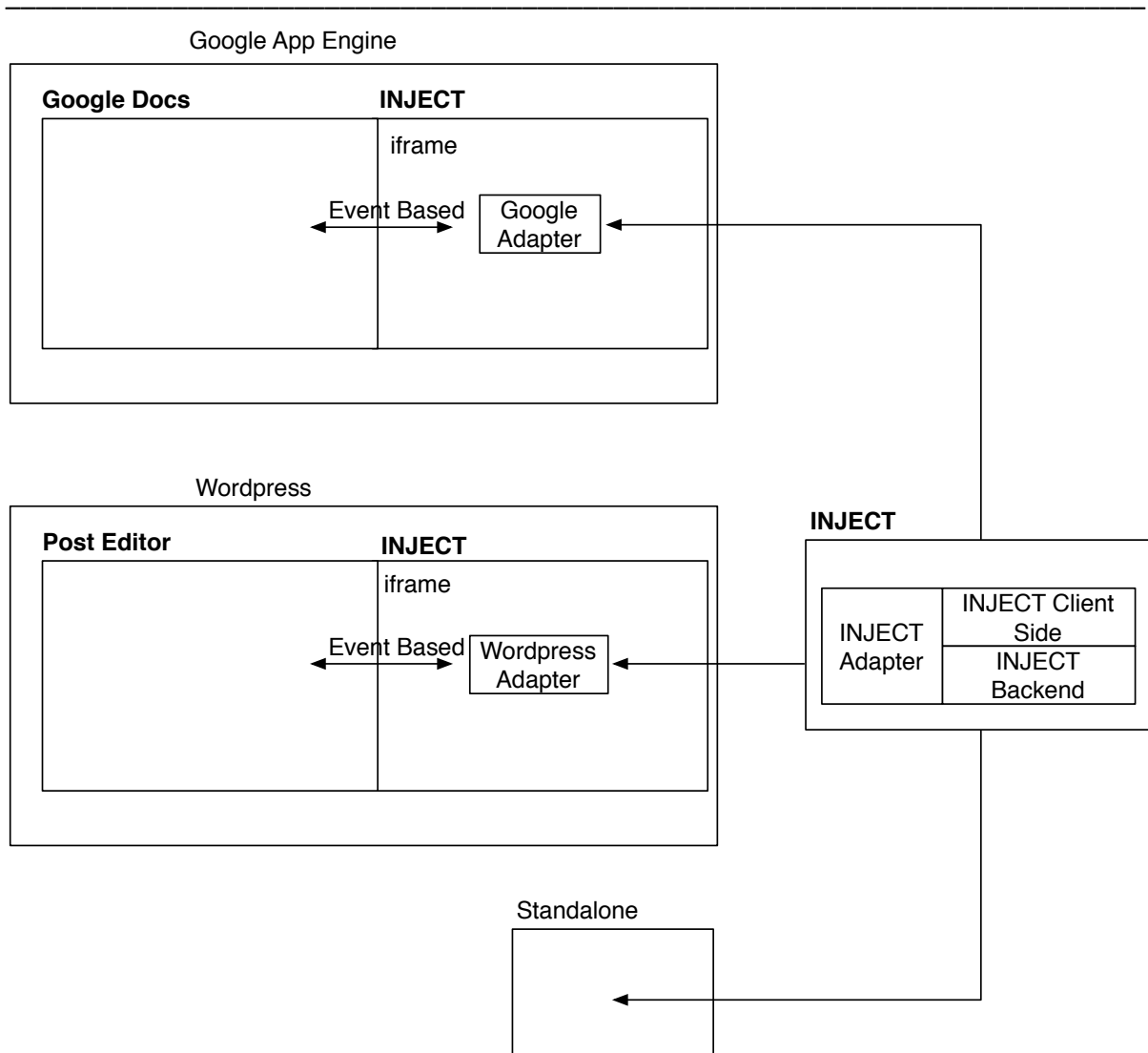


Figure 3. INJECT adapters

## 2.2 User Interaction Layer

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

1. Google Docs interface, which provides the INJECT functionalities through a plug-in / sidebar and exploits Google Docs as the editor for creating, editing, storing and deleting news stories. A Google Docs interface is expected to serve freelancing journalists (e.g., see scenario 1) who are accustomed to the free on-line editing tools provided by Google and wish to have easy access to INJECT functionalities.
2. CMS interface, which provides the INJECT functionalities through a TinyMCE plug-in / sidebar and exploits TinyMCE as the editor and associated popular CMSs such as Wordpress for creating, editing, storing and deleting news stories. This interface is expected to serve journalists working in newsrooms and are accustomed to write their articles in CMSs. TinyMCE has been selected because it is among the most popular CMS editors and in use by most INJECT use case partners.
3. Stand-alone interface, which provides the INJECT functionalities via the TinyMCE editor and associated INJECT plug-in / sidebar allowing journalists to use the INJECT

---

functionalities from within a Web browser and interact with other desktop applications using standard operating system capabilities. This interface is expected to serve 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 in Section 3.

## 2.3 Application Layer

The application layer of INJECT is currently 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 over 100 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. Moreover, the application layer will be augmented with user authorisation and authentication capabilities. Next, we describe components & services (i) – (iii) which are currently available in the M6 deliverable. The persistence component (iv) will be described in the M12 deliverable.

### Creative Search Service

The creative search service manipulates input text automatically to generate search query terms that form the input to 6 creative news angles that were reported by experienced journalists and leading digital experts in journalism as effective for developing creative news angles:

- I. **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;
- 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 as recommendations to stimulate journalists to create new story angles.

Depending on the creative news angle, the creative search service generates 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 that are passed to Presser as described in the next subsection. 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.

