



UNIVERSITY OF UTRECHT
DEPARTMENT OF INFORMATION AND COMPUTING
SCIENCES

Master Thesis in Business Informatics

A Reference Architecture for a Dynamic Competitive Intelligence Solution

Author:

Alex Cepoi

Supervisors:

dr. Marco Spruit
dr. Ad Feelders
Paul van der Hulst
Taco Hiddink

22nd August 2014
Academic Year 2013/2014

Abstract

In today's increasingly fast-paced business environment, Competitive Intelligence (CI) solutions are the key to enabling companies to stay on top of the changes in the competitive environment in which they are active, by leveraging actionable intelligence regarding the competitive landscape in an automated manner. It may come as a surprise then, that although CI is a few decades old, there is little knowledge available regarding the implementation of CI system. The aim of this thesis is to develop a modular technical architecture for such a system, serving as a reference for future implementation projects.

We perform a comprehensive literature review and exploratory interviews with companies and industry specialists, identifying the main requirements as well as the architectural fragments proposed using a coding technique based on grounded theory. We then construct a modular technical architecture, which could serve as a reference for CI implementation projects. A single use case analysis is conducted by implementing and creating and evaluating a prototype for a company active in the *maritime & offshore* industry. Lastly the architecture is evaluated by a group of experts and refined in a feedback loop.

Acknowledgements

I would like to *thank God I finished this thesis*, but also my supervisors, Jibes B.V., who facilitated my research internship, the interviewees and experts who agreed to participate as well as everyone who offered to help with feedback and guidance one way or another (Silja Renooij, Karl Werder, Arjan Verhoeff, Majid Bahrepour, etc.). Last but not least, I would like to thank my family and friends who put up with me.

Contents

1	Introduction	1
1.1	A brief history of Competitive Intelligence	2
1.1.1	Early beginnings	2
1.1.2	CI as an information system	3
1.1.3	CI in use today	4
1.2	Problem Statement	5
1.3	Research Trigger	6
1.4	Thesis Outline	7
2	Research Design	8
2.1	Research Questions	8
2.2	Research Method	11
2.2.1	Literature Review	12
2.2.2	Expert Interviews	13
2.2.3	Reference Architecture	13
2.3	Use Case Analysis	15
2.4	Expert Panel Evaluation	15
2.5	Overview: Process-Deliverable Diagram	15
3	Literature Review of CI	21
3.1	Key Intelligence Topics	23
3.2	Data Sources	27
3.3	Solution Requirements	29
3.4	Architectural Fragments	33
3.5	Techniques	44
4	Interviews	51
4.1	Interviews with companies	53
4.2	Interviews with CI professionals	54
4.3	Review	56
5	Reference Architecture	59
5.1	Design Challenges	59

5.2	Design Process	60
5.3	Architecture Design	63
5.3.1	Quality Attributes	63
5.3.2	Collection and Storage	65
5.3.3	Analysis	69
5.3.4	Information Dissemination	75
5.3.5	High-Level Architecture	76
5.4	Binding Time Decisions	77
5.4.1	Design binding time decisions	77
5.4.2	Implementation binding time decision	78
5.5	CI system implementation process	79
6	Analysis	80
6.1	Use Case	80
6.1.1	Overall Design	80
6.1.2	Topic Exploration	83
6.1.3	Client Feedback	93
6.1.4	Lessons Learnt	95
6.2	Expert Validation	95
6.2.1	Solution Requirements	96
6.2.2	Data collection	97
6.2.3	Analysis & Dissemination	98
6.2.4	Artefacts changes	98
7	Discussion	101
7.1	Limitations	102
7.1.1	Literature Review	102
7.1.2	Reference Architecture	102
7.1.3	Use Case	103
7.2	Future Research	103
8	Conclusions	105
	References	107
A	Papers Reviewed	113
B	Interview Protocol	116
C	Diagram Symbols	119
D	Topic Exploration – Sample Report	120

List of Tables

2.1	Activity Table	17
2.2	Concept Table	20
3.1	Concept Categories	23
3.2	Solution Requirements	30
3.3	Fragments Comparison Table	42
3.4	Fragments Activities	43
3.5	Techniques Descriptions	47
5.1	Design Decisions	62
5.2	Advantages and Disadvantages of using an analysis module as a postprocessor	70
6.1	Topic Exploration – Activity Table	90
6.2	Topic Exploration – Concept Table	91
6.3	Correlated Tags Evaluation	94
6.4	Rule and Change Detection Evaluation	94

List of Figures

2.1	Process-Deliverable Diagram of Research Project	18
3.1	Model of Key Intelligence Topics	26
3.2	Main architectural phases	34
3.3	Zhao and Jin (2011) framework for CI solution	35
3.4	Zhao and Jin (2011) social-network-based credibility evaluation model	35
3.5	Ziegler (2012) framework for CI solution	36
3.6	Wei and Lee (2004) event deduplication	37
3.7	Hu and Liu (2004a) opinion mining	39
3.8	Dai (2013) DAVID system	40
3.9	Liu, Shih, Liao and Lai (2009) event change detection	41
3.10	Model of Usable Techniques for Analysis	48
3.10	Model of Usable Techniques for Analysis (cont)	49
3.11	Model of Usable Techniques for Dissemination	50
4.1	Interviewees Table	52
5.1	A Very High-Level Overview	64
5.2	Collection Architecture Overview	69
5.3	Flowchart analysis module	72
5.4	Analysis Overview	73
5.5	Information Dissemination Overview	76
5.6	High-level Architecture Overview	77
6.1	Prototype Crawling	81
6.2	Prototype High-level Architecture	82
6.3	Prototype - Reference Artefact Coverage	83
6.4	Topic Exploration PDD	89
6.5	Experts Table	96
6.6	High-level architecture – before and after expert feedback	100
C.1	Flowchart symbols	119

Chapter 1

Introduction

The business environment has changed a lot during the last decades – constantly increasing competition due to globalisation, shorter product lifecycle, increased popularity of outsourcing as a means of cost reduction, these are just a few of the reasons why companies nowadays need to exploit a lot more information about the competitive environment on which to base their strategic decisions (Zanasi, 2001). They need to perform thorough analyses before committing company resources towards a product which may not last long on the market, due to fierce competition or any other reason.

It is no wonder than that a lot of academic activity has been carried out over the years in the domain of CI, a field which focuses on monitoring the competitive environment with the aim of providing actionable intelligence that will provide a competitive edge to an organization (Safarnia, Akbari & Abbasi, 2011). Competitive Intelligence is thus an umbrella term encompassing many topics, such as competitor analysis, customers and markets analysis (Wright, Pickton & Callow, 2002) or R&D trends (Zanasi, 2001) among others.

Traditionally, most of the research is conducted from a business and strategic perspective, focusing on executive attitude towards CI (Wright, Bisson & Duffy, 2012), business processes (Bose, 2008) and governance of CI projects (McGonagle & Vella, 2012), but remaining technology-agnostic. Not surprisingly, many managers perceived CI to be useful, but with a low return on investment, acknowledging that “CI is choked if only used at a strategic level” (Wright et al., 2002).

The field suffered a revitalisation with the recent proliferation of *consumer-generated media*, consisting of unstructured content in the form of opinions and feedback from a variety of forums, weblogs and social media (Glance et al., 2005). A study performed back in 1998 shows that 90% of all information needed by a company to make informed critical decisions was already available

on the Internet (Teo & Choo, 2001), but Oder (2001) noted that the software industry was “a long way from delivering a satisfying business or competitive intelligence solution”.

The large growth in the size of unstructured content created by consumers as well as the increasing availability of traditional media on the Internet have brought the necessity of quantitative analysis of this information via *Competitive Intelligence Capturing Systems* (Ziegler, 2012), tasked with collecting and performing analysis in an automated manner. Our work focuses on the technical architecture of such a system, which automatically collects, analyses and disseminates insights throughout the organisation based on its specifics and requirements.

1.1 A brief history of Competitive Intelligence

1.1.1 Early beginnings

Competitive Intelligence is an old concept, first appearing in the 60s under the name of *Competitive Data Gathering*. This is the first of the 4 stages of CI, focused primarily on data acquisition (as classified by Prescott (1995) in his extensive review of CI history). The second stage manifests itself under the term *Industry and Competitor Analysis*, seeing a shift in efforts from collection to analysis techniques, which still remain rudimentary. The late 80s brought the first reference to *Competitive Intelligence* and a focus on the role of information systems in CI and formalisation of business processes needed for a successful CI program. The fourth stage of CI, as predicted by Prescott (1995), would be called *Competitive Intelligence as a Core Capability*, with a change towards a more strategic use of CI and an emphasis on qualitative analysis techniques.

The paper’s predictions on the future of CI lost touch with recent developments, predicting shifts from quantitative to qualitative analysis, which make little sense in today’s world of big data (understandingly since it was written around the birth time of modern Internet). However it does highlight the large attention the field has received in its formative stages throughout the previous decades.

Rouach and Santi (2001) (among many others) decompose a typical CI program into 4 major phases:

#	Phase	Description
1	Planning	secure funding, identify intelligence requirements, identify data sources. . .
2	Collection	gather the data to be analysed from the various data sources
3	Analysis	perform analysis on the data in order to get intelligence
4	Dissemination	distribute the generated insights to the interested parties inside the company

The process of identifying corporate CI requirements (part of the planning phase) has been well treated in literature. As a result of extensive interviews, Herring (1999) identifies and describes three types of Key Intelligence Topics (KITs) that can encompass most executives' intelligence requirements: strategic decisions and issues, early-warnings for new market developments and key players in the market. He notes that "their needs were rather similar, only the specifics were different" and raises some issues that need considering like CI organisation, barriers to sharing intelligence or credibility evaluation. A different study performed by Fahey (2007) ends up with a slightly different classification into: marketplace opportunities, competitor threats, competitive risks and key vulnerabilities.

Topics like executives' perception of competitive intelligence and their involvement in company CI initiatives have also been popular among researchers (Wright et al., 2002; Smith, Wright & Pickton, 2010).

1.1.2 CI as an information system

At the beginning of the 21st century, it became apparent that most of the information needed to make strategic decisions is openly available on the Internet, and CI involved applying data mining techniques on public sources (Zanasi, 2001). With the growth of user-generated data and the popularity surge of web sources, the collection, analysis and dissemination phases would now be performed by sophisticated CI information systems, rather than by CI professionals.

de Oliveira et al. (2004) propose the use of text mining techniques in e-mail sources as a starting point for an automated CI solution. They argue for a tagging based approach which would at first identify the concepts present in the sources, and then use an association rule learning method in order to discover associations between them.

Zhao and Jin (2010a) advocate a three tier architecture, focused on crawling web sources, using Named Entity Recognition (NER) (a text mining tech-

nique) in order to identify the entities and relations involved in each text and mapping them into an ontology built for the competitive environment in which the company is active, due to their flexibility in modelling business profiles, events or business relations (Zhao, Jin & Liu, 2010). In two other papers, they argue for generating insights from the ontology and analysing their credibility using a social-network based model, but do not go into any technical details (Zhao & Jin, 2011, 2010b).

Dey, Haque, Khurdiya and Shroff (2011) agree with the idea of using an ontology for the conceptual modelling of entities and relationships, but they also propose using a method of topic-based clustering based on latent Dirichlet allocation and rule-based labelling system for each cluster. Such a labelling system would provide little scalability given the static nature of the rules, but suggest machine learning as a means to create an evolving system which fine-tunes itself subject to continued usage. This involves having the users of the system review its results (like misplaced labels), thus creating a training set, then used by the system in order to find the optimal parameter values which minimize its error rate. However, the idea is not detailed, nor tested in practice. They also recognize the importance of social media, but not as a means of credibility evaluation, but rather as a source for evaluating public opinion, brand popularity or competitor news.

Mikroyannidis, Theodoulidis and Persidis (2006) agree to the importance of using NER, but see entities as annotations in the original articles, not as part of an ontology, an idea endorsed by Ziegler (2012) as well.

1.1.3 CI in use today

Generally, Competitive Intelligence has been considered to be a domain mainly for large companies, as “82% of large enterprises and over 90% of the Forbes top 500 global firms adopt CI for risk management and decisions” (Xu, Liao, Li & Song, 2011). Many researchers disagree, suggesting that a CI “seems to be the key ingredient for success in today’s uncertain business environment”, irrespective of company size (Priporas, Gatsoris & Zacharis, 2005). They argue that a tailored competitive intelligence system is entirely achievable by most small companies on low budgets, but they do note that the ultimate goal should be a bespoke system developed in-house to cater specifically to company requirements (Wright et al., 2012, 2002). Vedder, Vanecek, Guynes and Cappel (1999) notes, however, that an “actionable, effective competitive intelligence requires a steady, ongoing program. Anything less sharply reduces the usefulness of the effort.”.

Adoption rate for Competitive Intelligence solutions is still much smaller than that of Business Intelligence solutions. We attribute this effect to the scarcer

and generally unstructured format of competitive intelligence, which consists mostly of interactions and relations, instead of hard, internally available data such as sales or profit numbers.

1.2 Problem Statement

Despite a growing academic interest in tackling the technical challenges of implementing a CI solution, there is little consensus today on how such a system should be architected. To this day, gathering competitive intelligence analysis still remains “a highly specialized activity and difficult to automate”, but all the more necessary in today’s business environment (Dey et al., 2011).

As seen above, some suggestions for implementing such a system have been presented (Zhao & Jin, 2010a; Dey et al., 2011), but they have not been validated in practice by a prototype or cross-analysed for similarity. In fact, few of them acknowledge the others’ work as related literature. Furthermore, issues like generating insights or analysing their credibility, however, have received little attention. Zhao and Jin (2011) sets forward an idea for using a social-network based model for credibility analysis of CI insights, but does not go into any details on how this model would be used, dismissing this as an “implementation detail”.

To the authors’ knowledge there is no in-depth analysis of what technological solutions should be used to meet the different requirements of the CI system components, or how one would architect a CI solution based on a set of requirements.

With such a big gap in literature, it is of little wonder that the development of a Web-based competitive intelligence system is still an ongoing-endeavour in most enterprises (Zhao & Jin, 2011). Like most decision support systems, competitive intelligence solutions should perform increasingly complex functions, such as reporting (presenting what happened), analysing (discovering why something happened), predicting (what will happen) and activating (the highest level of sophistication, helping managers achieve certain goals). Most companies, however, are either not aware of competitive intelligence altogether or stuck in the reporting phase on the evolutionary scale (Bose, 2008).

1.3 Research Trigger

The research trigger consists on a customer request of Jibes B.V.¹ from a client activating in the offshore & maritime industry. They were interested in a competitive intelligence solution that would effectively assess market threats and opportunities and propose courses of action based on trends and the delivery capacity of their competitors.

They argued that, in their industry, the capacity of honouring a request quickly has a huge impact in both the chance of winning an auction and in the contract cost, as most customers are willing to pay a premium price for a quick delivery. Since each sale involves high profits it is becoming increasingly important for the sales staff to properly assess the position of their competition in order to customize their offer so that they win the contract and at the same time maximize their profits.

Of course, designing such a system will likely be driven by the need to answer several such use cases. Zhao and Jin (2011) describe how presenting reports about competitors or events is a must for any competitive intelligence system, but identifying business relations is not so straightforward, mostly due to the fact that a lot of vendors go to great extent to hide their suppliers or customers from their competition. Most of the time though, this is the type of intelligence decision makers are mostly interested in, especially when devising corporate strategy.

The problem becomes even more complex when we consider the fact that requirements and implementation of a competitive intelligence system are largely dependent on the domain in which it is active (Zhao & Jin, 2011). It is thus very useful to realize what parts of such a solution need to be customized to a user's needs and what parts can be reused. Taking the example above, while a competitive intelligence solution for the maritime & offshore industry will require a very well thought-out analysis of their leads in order to ensure winning auctions, this will not be true in the case of low-margin markets, where other methods need to be used to ensure profitability.

The focus of our work will be the enterprise sector. There are other uses for CI solutions, such as regional benchmarking (for governments to monitor the competitiveness of economic regions), and even if our architecture could be adapted to a large extent for these scenarios, we restrict to analysing the enterprise sector for the sake of simplicity and having well-defined scope.

¹Jibes B.V., Sleepboot 13, 3991 CN, Houten, Netherlands

1.4 Thesis Outline

The thesis starts with the presentation of the research design, describing the research questions and research methods we used in order to achieve them. Chapter 3 presents a semi-structured literature review together with an overview of the most common key intelligence topics, solution requirements and architectural fragments identified, while chapter 4 covers the interviews we had with companies involved in the process of implementing a CI solution as well as CI professionals.

In chapter 5, we describe the designing process which lead to the reference architecture, together with the rationale for each design choice and binding time decisions, which are decision left for the architect to be made depending on project specifics. The analysis chapter describes our efforts in implementing a prototype and the topic exploration analysis module, which was the focus of the use-case. We describe the changes we made to an algorithm proposed in the literature, present sample results and do get feedback response from the client. The chapter ends with the expert assessment of the architecture and the changes they introduced.

Chapter 7 describes the main issues we encountered, limitations of our work as well as possibilities for future research, while the conclusions chapter presents the main deliverables of the research.

Chapter 2

Research Design

2.1 Research Questions

Taking into account the issues formulated in the previous chapter, our main research objective is:

To develop a technical reference architecture for a dynamic competitive intelligence system, which can be easily customized depending on a specific set of requirements.

When discussing software architecture representation, we use the concepts of structure (a set of architectural elements) and view (a particular representation of a structure). Bass, Clements and Kazman (2012) distinguishes between four major categories of structures:

module a collection of architectural elements with a well defined functional responsibility

component an architectural element with a well defined runtime behaviour; it is a principal unit of computation (e.g. services, peers, clients, servers, filters, etc.)

connector an architectural element which ensures interaction between components (e.g. call-return, process synchronization operators, pipes)

allocation a representation of how the system will relate to non-software structures in its environment (e.g CPUs, file systems, networks, development teams, etc.)

We will focus specifically on describing modules, components and connectors, since allocations are largely implementation details. For architectural views, we will be using flowcharts to describe the architecture and interactions.

The purpose of this reference architecture is to serve as a starting point for companies wishing to develop their own competitive intelligence solution. There are many types of reference architectures depending on their scope. Mellish et al. (2006) distinguish different types of classifications:

- abstract (designs and specifications) or concrete (low level implementation details)
- prescriptive (focused on one implementation) or flexible (allowing a range of implementations)
- functional specification (high-level comprehensive overview of possible modules and functionalities) or data specification (low-level reference of how data is actually exchanged between modules)
- task-oriented (system has a narrow scope) or generic (open-ended system, adjustable for a variety of purposes)

Our reference architecture aims to be abstract, since we don't want or see a need to enforce a specific technological stack to be used (with respect to specific projects and vendors), but we do want to define the different components and connectors in the system, how they process and exchange data. The system should have a certain degree of flexibility to the extent of allowing more implementations, but we don't focus on achieving completeness of possible implementation scenarios. Nor do we believe that to be possible at this point, given the little knowledge about the technical aspects of such a solution. We aim to make our architecture generic in order to be useful as a reference starting point for similar endeavours and we see no necessity for narrowing its scope.

There are three main phases in a competitive intelligence solution according to Bernhardt (1994) among others: data collection & storage, analysis and information dissemination. While we are not able to provide a comprehensive overview of all analysis modules possible, we focus on a high level architecture which takes into account the most common analysis requirements identified.

In the light of the above, the main research question this thesis addresses is:

What are the main components of a dynamic competitive intelligence system, how do they interact and what techniques can be used to implement them?

In order to help us in answering this research question, several sub-questions have been developed:

1. *What are the common requirements for a CI solution (the features of the solution and the types of insights to be produced)?*

We distinguish two sets of requirements that our solution should be

able to cater to: key intelligence topics and solution requirements. Key intelligence topics refer to the nature of the insights produced and are subject to the specifics of the market in which the company operates. Naturally, companies in a highly innovative market will be more interested in a CI solution focusing on new technology trends, whereas those in more traditional markets might be more interested in disruptions in their supply chain.

The second type of requirements we must satisfy are solution requirements, which refer to what the system should be able to do, rather than the type of intelligence to produce. Typical examples might be credibility evaluation or the possibility to match external information with internal data (such as sales data) in order to generate even more useful insights. Some companies will have a strict security policy within the organization, that generated insights need to pass before being presented to the user. This not only involves tracking the sources for each of the generated insights, but also deciding if and on what level of aggregation data can be presented. It is possible that these requirements can have an impact on the architectural structure of the system, so it is important to identify them before making key architecture design decisions. The resulting architecture has a modular design, so that each solution requirement can be implemented as a module which and integrated into the final solution at any point in its implementation.

2. *What components should be part of a competitive intelligence system and what is their role?*

Our first task is to split the system into modules each having separate areas of responsibility and then try to decompose each of them into components. The components should be loosely coupled to allow for quick enabling, adjusting, monitoring and re-use in other architectures. To exemplify, in the case of the framework proposed by Zhao and Jin (2011), we can distinguish four major components: an extractor for web sources on the Internet, a NER component which annotates entities in sentences in order to identify facts, a rule-based CI insight generator which processes facts and a credibility analysis module for assessing the credibility of the insights. Creating a high-level architecture starting from these components allows us to enforce a modular design, where unneeded functionality can easily be removed or implemented at a later stage.

This question seeks to identify all the possible components of a CI solution needed to satisfy the functional requirements identified above.

3. *What techniques can be used to achieve the objectives for each of the*

components identified?

Having identified the main components needed and their runtime behaviour and interactions, we proceed to analyse the state of the art techniques and how they can be combined in order to achieve the requirements for each component. For example, many techniques can be used for processing the text content extracted from web pages. While NER is an option, other popular techniques like tagging and rule-based pattern mining have been proposed (de Oliveira et al., 2004). One possible solution to improve the performance of NER is by cross-referencing the information extracted from one document with multiple sources (e.g. a news item reported on multiple newspapers).

As is the case with most design science problems, it is not possible to provide a complete overview of all techniques that can be used to achieve a particular purpose, but we strive to find an implementation design for the most common analysis modules (Hevner, March, Park & Ram, 2004).

Given the research questions presented above, the main deliverables resulting from this research would be:

- a list of the types of intelligence a CI solution should generate
- a list of common requirements for a CI solution
- a technical reference architecture for a CI solution (main deliverable)
- implementation details for most common analysis modules identified

2.2 Research Method

In light of the research questions described above, we now present the research method. Considering the main deliverable is a reference architecture, we follow the 7 guidelines proposed by Hevner et al. (2004) for design-science research. The research method is constructed with the purpose of creating an artefact, although the focus is more on the architecture itself rather than other artefacts useful in the development and use of the CI solution. We design the research method based on an iterative process with an evaluation plan which includes proof by construction.

The research will be conducted in several steps, which will be detailed in the subsections below.

2.2.1 Literature Review

A literature review has been performed in order to identify requirements as well as proposed architecture fragments and the techniques which can be used in designing a competitive intelligence solution. Some of the ideas put forward in literature have already been presented, but the purpose of this step is to perform a more thorough examination of the available research and to identify common ideas presented by researchers as well as issues on which they disagree.

In order to have a clear overview of the process, we will adopt a *snowball* reviewing method consisting of the following steps:

1. Start from the keyword “competitive intelligence”
2. Use the Google Scholar¹ search engine and extract first 10 results (excluding patents and cites), filtering out papers which fall out of scope.
3. For each paper, perform literature review based on grounded theory using a qualitative data analysis platform called NVivo².
4. For each paper, look up relevant papers which cite or are cited by it. If any new results are found, go to previous step.