### **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. Presser has three main components that are described in the following.

### **News Crawler**

Presser's News Crawler is responsible for accessing RSS feeds and downloading new news articles. We maintain a document that contains details of all RSS feeds that are currently used in INJECT. The News Crawler component loads the RSS feeds document and scrapes all RSS feed URLs. The remaining fields in the document are added to the scraping results so the original feed, region, etc. are stored for each article. RSS feeds are scraped every 12 hours apart from RSS feeds that have a specific flag set to True, which in this case are scraped more frequently, i.e. every 30 minutes. 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.

### **News Processor**

Presser's News Processor is responsible for processing raw news articles in 3 steps - Article Metadata Extraction, Named Entity Recognition, Natural Language Parsing. The first automatically removes any navigation links, ads and more undesired content from an article and extracts the main body of an article including embedded media such as links, images, videos etc. The second applies detects and extracts Named Entities from the article, e.g. people, locations organizations etc. All of these entities can provide interesting and important information on a piece of text and improve the overall analysis of an article. The third uses advanced natural language parsing techniques to identify key noun and verb phrases.

Once a news article has been processed, the service then uploads the processed results into an external Elasticsearch Cluster for post-search activities.

### **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.

### **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. With the news augmentation service, information is presented in the form of small chunks of content, usually a similar length to a tweet. These are the 'atoms' of news, and include profiles, quotes, charts, facts, definitions, statistics, data, maps, timelines, lists, videos, audio, and other embeddable content. When connected, these atoms form an ecosystem that can serve the same purpose as articles or other news formats, only more useful. 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.

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.

## 2.4 Data Layer

INJECT parses and semantically tags news stories from online digital news sources (including partner news archives) 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 in Table 1.

Table 1. Indexed News Sources

Source	Country
BBC	UK
Quartz	UK
The Guardian	UK
Telegraph	UK
FT	UK
The Times	UK
Sky News	UK
The Independent	UK
The Huffington Post	UK
The Huffington Post	US
Reuters News	UK
The Economist	UK
The New York times	US
Daily Mail	UK
The Wall Street Journal	US
The Washington Post	US
The Metro	UK
Herald Scotland	UK
Bloomberg	US
The Scotsman	UK
The Irish Times	Ireland

Irish Independent	Ireland
New Statesman	UK
Newsweek	US
The Daily Beast	US
Times Education Supplement	UK
BBC Mundo	UK
El Mundo	Spain
El Pais	Spain
Cinco Dias	Spain
CNN	US
CNN Money	US
London Evening Standard	UK
Birmingham Post	UK
Birmingham Mail	UK
Farming Life	UK
Belfast Telegraph	UK
Yorkshire Post	UK
Yorkshire Evening Post	UK
Manchester Evening News	UK
South Wales Evening Post	UK
Irish Examiner	Ireland
Herald Scotland	Scotland
The Mirror	UK
The Irish Sun	Ireland
Irish Daily Star	Ireland
The Sun	UK
Daily Star	UK
Daily Record	UK
Daily Express	UK
Los Angeles Times	US
Chicago Tribune	US
The Onion	US
Forbes	US
Fox News	US
Herald Tribune [International NY Times]	US

ABC News	US
Buzzfeed	US
Newsmax Media	US
U.S. News and World Report	US
The Globe and Mail	Canada
Toronto Star	Canada
New Zealand Herald	NZ
Dominion Post	NZ
The Sydney Morning Herald	Australia
The Brisbane Times	Australia
Herald Sun	Australia
The Daily Telegraph (Australia)	Australia
The Courier-Mail	Australia
Bangkok Post	Thailand
Jakarta Globe	Indonesia
South China Morning Post	Hong Kong
Der Spiegel International	Germany
Ekathimerini	Greece
Dutch News	Netherlands
Krakow Post	Poland
Portugal Resident	Portugal
The Local Newspaper	Sweden
Connexion Newspaper	France
Le Monde	France
Le Monde Diplomatique	France
EuroFora	EU
Friedl News	Austria
New Europe	Belgium
Copenhagen Post	Denmark
News of Iceland	Iceland
Finnbay Newspaper	Finland
North Cyprus News	Cyprus
Prague Daily Monitor	Czech Republic
Daily News Egypt	Egypt
The Punch	Nigeria

Business Day Live	South Africa
Independent Newspaper	South Africa
Mail and Guardian	South Africa
Bhutan Observer	Bhutan
Financial Express	India
Business Standard	India
Economic Times	India
The Indian Express	India
Live Mint [INDIA]	India
Stavanger Aftenblad	Norway
Bergens Tidende	Norway
Dagbladet	Norway
Verdens Gang (VG)	Norway
Dagens Næringsliv	Norway
NRK	Norway
Aftenposten	Norway
Le Figaro	France
BFMTV	France
Le Parisien	France
Le Express	France
L'OBS	France
Le Point	France
Les Echos	France
CBS	Netherlands
SCP	Netherlands
NU	Netherlands
Al Jazeera	Qatar
FD	Netherlands
Adforrmatie	Netherlands
Eerste Kamer	Netherlands
Europees Parlement Nieuws	Netherlands
Daily Nation	Kenya
Vanguard	Nigeria
The Namibian	Namibia
News24	South Africa

---

Technology Review	USA
Ars Technica	USA
WIRED Business	USA
WIRED Culture	USA
WIRED Science	USA
CNET News	USA
9 to 5 Google	USA
9 to 5 Mac	USA
The Verge RSS	USA
APPLE Press Releases	USA
Daring Fireball	USA
NYTIMES Tech	USA
Venture Beat	USA
The Register Headlines	UK
The Register Business	UK
The Ken	India
Heise News (german)	Germany
Mashable	USA
Hacker News (3rd party)	USA
The next Web	USA

---

### 3 INJECT User Interfaces

---

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

- Google Docs Add-on interface;
- CMS interface, e.g. for WordPress and other, often custom-made, content management systems;
- 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. They 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 elements of the INJECT user interfaces, and reports the rationale for these elements. It focuses on the Google Docs Add-on version of the interface, but includes parallel examples from other tools, to demonstrate these other interfaces.

#### 3.1 Google Doc Interface

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 4. 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.

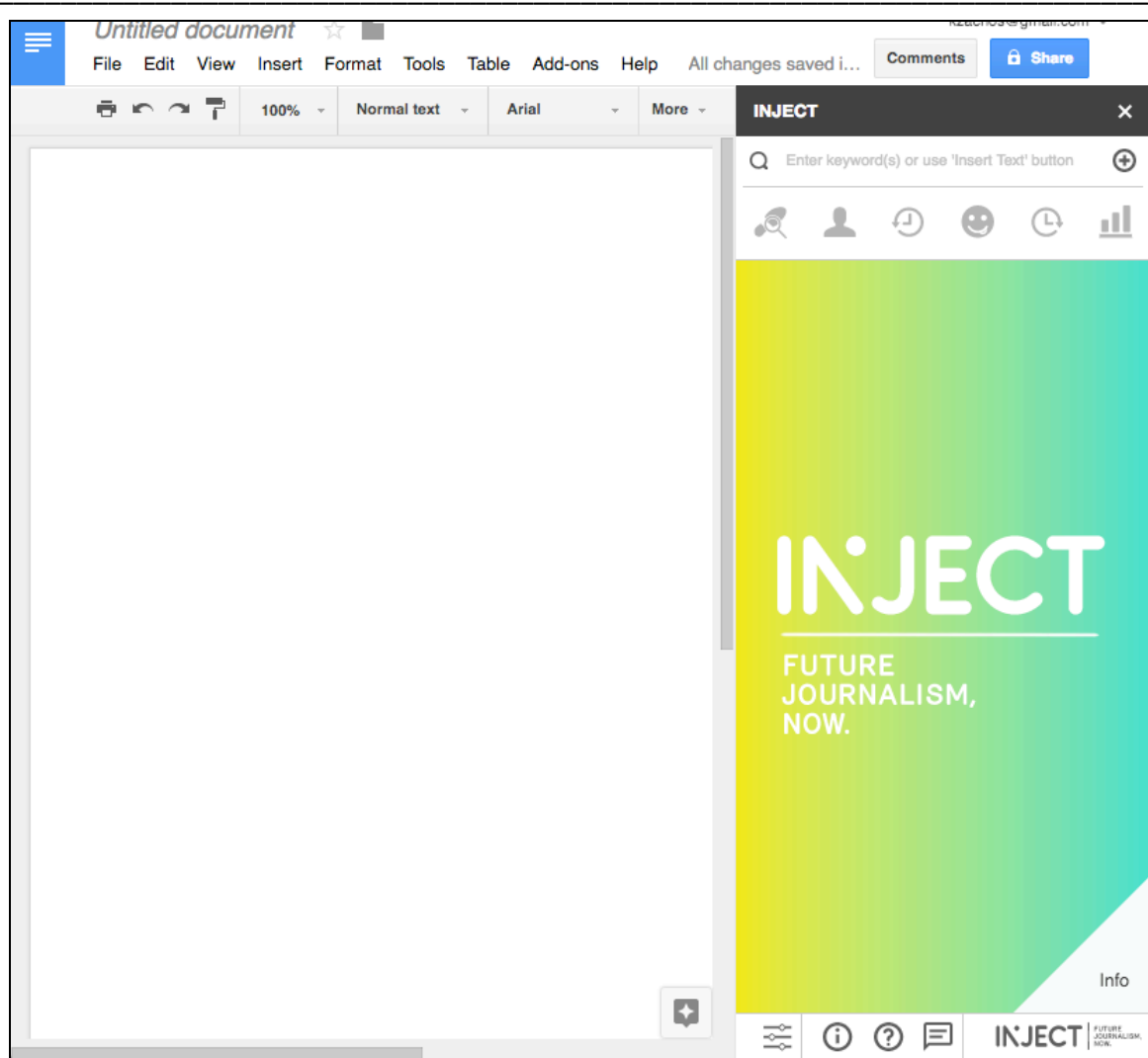


Figure 4. The INJECT start sidebar, as it opens in Google Docs

The journalist is free to work as s/he wants. 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 US presidential election conspiracy. She has already typed these words in the text editor. She highlights the text in the editor, then clicks the + feature in the right-hand top corner of the sidebar. In response to this request, INJECT parses and extracts keywords from the entered text, and displays these keywords, as shown in Figure 5.

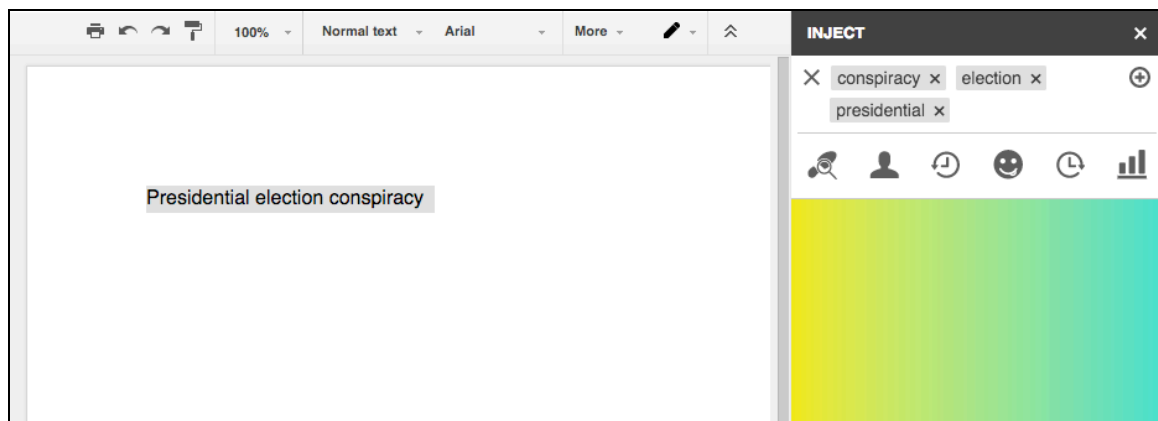


Figure 5. The INJECT sidebar, displaying extracted keywords from highlighted text in the text editor