We performed the above sequence twice. On the first round, we used open coding in order to identify the core concepts. In the second round, we repeated the procedure, searching for “competitive intelligence” papers published after 2009, in order to identify more recent work on the field and to limit the search engines’ bias for older articles (due to their likelihood of having a higher citation count). On the second pass we also switched to using selective coding, as we have already identified the core set of concepts. Due to the huge number of papers which have no or little relevance to the technological side of CI solutions (focusing rather on topics such as business processes or necessary changes in company culture), we only analysed technically-relevant papers when looking at citations, going as far as 4 levels in depth.

This structured process should ensure we have identified most important literature research, and follows a protocol which is to a certain extent reproducible (although largely reliant on the behaviour in the search engine).

¹Google Scholar, <http://scholar.google.com/>

²NVivo, http://www.qsrinternational.com/products_nvivo.aspx

2.2.2 Expert Interviews

The literature review has exposed the key intelligence topics of the industry as well as some of the functional requirements for a CI solution. In order to make sure these are still aligned with recent developments and in order to better explore the technical requirements of a CI solution, we performed expert interviews with both companies involved in implementing such a programme and CI professionals. The selection criteria include expertise, lack of conflicts of interest and, of course, willingness to participate.

Due to the immaturity of the market in what regards CI solutions and to the complex and open-ended nature of the topic, we decided against a quantitative method for exploring these requirements. The use of semi-structured interviews will allow the participants to talk freely and presents a greater opportunity for exploring the topic. Their suggestions will be noted and later cross-analysed with the ideas identified in literature.

2.2.3 Reference Architecture

Now that we are aware of the functional requirements of a CI solution in the industry as well as main ideas proposed by the academia, we combine the two inputs in order to create the architectural design.

Method

The first step in architecting any information system is gathering requirements. By using the coding method in grounded theory, we scan the scientific papers and the interview transcripts and extract the concepts describing KITS, solution requirements or techniques to be used in implementation and label them. This process is iterative, as labels are continually refined and duplicates are merged. This allows for a clear overview of which ideas are endorsed more in the implementation of a CI solution and allows us to see which of them are used together.

Once this is done, we construct a reference architecture by following the attribute-driven design method proposed by Bass et al. (2012). This involves first decomposing the design into smaller parts, which we design individually. In designing each part, we first gather all requirements identified above which are relevant for the respective part, we generate a design and test it in a prototype. A thorough evaluation of all implementation and design decisions is out of the scope of this thesis and likely to be subjective and incomplete, given the large variety of use cases in which it can be applied and other influencing factors (Hevner et al., 2004), but we plan to evaluate

at least the ones presented in the literature, and record the rationale behind the recommendations made. This would serve as a reference for future implementation endeavours and allow companies to evaluate how well our model fits their specific use case.

Design Quality Attributes

Any architectural design must satisfy both functional requirements and quality attributes. A functional requirement is simply a property which states a particular function or qualification of a system (Bass et al., 2012). A quality attribute is defined as measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders; effectively a qualification of a functional requirement or sometimes called a *non-functional requirement* (Bass et al., 2012).

There are many quality attributes. Among the most commonly used in software architecture we can mention: availability, interoperability, modifiability, performance, security, testability, usability (Fricke & Schulz, 2005; Bass et al., 2012). The most important quality attributes we are interested in are availability, modifiability and testability, as the others either depend a lot on implementation details (e.g. performance is mostly dependent on chosen analysis algorithms) or the existence of a particular context (e.g. interoperability with existing systems).

Modifiability is very important as it is unlikely that a specific company project will have an exact match with the requirements we identify. The architecture should allow for customization in what regards adding or removing certain functionality as well as customize the data flow and storage in the system. Loose coupling (lack of complex dependencies between modules) and high cohesion (each module has a few responsibilities and full control of those responsibilities) are the key features we focus on in order to achieve this. Since most CI requirements have good commonality and generally differ only in specifics (Herring, 1999), we see little benefit in discarding modifiability. We emphasize this aspect because implementing a CI solution, like BI solutions, requires a company-wide effort and needs to be a continuous and ongoing process, constantly leveraging new possibilities for generating intelligence (Vedder et al., 1999), so it will likely suffer many changes across its lifetime.

Testability is another way of achieving modifiability by forcing components (especially analysis modules) to be individually run and tested in a sandbox environment. This not only eases for development of separate functionality, but creates an abstraction layer between each module and the rest of the system.

Availability is an important quality attribute in any large, enterprise-wide

information system and ensures that the system is up and running even in case of failures in unrelated parts of the system.

2.3 Use Case Analysis

Our primary method for evaluating the artefact is by performing a single use-case analysis on one of the clients of Jibes for a CI solution. This involves implementing a prototype and assessing how well it performs. Due to the large amount of work involved in this operation, we decided to perform a single use-case study. While this is insufficient to rigorously assess all aspects of the architecture, it will at least provide an opportunity to test our artefact and see how well it can be applied in practice, and is also a part of the attribute-driven design method by Bass et al. (2012) which we have chosen to follow.

We start from a set of project requirements, adapt our reference architecture and implement a prototype. Given the constraints of the research project, it is not feasible to perform the use case analysis on all the aspects of the architecture. For example, a security audit of the entire solution or evaluating the system under prolonged continued usage is out of the scope and budget of this research. We evaluate its overall performance by asking the beneficiary to rate the correctness of the insights generated by the system.

2.4 Expert Panel Evaluation

The architecture model is then assessed by an expert panel of professionals with experience in system design and architecture. They have been asked to evaluate and give feedback on the reference architecture and their assessment serves as a second validation method. The validation follows a semi-structured format as they are asked to comment on the main quality attributes as well as offer suggestions for improvement.

Suggestions are analysed and incorporated back into the architecture.

2.5 Overview: Process-Deliverable Diagram

In order to create a more insightful view of the main stages of the research project and how they interoperate, we use a method developed by van de Weerd and Brinkkemper (2008), called method engineering. The output is a process-deliverable diagram (PDD), which depicts the activities which will be carried out, the deliverables resulting from them, as well as how

they all contribute to the main deliverable. This should allow for a better understanding of the main phases of the research and the role of each activity in the research project.

The PDD of this research project is depicted in Figure 2.1. The activities presented in this PDD are further described in the activity table of Table 2.1. A description of the concepts present in the PDD can be found in Table 2.2.

Activity	Sub-Activity	Description
Develop Workplan	Define Problem Statement	Describe the issues that will be addressed by this thesis (lack of an architectural blueprint for a CI solution).
	Initial Literature Review	Perform literature study in order to have a clear understanding of what and how well the issue has been treated in literature.
	Develop Research Approach	Describe how will the research project will be structured in order to tackle the issues identified.
	Write Long Proposal	Document proposal for research project.
Literature Review	Identify KITs	Identify in literature the specific use cases that companies require of a CI solution to provide insights for (e.g. competitor investments).
	Identify Solution Requirements	Identify in literature what are the specific requirements for a CI solution, which affect the entire system, rather than the output produced. (e.g. credibility analysis of facts).
	Identify Techniques and Architecture Fragments	Identify techniques proposed in literature for implementing a CI solution (e.g. NER for identifying competitors in a news article) or larger architecture fragments.
Conduct Interviews	Assess KITs	Discuss with interviewees whether the KITs identified in literature are still valid for the today's industry.
	Assess Solution Requirements	Discuss about the CI solution requirements identified, whether they are representative and if there are others which were not identified.

	Assess Proposed Techniques	Discuss the main techniques identified or used in the industry in existing CI solution.
Create Architecture	Decompose Architecture	Based on the list of requirements and suggestions from the focus group, identify the main components and modules that form a CI solution, and how they interact.
	Create Implementation Plan	Taking into account the main components identified above, and proposed architectural fragments, create an implementation plan for each of the components, filling in the blank spots where necessary.
Use Case Analysis	Analyse Requirements	Identify the specific intelligence and solution requirements for the project.
	Adapt Architecture	Create a situational technical architecture by adapting our reference architecture to the specific requirements of the project at hand.
	Implement Prototype	Implement the actual system based on the specifications of the situational architecture.
	Evaluate Prototype	Test the prototype in a controlled environment by having the client assess the accuracy of its output.
Conduct Expert Assessment		Conduct expert panel evaluation, assessing whether the architecture satisfies requirements and quality attributes, and collect suggestions for improvement.
Finalize thesis project	Finalize thesis	Write the project results, documenting method, results and conclusions.
	Write Paper	Based on the thesis document, write scientific paper, subject to publishing.
	Create Final Presentation	Create the presentation to be used in the thesis defence.

Table 2.1: Activity Table

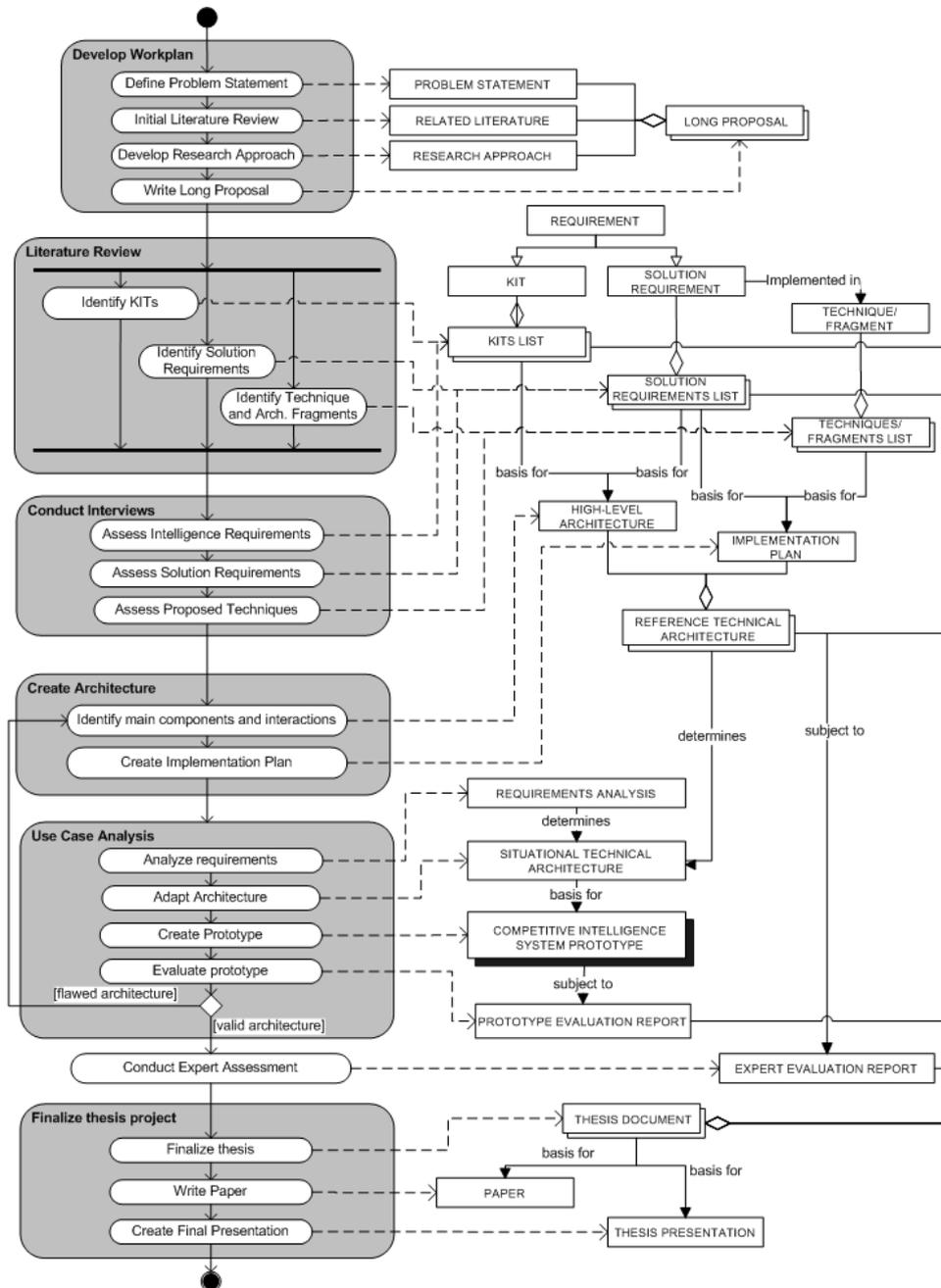


Figure 2.1: Process-Deliverable Diagram of Research Project

Concept	Description
PROBLEM STATEMENT	The real-world issue that this research project attempts to solve.
RELATED LITERATURE	Superficial literature review of the topic.
RESEARCH APPROACH	A description of the research method employed in attempt to solve the issue.
LONG PROPOSAL	A comprehensive report documenting problem statement, research questions and research method.
REQUIREMENT	Specific need that a project must satisfy – can be described as a use case, user story, etc.
KIT	Key Intelligence Topic, an example of information type that the CI solution must produce. (e.g. changing customer preferences).
SOLUTION REQUIREMENT	Functional requirements which affect the CI solution as whole, rather than it's output (e.g. credibility analysis).
TECHNIQUE / ARCHITECTURE FRAGMENT	Specific approach to solve a technical problem (e.g. a process based on association rule learning implementing event change detection).
HIGH-LEVEL ARCHITECTURE	An overview of the major components & modules which would satisfy the solution requirements identified, as well as their role and the data exchanges between them.
IMPLEMENTATION PLAN	Technical implementation design(s) for each of the components identified above. This will be constructed by using deductive reasoning, starting from the identified requirements and techniques proposed in literature.
REFERENCE TECHNICAL ARCHITECTURE	Deliverable consisting of the high-level architecture and the implementation plan of each of the components.

REQUIREMENTS ANALYSIS	A deliverable documenting the requirements of the company in the implementation plan of their CI solution.
SITUATIONAL TECHNICAL ARCHITECTURE	An adaptation of a reference technical architecture in order to cater to the project requirements.
COMPETITIVE INTELLIGENCE SYSTEM PROTOTYPE	The actual implemented system, capable of satisfying the intelligence requirements.
PROTOTYPE EVALUATION REPORT	A deliverable documenting the user feedback from the client regarding the evaluation of the prototype.
EXPERT EVALUATION REPORT	An assessment of all the comments made by the expert panel regarding the technical architecture, regarding its adherence to quality attributes and feasibility of implementation.
THESIS DOCUMENT	the entire project report, documenting all aspects of this research endeavour.
PAPER	scientific paper summarising thesis results.
THESIS PRESENTATION	The presentation which will be used during the thesis defence.

Table 2.2: Concept Table

Chapter 3

Literature Review of CI

Subject to the literature review process presented above, we performed a structured literature review, starting from a keyword and following back and forward citations based on relevancy (what is commonly known as the *snowball* method). While the first 10 direct results were all selected, only technical articles were selected from the large set of cited and citing publications.

This was motivated by the scarcity of technical articles on implementing Competitive Intelligence solutions compared to publications handling different aspects such as business requirements, historical overviews, adoption research on different markets, etc. While we considered these important as well, we found them to be of marginal use for the purpose of building a technical architecture.

Despite our attempt to be as impartial as possible, we have to acknowledge the fact that our judgement on the selection of the articles based on the technical aspects of implementing such a project can be subject to personal bias.

The papers were selected in two rounds: the first one by snowballing from the keyword “competitive intelligence”, resulting in 35 papers being selected, and the second by starting from a search for the same keyword, but for publications after 2009. The latter round resulted in the selection of 23 different publications. The motivation for this decision was that our initial round was largely composed from a series of older articles with a large number of citations, but a lot of which were outdated and mostly tackling the business perspective of CI. The full list of papers reviewed in each round is presented in Appendix A.

While we were originally planning to use a keywords based approach for selecting the relevant papers, we found we have gathered enough literature

after just the two rounds presented above using a *snowball* approach to literature gathering.

The literature review process hence resulted in the systematic evaluation of 58 papers, although many others not part of this process were also reviewed, i.e. in case they were treating a CI component more in depth. There is a large consensus among all publications in what regards the three major phases that should make up any competitive intelligence solution. These are (according to Bernhardt (1994) among others):

1. Collection and Storage

Identify available sources, extract and pre-process the raw data and store for convenient future access.

2. Analysis and Interpretation

Process the available data, run various analyses and extract insights from the data. This step is the most elaborate and transforms information into intelligence.

3. Information Dissemination

Distribute the right insights to the right employees all throughout the organisation.

Some authors add two other phases to the ones above. One is the initial step of planning which deals with the organisational aspects of assessing requirements and resources and directing the competitive intelligence project (Norling, Herring, Rosenkrans, Stellpflug & Kaufman, 2000) and the other is the final step of the feedback loop, which implies the constant tuning of the system to better adapt to company requirements and its users (Vedder et al., 1999). However, we chose to ignore these for now due to their lack of technical relevancy from an architectural point of view.

Concepts were extracted from the article based on grounded theory with the help of NVivo platform. This allowed us to keep track of which concepts were present in which sources and to be able to reclassify and organise them in multiple iterations as more and more concepts were added.

By using this reviewing approach, we eventually identified three categories into which to classify our concepts, as described in Table 3.1. We ended up with the same categorisation of concepts as Dai (2013) with respect to *Applications*, *Techniques* and *Foundations*, to which we added *Key Intelligence Topics* as opposed to their overarching objective of *Knowledge Discovery*. For the purpose of constructing our architecture we will focus especially on the applications (in order to analyse its main modules) and techniques (in order to describe an implementation plan for each of the modules).

We first present the key intelligence topics we identified in literature, that is the different kinds of intelligence that companies can or might be interested

Category	Description	Sample Concepts
Key Intelligence Topics	The pieces of information the solution should produce	Products & Features Patent Trends Government Tenders Job Postings
Applications	Functional solution requirements	Event Detection Credibility Evaluation Visualisation
Techniques	Individual algorithms and procedures which combined make up a solution functionality	Neural Networks Clustering Decision Trees
Foundations	Broad scientific areas of which techniques are part of	Statistics NLP

Table 3.1: Concept Categories

in, as well as the common types of data sources for obtaining them. We then look the the functional requirements of CI solutions. Lastly we present the main fragments of CI solutions implementations we find as well as the techniques which can be used to implement its various components.

3.1 Key Intelligence Topics

There are many ways of classifying the intelligence requirements of organisations, and likely all of them would be perfectly valid ways of doing so, with differences generally being subtle. Due to the interrelatedness of these KITS, it would be hard to argue for a specific classification of them and this exercise would likely not be productive.

Herring (1999) noticed that after performing extensive interviews with executives in almost every industrial sector, he found their intelligence requirements to be surprisingly similar, and only differing in specifics. He classified their needs in three large categories:

Key Players in the Marketplace

In-depth profiles of all entities participating in the marketplace, be they competitors, customers, suppliers, etc. This is probably the most basic requirement of any dynamic CI solution and the first it should

tackle.

Early-Warning Topics

Identify changes or possible changes in trends regarding all activities performed. This includes research or technological breakthroughs, changes in policies and regulations, sudden changes of interest of competitors, etc.

Strategic Decisions & Issues

This involves decision support for upper management regarding the strategic decisions and direction for the company. One of the most common implementations in this area is scenario analysis.

Another study performed by Fahey (2007) classifies these CI requirements slightly differently in 4 strategic inputs:

Marketplace opportunities

One big objective of any CI solution is to extend current business opportunities for improved market share, as well as explore new marketplace opportunities by tracking regulatory or technological developments in the fields in which the company is active with the intent of providing products which meet changing customer needs or possibility for expansion.

Competitor Threats

A CI solution must be able to timely identify competitor threats and ensure that the company's strategy wins against rivals.

Competitive Risks

Risks for carrying out a company's strategy are not determined only by competition, but also by changes in the marketplace itself, driven by changing customer preferences, suppliers or governmental regulations.

Key Vulnerabilities and Live Assumptions

Every corporate strategy is based on several assumptions about the evolution of the industry the company works in. It is also susceptible to certain scenarios that could critically affect it, without it involving the market it is active in or its competitors. Issues like possible publicity hits or strategic actions of non-competitors which inadvertently affect the company are typical examples of such cases. Identifying and monitoring these vulnerabilities as well as contingency plans to apply in these scenarios is the key to creating strategic advantage.

Safarnia et al. (2011) did a thorough evaluation of these, concluding that they all positive influence on the company's competitive advantage.

In the end, the CI program manager would have to prioritise these KITS depending on their usefulness to the users and the feasibility of implementa-

tion, given the current status and resources available. Our literature review resulted in identifying the following Key Intelligence Topics as depicted in 3.1.

As can be seen in the figure, we take a different approach and classify not by the types of insights, but rather by the type of entities we wish to monitor. We therefore identify three main types of entities we would like to extract intelligence about: companies (monitor the competitive position of competitors, companies in the supply chain, B2B clients, etc.), people (be they B2C customers, key employees, stakeholders, etc.) and markets (for general information regarding research activity & trends, regulators, policy makers, etc.).

Note the fact that while in most B2C businesses, analysis and monitoring of customers falls under typical BI solutions (analysing sales, customer segregation, etc.), in B2B businesses it is common that this falls into the CI category, since it involves much more rapport with the customers and this data cannot be treated quantitatively.

We opt for this classification of KITs because information about *competitors* is more descriptive than i.e. *marketplace opportunities* and maps better to the kind of information that would be extracted, whereas *marketplace opportunities* identifies the type of insights that would result as an output of the analysis phase. The impact of this choice is minimal as the insight list should be fairly complete and can be reclassified under any of the above schemes.