INJECT allows the journalist to review edit, add and delete each keyword directly in the top of sidebar shown in Figure 3.2, before launching each creative search. 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 searches. The INJECT toolbox feature is shown on the left-hand side of Figure 6.

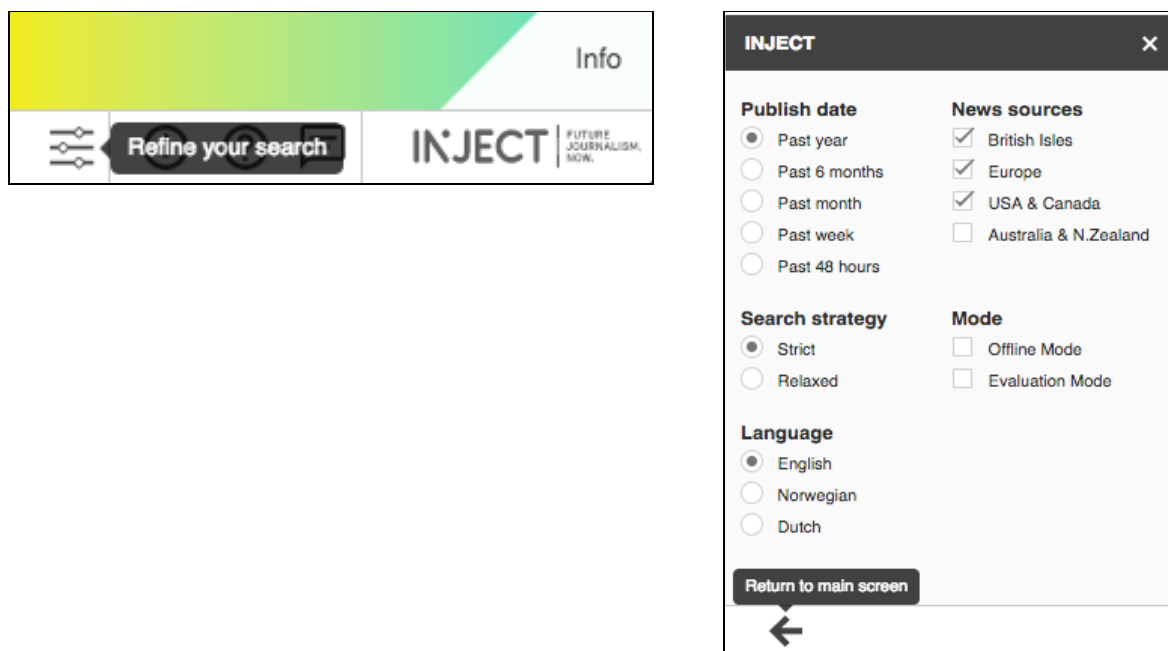








Figure 6. The INJECT toolbox and creative search parameters

The INJECT toolbox offers the journalist parameters to control the creative search, based on the publication date, location and language of the original news stories, see the right-hand side of Figure 6. S/he can also choose strict (directed) or relaxed (undirected) creative search alternatives, as well as the mode for use (for example for design evaluation work). At any time, the journalist can return to the main screen, and the INJECT sidebar, to continue working.

To launch creative searches in INJECT, the journalist simply clicks on the 6 creative search icons underneath the sidebar's query terms, as visible in Figure 4 and Figure 5. Each icon was

designed to communicate the specific creative news angle, and a simple hover-over also displays the text name of the creative news angle. The 6 icons and creative news angles are shown in Table 2.

Table 2. The INJECT creative search icons

	Backing & Evidence – search for news articles that report on quantitative evidence that are associated with the requested information
	Individuals, the key players – search for news articles that identify individuals who are associated with the requested information
	Causal, the background and back story – search for news articles that report on the causal back story associated with the requested information
	Quirky and satire – search for digital cartoons that are associated with the requested information
	Ramifications and the future – search for news articles that report on the future implications that are associated with the requested information
	Data visualisations, charts and infographics – search for news articles that include different forms of visual representations that are associated with the requested information

INJECT presents results from each creative search in the sidebar, underneath the creative search icons. Figure 7 shows a segment of the INJECT sidebar that presents news articles and individuals retrieved in response to the query terms. The icon of the selected creative news angle remains highlighted, in green – in this case the journalist used the *individuals/key players* creative news angle. The journalist can re-launch the creative search at any time using the *refresh* icon that is presented before the creative search icons. The sidebar presents retrieved information in chunks of 10, and the journalist can scroll down the sidebar to access the information that is not visible directly in the sidebar. For each retrieved individual, the sidebar presents the following, to provide sources of creative inspiration to the journalist:

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

For each retrieved news article, the sidebar presents the following, also to provide sources of creative inspiration to the journalist:

- The news article’s headline, which the journalist can click on;
- The source of the article, and where it was published;
- All or part of the first sentence from the article;
- A set of interactive colour-coded entities that INJECT extracts from the article. These entities are of 4 types: people, places, events and organisations.