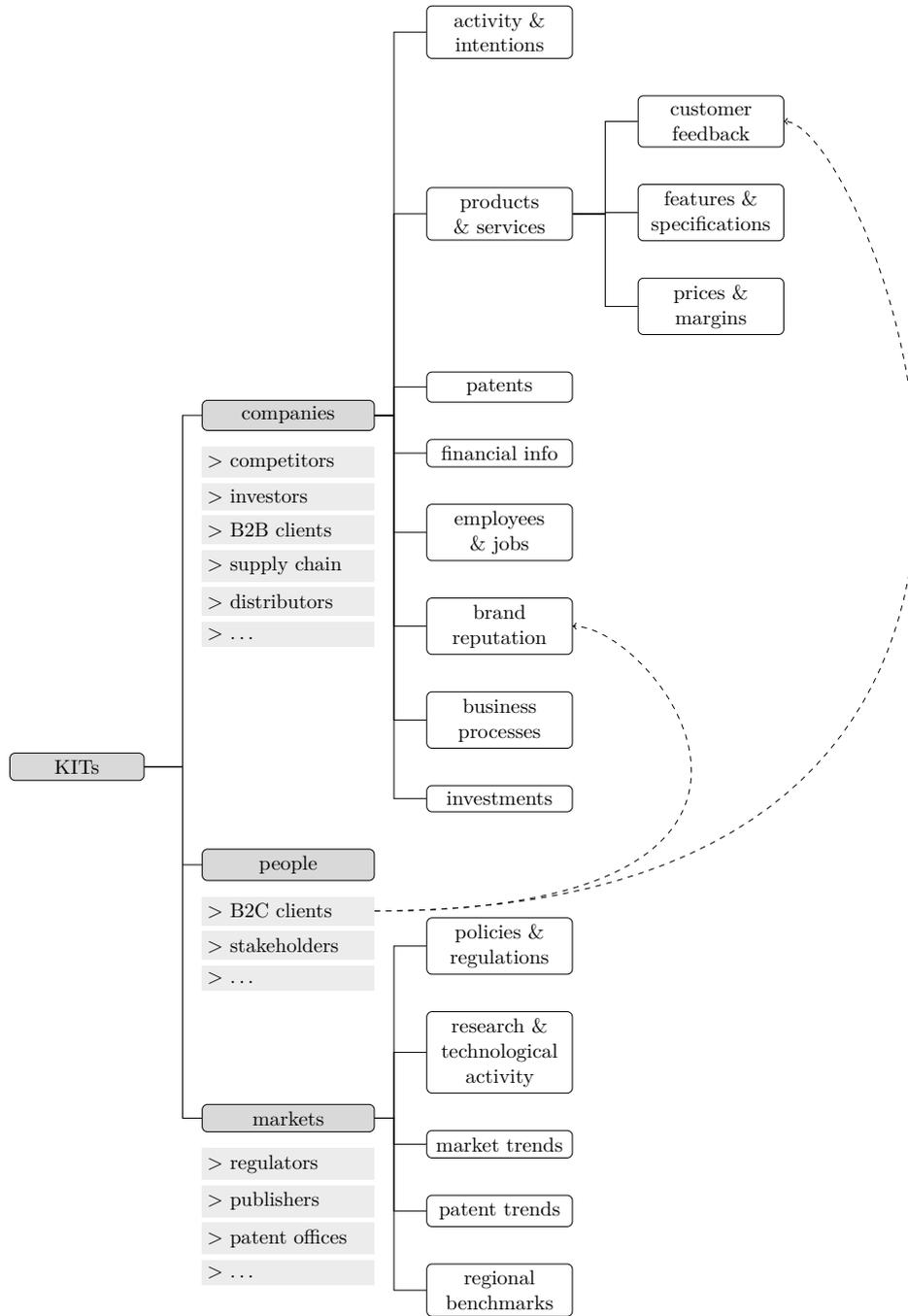


Figure 3.1: Model of Key Intelligence Topics

3.2 Data Sources

Historically, CI solutions have focused on exploiting unstructured information from the Internet, and it has been shown that 90% of the information needed to make critical strategic decisions is available as public data on the Internet (Teo & Choo, 2001). A CI solution must be able to integrate data from both internal and external sources (Kong, Fu, Zhou, Liu & Cui, 2007), and recently it is becoming more and more important to be able to integrate information extracted from structured data sources (e.g. data warehouses) with unstructured data (e.g. the web) (Fan, Wallace, Rich & Zhang, 2006).

The possible data sources we identified in our literature review are described below:

News sources

Traditionally the most common source for any CI solution has been news articles, newspapers or any other type of content approved by a publisher. These generally cover events and analyses from a more or less objective perspective, but may also include editorials and opinion articles, so proper assessment of subjectivity is critical for a good interpretation of the content (Xu et al., 2011).

User-generated content

Whether they are forums (or any sort of discussion groups), newspaper articles, blogs or social media sites, all these sources of user-generated content can be the source of new insights. These can provide opinions, feedback, insights into customer perception (Bose, 2008).

Commercial & open databases

There are many commercial online databases which contain large amounts of well-organized data on various subjects (Chen, Chau & Zeng, 2002) and, more recently, there is a large increase in the availability of open data. They can be of any form, ranging from trade journals, periodic market analysis reports, government tenders systems or company annual reports (i.e. financial statements). These are among one of the most important sources used by CI professionals (Chen et al., 2002).

Competitor websites

The public websites of competitors or other entities which a company comes in interaction with are a major source of valuable information. These can contain information about products & services, features and specifications, job opportunities, but also press releases and other insights about the direction in which the company is going. Although there is the obvious danger of extracting biased information, these websites should be, at the very least, the most comprehensive source

for information about a company's services (Xu et al., 2011).

Academic publishers & patent databases

Patent databases are a free and possibly one of the most reliable sources of technical information available (Zanasi, 2001) and, together with academic publishers, are the best source to go to in order to get an in-depth analysis of the future direction in technical innovation. By analysing these sources, one could gain insights regarding research trends, but also into your competitors' strategy, by analysing their patent applications and acquisitions. Academic publishers similarly present interest as their data can reveal the subjects of academic interest and implicitly the ones where scientific breakthroughs are likely in the near future.

Data warehouses

Integrating company (usually structured) data into the analysis is a very commonly cited requirement of a CI solution and, although this is usually the realm of BI solutions, we cannot dismiss the fact that few insights can be created without integrating both types of available data. Fan, Gordon and Pathak (2006) call this process of combining data mining with text mining techniques *duo-mining*.

Internal email systems

There have been some suggestions proposing that a CI system should also have integration with the internal email systems used within the company, so it can do various analysis (like sentiment analysis) on the interaction with the various parties (Bose, 2008). It is important to take into account, however, the threat this poses to employees' privacy and whether the benefits are worth it.

Direct enquiries

Another suggestion (albeit from an article published in 1994) mentions the use of cold calling the competitors offices (or various parties with which they interact with) in order to gather information (Bernhardt, 1994). While the morality of doing so is questionable, this source presents little importance for our technical architecture since these findings would have to be codified as a report in one of the companies internal wikis.

While it is important to train the solution on the actual dataset to be used in production, some standard datasets can be used for training and evaluating the system's performance. For news articles, the most commonly used are the TREC¹ dataset collections (the first of which is the Associated Press dataset). Most patent databases are freely available for download, e.g.

¹Text Retrieval Conference, <http://trec.nist.gov>

WIPO², USPTO³ or EPO⁴. DMOZ⁵ is the largest taxonomy classifying more than 4 million websites (Ziegler & Skubacz, 2006).

3.3 Solution Requirements

Simultaneously with extracting the KITs from literature we also identified the functional requirements for a CI solution. While the list is not meant to be a complete reference, we found that we did not uncover any new requirements during the interviews process. This gives us confidence in constructing an architecture based on the requirements we identified.

We classified the functional requirements into the three main phases of a CI solution (as described above) for a better overview. An overview is presented in Table 3.2, and we expand on them below.

1. *Track changes in posts*

In the context of solution requirements regarding the extraction phase, tracking changes in web pages is a commonly mentioned feature. This firstly involves monitoring newly added content in the data sources, since our solution is supposed to inform users on new developments without presenting duplicate information. Secondly, Chen et al. (2002) mentions the usefulness of this feature also in the context of user comments, which can accumulate long after the publication of an article.

2. *Collect both structured and unstructured data*

Initially, CI systems were designed to parse unstructured data, mostly as Internet started growing in popularity. More recently, however, emphasis is being put on the importance of integrating the unstructured data from public sources with structured data from data warehouses (Rao, 2003).

3. *Credibility analysis*

One of the most quoted features of a CI solution would be analysing the credibility of the extracted facts. Given the free nature of the Internet as well as the subtleties of natural language, it is becoming increasingly important to assess the reliability of extracted facts before trying to create insights from them. One suggestion would be to cross-reference sources which independently report on the same facts in the idea that the number of independent source positively correlates with reliability

²World Intellectual Property Organization, <http://www.wipo.int>

³United States Patent and Trademark Office, <http://www.uspto.gov>

⁴European Patent Office, <http://www.epo.org>

⁵DMOZ, Open Directory Project, <http://www.dmoz.org>

Phase	#	Functional Requirement	Sources (e.g.)
Extraction	1	Track changes in posts	4 (Chen et al., 2002)
	2	Collect both structured and unstructured data	2 (Rao, 2003)
Analysis	3	Credibility analysis	7 (Zhao & Jin, 2011)
	4	Topic exploration	5 (Ziegler, 2012)
	5	Document summarisation	2 (Bose, 2008)
	6	Scenario simulation	2 (Bose, 2008)
	7	Event deduplication	2 (Chen et al., 2002)
	8	Integrate structured and unstructured data	2 (Fan, Wallace et al., 2006)
	9	Product comparison	1 (Xu et al., 2011)
	10	Predict competitor data	1 (Cobb, 2003)
	11	Company, people and markets profiles	1 (Bose, 2008)
	12	Multilingual support	1 (Chen et al., 2002)
Dissemination	12	Ad-hoc querying	8 (Mikroyannidis et al., 2006)
	14	Monitoring & alerts	5 (Fan, Wallace et al., 2006)
	15	Personalised information routing	5 (Fan, Gordon & Pathak, 2006)
	16	Information visualisation	5 (Rao, 2003)
	17	Newsletter summaries	1 (Prescott, 1995)
	18	Security policies	1 (Bose, 2008)

Table 3.2: Solution Requirements

(Bernhardt, 1994). Zanasi (2001) suggests we also check data inertia in order to analyse how does one piece of information reported in one source get referenced by other sources, instead of being independently reported, and how does this influence credibility. Zhao and Jin (2011) is proposing a graph model for calculating credibility of generated insights based on the reputation of sources and the credibility of the facts extracted, but does not go into details of how to accomplish this.

4. *Topic exploration*

Topic exploration is an effective way of discovering similar or commonly used together concepts with the purpose of collecting new insights. By statistically analysing the co-occurrences of terms and concepts, one can discover patterns, trends as well as disruptors in specific markets. Ziegler, Skubacz and Viermetz (2012) proposes using this approach to explore customer feedback in relation to different features of a product.

5. *Document summarisation*

Another common mentioned feature is document summaries, which simplify understanding an document's topic by capturing its key points. This saves time in assessing its contents and helps in previewing the multiple sources for a specific event.

6. *Scenario simulation*

Prescott (1995) mentions that scenario analysis to be one of the most useful exercises a company could perform in order to assess the strategic implications of their actions. This involves finding patterns with historic data recorded from the past and predicting how would the environment react to a specific scenario. Analysing best and worst case possibilities allows management to make better informed decisions (Bose, 2008).

7. *Event deduplication*

Some news is reported by a lot of sources, some only by a few. It is paramount therefore that a CI system which aggregates data from multiple sources be able to distinguish when the same event is reported by multiple sources. This drastically reduces the amount of information the user is immediately exposed to, at the same time allowing him to drill down and investigate a particular event.

8. *Integrate structured and unstructured data*

While traditional BI systems concern themselves with extracting insights from the structured data in data warehouses, CI has generally concerned itself with unstructured data. Recently, however, great emphasis is being put on performing analysis capable of cross-correlating the two types of data, a concept called *duo-mining* by Fan, Wallace et al. (2006), which identifies it as a major future trend in CI.

9. *Product comparison*

Another useful functionality of a CI system is its ability to compare 2 products side by side based on customer feedback received regarding their features. Since these reviews is usually rich in comparative opinions, it is important to investigate the general public perception on how one company's products fare when put head to head to those of its competitors (Xu et al., 2011).

10. *Predict competitor data based on similar competitors*

Sometimes it is useful to estimate some missing information about a competitor by interpolating data we have on similar competitors (Cobb, 2003). Given a large amount of data, estimates for missing data for a particular company become more and more accurate, especially compared to human analysis.

11. *Company, people and markets profiles*

Building profiles for various entities in the marketplace is probably the most basic and common of functional requirements for any CI system. This information should be structured, historical and should be a building block for most of the more complex analyses (Bose, 2008). The reason it scores so low in our ranking is probably that most authors assume this is a basic feature of any CI solutions.

12. *Multilingual support*

Before any text analysis is performed the language must be identified in order for the algorithms to work correctly. If it is necessary to perform analysis on other languages, it is important to note the increased difficulty in finding proper implementations for text mining tools and corpuses.

13. *Ad-hoc querying*

Performing ad-hoc queries, whether in natural language or through an interface, is perhaps the most convenient and common method of extracting information from the CI system.

14. *Monitoring & alerts*

While ad-hoc queries are useful on their own, it is likely that an user is interested in monitoring certain topics closely related to his area of expertise or interests as well as being alerted when certain events of potentially high impact occur.

15. *Personalised information routing*

While monitoring certain topics is a useful way of staying up to speed, it has been reported that this can easily overwhelm the users with too much information and that most users do not know how to formulate persistent queries or topics to follow that properly represent their long-term information needs (Fan, Gordon & Pathak, 2006). They propose a different approach, by both allowing these subscriptions, and

adjusting preferences based on their usage behaviour. This type of personal information routing logic is a commonly requested feature in CI solutions.

16. *Information visualisation*

While out of the scope of this thesis, intelligence information visualisation for generated insights is another feature commonly mentioned in the literature. A well created visualisation can sometimes be the best way of conveying a message, and which type of visualisation to choose is not always obvious (e.g. competitor acquisition patterns by animated pie charts, research trends by bar charts).

17. *Newsletter summaries*

Periodic newsletters are a useful companion to ongoing monitoring in avoid missing important news due to overwhelming information. They are also useful as periodic review of the main events.

18. *Security*

Security policies inside companies regarding the access and use of CI solutions seem to be one of the least mentioned aspects of a proper implementation, most likely due to the perception that CI is a company-wide effort that all employees should have access to. Bose (2008) does mention though the importance that this data does not fall into the hands of competitors as that may expose intrinsic knowledge regarding the company's interests and strategy.

3.4 Architectural Fragments

As seen in the previous chapter, the three main phases of a competitive intelligence solution (according to Bernhardt (1994) among others) are data collection, data analysis and information dissemination. Data collection will be the phase responsible with extracting data from internal or external sources into a collection of entities we call posts. We define an post as a abstract article or unitary snippet of text together with all its metadata (e.g. date, author, source, user comments). The analysis phase will be running various algorithms in order to to create CI insights, which will then be disseminated to the end-users.

An overview of these phases can be seen in Figure 3.2. The only enhancements we presented is the distinction between pre- and post-processing and splitting the analysis phase into post-processing and the actual analysis. We consider pre-processing the elimination of clutter from data sources (i.e. in the case of the Web, we can have ads, navigational elements) while post-processing would deal with the annotating of the content (e.g. identifying part of speech,

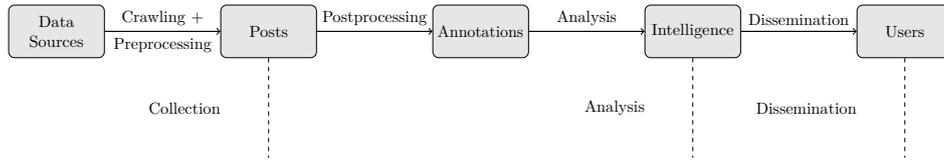


Figure 3.2: Main architectural phases

or named entities). Generally pre-processing should yield nearly perfectly accurate results, while the results of the post-processing activity are more subjective and its quality is hard to quantify.

Of course this is a high-level conceptual view which does not deal with issues like processes or data storage. We impose this division now in order to better be able to compare the different fragments presented in literature, but we will be using this throughout the remainder of this thesis.

We have identified a few fragments for CI implementations presented throughout the literature. We will be using flowcharts to represent architectural fragments throughout this thesis, the main symbols used are presented in Appendix C. Zhao and Jin (2011) proposes an architecture reliant on the extraction of entities and relationships using Information Extraction techniques (named entity recognition and entity relation extraction, aided by a domain dictionary) and modelling them onto an ontology. Using a rule-based technique, intelligence insights would be generated and be subject to a credibility analysis process (see Fig. 3.3). While not going into detail on how this process would work, he proposes a social-network-based evaluation model, where the credibility of each insight would depend on the credibility of the facts it constitutes from as well as the credibility of the sources from where these facts have been extracted (see Fig. 3.4). In the figure we have sources represented as yellow circles, facts as ellipses and intelligence as blue squares. The dashed arrows represent references between sources and are useful to identify if a specific piece of information is reported independently or citing another source. The results, annotated with the credibility data, would be exposed to a query processor and presented to the end-users.

Ziegler (2012) proposes a similar architecture, but proposes using a plain storage for posts. Indexes would be build this data by preprocessing the content (remove stopwords, perform stemming) and weighing the resulting terms. An inverted index would be built for the identified terms and calculated weights. All analysis generated data (resulting from trend detection, sentiment analysis, etc.) would be stored in a separate annotation storage system (see Fig. 3.5). He supports the idea of independent unsynchronized annotators which write their results to the same annotation layer database.

Dai (2013) identifies two main types of analysis which can be performed:

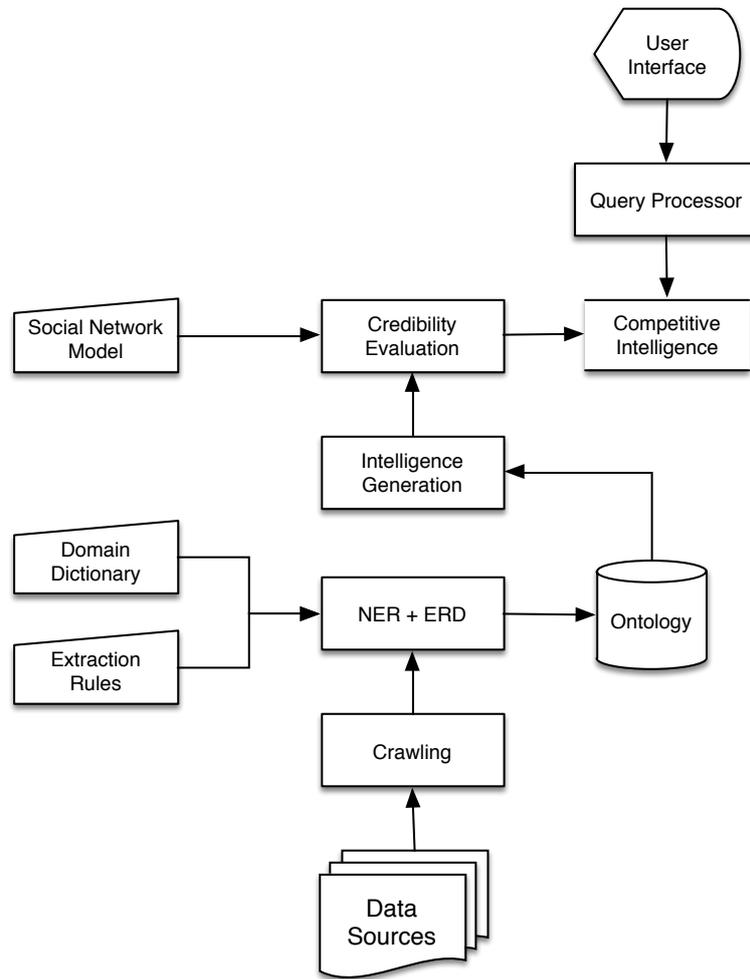


Figure 3.3: Zhao and Jin (2011) framework for CI solution

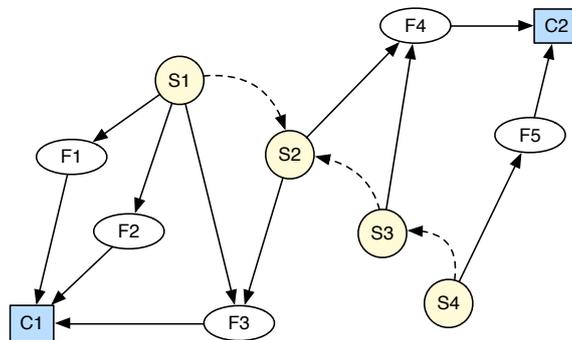


Figure 3.4: Zhao and Jin (2011) social-network-based credibility evaluation model

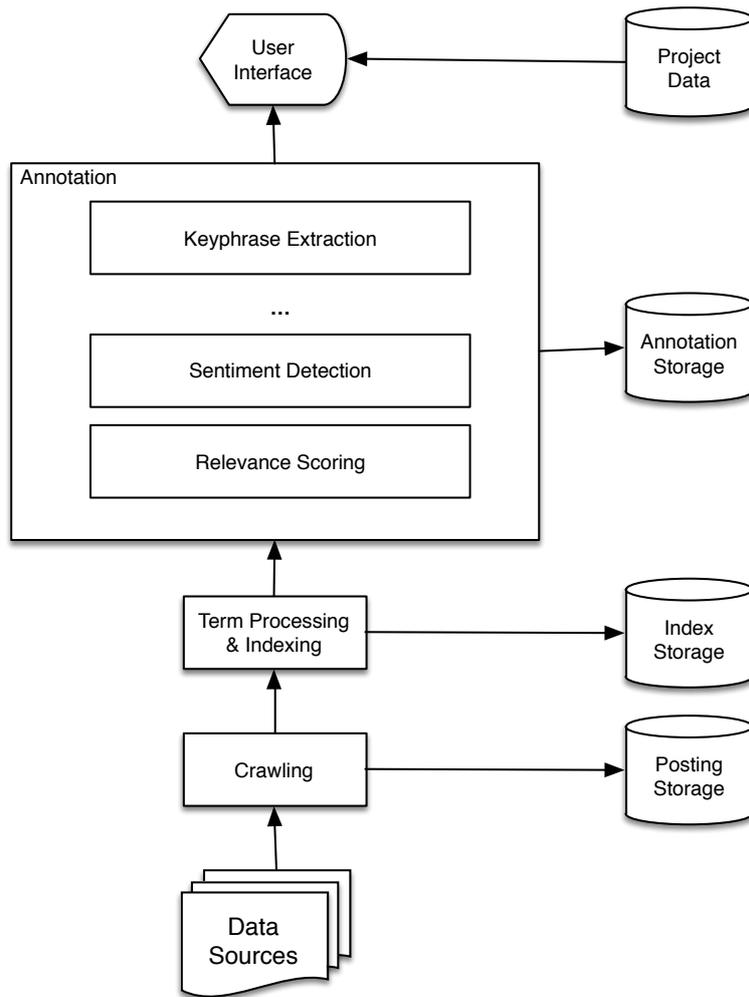


Figure 3.5: Ziegler (2012) framework for CI solution

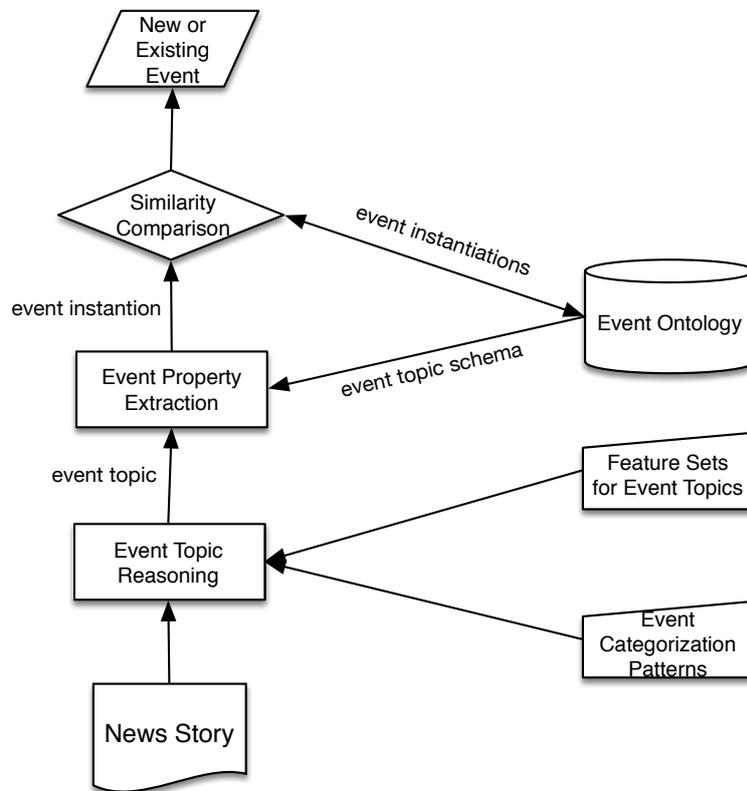


Figure 3.6: Wei and Lee (2004) event deduplication

opinion mining and event detection. The event detection system is useful for clustering posts from multiple data sources and reporting them as single events (as presented in Fig. 3.6). It starts with identifying the main event topics (types of events), described by when, who, where and what (i.e. company acquisition or product launch) and classifying each event into the one of the topics. In the event property extraction phase, the actual values pertaining to the event type are extracted and then compared with other events for similarity. This final steps decides if the event is new or existing.

The opinion mining system starts by part-of-speech (POS) tagging the content of the posts (annotating each word with its part-of-speech). Using association rule mining, we mine frequent features (set of words which appear in the same document), after which we prune useless features by means of compactness pruning (removing features whose words are regularly found to be very far apart) and redundancy pruning (removing features which are included in other features). Afterwards, we try to locate infrequent features by examining words with an opinion word but no frequent feature and selecting the nearest noun phrase to the opinion word. This mechanism aims to extract features

which are very rarely mentioned in a product review. The last step involves identifying for each feature the opinion direction (positive or negative) by checking the opinion words used. The whole process is depicted in Fig. 3.7.

Using the systems mentioned above as building blocks, Dai (2013) proposes a high-level CI solution which she calls *Data Analysis and Visualisation Aid for Decision Making* (DAVID), presented in Fig. 3.8. The design consists of a preprocessing step which does natural language processing of posts, feature extraction which comprises of named entity recognition, event detection and opinion extraction system (as presented above), a processing step which does all sorts of analyses and lastly a knowledge discovery phase which handles monitoring and trend analysis on top of the results of these analyses. It makes use of a document knowledge base (taxonomy of known competitors, products, events) to improve the results of the feature extraction and trend analysis.

Liu et al. (2009) proposes a solution to be used for trend analysis, or what he calls *event change detection* (Fig. 3.9). It is very similar to the design of Wei and Lee (2004) with the additional step of performing an analysis of the changes in association rules between two periods of time, described at length in Song, Kim and Kim (2001). The system analyses differences in the conditional and consequent parts of the rules in order to detect emerging patterns, unexpected changes patterns and added/perished rules. The last phase is estimating the degree of the change, for which some formulas are proposed in each case.

Since it is not trivial to analyse the similarities differences between these fragments, we employ a technique similar to the super method proposed by van de Weerd, de Weerd and Brinkkemper (2007) for comparing the 5 fragments. First we extract the activities for presented in each of the fragments and then we develop a comparison table by specifying all activities and comparing them across all the other fragments. We mark the presence of each activity with either ✓(if they are present), ✗(if they are absent), *n/a* if they are not applicable to the current fragment (Wei and Lee (2004) and Hu and Liu (2004a) are but subfragments of a complete CI solution) or a custom string (which means they are present under a different name or are of a specific type). The comparison table is present in Table 3.3 and a description of the activities can be found in Table 3.4 .

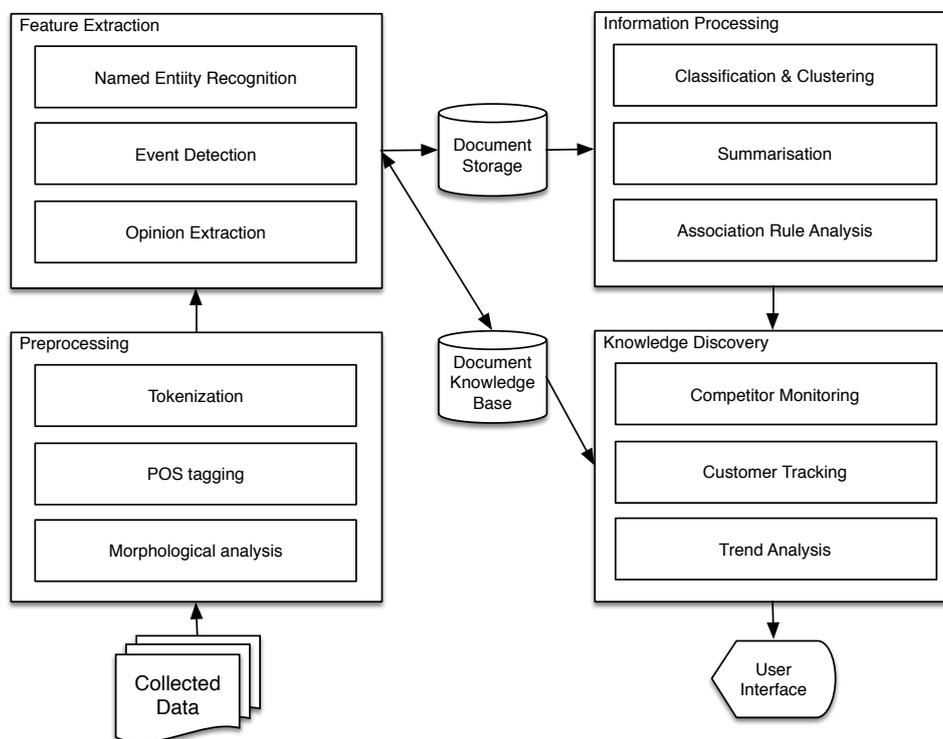


Figure 3.8: Dai (2013) DAVID system

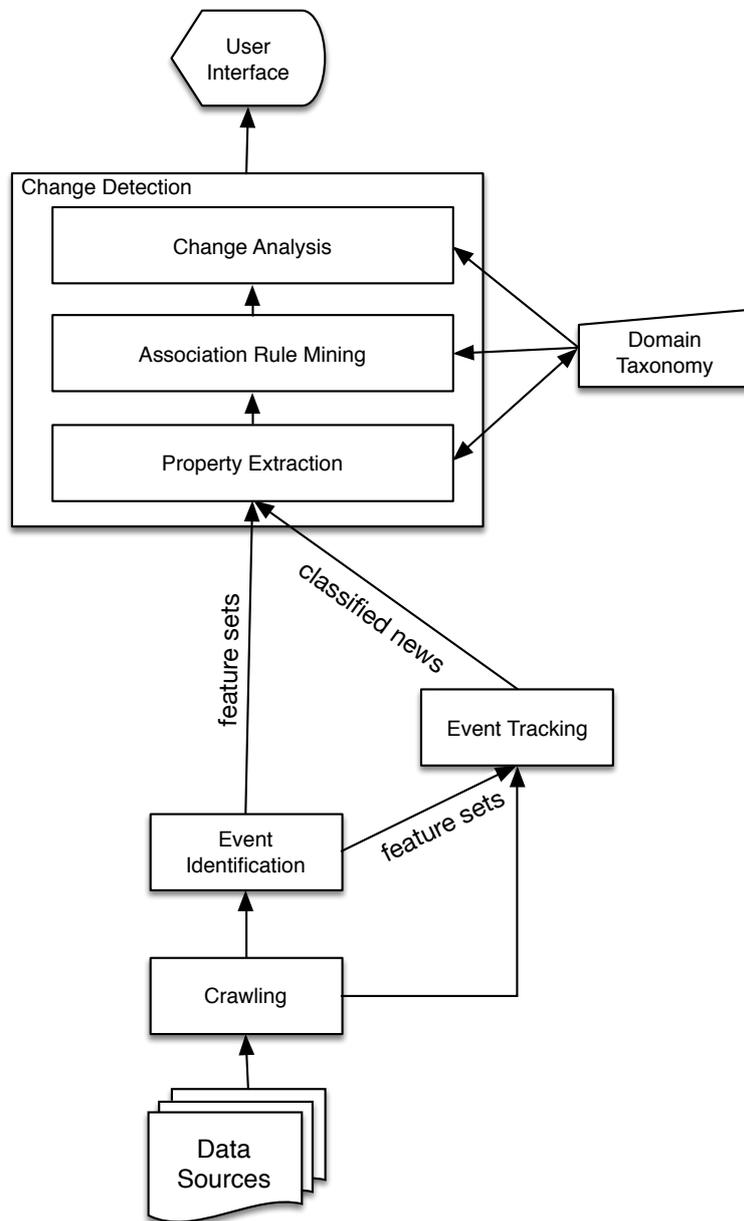


Figure 3.9: Liu et al. (2009) event change detection

Action/Fragment	Zhao and Jin (2011)	Ziegler (2012)	Wei and Lee (2004)	Hu and Liu (2004a)	Dai (2013)
Preprocessing					
Detect Content	✗	DOM-based learning	n/a	n/a	n/a
Post Storage	✗	✓	n/a	✓	✓
Postprocessing					
POS Tagger	✗	✓	✗	✓	✓
NER + ERD	✓	✗	✓	✗	✓
Entity Storage	ontology	annotations	n/a	✗	✓
Analysis					
Event Deduplication	(implied)	✗	✓	✗	✓
Credibility Evaluation	SN Model	✗	n/a	✗	✗
Opinion Mining	✗	polarity	n/a	✓	(postprocessing)
Trend Detection	✗	co-occurrences	n/a	✗	✓
Results Storage	✗	annotations	n/a	n/a	Knowledge Base
Dissemination					
Adhoc Querying	✓	✗	n/a	n/a	✗

Table 3.3: Fragments Comparison Table

Activity	Subactivity	Description
Preprocessing	Detect Content	Automatically detect content from webpages (eliminating ads, navigation links, etc.).
	Post Storage	Store article metadata and content immediately after crawling.
Postprocessing	POS tagger	Perform part of speech tagging on article text, this will annotate each word with the part of speech it represents (i.e. noun, verb, adjective).
	NER + ERD	Perform Named Entity Recognition and Entity Relation Detection on article text (this will identify organisations, people, location and relationships between them).
	Entity Storage	Store previously identified entities and relationships.
Analysis	Event Deduplication	Group articles describing the same event/incident together.
	Credibility Evaluation	Evaluate how credible the a hypothesis is based on existing set of data.
	Opinion Mining	Identify the sentiment present in the article, may be polarity (positive/negative) or a wider range.
	Trend Detection	Identify patterns and trends in events across a timespan.
Dissemination	Adhoc Querying	Allow user to explore the available data/information by performing specific adhoc queries.

Table 3.4: Fragments Activities

3.5 Techniques

We grouped the techniques we found throughout literature into two main sections regarding their use: analysis and dissemination. The data collection part of a CI solution has not received too much attention, but that would likely be due to the fact that it is, at the same time, the most straightforward piece of the puzzle, consisting mostly of focused or automated crawlers (Ziegler, 2012).

For analysis, we identified four types of algorithms (natural language processing, statistical analysis, machine learning and heuristic algorithms). Many types of techniques are available for each of these types, depending on what type of analysis is desired. Data modelling is not well treated topic in research, but proposed solutions include using an ontology, flat databases with full-text indexes, xml databases (useful for storing various degrees of annotated data) or keyword networks (taxonomies).

A comprehensive list of analysis techniques is presented alongside data modelling options in Fig. 3.10. A table with more detailed descriptions and references can be found in Fig. 3.5 and in the glossary section.

Topic	Technique	Description
Information Extraction	NER	Named Entity Recognition concerns itself with identifying entities in a text (like persons, organizations, locations, time, etc.). It is implemented either using a <i>gazetteer approach</i> , i.e. making use manually constructed taxonomies, specifications (TIMEX2, TimeML) or by using chunkers pre-trained on a corpus (Ziegler, 2012).
	ERD	Entity Relation Detection must identify relationships in text between the aforementioned entities. There are generally two types: rule-based methods (i.e. DOM minimal distance (Zhao & Jin, 2009)) and classification-based methods (which can be feature-based or kernel-based) (Xu et al., 2011).
Information Retrieval	semantic search	capacity of a search engine to understand intent and contextual meaning, answer natural language queries, etc. (Fan, Wallace et al., 2006).

	latent Dirichlet allocation	a generative probabilistic model to determine if a document is relevant to a particular topic (Dey et al., 2011).
summarisation	template instantiation	Summarisation technique involving identifying and extracting certain core entities and facts in a document, packaged in a template; it assumes domain knowledge (Hu & Liu, 2004a).
	passage extraction	Summarisation technique involving identifying segments of text which are representative to the document content (Hu & Liu, 2004a).
opinion mining	comparative opinion mining	opinion mining specialised on texts comparing various products on specific features; various implementations have been proposed and cross-analysed (Xu et al., 2011; Jindal & Liu, 2006)
	sentiment analysis	processing of texts with the purpose of identifying subjective position of the author (usually polarity: positive/negative) (Xu et al., 2011).
	subjectivity analysis	processing of texts with the purpose of identifying whether sentences are subjective or objective (Hu & Liu, 2004a).
	opinion holder detection	identification in a subjective sentence of the entity who holds the opinion (Kim, Jung & Myaeng, 2007).

statistical analysis	tf-idf	a numerical statistic that is intended to reflect how important a word is to a document in a corpus; composed of term frequency and inverse document frequency, it increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others
	co-word analysis	the exploration of patterns involving the co-occurrence of different entities within the same document (Vaughan, Yang, Chen, Liang & Li, 2010)
heuristics	genetic algorithms	a search heuristic process which mimics the process of natural selection in evolution (Cobb, 2003).
	particle swarm optimisation	heuristic optimised to search a very large space of candidate solutions; it has been proposed to be used as a content extraction (signal/noise detection) in web pages (Ziegler & Skubacz, 2012)
machine learning	association rule learning	a popular and well researched method for discovering interesting relations between variables in large databases (Liu et al., 2009)
	clustering	the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters) (Chen et al., 2002).

classification	the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known (Zanasi, 2001).
----------------	--

Table 3.5: Techniques Descriptions

The dissemination model is generally split into the selection and ranking phases, selection involving the generation of candidate results which are then ranked. We identified three types of selection mechanisms: topic subscriptions (user subscribes to a predefined system topic), system persistent query (the selection algorithm is improved over time, but globally for all users) and user persistent query (the selection is personalised to learn user preferences and interests in certain areas). Several ways of implementing each of these phases have been proposed and are presented in Fig. 3.11.

Fan, Gordon and Pathak (2006) did a thorough evaluation of the different methods and proposes a method combining the use of Robertson's selection value for the selection phase and a genetic programming approach through several terminal ranking functions (mostly variations of term frequency and domain frequency), in order to discover the function which yield best results for each user. The method yielded marginally better results over using a support vector machine, which was the baseline for testing. We are not presenting a more detailed overview of all methods proposed since this would be a lengthy and superficial exercise, but we invite the reader to consult Fan, Gordon and Pathak (2006), which contains a complete analysis of all these methods.

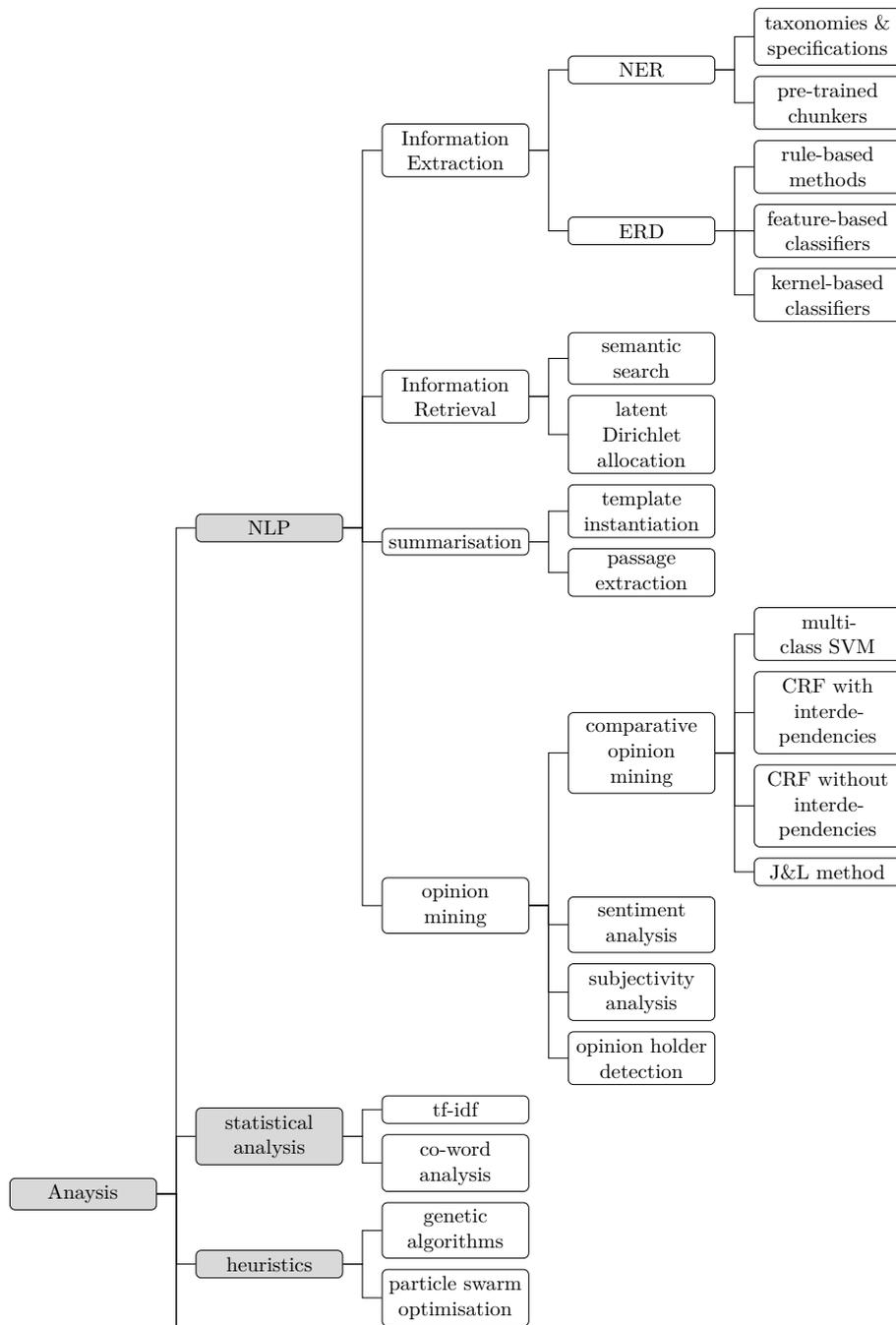


Figure 3.10: Model of Usable Techniques for Analysis

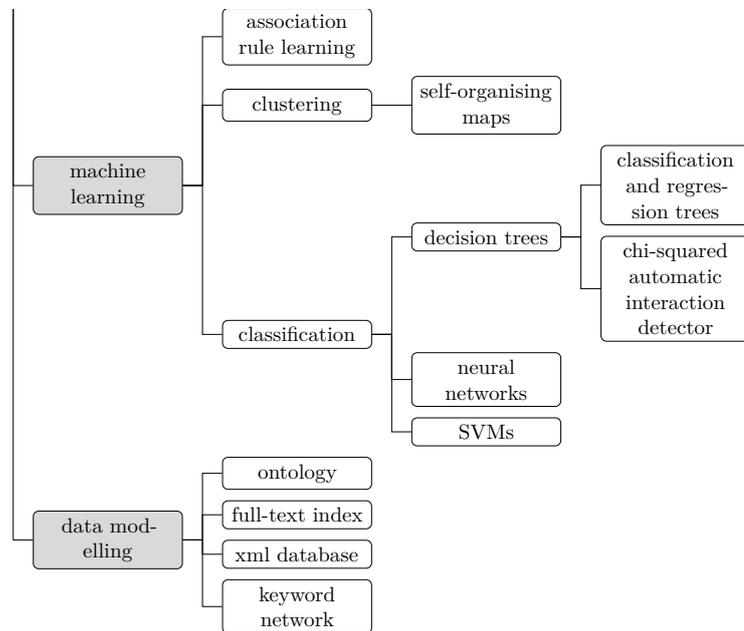


Figure 3.10: Model of Usable Techniques for Analysis (cont)

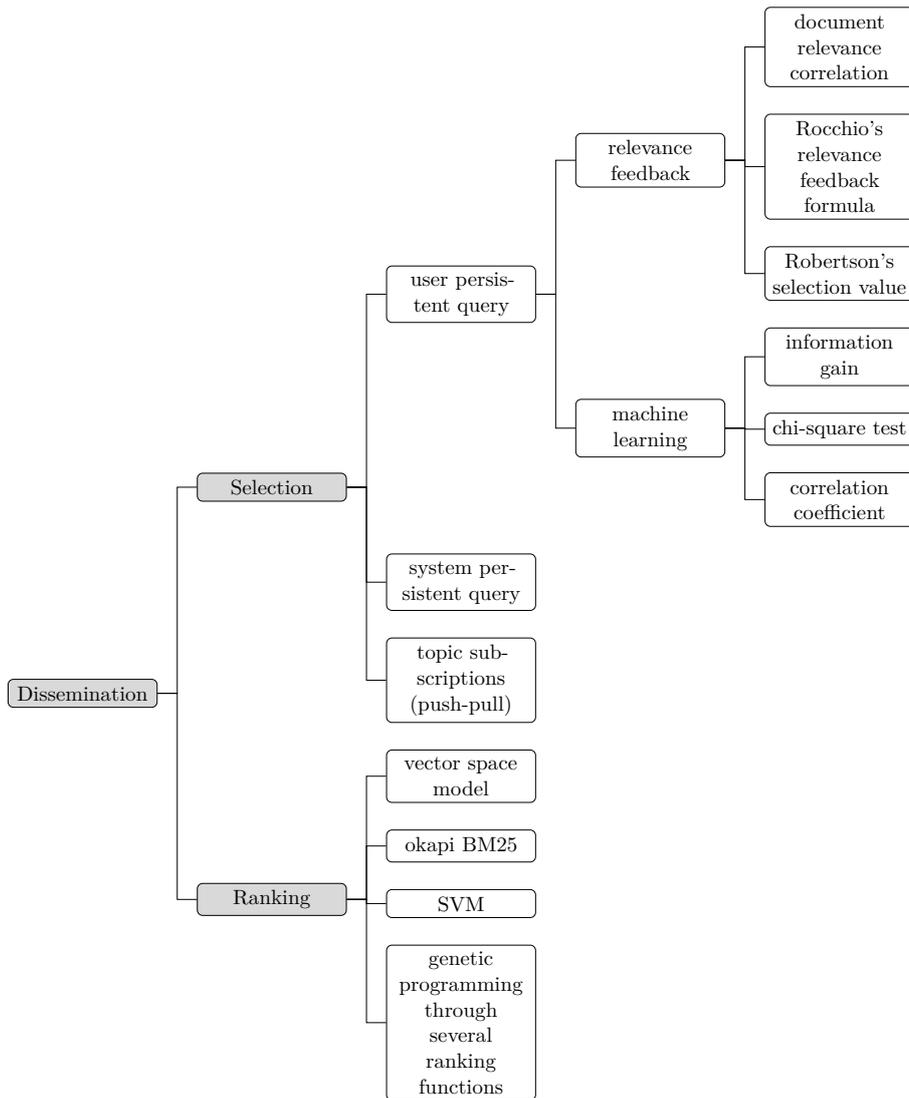


Figure 3.11: Model of Usable Techniques for Dissemination

Chapter 4

Interviews

As mentioned above, due to the immaturity of the market when it comes to automated competitive intelligence gathering and analysing, we used purposive sampling (more specifically typical case sampling) for selecting the participants for interviewing. The goals of the interview were:

1. Assess identified KITs and check if there are missing ones
2. Assess identified solution requirements and check if there are missing ones
3. Identify current implementations for CI and their high-level architecture
4. Issues encountered and lessons learnt

To this extent we developed an interview protocol, presented in Appendix B.