Moreover, for each retrieved political cartoon that is retrieved using the Quirky and satire creative news angle, the sidebar presents the cartoon caption and a thumbnail image of the cartoon that can be clicked on to enlarge, if the journalist wishes it.

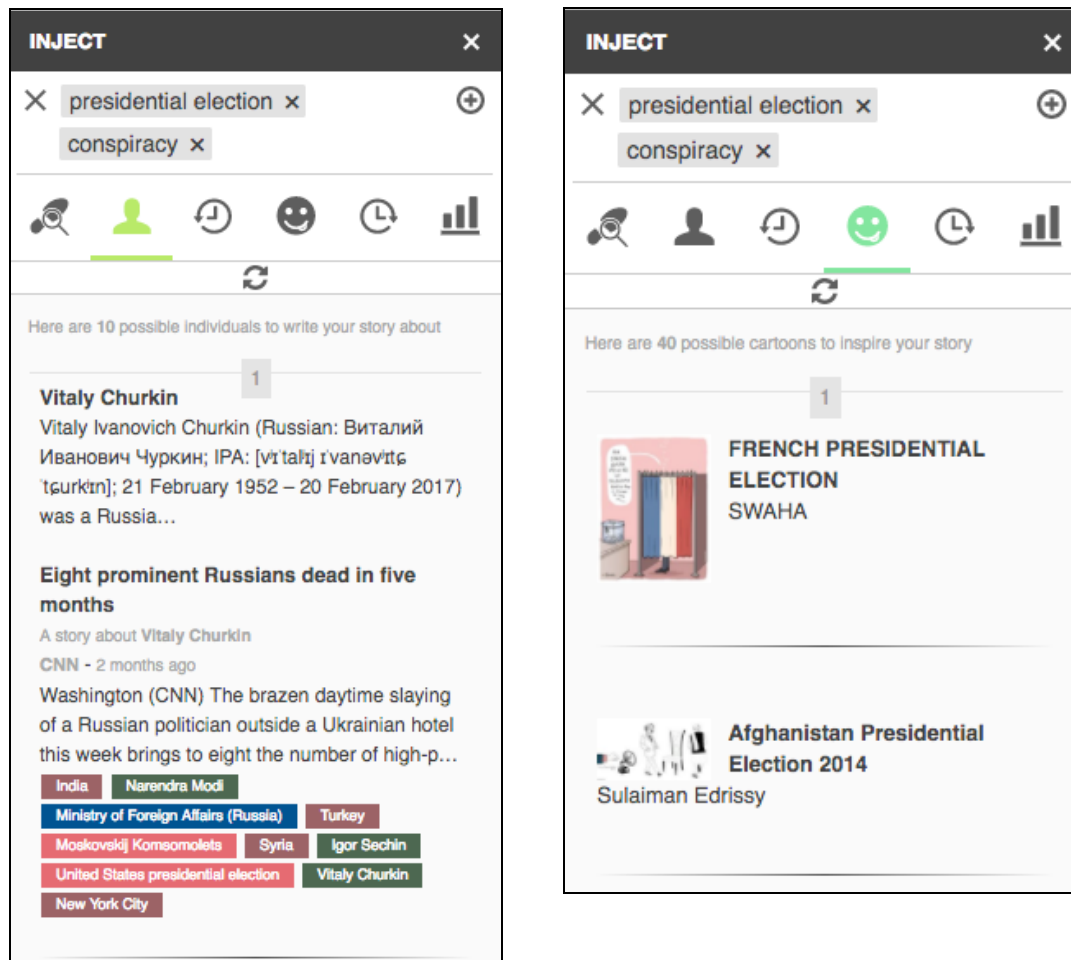


Figure 7. The INJECT sidebar. On the left the sidebar presents information about one individual and one news article related to the journalist’s request. On the right the sidebar presents political cartoons that were retrieved using the quirky and satire creative news angle

The journalist can then explore more information about a presented individual or a presented news article. When the journalist clicks on the individual’s name, INJECT opens a new dialog box over the editor tool. The dialogue box provides a photograph of the individual and a text description of the individual, extracted from that individual’s Wikipedia description, as shown in Figure 8.

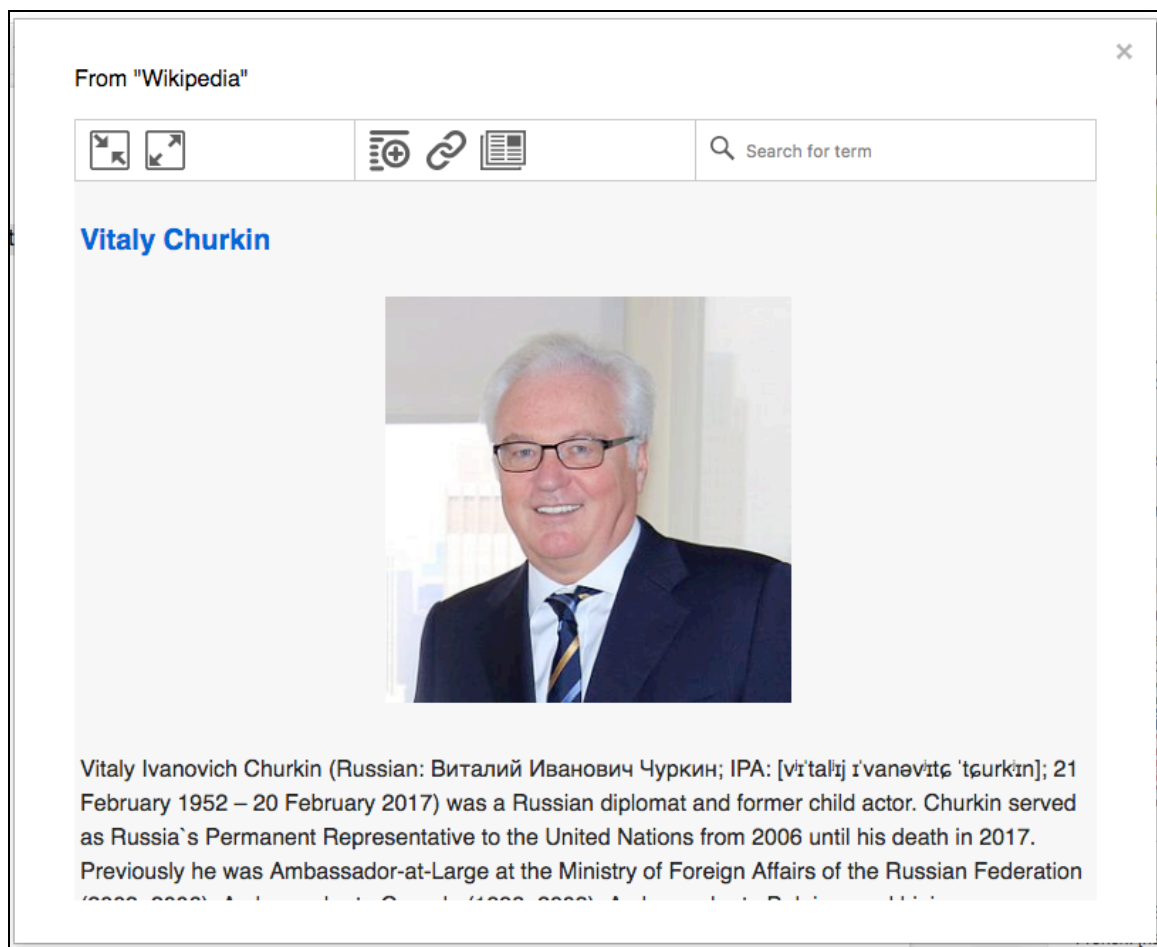


Figure 8. INJECT's individual dialog box

The dialogue box also offers the journalist different capabilities, which are presented across the top of the box. These capabilities were implemented to facilitate creative thinking and to enhance journalist productivity when s/he uses INJECT. The journalist can enlarge or reduce the size of the box, so that s/he can continue to work in the text editor. The journalist can click on the *newspaper* icon to open the original news article at source, in a new browser tab. The journalist can click on the *link* icon to incorporate a full reference link to the news article in the text open in the text editor. And the journalist can incorporate highlighted text from the news article directly into the text open in the text editor.

Furthermore, INJECT support the journalist to undertake refined keyword searches of the text for each single individual and news article, using the *Search for term* field in the dialog box. Not only does this feature highlight the presence of the creative search query terms in the article, to guide the journalist to relevant information in longer news articles, but it also allows the journalist to examine each news article more productivity within the INJECT tool. Figure 9 depicts the news article dialog box, with a portion of the news article text highlighted.