We approached two groups: companies (CI solution users) and professionals (CI solution vendors or consultants). The criteria used for selecting them include: perceived usefulness (companies which we thought would benefit from processing unstructured public data for competitive intelligence), size (medium or large companies whose budget permit implementing a CI solution), proximity (companies which are based in Netherlands/Europe were preferred for the convenience of arranging a face-to-face interview and timezone synchronisation) and professional network (companies with whom the authors had connections with or were referred to).

We realise that such a sample cannot be statistically representative of the entire population, however, we find greater utility in conducting interviews with companies which are more likely to have experience with implementing such a solution. We approached people which we were either referred to (snowball sampling) or which have made their experience in competitive intelligence, technical architectures or unstructured data solutions, publicly

Name	Qualification	Years Experience
Company C1	Digital Publishing Services	–
Company C2	Analysis and Statistics Services	–
Arent van ‘t Spijker	BI & CI Consultant	10
Nico Buwalda	BI & CI Consultant	20
Drs. Ing. Alain Wille	SI & CI Consultant	6

Figure 4.1: Interviewees Table

available. The reason for this approach is motivated by a prior attempt to contact companies through their general enquiry channel which proved to be completely unsuccessful.

Candidates were approached either by phone or email and we sent a reminder email after two weeks in case of lack of response. We invited people from 8 companies to be part of our study and 7 professionals and we received confirmation from 2 companies and 3 professionals, which represents a response rate of 25% and 43% respectively. The numbers entertain the idea that CI professionals and vendors are more inclined to participate in such a research than potential customers, which may not come as a surprise.

A total of 6 interviews were conducted, 4 of them face-to-face and 2 by phone (or equivalent conferencing tools) and lasted an average of 45 minutes. The participants were given an interview consent form to sign which reaffirmed their rights and mentions our intention to record the interview for personal use and publish quotes for research purposes. Companies representatives agreed to the interview on the condition of anonymity. The consent form is annexed in Appendix B along the interview protocol. In the case of phone interviews, however, no consent form was signed, but interviewees were asked for verbal consent.

The list of companies and professionals along with their qualifications is presented in Table 4.1. For the purpose of protecting any sensitive information regarding the companies they work for, we do not disclose the names of the interviewees from specific companies.

Below we present a summary for each of the interviews discussing the main insights related to the architecture design of CI solutions.

4.1 Interviews with companies

Company interview – C1

Company C1 offers many services in the publishing industry. They have IT department of around 50 employees and they outsource or acquire external services and solutions as possible.

The KITs we identified are monitoring competitor and customer actions, research activity and patents. They hold the belief all CI data should be known throughout the company, so there is no need for enforcing security policies. Solution requirements include analysing impact factor and trends per research domain and credibility analysis for all data (however, they already know who are the major opinion influencers which they consider credible and consider treating the others statistically).

To this end they have acquired an enterprise storage solution with built-in annotations, analytics, full-text search and triple-store support (an xml document-based solution called MarkLogic), where they are storing various intelligence reports created or bought. They lack an automated way of importing public data or analysing data, but they are actively looking into implementing such a solution (personal communication, Company C1, 8 jan 2014).

Company interview – C2

Company C2 is offering analytics and statistics services. They have an IT department of 300-500 employees, but it has a support role for the other departments.

They are running a CI solution which automatically collect data from a wide range of sources, but most of them are reliable news sources (no user-generated content). The collected data is processed by several *static queries* by means of content and source (no entity extraction is performed). Several intelligence workers will process each of these queues and generate intelligence reports in the form of newsletters which get distributed as needed throughout the company. They are actively looking into a solution for reducing the human workload involved in this process. The most important solution requirements they have presently are event deduplication and credibility analysis. They offer a lot of (statistical) analyses (being their core business), but most of them are currently manually created (personal communication, Company C2, 13 feb 2014).

As separate projects, they have prototyped a social media analysis system (performing sentiment analysis) and and or collecting data from various

public sources (e.g. job boards, flights, products from various categories), however some of them have been discontinued on the grounds of not being statistically representative. Due to the nature of how this data is being analysed they prefer keeping the different types of sources in separate systems (personal communication, Company C2, 27 jan 2014).

4.2 Interviews with CI professionals

Arent van 't Spijker, BI & CI Consultant – P1

Arent van 't Spijker is a consultant with over 10 years experience in competitive intelligence and the creator of one of the first SaaS CI solutions, Astragy.

He mentions that in his work he has yet to identify any unique functional requirements from customers and that their needs are generally similar and only differing in specifics, similar with what was mentioned by Herring (1999).

For this reason, he suggests there is no need to develop a bespoke system and that a modular system which can be tailored for each specific customer should be sufficient. He advocates the use of a standard funnel for data collection from a series of sources and implementing a library of processing modules (*filters*) with single purpose (following the separation of concerns principle). These modules can be grouped and chained in order to cater to the diverse needs of the customer. This maximizes reusability of development effort and caters for a wide range of requirements.

One of first decisions one has to make in the architecture design is whether or not to store the full content or just document metadata. More recently, there is the alternative of storing content in a distributed filesystem (a concept hardly existent during the development of Astragy), but he mentions he would likely make the same decision on discarding content and only storing metadata and a text summary for each post.

Arent advocates the use of a flat database which only stores relevant content as a means of keeping size under control and having the modules running quickly through the entire dataset (since these operations would execute often). He entertains the idea of using a non relational database (ontology based or graph database) for storing intelligence data, but feels that by doing so, one is trading flexibility for speed and ease of use. He generally recommends keeping a flat database if the users are analytically inclined (i.e. comfortable with writing complex queries) and a graph-like database system for typical business users who are interested in standard reports.

Another specific design consideration of Astragy was performing named entity recognition only on known entities, which should be provided by users by means of a taxonomy, a suggestion we also found in Zhao and Jin (2013). On the same note, he identifies the current major vendors of CI solutions (e.g. ComIntelli, Digimind, Cipher CI, Global Intelligence Alliance) and notes that all these solutions use a taxonomy driven named entity recognition technology, except for Digimind which instead offers a powerful querying tool designed specifically for analytical users.

Arent mentions that most customers do not have security-related requirements, and only one of his clients wanted to protect some set of reports from the rest of the company. Also, there are a few existing solutions which do allow this sort of granularity in security policies. Another suggestion was that real-time monitoring triggers kills system performance and that it would probably be wise to batch these operations and offload them to a different server. He questions the actual real-time necessity and suggests running all these triggers on a fixed interval though (degree of minutes) (personal communication, Arent van 't Spijker, 18 feb 2014).

Nico Buwalda, BI & CI Consultant – P2

Nico Buwalda is a consultant with many years experience in BI and CI solutions. He has been working on CI projects since the time the intelligence gathering was mostly a manually performed activity and the main method of data collection was performing interviews, and he notes the recent increase in demand for CI solutions as technology enables for more and more types of automated analysis. He puts an emphasis on creating the CI solution design starting from the requirements, identifying data sources and then designing the architecture and processes.

The KITs he identifies as being the most important are competitor, client and suppliers profiles, research activity and SWOT based portfolio management. Solution requirements include, first of all, automatic event deduplication, even change detection, trend detection and integration with internal data sources (like ERP systems). He suggested event deduplication should be part of the data collection phase as each post is compared for similarity to those previously collected (largely based on title and timestamp) (personal communication, Nico Buwalda, 19-20 feb 2014).

Drs. Ing. Alain Wille, SI & CI Consultant – P3

Alain Wille is currently leading the development and implementation of a CI solution called MI7TM together with the Strategic Intelligence firm *Rodenberg*

Tillman & Associates. MI7 is a tailored solution with many reusable modules. As external data sources they use, among others, Digimind & LexisNexis data (metadata and text summary for each post), but integrate this data with other customer sources (internal or external). The focus of the solution is a combination of non-stop data collection and a powerful adhoc querying and visualisation tool, as he mentions that it is more useful to present data to the user in various ways to help him understand trends (timelines, graph networks, geographic maps) rather than try to automatically detect them. He still finds the field of corporate & strategic intelligence to be immature and selling CI solutions as difficult as top management (especially in Europe) are not yet aware of the importance of intelligence in daily business practices, despite studies suggesting otherwise.

Personalised information routing and granular security policies for different types of users are built into the MI7 system. MI7 also uses a relational database where data is pre-filtered when importing. It employs a named entity recognition tool which takes as input a domain-specific taxonomy containing information about competitors, markets, suppliers, etc. Credibility analysis is performed at the source level. It could also be performed at the post level, but he is not sure how good of a performance such granular filters would have.

The querying system empowers a powerful semantic analysis technique which builds system-queries based on the context of the keywords supplied by the user and the existing data. This query is enriched by means of technique called latent semantic analysis, which finds related keywords to those already used by analysing their co-occurrence in a document corpus. This effectively achieves finding results which have no exact match, but have a strong connection with the keywords used in the query, allowing the querying tool to “read between the lines” (personal communication, Alain Wille, 27 feb 2014).

4.3 Review

In order to help in the development of a reference architecture, we synthesise the main design choices and solution requirements we identified in the interviews performed. While we did not uncover any new solution requirements, we were able to identify the ones which are more common in the industry as well as discover the design decisions in existing implementations.

Design Choices

1. Annotation based storage

In what regards storage, a relational databases seem to be the favourites in industry implementations and results returned by various modules are stored as annotations to each post (personal communication, Company C1, 8 jan 2014; Arent van 't Spijker, 18 feb 2014; Alain Wille, 27 feb 2014).

2. Modular System

Commercial solutions mentioned use a modular system where modules can be added and customized by the client with respect to implementation or scheduling (personal communication, Arent van 't Spijker, 18 feb 2014; Alain Wille, 27 feb 2014).

3. Store metadata and summary

All solutions mentioned store metadata and post summary instead of the full content of a post (personal communication, Arent van 't Spijker, 18 feb 2014; Alain Wille, 27 feb 2014).

4. Gazetteer approach to named entity recognition

We mentioned in the previous chapters that there are two approaches to NER implementations: a gazetteer approach (based on taxonomies and specifications) and an automated approach by using chunkers pre-trained on a corpus. All commercial solutions mentioned employ a gazetteer approach to named entity recognition, when this is employed. This seems to be the preferred approach to NER across all major CI vendors (personal communication, Arent van 't Spijker, 18 feb 2014).

Solution Requirements

1. Collection

(a) Integration with Internal Data Sources

Many companies have internal knowledge repositories which they need they want to make use of when designing a CI solution (personal communication, Company C1, 8 jan 2014; Nico Buwalda, 19-20 feb 2014; Alain Wille, 27 feb 2014).

2. Analysis

(a) Event Deduplication

Identifying which sources report the same event has been reported as being one of the important features of any CI solution (personal communication, Company C2, 13 feb 2014; Nico Buwalda, 19-20 feb 2014)

(b) Topic Exploration (trends analysis & event change detection)

Topic exploration techniques like co-word analysis and identifying trends and changes in the industry are an important part of CI solutions (personal communication, Company C1, 8 jan 2014; Nico Buwalda, 19-20 feb 2014).

(c) Credibility analysis

Credibility analysis is a common requirement of a CI solution, and sometimes identified as the most important (personal communication, Company C2, 13 feb 2014). In some cases, however, it is possible that the credible sources (and opinion influencers of the industry) are well known so there is no need for a very complex credibility analysis logic (personal communication, Company C1, 8 jan 2014; Alain Wille, 27 feb 2014).

3. Dissemination

(a) Ad-hoc Querying (system persistent queries)

While all CI solutions implement an ad-hoc querying system (this being one of the most basic ways of extracting data out of it), some use different techniques to enrich queries and provide results which have not been specifically requested but are tightly interconnected (personal communication, Alain Wille, 27 feb 2014).

(b) Monitoring & Alerts

Realtime monitoring and alerting is a requirement some companies have, although the definition of realtime in CI solution might be usually of in the minutes/hours range (personal communication, Arent van 't Spijker, 18 feb 2014).

(c) Information Visualisation

Information visualisation is a very easy way to identify trends, as it is usually easier to do plot data on a map or timeline and let the user spot them than try to employ more sophisticated techniques (personal communication, Alain Wille, 27 feb 2014).

(d) Security

Security is not encountered in CI solution, as the information is deemed useful to be known by all company employees, although sometimes there are requests to restrict access to certain knowledge (personal communication, Arent van 't Spijker, 18 feb 2014).

Chapter 5

Reference Architecture

We started by performing a literature review on CI solutions in order to identify the most important requirements, techniques proposed for implementation as well as architectural fragments proposed either for a CI solution or analysis module. We then performed interviews with both companies interested in developing a CI solution as well as CI professionals who have been involved in the design and implementations of CI solutions. Some of the design decisions presented in this chapter originated as a result of the use-case analysis and expert evaluation. In order to keep redundant iterations to a minimum, we present the final versions here, but we do describe extensively the changes involved in the next chapter.

This will allow us to combine the ideas proposed in the academia together with the design decisions and insights from industry in our construction of a reference architecture.

5.1 Design Challenges

Even before we started, we realised the need for a modular architecture which can be adapted for each customer though we did not realise how much impact can a module have on the design.

For example, a basic CI solution would involve crawling and indexing a set of sources and performing ad-hoc querying on the index. When it comes to analytics, however, we notice a large difference in possible requirements and implementations. Certain solution requirements (i.e. event change detection) require performing NER on all articles beforehand, while opinion mining (at least as proposed by Hu and Liu (2004a)) does not require named entities at all. Also techniques such as NER (both the ones based on a taxonomy or a

pre-trained classifier) can be run at the post level, whereas other techniques such as TF-IDF tagging needs to run on the entire collection of posts.

During the use case, we discovered modules can run on prefiltered dataset at query time. Fortunately regardless on how they are run, the logic employed by these modules remains similar at large (personal communication, Arent van 't Spijker, 18 feb 2014), so we only need to provide flexibility in customizing these modules in order to fit a customer's needs (Kiczales et al., 1997).

We realised it would be impossible to recommend a recipe for each module and how they should work, and, even if we did, the continued research in each of those areas would make it obsolete very fast. We restricted ourself to how the overall architecture should be done, how modules should interact and document identified modules in the literature.

5.2 Design Process

To help us in our process of designing a reference architecture, we use the Attribute-Driven Design method proposed by Bass et al. (2012). This involves an iterative process in which the architecture is decomposed into parts and each part is designed and tested individually. The 3-step design strategy is summarised below:

1. Decomposition

This step involves splitting the design into smaller parts which will be designed individually.

2. Architecturally Significant Requirements (ASRs)

Gather ASRs from requirements documents (literature review), stakeholders (interviews) and business goals (purpose of the system).

- (a) Functional Requirements

Functional requirements define the capability of a system to provide certain functions and meet certain needs when used in specified conditions (ISO 25010).

- (b) Quality Attribute Requirements

A quality attribute requirements are qualifications of the functional requirements describing how a system should react to an event and how to measure its reaction (e.g. how fast, how resilient). These requirements should testable and unambiguous.

- (c) Constraints

Constraints are decisions which have already been made, usually by other parties (i.e. management) due to external factors. Common examples are the use legacy systems or certain programming languages/technologies.

3. Generate and Test Process:

- (a) Choose a part of the system to design.
- (b) Marshal all the architecturally significant requirements for that part.
- (c) Create and test a design for that part.

Creating the design involves a series of design decisions which need to be taken at each step regarding architectural elements (modules, components and connectors). The categories of design decisions identified by Bass et al. (2012) are presented in Table 5.1.

Design Decision Category	Description
Allocation of Responsibilities	Assigning responsibility (e.g. include functional requirements, quality attributes, etc.) to elements.
Coordination Model	Identify elements which must coordinate and choose a mechanism to coordinate them.
Data Model	Determining if/how the data is going to be stored, and if/how it is going to be mapped to objects in memory.
Management of Resources	Identify hardware requirements for each element and optionally plan for scalability/availability.
Mapping among architectural elements	Identify relationships between architectural elements.
Binding time decisions	Identify design decision which can be late binding (taken at implementation or runtime). This freedom gives flexibility to the design.
Choice of technology	The premade decisions regarding choice of technologies and tools, specific to each company.

Table 5.1: Design Decisions

5.3 Architecture Design

Now that we described the design process, we start by decomposing the system into manageable parts. The three main phases of any CI solution are Data Collection, Analysis and Dissemination (as depicted in Figure 3.2), so we design them individually. The interaction between these components can be visualised in Figure 5.1, though we will explore the rationale behind the interaction between them over the course of the next subsections.

The main functional solution requirements have been identified using literature review and interviews. The list presented in Table 3.2 remains comprehensive. Since we are trying to develop a reference architecture, we have no predefined constraints we need to cater for.

5.3.1 Quality Attributes

The seven most important quality attributes as identified by Bass et al. (2012) are:

1. Availability

Availability describes the way a system can prevent, detect and recover from failures. The main requirement we impose with respect to availability is that the system can fully recover from failures in individual modules or components (e.g. crawlers, postprocessor or analysis modules). The system should also detect any errors caused by faulty user-provided data.

2. Interoperability

We currently ignore interoperability as we have yet to identify a need to interoperate with an existing system. The architect should revisit the design in order to build interoperability layers should they be required.

3. Modifiability (user-provided taxonomies, custom modules)

The main modifiability requirements with respect to the design is the possibility of building custom analysis and postprocessor modules as well as customizing the scheduling logic between them.

4. Performance

It is unlikely to be able to properly define performance requirements without prior knowledge of database size, which is hard to define, especially for a reference architecture, and without a specific budget or hardware infrastructure (to limit the flexibility on horizontal or vertical scaling). As a general rule the average time between post updates

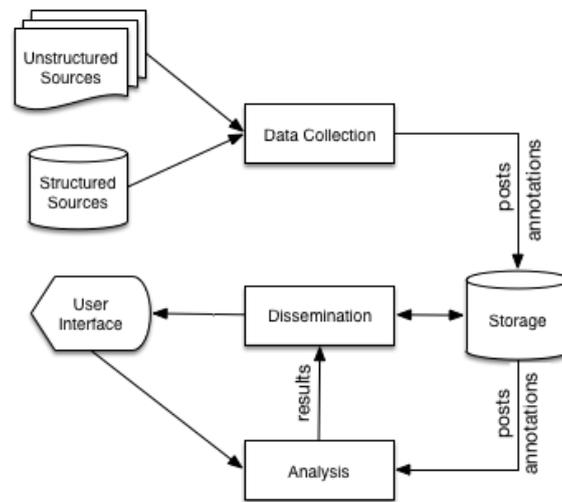


Figure 5.1: A Very High-Level Overview

(recrawling of an article) should be a few orders of magnitude less than the amount of time it takes for all postprocessors and modules to run divided by the number of posts. This ensures that a module is not rescheduled while it is already running and accumulate a large number of pending jobs. Trade-offs need to be made between the frequency of updates and the running time of the modules.

5. Security

Security is out of the scope of this thesis.

6. Testability

All crawlers as well as postprocessor and analysis modules should be decoupled and it should be possible to test them in an isolated environment (*sandbox*). Dissemination modules should also be subject to integration tests.

7. Usability

UX/UI is out of the scope of this thesis.

To sum up, the quality attribute requirements we are interested in availability, modifiability and testability with respect to crawlers and postprocessor and analytics modules.

5.3.2 Collection and Storage

Data collection is first phase of a CI solution and is responsible with extracting and storing data.

The main functional requirements for the data collection, as presented in Table 3.2, are:

- F1 Track changes in posts
- F2 Collect both structured and unstructured data
- Q1 The component should be fault tolerant and not impact the rest of the system.
- Q2 Changes in data collection logic should not have an impact on the rest of the system (e.g. frequency, crawling speed, etc.).
- Q3 Data collectors should be able to run in isolated environment for the purpose of testing (Bass et al., 2012).

For simplicity, we prefixed the functional requirements with F and quality attribute requirements with Q.

Collection

Data collections starts with identifying all the data sources which need to be integrated into the system and obtain rights to access them (Bose, 2008). The systems can be public (like most of the information available on the web) or private (internal knowledge repositories)

From each of these sources a set of individual items which we call generically posts will be extracted. They can be articles, forum posts, industry reports, etc. We make a distinction between a post, an abstract entity, and a post item, the representation of a post in a structured format, but for all practical purposes we use the terms interchangeably when there is no risk of confusion.

We distinguish two types of components which handle the collection part of a CI solution: either a crawler (in case the input is an unstructured data source) or an Extract, Transform, Load (ETL) tool (for ingesting structured data), as suggested by F2. The crawlers and ETL tools would run independently of one another, thus satisfying Q3.

We define a crawler as a component which can execute actions and traverse any unstructured data source in a predefined way. While a web crawler is probably the most well-known example, a generic crawler should be able to extract data from other types of data sources (all sorts of data APIs, rdf, etc.).

A web crawler would be responsible with authenticating itself on the web source (if needed), following a predefined traversal logic and extracting from each post the metadata (e.g. title, author, etc.) and content. There are, to the authors' knowledge, three ways of implementing the extraction part of the process.

The simplest option is to use manually-provided selectors in order to get to the needed information. It can thus be told how to get to the article and what to extract on each page (e.g. an XPath for title, content, author, etc.). This has the advantage of being probably the most accurate at the expense of constantly having to adjust the selectors and logic on possible site layout changes.

Second method is to provide for each website a template of what to extract and let the algorithm figure out how to apply this for further pages. Some implementations have been proposed, and they work very well even with only one template given, but may have trouble with cumbersome websites (Zhai & Liu, 2005).

Third option is to employ a fully automated algorithm such as the one proposed by Ziegler and Skubacz (2012) which uses a technique called particle swarm optimisation to analyse the DOM structure of a webpage and distinguish whether it is signal (e.g. content) or noise (e.g. navigation elements, ads).

Once the crawler has extracted the relevant information, the post items enter through a shared pipeline of preprocessors which have the task of cleansing the data, operating at a global level (same logic across sites). Examples include stripping html tags, normalize dates, etc. Preprocessors would only be responsible for data cleansing, and operate the same across all sites, thus guaranteeing a high cohesion.

Optionally, we can at this point filter out duplicate (or rather unchanged) post items (resulting from both preprocessors and ETL tools) by looking up a hash value in the database. This is useful in case the next process (postprocessors and filters pipeline) is lengthy and if we expect a lot of posts to be recrawled frequently. If the hash exists and its corresponding post item in the database is identical, we can discard the post at this stage as it has already been stored and up-to-date.

The post items now enter a pipeline of postprocessors and filters before being finally passed to the storage middleware. The role of postprocessors is to implement essential analysis-level processes (NER could be a candidate here) during the collection phase, as opposed to independently later. The role of the filters is to discard the posts which are not relevant for the user.

One usage example is a NER postprocessor and a filter which discards

all posts in which no named entities have been detected. It is important to have very good filters in order to keep the size of the database (and subsequently the running time of the analysis algorithms) in check (personal communication, Arent van 't Spijker, 18 feb 2014). We will discuss more about postprocessors during the in the analysis section and what are the reasons of including some of the analysis at this stage.

Storage

The final step of the collection phase would be storing the extracted posts. All our interviewees and architecture fragments we identified in literature have a form of storage at this level, except Zhao and Jin (2011) which suggests (albeit not explicitly) only storing the entities and relationships identified after a subsequent NER postprocessor. However, he only designs a specific module (extracting insights using “rules” which he does not describe or elaborate on). In order to allow for flexibility in the requirements of postprocessor and analysis modules as well as to isolate the collection and analysis modules, it is essential to store the posts data.