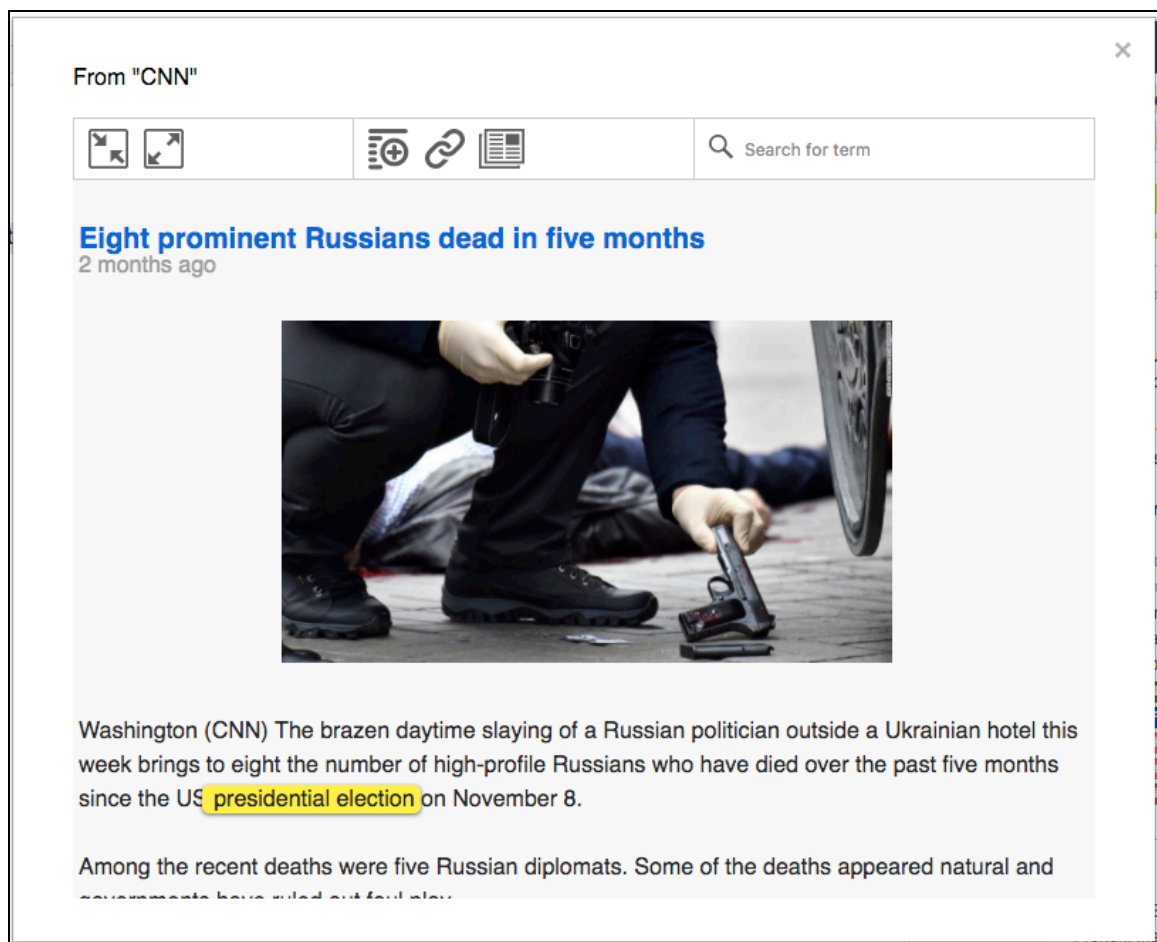


Figure 9. INJECT's news article dialog box

INJECT also offers other interactive creativity support within the sidebar, based on the entities that INJECT extracts from each retrieved news article. First, it provides one recommendation for each extracted entity to guide the journalist to discover one or more new angles on the news story. To access these recommendations, the journalist only has to move the cursor over the entity, and INJECT presents the recommendation as a simple hover-over feature – no clicking or other form of interaction is required. The left-hand side of Figure 10 presents one such recommendation, linked to the extracted entity *ABC News*. 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.

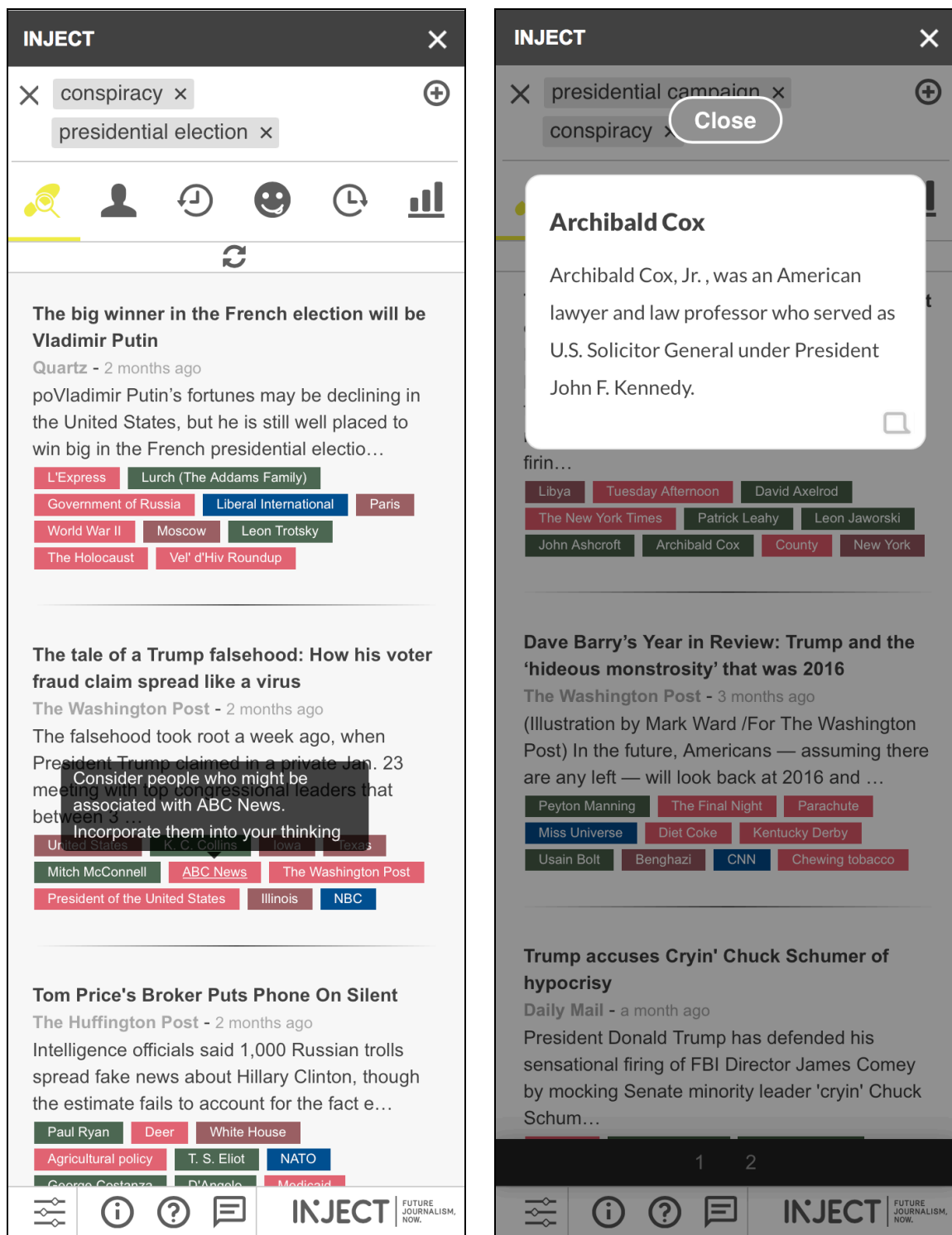


Figure 10. INJECT sidebar features based on extracted entities associated with retrieved news articles – on the left, the recommendation feature, on the right, the concept definition feature

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. The right-hand side of Figure 10 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.

### 3.2 CMS & Standalone Interfaces

Another version of INJECT user interface is the TinyMCE version. TinyMCE is a popular HTML WYSIWYG editor compatible with most CMSs. After installing and configuring properly the INJECT plugin for TinyMCE, the user will get an “INJECT” button inside the editor as shown in Figure 11; 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.