The first design choice identified by our interviewees is regarding what information to store (personal communication, Arent van 't Spijker, 18 feb 2014; Alain Wille, 27 feb 2014), as they suggest storing article metadata (e.g. title, author, date, source, etc.) and a summary of the article content. One of our interviewees questions whether he would still store summaries in favour of the full-text should he be re-implementing a CI solution, arguing this should be a binding time decision. He suggests summaries should still suffice for most purposes. Given that some modules we identified in literature do require the full-text content (e.g. Hu and Liu (2004a) in Figure 3.7) and the constantly decreasing price of storage space (Walter, 2005), we argue there is little benefit in discarding the full-text content. However, the decision has little impact over the architecture and remains a binding time decision.

In order to properly update existing data, a unique identifier needs to exist for each post. This unique key would be a combination of the source and an unique *site_id* (which can be chosen at discretion). For example, for some websites the url may serve equally well as a *site_id* while other websites may provide a custom identifier for each post. This needs to be chosen at implementation time, depending on the sources used. The key (*source, site_id*) is thus sufficient to uniquely identify (and thus update) records. Thus records whose keys are not found in the storage system are inserted, and records whose keys are will be overwritten by the new versions.

The issue with when to perform the inserts/updates is tightly related to the question of coordination between collection and analysis modules. The two

approaches to handling this coordination within a CI solution are:

S1 insert/update each post immediately as it is crawled

S2 store newly crawled posts in separate snapshots and import them at the end of the collection process

Option S1 has the advantages of being less complex and more seamless, but may cause the results of analysis modules to be originate from an older set of posts (if crawling is being performed while analysis is also running). It also creates issues if analyses are meant to be run on the complete dataset rather than on a subset. Option S2 is the preferred approach as it mitigates these issues, but is slightly more complex.

At any rate we leave this as a binding time decision since it is dependant on many variables which are client-specific, like the size of the dataset, how often new posts are published, what is the running-time of the analysis modules, etc.

While many types of storage engines may do the job, especially for smaller volumes (we used a plain-text JSON file in our prototype evaluation), a column- or document-oriented database would work best, due to the flexibility in storage schema (personal communication, Company C1, 8 jan 2014), and some of these also provide support for multiple versions of data (*Multiversion concurrency control*), which would satisfy F1 (Cattell, 2011). A column-database also provides a more efficient way to store and update annotations than document-databases by providing a higher throughput when multiple operations are performed on a single row at the cost of a slight increase in complexity (Cattell, 2011).

Another suggestions from (Zhao & Jin, 2011) would be to use an ontology or a graph database to model the posts as well as the entities discovered. This would allow to easily model relationships between entities and the posts they are referred in, at the cost of storage space and increased complexity when inserting, updating or deleting articles (personal communication, Arent van 't Spijker, 18 feb 2014). In the end, our design is not reliant on this choice so this remains to a binding time decision, as the architect needs to evaluate the trade-offs depending the size of data sources and the volume of the queries which the analysis modules would perform. Unfortunately, it is not possible to give specific recommendations at this point without performing extensive tests on variable sizes of databases and scenarios.

An overview of the collection phase of the architecture is presented in Figure 5.2. The dashed line around duplicate filtering symbolise that the process is optional.

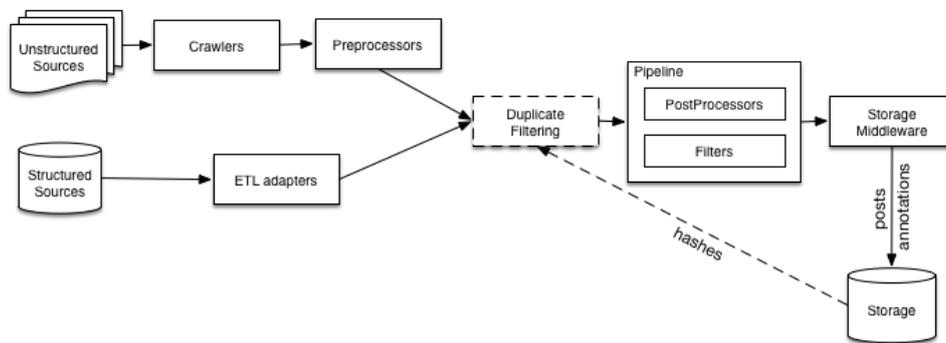


Figure 5.2: Collection Architecture Overview

5.3.3 Analysis

The functional requirements of the analysis phase have been extracted in 3.2. Except for *multilingual support*, which involves adapting the analysis algorithms for other languages, each of the requirements refer to a distinct analysis module. As for the quality attributes, we have:

- Q1 Failures in one analysis module should not impact analysis modules which do not depend on it or any other part of the system.
- Q2 Changes in one analysis module should not impact any other part of the system. Also, it should be easy to customize the scheduling logic between analysis components (personal communication, Arent van ‘t Spijker, 18 feb 2014; Alain Wille, 27 feb 2014).
- Q3 Each analysis module should be able to be run and tested individually.

One of the most important requirements identified in our interviews was the need to offer complete flexibility with respect to scheduling analysis modules, because even though customers have similar requirements, they are not identical (personal communication, Arent van ‘t Spijker, 18 feb 2014; Alain Wille, 27 feb 2014). They suggest that modules themselves should be customizable as well as the scheduler which handles how they are run (personal communication, Arent van ‘t Spijker, 18 feb 2014), which motivates the quality attribute Q2.

Postprocessors

Another requirement was to be able to effectively filter out irrelevant articles before, and a point was raised regarding whether articles where no known entities are identified should be stored (personal communication, Arent van ‘t Spijker, 18 feb 2014). Such a requirement would necessitate the NER

Advantages	<ul style="list-style-type: none"> ✓ article content and analysis output are always in sync ✓ all articles analysis output at all points in time ✓ analysis output can be used by filters to discard irrelevant articles (thus reduce storage requirements) ✓ reduces access to the storage (since article content and analysis output are stored in one query)
Disadvantages	<ul style="list-style-type: none"> ✗ they operate at post level (no information about other posts) ✗ may decrease collection speed (due to additional overhead of the postprocessor) ✗ limits the possibility for a scheduler to schedule the module in parallel with others (since it is no longer subject to the scheduler)

Table 5.2: Advantages and Disadvantages of using an analysis module as a postprocessor

happens before the storage (in the collection phase), hence the concept of postprocessors. Even in literature we identified the use of some analysis modules (like POS tagging) can happen either before the storage point (e.g. Fig. 3.8) or after (e.g. Fig. 3.7).

Any analysis module which operates at the post level (without information about the rest of the posts in the storage) can be implemented both as an analysis or postprocessor module. Generally, postprocessors are very useful where they can be implemented as they greatly reduce data consistency issues, but for the sake of completeness the advantages and disadvantages of this choice are presented in Table 5.2.

Converting analysis modules which qualify (e.g. NER, POS tagger) to postprocessors is thus recommended, but still left as a binding time decision.

Scheduling

It is common for analysis modules to have dependencies between themselves (personal communication, Arent van 't Spijker, 18 feb 2014), so we need to figure out a way to schedule them in the correct order. The problem of scheduling starts by receiving a directed acyclic graph (DAG) of modules (together with their dependencies) and a list of modules which need to be

run. We start with the assumption that there are no loops in the graph (acyclic), but an option to write a custom scheduler should be possible if loops are not avoidable.

Task scheduling is one of the most researched (NP-complete) problems in computer science and there are many heuristics proposed to solve it (Kwok & Ahmad, 1998). It gets even more complicated when we consider running the tasks on multiple processors or distributed on a MapReduce cluster, as we introduce costs to each task (to signify the computing power required) and even to edges (symbolising the cost of moving the data to another node or processor) (Polo et al., 2010). However most of these algorithms assume a large graph with many dependencies (for reference Kwok and Ahmad (1998) did a performance comparison of them running on graphs with 50 to 500 nodes). Whereas we expect a graph with a maximum of 10 nodes (all solution requirements as we identified in 3.2) and generally few dependencies. For our usecase running the modules sequentially in a topological order (the most basic algorithm for task scheduling) should be a more than adequate solution. Better performing but more sophisticated options are, of course, always available if needed and are described in detail in the sources mentioned, among many others. In the case of a small number of modules or if dependencies are very straightforward, it is possible to manually specify for each module the chain of module dependencies which need to run beforehand.

Since most users would want to be kept in the loop on a daily basis, it is sufficient to run the interdependent chain of modules nightly (personal communication, Arent van 't Spijker, 18 feb 2014). However, during the process of our usecase we discovered that it would be extremely useful to run certain analysis modules (in our case the topic exploration module) on a subset of a data. Hence, we need to adapt our design to cater for on-demand runs of analysis modules and modules should expose the possibility of filtering the data before running.

In order to ensure the quality attributes Q1 and Q3, we propose running each module as separate processes spawned by the scheduler. Upon completion the scheduler checks the results and possible schedules the next module in topological order. This ensures that a failure in a module cannot impact will not cause failures in others and that a module can be run individually by spawning the process (thus easily tested).

One issue that was raised during the expert validation was preserving data consistency between analysis modules, since new posts can be imported while a dependency module is running and the next module would then find new posts for which the dependent module has not run. In order to cater for this the scheduler needs to index the items present before the chains of analysis modules is scheduled and use that subset for all modules it schedules. This

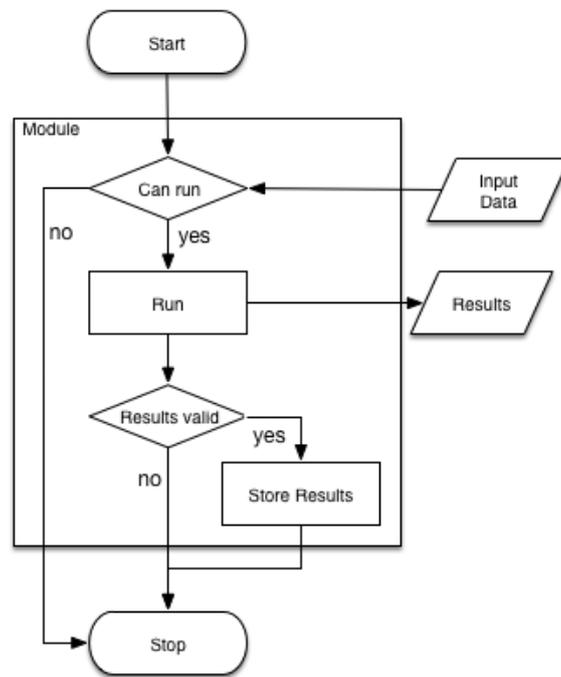


Figure 5.3: Flowchart analysis module

can be achieved in an number of different ways, among which we mention creating a snapshot of post item identifiers and storing a timestamp and filtering based on a it in each module.

The standard interface for implementing a module is very basic: the module checks if all prerequisites for running are met, it runs the analysis code. The results are then checked for validity and then stored, as depicted in Figure 5.3. The results can then be piped as input to another module.

An overview of the analysis phase of a CI solution is presented in Figure 5.4.

Module Implementations

When we first embarked on this research path, we assumed that main deliverable (reference architecture) would contain a proposition for implementing the most common analysis modules. Although the requirements are largely similar and differ only in specificity (Herring, 1999), it is important to keep a flexibility in the implementation of modules and filters and how they are chained (personal communication, Arent van ‘t Spijker, 18 feb 2014).

We also realised there are many valid ways of implementing the same functionality, a lot of them being or becoming research fields on their own. An

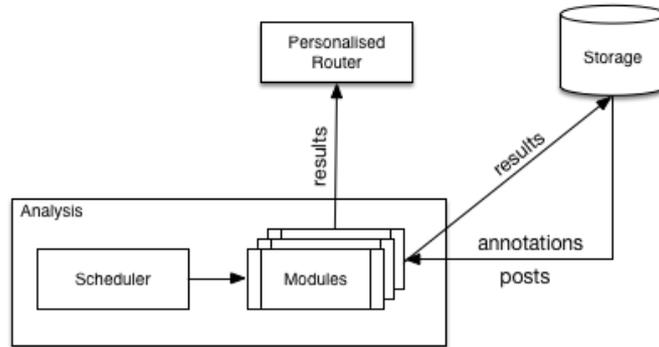


Figure 5.4: Analysis Overview

analysis module would generally be a combination of different techniques, many of which have been identified in Figure 3.10. The most common analysis modules can be seen in the Analysis section of Table 3.2, all of which bar *Multilingual support* would map to one of the modules.

Ultimately we gave up on the idea of providing a reference implementation for each module on the account that endorsing a particular implementation over another cannot be done without previously implementing multiple use cases and it would quickly becoming obsolete. A decision was thus made in favour of a higher level architecture which lets the architect choose the appropriate module implementations given the exact requirements and constraints of the project. We do, however, want to summarise the module implementation ideas discovered throughout our research. These remain areas of future research:

- Named Entity Recognition

NER can be implemented both as a module or as a postprocessor. They can employ both a gazetteer approach (based on a taxonomy of companies, products, etc.) or in a more automated manner by using a pre-trained chunker. As found out during the interviews, all commercial solutions employ a gazetteer approach, due to simplicity and generally better accuracy. Furthermore the taxonomy can later be amended in order to fine-tune the results. During our use-case analysis we used both and even a hybrid approach, but found the pre-trained NER to underperform (at least without extensive optimisations).

- Credibility Analysis

Credibility analysis of the information extracted is an often mentioned feature of a CI solution, but we have yet to find any mention of how this could be implemented besides Zhao and Jin (2011). He suggests using a social network module 3.4, but fails to provide any details on how to

exploit that model. Many sources in literature discuss the importance of credibility analysis but stop short of providing an implementation (Chen et al., 2002), while others suggest various techniques to assess credibility by the quality of the source website (Fogg et al., 2001). Some interviewees suggested performing credibility analysis at a basic level simply by assigning scores to the various sources, or by allowing users to rank them and aggregating the scores (personal communication, Company C2, 13 feb 2014; Alain Wille, 27 feb 2014). Alas, credibility analysis in a CI solution remains a possible direction for future research.

- Topic Exploration

The most sophisticated algorithm in topic exploration revolves around event change detection which aims to identify trends and pattern changes in the dataset. An algorithm has been proposed by Liu et al. (2009), by adapting a similar algorithm proposed by Song et al. (2001) but focused on exploring changes in buying habits of customers. We explored this more in-depth during the use-case analysis, noting the adjustments we made in order for this method to work with our data as well as the evaluation feedback made by the client (see Figure 6.4).

- Document Summarisation

Text summarisation has been an extensively researched topic and remains a research field on its own (Mani et al., 1999). There are two approaches to document summarisation: template instantiation (extracting core sentences from a document based on a template) and passage extraction (identifying segments of text representative to the document content) (Hu & Liu, 2004a).

- Scenario simulation

Scenario simulation is probably the most sophisticated requirement and we have not identified any suggestions for automating it. However, other types of analysis (e.g. topic exploration) may be helpful in aiding decision making during a scenario simulation exercise.

- Cluster events from multiple sources

Deduplicating the same event reported in multiple sources is one of the most basic yet simple analysis. Wei and Lee (2004) proposes an incremental approach of dealing with this issue by classifying all new posts as belonging to a previous event or a new one. Another approach involves clustering articles based on a similarity measure (date, title and entities involved can be good candidates).

- Product comparison

Comparing two products by analysing customer feedback (including comparative statements) is a common task and well treated by Xu et al. (2011).

- Predict competitor data

In cases where a lot of numeric data is available and inferring some missing values is required, the nearest neighbour method (similar to memory-based reasoning) can be used to fill in the gaps (Cobb, 2003).

5.3.4 Information Dissemination

We spent little time discussing information dissemination, partly because the topic is generic, not specific to a CI solution. During our literature review we identified the main methods of delivery to be ad-hoc queries, monitoring, alerts and digests.

The main methods of enhancing the results are personalised information routing and visualisation techniques. Fan, Gordon and Pathak (2006) describes at great length the techniques which can be used for information routing, which is composed of persistent query (representing a user's long standing information requirements) and a ranking function. He argues that a persistent query can either be a user-provided PQ (users manually specify their interests) or a system-constructed PQ (where the system learns the user's preferences) and that the ranking function could also be personalised using a genetic programming approach.

One of our interviewees mentioned the use of a technique called *latent semantic analysis* in order to enhance queries to include strongly related concepts (personal communication, Alain Wille, 27 feb 2014). He also advocated for the use of visualisation techniques to convey and help users explore and arrive at certain insights, mentioning they tend to have a higher success rate than employing complex analysis algorithms. Information visualisation techniques has not been part of our focus.

Fan, Gordon and Pathak (2006) performs a thorough comparison of the possible algorithms which can be used for information routing and evaluates their performance. Whatever the choice, the architecture remains the same: a personalised information router component which gets results either from the database or from the analysis modules and decides to which of the users to send them. The overview is presented in Figure 5.5. Due to time constraints and the complexity involved in evaluating a personalised information router, this component was not tested during the use-case analysis, which was limited to a simple command line interface for ad-hoc querying results and scheduling module reports.

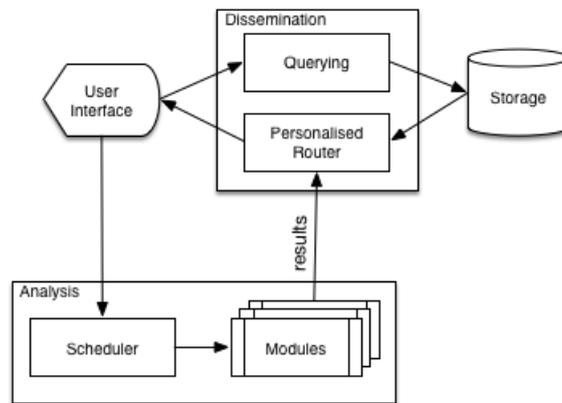


Figure 5.5: Information Dissemination Overview

5.3.5 High-Level Architecture

By combining the architecture fragments for the three phases described, above we end up with an overall picture of the high-level overview from Figure 5.6.

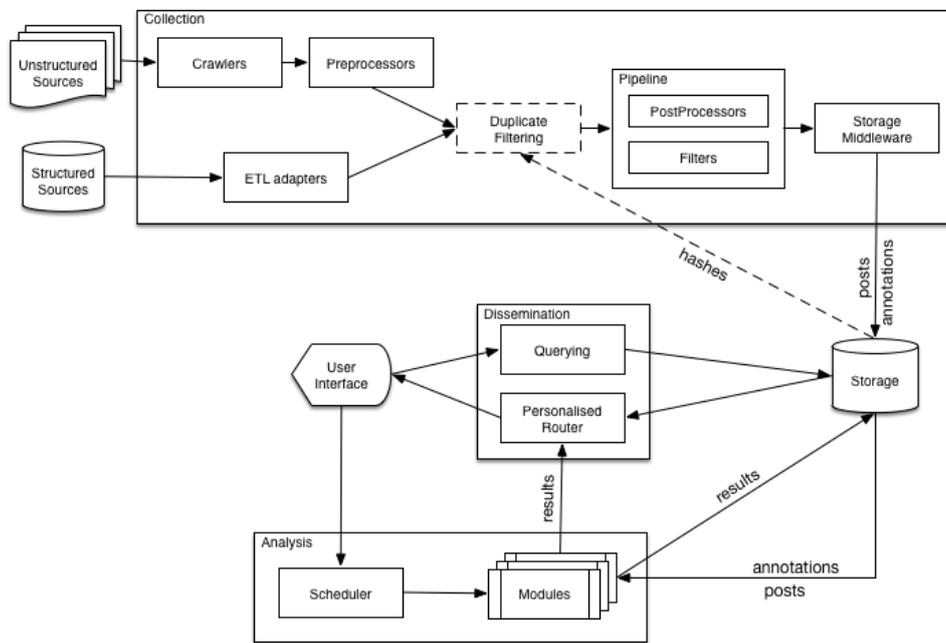


Figure 5.6: High-level Architecture Overview

5.4 Binding Time Decisions

As mentioned at the beginning of this section, we leave some architecture design decision to be taken based on the specifics of the project at hand (binding time decisions). We distinguish between two types of decisions: system design decisions, which have an impact on the entire system (e.g. in flow of data, running-time, etc.) and implementation decisions, which reflect how a particular component is implemented and have no meaningful impact on the rest of the architecture.

Below we present the binding time decisions identified, grouped in the two categories.

5.4.1 Design binding time decisions

- Track changes when recrawling

If one of the requirements is to monitor the evolution of the attention a post gets from users or just be able to get updates made by the author, a schedule for recrawling and way to only recrawl articles from a fixed period (it is unlikely one wants to check articles from years ago) is necessary and needs to be performed on a site-by-site basis. We

consider this a design decision, since it has an impact on the storage middleware and database schema.

- Storage: store module output as annotations or by means of an ontology (graph database)

All our interviewees adopted a simple annotation-based storage approach, while academics advocate either that (Ziegler, 2012) or an ontology-based storage system (Zhao & Jin, 2011). Annotation based storage is flexible and makes updating simple (personal communication, Arent van 't Spijker, 18 feb 2014) while also allowing text analytics solutions (personal communication, Company C1, 8 jan 2014). Ontologies provide faster entity-oriented read queries (i.e. it can find all posts mentioning a competitor without performing a full scan of the dataset) at the cost of more expensive writes (updating the list of entities in a post requires additional queries to add/prune relationships). In the end it does not have a meaningful impact on the architecture and remains a binding time design decision.

- Run some analysis modules as postprocessors

As mentioned above, some analysis modules can be run both as modules or postprocessors if they operate at the post-level, as we've also seen in literature. The advantages and drawbacks have been enumerated in Table 5.2, as this still remains a binding time decision, to be determined by project context.

5.4.2 Implementation binding time decision

- Webpage automatic content detection or customised crawlers

If the number of data sources is large enough that writing custom crawlers becomes infeasible and the data is not in a structured format (i.e. rss feeds), then one needs to look into algorithms which can automatically detect the interesting content on each webpage (e.g. title, author, full-text, etc.).

- Store full-text or summary along post meta-data

Storing post summaries instead of full articles has been advocated by some industry specialists (personal communication, Alain Wille, 27 feb 2014). It is still questionable whether given the recent surge in storage capabilities, it is worthwhile to discard full-text content in favour of summaries (personal communication, Arent van 't Spijker, 18 feb 2014). This is not a design decision, since all modules and algorithms would operate identically on a post's summary or full-text content.

- Event Deduplication: incremental clustering or reclustering?

Another implementation choice is how clustering similar posts into the events should be performed (a requirement we named *event deduplication*). The two main approaches are incremental clustering (deciding for each incoming post if it depicts a new or existing event) and re-clustering (on each new batch of posts, discard previous clusters and regroup posts into events), but hybrid methods can be employed as well (Can, 1993).

Incremental clustering is much faster but resulting clusters are hugely influenced by the order posts are evaluated and quality degrades after relatively small increases in the size of the database (25-50%) (Can, 1993). Wei and Lee (2004) suggests such an incremental clustering approach. Re-clustering, on the other hand, is slower but should in principle yield a better accuracy.

5.5 CI system implementation process

As the number of deliverables is extensive, we compiled below a short process for combining them in the implementation of a CI system. Note we focus on the steps in the technical implementation process of a CI system, rather than on business processes of implementing a CI program. We identified the following steps:

1. Identify KITs (Figure 3.1)
2. Identify data sources based on KITs (Section 3.2)
3. Identify solution requirements (Table 3.2)
4. Create situational architecture
 - (a) start from reference architecture (Figure 5.6)
 - (b) make binding time decisions based on database size and usage estimates and solution requirements (Section 5.4)
 - (c) further adapt architecture based on project context and constraints (e.g. existing technology stack, solution requirements which are not treated, etc.)
5. Implement situational architecture
6. Implement analysis modules (check suggestions in subsection 5.3.3)

Chapter 6

Analysis

6.1 Use Case

In order to analyse the feasibility and usefulness of implementing a CI solution using the architecture presented in the previous chapter, we developed a prototype which tries starting with it as a reference. It was also of help in refining some design decisions as we found discrepancies between our understanding of how such a system should be implemented, the designs proposed in scientific papers/interviews and our effort in instantiating such a design.