#### INJECT plugin for TinyMCE

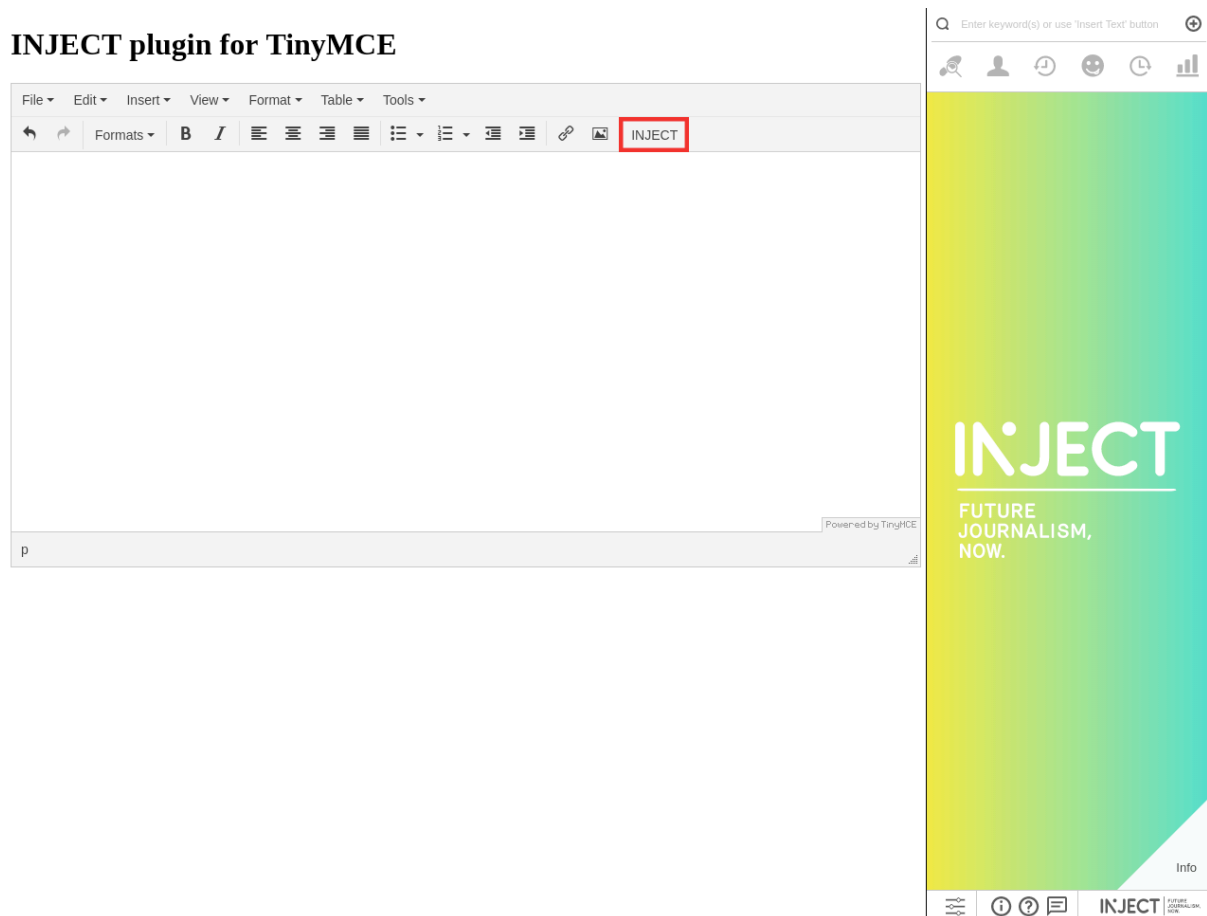


Figure 11. INJECT plugin for TinyMCE

Finally, we show the integration of INJECT with WordPress. 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 12, INJECT sidebar is launched inside WordPress and users can use INJECT exactly as described above.

The screenshot shows the WordPress admin interface for the INJECT plugin. The page title is "INJECT plugin for WordPress". The left sidebar shows the "Pages" menu. The main content area displays a text editor with a red box around the "INJECT" button in the toolbar. The right sidebar shows the "Publish" section with status "Published", visibility "Public", and a "Featured Image" section. A large vertical banner on the right side of the page reads "INJECT FUTURE JOURNALISM, NOW."

Figure 12. INJECT user interface for Wordpress

---

## 4 Summary and Outlook

---

WP1 has undertaken technical work to address the user requirements and research ideas, and to integrate these requirements and ideas into a single coherent architecture incorporating existing software available by City University, ICCS and Explaain. The current deliverable documents the status of the INJECT architecture and describes the system's main building blocks and functionalities, as these are available at M6 of the project.

The architecture identifies the following key components divided into three main layers: (a) User Interaction Layer which comprises a Google Docs Add-on interface, a CMS interface and stand-alone version of INJECT. (b) Application Layer which currently implements (i) the *creative search service* that manipulates text from the editor to generate search queries that form each input to the 6 implemented creative news angles; (ii) the *news aggregation and content extraction service*, called Presser, which searches over 100 news sources using creative search queries generated by the creative search service; (iii) the *story augmentation service*, which parses and enriches search results with Explaain cards. Moreover, a fourth component, so-called the *persistence component*, is planned for M12. (c) Data Layer which provides access to internal and external resources and content repositories.

The final version of D1.2 will implement the architecture presented herein by refining the existing user interfaces and components based on on-going user evaluations and testing. Moreover, the final version of D1.2 will include additional components supporting user authentication and authorisation, persistency and social media search.

---

## 5 References

---

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

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.

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](http://open-source.ebswift.com/WordNet.Net).