A project has been implemented in collaboration with Jibes for one of their clients which is active in the maritime & offshore industry. Their main requirements for a CI solution were to be able to quickly explore articles based on competitors, markets and areas as well as identify patterns and trends over the years in the large amount of available data.

6.1.1 Overall Design

When we started the use-case design a commercial product for information collection was already being developed. In order to bootstrap the process and due to time constraints we decided to use it as a part of our CI solution prototype process. While it does not perfectly match the architecture presented in the previous chapter, it serves as an example of how existing systems can be used as a “collateral” in developing a CI solution starting from a reference architecture (Bass et al., 2012).

The overall architecture consists of:

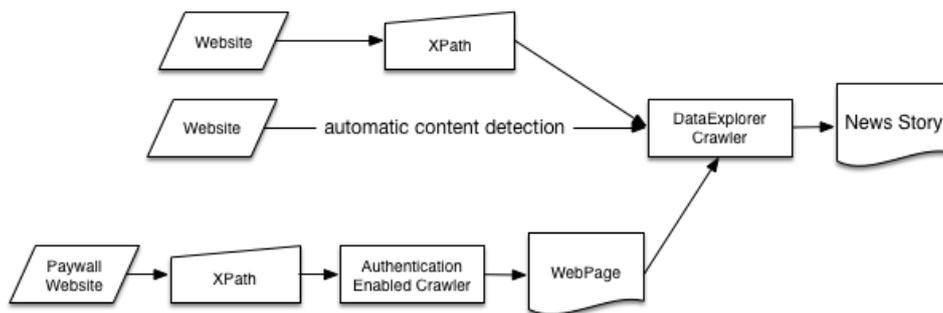


Figure 6.1: Prototype Crawling

- IBM DataExplorer, a commercial solution which does crawling and NER
- Custom ETL (extract, transform, load) tool to extract data from the DataExplorer system
- Custom tool which queries data and performs topic exploration

The data sources used are:

- Upstream ¹
- Asiasis ²
- Maritime Journal ³
- Internal Wiki

The crawling system consists of a set of crawlers which scrape data from the sources above by either automatic detection of content and post metadata (e.g. author, date) or specific XPath selectors (an XML querying language) for those attributes where automatic detection failed. Due to the lack of proper form authentication support in DataExplorer we had to implement a custom crawler which authenticates itself and crawls the content in advance, exposing the unauthenticated content to DataExplorer's own crawling engine. This is a hack and should in general not be needed, but it was the way we chose to overcome the limitations of an inflexible commercial tool. The three types of methods used are presented in Figure 6.1.

In the next step, NER has been performed on the extracted articles using either a gazetteer approach (handcrafted taxonomy) or a pre-trained classifier (provided by the GATE project ⁴). Using these techniques we identified:

¹Upstream, <http://www.upstreamonline.com>

²Asiasis, <http://www.asiasis.com>

³Maritime Journal, <http://www.maritimejournal.com>

⁴GATE Project, <http://gate.ac.uk>

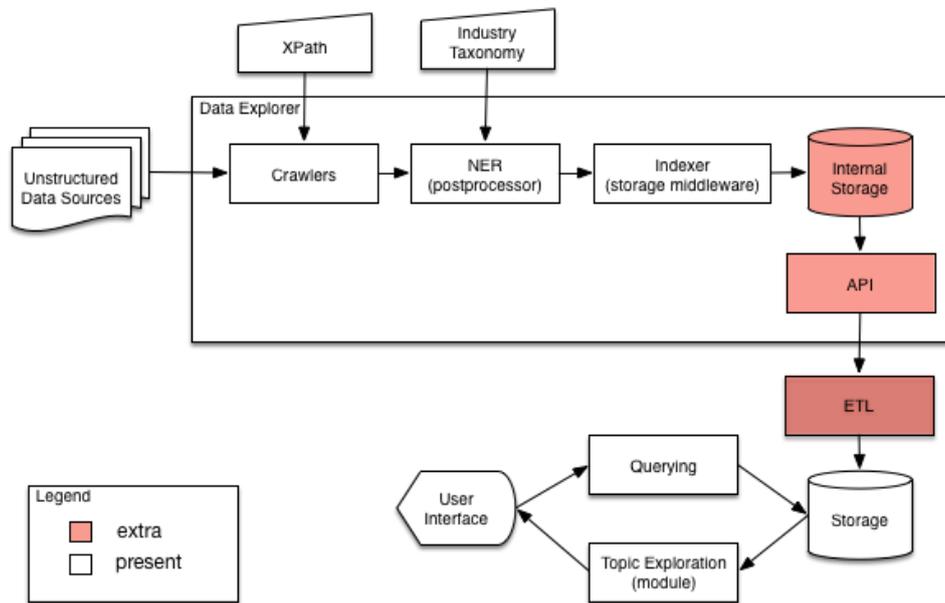


Figure 6.2: Prototype High-level Architecture

- Organisations (using both a gazetteer and a pre-trained NER)
- People (using a pre-trained NER)
- Areas (using a gazetteer NER)
- Markets (using a gazetteer NER)

For the gazetteer NER, we made use of two taxonomies which were provided by the client, one for areas (area > country, e.g. "Asia > Korea, South"), and another for markets (domain > market, e.g. "Seagoing Transport > LNG Tankers").

The second step is exporting the data from the DataExplorer system via the built-in REST API. A custom ETL tool was built using the *scrapy* crawling framework, which authenticates and runs sequential queries against the API extracting the data. As the number of articles was small (16696) we opted for the simplicity and convenience of storing the articles in a flat file in the JSON format.

The last part of the system consists of a topic exploration system run by a real-time querying tool, described in detail in the next subsection. An overall picture of the technical architecture of this prototype is depicted in Figure 6.2.

The architecture is similar to the our reference architecture, with the following amendments:

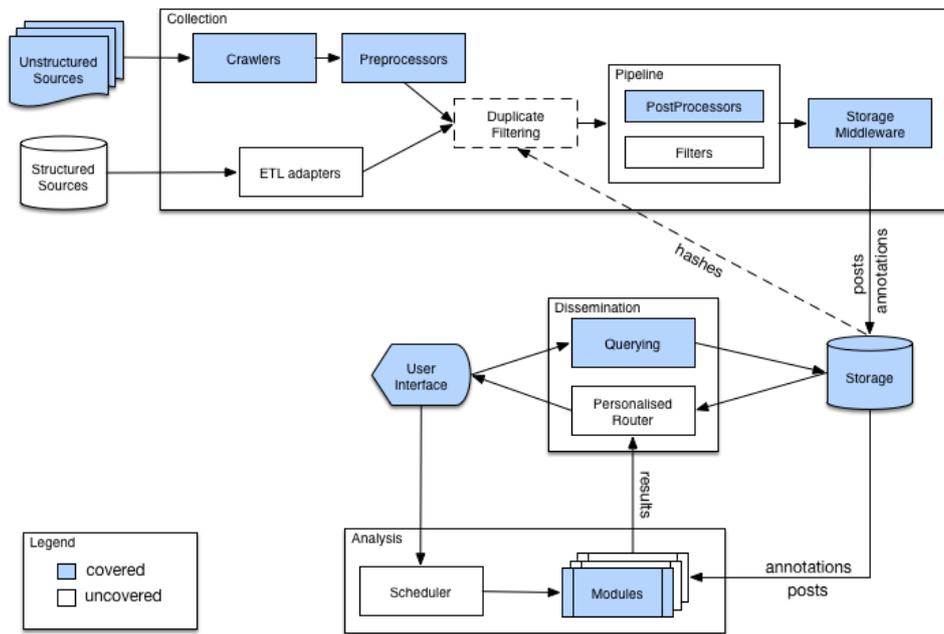


Figure 6.3: Prototype - Reference Artefact Coverage

- there is no complex module for scheduling modules, since topic exploration is the only module implemented
- another crawler is used to preprocess results for DataExplorer crawler in the case of some paywall websites
- the need for an ETL tool which gets data through an API, due to lack of transparent access to the storage engine (marked with red in Figure 6.2)

Figure 6.3 shows the coverage of the reference architecture by our prototype.

6.1.2 Topic Exploration

Ziegler (2012) mentions the usefulness of analysing co-occurrence for topic exploration, so we start with exploring the top co-occurrences between entities in order to get a basic understanding of the market. We then adopt and modify the event change detection algorithm proposed by Liu et al. (2009) in order to exploit the strongest association rules between tags as well as how they evolve over time (identifying here trends as well as unexpected new/perished/changed rules).

All collected data is from years 2012-2013. The topic exploration technique can run on the entire dataset or a subset of it (for example if we want to

specifically target an area, organisation, etc.).

We start from a set of posts and a predefined set of features (for this module we used "areas", "markets", "orgs" and "people"). The first step is to generate tags for all posts (a tag being a feature-value pair).

An example post item is presented below:

```
{
  "lang": "english",
  "title": "Brent falls under $106",
  "url": "http://www.upstreamonline.com/live/
    article1355570.ece",
  "text": "Oil fell under $106 a barrel on Wednesday to
    a six-week low as concerns eased about [...]",
  "publish_date": "2014/03/19",

  "annotations": {
    "areas": ["Asia > Russia", "Commonwealth of
      Independent States > Ukraine"],
    "markets": ["Offshore Oil & Gas"],
    "orgs": ["US Federal Reserve", "American
      Petroleum Institute", "Energy Information
      Administration"],
    "people": ["Christopher Bellew", "President
      Vladimir Putin", "Brent"],
  }
}
```

This post contains 9 tags (["areas", "Asia > Russia"], ["areas", "Commonwealth of Independent States > Ukraine"], ["markets", "Offshore Oil & Gas"], ...) grouped in 4 features ("areas", "markets", "orgs", "people").

The next step is what we call *tag promotions*, a process which consolidates the tags with the same meaning into one. Example promotions include acronyms (*OSE* becoming *Oslo Stock Exchange*) or named entities identified by the classifier being “promoted” to the named entity specified in the taxonomy (*Hyundai Heavy Industries* becoming *Hyundai*).

As part of topic exploration we first identify the top 20 tags for each feature. For example, these are the top organisations which are co-mentioned with *Norway*:

```
89 |+++++++| ["orgs", "Hyundai "]
57 |+++++++| ["orgs", "Daewoo "]
56 |+++++++| ["orgs", "Samsung "]
34 |+++++++| ["orgs", "Rolls-Royce "]
25 |+++++++| ["orgs", "Bergen Group "]
24 |+++++++ | ["orgs", "Norwegian Continental "]
23 |+ +++++ | ["orgs", "Shell "]
```

```

20 | +++ +++ | ["orgs", "independent"]
19 |      +++++ | ["orgs", "NYSE"]
14 | ++ ++ ++ | ["orgs", "Oslo Stock Exchange"]
14 | +++ +++ | ["orgs", "Hyundai Samho Heavy Industries"]
14 | +++++   | ["orgs", "STX OSV Holdings"]
11 | +++++ ++ | ["orgs", "Statoil"]
10 | ++++++++ | ["orgs", "TTS Group ASA"]
10 |   +++ ++ | ["orgs", "The Group"]
10 | + + +++ | ["orgs", "The Company"]
10 |   +++++ | ["orgs", "Bergen Group ASA"]
 9 | ++++++++ | ["orgs", "BP"]
 9 | + +++++ | ["orgs", "CATERPILLAR"]
 9 | +++ ++ | ["orgs", "Maersk"]

```

At the beginning of each line is the total number of articles tagged with both *Norway* and that specific organisation, as well as a list of occurrences, e.g. |+++ +++|. In this example, an occurrence was found in each of the quarters except Q4 2012; there are 8 quarters for 2012-2013 period, and each + signifies that there was at least one occurrence for that quarter.

The second step is to divide all articles into quarters (3 calendar months periods) and generate rules using the association rule learning technique (we used the Apriori algorithm) for each of the quarters.

Rigorously, let $\mathcal{I} = \{x_1, x_2, \dots, x_n\}$ be a set of distinct tags and \mathcal{D} a multiset of \mathcal{I} (a set of transactions T , not necessarily distinct, where $T \subseteq \mathcal{I}$). The support of a tag set X is the percent of tag sets in \mathcal{D} which include it:

$$\text{supp}(X) = \frac{|\{T \in \mathcal{D} \mid X \subseteq T\}|}{|\mathcal{D}|} \quad (6.1)$$

We define an association rule as an expression $X \implies Y$ where $X, Y \subseteq \mathcal{I}$ are tag sets with $X \cap Y = \emptyset$. Then we can define two important measures for an association rule (Agrawal & Srikant, 1994):

$$\text{supp}(X \implies Y) = \text{supp}(X \cup Y) \quad (6.2)$$

$$\text{conf}(X \implies Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)} \quad (6.3)$$

We informally call for a rule $X \implies Y$, X to be the conditional part and Y the consequent part. The support of a rule thus represents the the percentage of elements in \mathcal{D} which include both its conditional and the consequent part, whereas the confidence of a rule represents the probability that a element from \mathcal{D} which include its conditional part also includes its consequent part.

In order to derive association rules we use a standard Apriori algorithm for sparse datasets (Agrawal & Srikant, 1994), which despite its simplicity is one of the best algorithms for association rule learning (Hipp, Güntzer & Nakhaeizadeh, 2000). The algorithm can be applied on the entire dataset or on a pre-filtered subset, i.e. by selecting only articles which contain a specific feature. One downside of the Apriori algorithm is that its performance is determined by a preselected minimum support threshold used which is not trivial to provide, given that it depends a lot on the characteristics of the dataset, so it can be a trial and error process. For the reference we usually generate rules with minimum confidence of 0.1 and a minimum support in the range from 0.01 (in case of the entire dataset) to 0.1 (when querying a reasonably common tag).

The output of the algorithm are rules in the form:

```
0.84 |+++++++| ["markets", "Seagoing Transport > LNG
  Tanker"] -> ["areas", "Asia"]
0.73 |+++++++| ["markets", "Seagoing Transport > LNG
  Tanker"] ["areas", "Asia"] -> ["areas", "Asia > Korea
  , South"]
```

The first one means that 84% (the confidence) of all articles tagged with *LNG Tanker* (LNG stands for *liquefied natural gas*) will also be tagged with *Asia* and that 73% of all articles tagged with *LNG Tanker* and *Asia* are also tagged with *South Korea*.

For the rest of this document, we will assume that this correlation implies causation. While this is a statistical fallacy (false causality), we do this for the sake of simplicity in natural language interpretation and because each of these correlations must be examined by a competent user for a possible causation. So for the sake of rigour, we can formulate the first rule as a hypothesis “84% of LNG Tanker business is done in Asia”, backed up by the statistical fact that if an article has been tagged with *Seagoing Transport > LNG Tanker*, then there is an 84% probability that it was also tagged with *Asia*.

Notice how the two rules are completely independent of one another and do not overlap: the former rule mentions nothing about how much business is being done in South Korea, while the latter does not measure how significant the the business done in Asia is (which is incidentally exactly what the first rule measures). Put another way, one might find the fact that “73% of LNG Tanker business done in Asia was actually in South Korea” meaningless if it were the case that “the total percentage of LNG Tanker business done in Asia is 0.1%”. These two rules seems to suggest that South Korea is a leading authority in the LNG market, a fact which was later confirmed to us by the client.

The problem at this point is that the association rule learning is going to generate a lot of such rules (in our experiments a couple hundred per quarter), so it may get complicated to analyse them all and spot trends or changes in the marketplace. The final step is the event change detection algorithm which compares the rules mined from two successive quarters in order to identify new, perished or growing trends as well as unexpected changes. For each of these changes, Liu et al. (2009) also propose ways to calculate the *degree of change* (marked as *deg* below), a measure which should quantify how important the change is considered to be (the higher the value the more important the change). Below we present each types of changes, along with some examples:

1. Emerging

Emerging rules are the ones that have been found also in the previous quarter and are seeing an increase in support in current quarter (we use a threshold of 20%, i.e. a ratio between the two values of 1.2 or larger).

```
> emerging 2013 Q3 => 2013 Q4
  deg +1.05 ["areas", "Asia > Russia"] -> ["areas", "Arctic Region"]
  deg +1.05 ["areas", "Arctic Region"] -> ["areas", "Asia > Russia"]
  deg +0.78 ["areas", "Asia > Russia"] -> ["markets", "Offshore Oil & Gas"]
```

2. Added

Added rules are rules that have been found in this quarter but were not found in the previous one.

```
> added 2012 Q4 => 2013 Q1 (query: LNG Tanker)
  deg +0.20 ["markets", "Seagoing Transport > LNG Tanker"] ["areas", "Asia > Japan"] -> ["areas", "Asia > Korea, South"]
```

3. Perished

Perished rules are the ones that have been found in previous quarters, but not in the current one.

```
> perished 2012 Q4 => 2013 Q1 (query: Shell)
  deg +0.21 ["markets", "Offshore Oil & Gas"] ["orgs", "Shell"] -> ["areas", "Asia > Korea, South"]
```

4. Consequent change

Consequent change rules are unexpected changes in the consequent part of a rule. The rule found in the previous quarter is not found in the current one, but instead there is a rule with a very similar conditional part and a different consequent part.

```
> consequent change 2013 Q1 => 2013 Q2
  deg +1.05 ["areas", "Europe > Norway"] -> ["
    areas", "Asia > China"] ==> ["areas", "
    Europe > Norway"] -> ["areas", "Asia > Korea,
    South"]
```

5. Condition change

Condition change rules are unexpected changes in the condition part of a rule. They operate similarly to the consequent changes.

```
> condition change 2012 Q4 => 2013 Q1
  deg +1.01 ["markets", "Ship Repair"] -> ["areas
    ", "Asia > China"] ==> ["markets", "Seagoing
    Transport > Container Vessel"] -> ["areas",
    "Asia > China"]
```

After generating the changes, we filter out the ones where the consequent part can be inferred from the conditional part regardless of the dataset. Including these self-evident rules would lead to a lot of false-positives, like the one finding that there is a lot of wherever *China* is mentioned *Asia* is mentioned as well:

```
["areas", "Asia > China"] -> ["areas", "Asia"]
```

Note we do keep rules which describe the inverse rule, suggesting *Asia* does a lot of business with *China*, although those can be filtered out as well by using an option we called *aggressive filtering*.

Finally we classify the most popular rules by averaging their confidence across quarters. This creates a bias for showing the most important trends which have been happening for the entire period of 2 years. For example the most prominent trends identified with the entire dataset are:

```
1.0 |+++++++| ["areas", "Southeast Asia > Hong Kong (
  SAR)"] -> ["markets", "Environmental Safety &
  Control > Search and Rescue Vessel"]
0.994 |+++++++| ["areas", "Southeast Asia > Hong Kong (
  SAR)"] -> ["markets", "Defence & Security > Search
  and Rescue Vessel"]
0.946 |+++++++| ["orgs", "Daewoo"] ["orgs", "Hyundai"]
  -> ["areas", "Asia > Korea, South"]
```

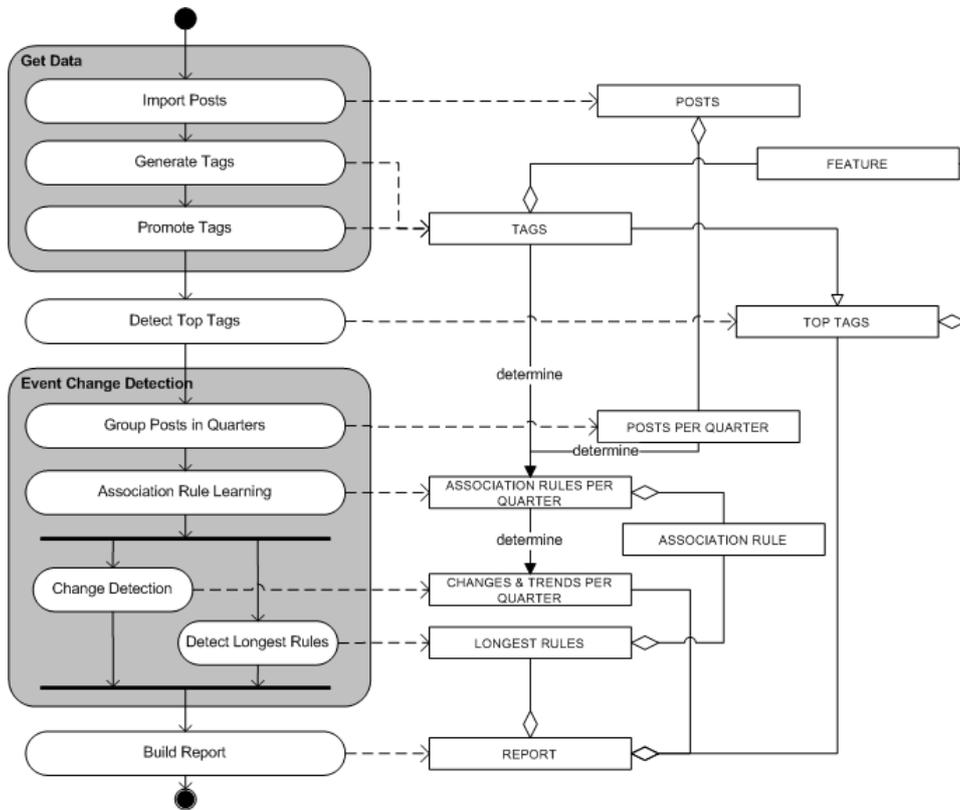


Figure 6.4: Topic Exploration PDD

These suggest Hong Kong’s interest in search and rescue vessel markets (both in environmental safety & control and defence & security) as well as Daewoo and Hyundai doing business together in South Korea.

A process-deliverable diagram of the topic exploration module is presented in Figure 6.4. The activities are further described in the activity table from Table 6.1 and the concepts are explained in Table 6.2. A complete sample report is presented for reference in Appendix D.

Activity	Sub-Activity	Description
Get Data	Import Posts	Retrieve all posts from storage.
	Generate Tags	Generate a list of all tags present in the posts.
	Promote Tags	Merge together tags which refer to the same concept.
Detect Top Tags		Create a list of top 20 tags for each feature.

Event Change Detection	Group Posts in Quarters	Categorize posts into a list of quarters (or any timeframe deemed useful).
	Association Rule Learning	Perform association rule learning (we used the Apriori algorithm) on each set of posts (corresponding to each quarter).
	Change Detection	For each set of successive quarters identify emerging, added, perished and unexpected rule changes.
	Detect Longest Rules	Identify rules that have the highest average confidence score across all quarters.
Build Report		Present a complete report containing top tags, longest running rules and changes detected.

Table 6.1: Topic Exploration – Activity Table

Concept	Description
POST	abstract article or unitary snippet of text together with all its metadata (e.g. date, author, source, user comments)
TAG	a feature-value pair with which a post can be associated
FEATURE	a type of tag with which a post is classified
TOP TAGS	List of most commonly identified tags in a set of posts.
POSTS PER QUARTER	A categorization of each post based on the quarter it was published on.
ASSOCIATION RULES PER QUARTER	A set of association rules identified for each set of posts.
ASSOCIATION RULE	Pattern which expressed the fact that if a subset of tags is present in a post, then there is a certain probability that another subset of tags is also present (see 6.3).
CHANGES & TRENDS PER QUARTER	A set of one or two rules (if two, they must be from successive quarters) which express an emerging, added, perished or unexpected change pattern.

LONGEST RULES	Set of association rules which have the highest average confidence scores across all quarters.
REPORT	Deliverable containing top tags, longest running rules and changes identified.

Table 6.2: Topic Exploration – Concept Table

We will not represent the plethora of formulas presented by Liu et al. (2009) that compose the event change detection logic (those can be easily looked up in the original paper), but we like to note the modifications we made to the algorithm to suit our case. These are:

1. Allow for multiple values for the same feature.

Both Song et al. (2001) and Liu et al. (2009) acknowledge the importance of having tags grouped by features (e.g. organisation, market, etc.). However, they both assume that each post will be tagged by maximum one value for each feature, which was not the case in our dataset where it was common to identify multiple companies or areas in the same article.

In order to overcome this limitation, we replaced calculating the $\eta_H(A, B)$ function between values A and B of the same feature with $\delta_H(V_A, V_B)$ which computes the difference between a set of values V_A and V_B :

$$\delta_H(V_A, V_B) = \frac{1}{|V_A| + |V_B|} \left(\sum_{A \in V_A} \min_{B \in V_B} \eta_H(A, B) + \sum_{B \in V_B} \min_{A \in V_A} \eta_H(B, A) \right) \quad (6.4)$$

The formula above calculates the minimum difference (maximum similarity) for each of the values and averages the results. So Liu et al. (2009)'s original formulas

$$l_{ijk} = 1 - \eta_H \left(v(r_i^t, A_{ijk}), v(r_j^{t+k}, A_{ijk}) \right)$$

$$r_{ijk} = 1 - \eta_H \left(v(r_i^t, B_{ijm}), v(r_j^{t+k}, B_{ijm}) \right)$$

now become:

$$l_{ijk} = 1 - \delta_H \left(v(r_i^t, A_{ijk}), v(r_j^{t+k}, A_{ijk}) \right)$$

$$r_{ijk} = 1 - \delta_H \left(v(r_i^t, B_{ijm}), v(r_j^{t+k}, B_{ijm}) \right)$$

with δ_H defined based on η_H as can be seen in (6.4).

2. Only check for event changes between rules with the same features in both condition and consequent part

While not explicitly mentioned in Liu et al. (2009), we find that unexpected changes where the features differ provide little to no value to the user, so we are discarding them. As a side benefit, this also provides a large decrease in running time, as we only need to analyse changes between rules with the same set of features in both the conditional and the consequent part. In our informal tests, this reduces the average running time for the change detection algorithm from 40 seconds to 1.

3. Simplify similarity calculation between two tags

As mentioned above, we used two taxonomies for better evaluating the similarity between *areas* and *markets* tags, as is suggested by Liu et al. (2009).

However both our taxonomies were only two levels deep, so it became redundant to evaluate the common path in a taxonomy graph. Hence we adopted a much simpler formula for calculating η_H : the Jaccard distance, the coefficient often used in measuring dissimilarity between sets (Ziegler, 2012) which calculates the percentage of non-mutual concepts.

$$\eta_H(A, B) = 1 - \frac{|P_A \cap P_B|}{|P_A \cup P_B|} \quad (6.5)$$

where P_A is the set of concepts on the path from root to node A and P_B the set of concepts on the path from root to node B .

By using this formula, we get different similarities depending on the number of common concepts and the total number of concepts. For example, from high to low, we have:

Tag A	Tag B	sim
"Deepsea Mining > Barge"	"Deepsea Mining > Barge"	1.00
"Deepsea Mining > Barge"	"Deepsea Mining"	0.66
"Deepsea Mining > Barge"	"Deepsea Mining > Pontoon"	0.50
"Deepsea Mining > Barge"	"Offshore Wind > Sea Tug"	0.00

For reference, the original formulas by Liu et al. (2009) were:

$$\eta_H(A, B) = \frac{Max\left(\sum_{L_i \in P_A} WL_i, \sum_{L_j \in P_B} WL_j\right) - \sum_{L_k \in P_{comm}(A, B)} WL_k}{Max\left(\sum_{L_i \in P_A} WL_i, \sum_{L_j \in P_b} WL_j\right)} \quad (6.6)$$

where $P_{comm(A,B)}$ is set of concepts on the common path between P_A and P_B .

Since our markets taxonomy was non-hierarchical, the formula we used provided non-zero similarity for markets in different domains, i.e. in the case of "Offshore Wind > Research Vessel" and "Offshore Oil & Gas > Research Vessel", which is what the customer intended. With the exception of this case, both formulas would yield the same results for two-level deep taxonomies.

4. Slight changes to thresholds

Liu et al. (2009)'s method involved the use of certain thresholds, hardcoded values which may need to be tweaked by the user. We ended up using the same thresholds for change detection, however due to a large number of unexpected changes compared to the rest of the types of changes (which are partly due to the use of multiple values per feature), we used half of the value of θ_{em} for evaluating similarity for unexpected conditional/consequent changes. We also ignored rules which yield a degree of change of less than 0.1 in order to reduce the number of changes detected.

6.1.3 Client Feedback

In order to assess the correctness of the insights generated by the system we had the client evaluate them. While we originally wanted to perform a hypothesis-based knowledge discovery method, he could not devise any particular hypothesis from the last 2 years (on which we had available data), but did suggest we look at *Patents & R&D & Innovation* and *LNG Tankers* markets for insights. We created reports on them and opted for an evaluation of the insights generated by means precision and recall (for correlated tags) and by using a likert scale (for longest running rules and changes).

For the top 20 correlated tags in each report the client provided figures for the number of tags that are wrongly presented as the most important and another for the number of items which they expected to find in the top 20, but didn't. This way we were able to calculate the precision and recall scores. In evaluating the correctness of the longest running rules and of the changes identified, we used a 5 point likert scale (1 - incorrect/not meaningful, 5 - correct) in order to evaluate the perceived correctness of the algorithm's output.

The aggregated results for assessing the relevancy of correlated tags given the two queries are presented in Table 6.3 while the correctness of change detection logic is displayed in Table 6.4. The numbers in after a slash (/)

	wrong (/40)	missing (/40)	precision	recall	F_1
Areas	0	0	100%	100%	1.00
Markets	9	1	77%	97%	0.86
Orgs	8	8	80%	80%	0.80
People	32	32	20%	20%	0.20

Table 6.3: Correlated Tags Evaluation

	1	2	3	4	5	#	avg	std
common rules	3	0	2	1	24	30	4.43	1.25
emerging	2	0	0	1	6	9	4.0	1.63
consequent	7/0	0	0	4	2	13/6	2.53/4.33	1.69/0.47

Table 6.4: Rule and Change Detection Evaluation

represent the corrected values by (later) fixing the mistakes in the taxonomy which caused bad correlations to be found.

Additional comments made by the evaluator were:

- mistakes in taxonomy lead to the detection of most of the incorrect patterns
- input data quality is important for good results
- these were evaluated for correctness, and while insightful for laymen and people new to the industry, they are not news for domain experts
- the quality of changes depends largely on the level of expertise of the evaluator
- more interesting trend changes happen over longer period of times (years of even decades), but unfortunately we don't have news data dating that long ago

Looking at the results, we notice that the only named entity which does not make use of a gazetteer approach to NER performs by far the worst. This seems to relate to the fact that industry specialists only using taxonomy-driven NER in their solutions (personal communication, Arent van 't Spijker, 18 feb 2014), and should not come as a surprise since taxonomy-based NERs are designed to maximize performance on a particular dataset, but do not work on a different one (Nadeau & Sekine, 2007).

These results suggest there is value in such a system, but it can be difficult to filter out from the huge number of changes and rules the ones that a particular type of user might be interested in. Users not very familiar with

the topic might find even the obvious relationships and changes interesting, while domain experts might not find much value in them. It would have been interesting to evaluate the responses we got against the lower ranking changes (changes with a lower *deg* score), however we did not manage to do that due to time constraints and added difficulty for the expert to properly evaluate if a result is *insightful* or incorrect.

We do hold the belief, however, that the current results show the promise of useful results in the implementation of CI solutions and that the reference architecture is useful guidance in implementation of such projects in the future, if companies are willing to commit the amount of effort necessary.

6.1.4 Lessons Learnt

During the development of the prototype, several facts became apparent that were not accounted for in our original architecture draft. These prompted some changes to reach the final version presented in the previous chapter.

- some analysis modules can run as postprocessors
- analysis modules should be able to run on a prefiltered set
- analysis modules should be able to run at query-time
- given good filters/data sources, an annotation-based storage system will provide good performance & flexibility in schema

6.2 Expert Validation

In order to assess the architecture, we also conducted interviews with experts and recorded their feedback and suggestions for improvement. Many of the points they made have been already integrated into the final architecture, but we will mention here what the changes this process involved. The high-level architecture before and after expert feedback was integrated is presented in Figure 6.6 (changed components are highlighted with red).

We interviewed a total of 3 people, presented along with their experience in Table 6.5.

Generally the architecture has been well received, but many suggestions for improvement have been proposed and implemented.

Name	Qualification	Years Experience
Rob Guikers	Solutions Architect	19
Drs. Ing. Alain Wille	SI & CI Consultant	6
Stelian Nastase	Solutions Architect	3

Figure 6.5: Experts Table

6.2.1 Solution Requirements

The first issue raised by Rob Guikers was regarding one of our data collection solution requirements originally called *detect changes when crawling*. He made the case that this detection is not done during the crawling phase, but should actually be an analysis process to assess the impact of the changes. The extraction phase would be responsible, though, for storing each new version of the page. To accommodate this, we changed the name to *track changes when crawling*, which keeps much of the old functional responsibility of storing the new versions of posts. A simple complementary analysis module could be developed to assess the impact of the changes (e.g. in order to notify users in case of a surge of comments on an old thread).

Stelian Nastase mentioned the initial requirement of a CI solution to create *competitor profiles* is too restrictive, since one would likely be interesting in creating profiles for companies in the supply chain or a company one is interesting in acquiring for example. Furthermore, similar profiles would have to be constructed for people or markets. Therefore our initial requirement of CI profiles has been changed to *company, people and markets profiles*. He also stressed the importance of monitoring the supply chain, not just immediate suppliers as well as differentiating between B2B and B2C customers, which we did by adjusting our list of KITS in Figure 3.1.

In our original design, we mentioned one of our solution requirements to be *Integration with structured data from internal data sources*. Stelian Nastase correctly pointed out that the differentiation between internal and external system is not an issue from an architectural perspective, which should focus rather on the integration of structured and unstructured data when performing various analyses. While this is no trivial task, he suggests it is essential in a world where most companies have a lot of structured data and are only now delving into unstructured data realm. We agree that it was unreasonable to make the assumption that internal systems are generally structured data, so we changed the original requirement *use both internal and external data sources* to *use both structured and unstructured data sources*.

6.2.2 Data collection

Stelian argued that we put too much focus on crawling Web pages, when data can be present on a variety of formats (e.g. rdf) and there is a growing usage of external data services (Data-as-a-Service). This was also confirmed by Alain who noticed the abundance of third party data sources. We fixed this issue by changing the distinction between web and internal systems to a distinction between sources with direct database access and sources exposed via an intermediary protocol. Sources with direct database access will be using an ETL tool, while sources exposed via an intermediary protocol (e.g. Web pages via http protocol, ontologies via rdf) will be using a crawler (which is not necessarily a web crawler anymore).

One flaw in our original design is that we originally suggested a pipeline of postprocessors followed by a pipeline of filters in the data collection phase of our architecture. It was pointed out by both Rob Guikers and Alain Wille that it would be common to have filters and postprocessors intermixed in one larger pipeline, which we agree to and have changed accordingly.

Another issue raised by Rob was regarding unnecessary runs of pre- and postprocessors in case there are no changes to an article, so he proposed adding a hash-based caching layer before the preprocessor stage. We agree this was an useful addition with the mention that we placed this caching before postprocessors for two reasons:

1. in the case of webpages, placing a caching layer before preprocessors could easily cause cache miss for dynamic web pages (due to changing ads, dates, etc.).
2. preprocessors do not run for structured data
3. preprocessors should generally be faster than a database query for a cache

So, while we did add a caching component before the postprocessors and filters pipeline, it remains optional. Another caching layer can be added at a preprocessor level if it is deemed useful depending on the scenario at hand.

Stelian mentioned that the number of data sources is likely to vary wildly between B2B and B2C businesses and that for B2C businesses may require more aggressive filters, but it is hard to give any generic recommendations. Alain suggested that in some cases we might not want to filter out posts with no entities detected as the entities may not be known beforehand (and detected later). He added that is important to conduct an audit of available data sources before embarking on implementing a CI solution.

One of the most important issues that was raised was regarding data consist-

ency when data ingestion and analysis modules are running simultaneously. Stelian raised two issues here:

1. Analysis needs to happen on the complete dataset and importing it is lengthy; how to avoid analysis running on parts of the dataset?
2. An analysis module A requires another module D having had run beforehand. So D starts running while new data is collected into the system and by the time D finishes and A starts, the system contains new posts upon which D has not run yet.

He added there are a few ways of dealing with this issues, the most common being using snapshots (both by ETL scripts when importing and by analysis modules) or timestamps (by making analysis modules ignore posts newer than the specific point in time when data collection started). While using snapshots seems the most robust option in order to safeguard from unforeseen consistency issues, Stelian mentions there may be other (simpler) options available to ensure consistency depending on the specifics and constrains of the project.

6.2.3 Analysis & Dissemination

Stelian noticed that in the case chained analysis modules, it is very important to make sure the results produced are correct and that all the data is received correctly (what he calls *algorithm credibility*). This prompted us to add a second check phase after in the running activity of an analysis module in Figure 5.3, though the nature of these checks would depend on a module-by-module basis.

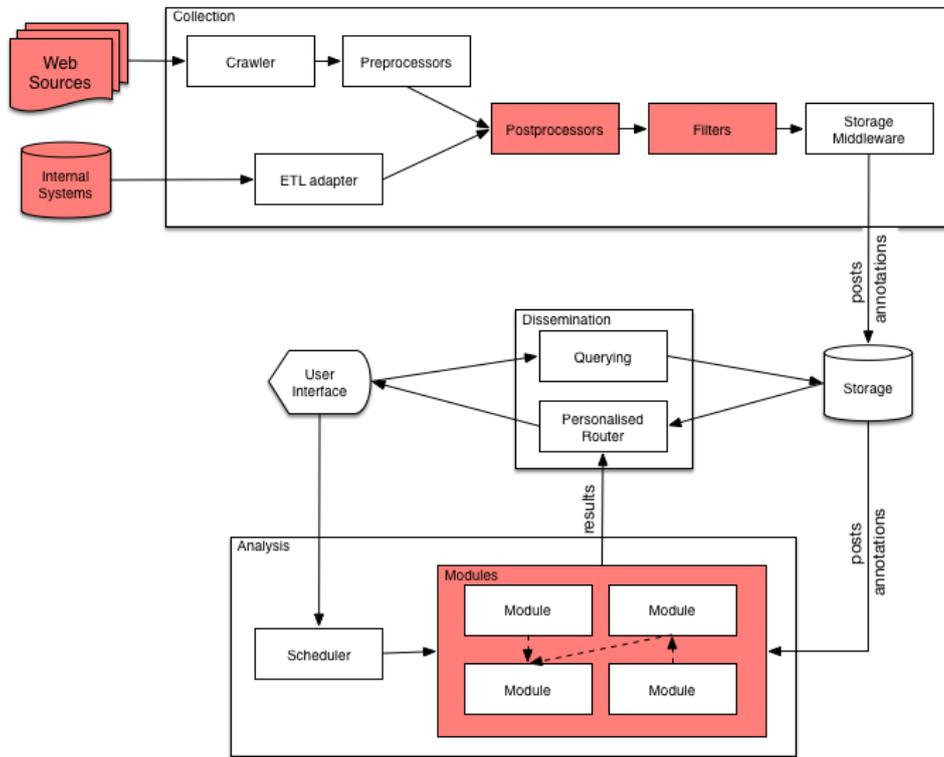
He also mentioned the name *scheduler* makes one think about time-based scheduling which would be prone to errors and that for simple CI systems (with only a couple of modules), having a simple flow of modules might be easier (foregoing the scheduler altogether), as we have seen in our prototype. He also emphasised the importance of the a personalised information router which is paramount for a good CI system. Alain suggests that automated analysis is not at the point where it is useful in practice and makes a case for a CI system which with a 24/7 data collection and a powerful adhoc querying and visualisation to help the user understand the trends rather than automatically detect them.

6.2.4 Artefacts changes

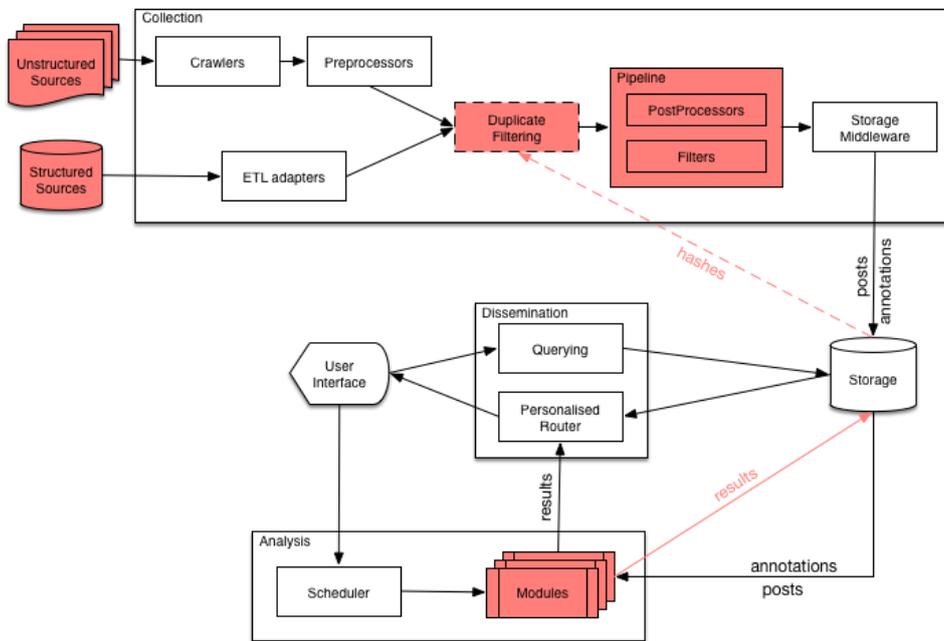
Rob suggested the name reference architecture might be confusing because it's makes one thing more about basic operations supported by an IT/IS

department for applications rather than an architecture to be used by the template for a particular domain. He suggested an *application process flow* as a more representative term, but we believe it would be too simplistic and exclude common analysis modules or implementation details.

He also proposed splitting binding time decisions between system design decisions and implementation decisions, as we did not originally make this distinction.



(a) before



(b) after

Figure 6.6: High-level architecture – before and after expert feedback

Chapter 7

Discussion

The research process generally went smooth, although suffered some changes in the process. The main issues encountered were with respect to the abundance of business-oriented research in CI compared to amount of technical details and with finding experts with knowledge of CI solutions (a niche market). We dealt with this by using a technique called *snowballing*, by following references treating technical aspects in literature and by soliciting references with knowledge of the CI solution from our interviewees.

Another issue encountered was complexity of such a system, since as we delved into the research more, we discovered many of its modules to be research topics in their own right. To cater for this variability we limited ourselves to a higher level architecture which focuses more on the interaction and requirements of the analysis modules rather than their implementation details, however we did take into account the requirements identified for implementing the most common modules.

Also the design was slightly changed as we originally were considering doing the use-case analysis after expert-validations, which we reconsidered when we decided to follow the attribute-driven design method of Bass et al. (2012), which tightly couples generating the artefact and testing it in implementation.

Our reference architecture is more flexible than the framework proposed by Zhao and Jin (2011) (in Figure 3.3), which assumes a CI solution only focusing on basic rule-based intelligence generation from identified named entities, while catering equally well for that scenario. It is also more descriptive than the architectures from Ziegler (2012) (Figure 3.5) or Dai (2013) (Figure 3.8), which specify little more than the possible analysis modules. Finally, our reference architecture is the result of extensive literature review study as well as expert interviews and the rationale of all design decisions as well as binding time decisions are recorded, making it a more useful starting point

for future research endeavors.

7.1 Limitations

7.1.1 Literature Review

The main limitations of the literature review process have to do with reproducibility. The process of selecting the top results from Google Scholar is not reproducible, both due to a temporal perspective (e.g. new papers being added) and algorithmic perspective (modifications made to the platform cause papers to be frequently re-ranked).

Results from Google Scholar which were either not publicly available or through the library subscriptions of *Utrecht University* were skipped. The issue is mitigated, however, by the fact that they account for less than 10% of the total number of papers reviewed. Another issue is that selection of subsequent articles as we performed snowballing through the references (i.e. cited and cited by articles) was biased and subject to authors' evaluation of relevancy (as we tried to favour papers tackling the technical aspects of CI implementation which tended to be scarce).

NVivo counting of references is not a very rigorous method, as the process of determining the most cited requirements or techniques could have been subject to bias. When a snippet of text would be catalogued as a reference to a requirement or technique is slightly subjective, as a lot of articles were blurring the lines between manual and technology-aided analysis or being unspecific regarding the category of techniques which would be used in implementing them.

7.1.2 Reference Architecture

We tried to keep good traceability of the considerations which lead to the design decisions we took in the reference architecture design, but a lot of design decisions were left open to be made at binding time, mostly due to their dependency of external project-specific circumstances, but also due to lack of time to perform extensive implementation cases and develop a decision-model to help in each case.

Another issue is that while our architecture if designed to be flexible, it caters mostly to the solution requirements we identified in literature or during the interviews. Although we consider them to be rather comprehensive (we have not identified any new requirements during the interview process that were not already found in literature), the need for future modules and specific

interactions between may require changes in the architecture to accommodate them.

The architecture would require more use-case analysis performed in order to be validated (see coverage in Figure 6.3) but that was not possible due to time constraints and the large amount of effort required in the process.

7.1.3 Use Case

The users of event change detection system need to be analytical, and understand support, confidence and other statistical measures for evaluating association rules for better understanding of the environment, lest they draw wrong conclusions (though a good UI may make understanding these much easier).

Change detection logic does not currently take into account confidence when evaluating the degree of how significant the change is and does not deal well with cut-off points. For example, if we use a minimum support of 0.1 for detecting association rules and the support for a rule is 0.11 in one quarter and 0.09 in the next, the rule will be filtered out in the next quarter, thus making it possible for the event change detection algorithm to identify it as a perished rule, even though the fluctuation was relatively small. Improving the event change detection algorithm to handle edge cases like these is likely to greatly increase the quality of generated insights, but is left for future research.

Changes in maritime industry are slow (it's a traditional industry), insights would probably be more insightful if analysis was performed over 10 or more years of data and we tried to identify changes between larger periods of time (1 year as suggested by the client), but this was not an option due to the availability of historical data only from the last two years.

7.2 Future Research

Being such a vast topic, naturally there are many possible opportunities for future research. Among those we mention:

1. Storage model (column- vs document- vs graph-oriented)

As mentioned above, there are a few storage models for handling the data requirements of a CI solution. An extensive analysis of the differences between them can be performed and tested against real-world use. A decision model could also be created to help architects choose the best option given the characteristics project at hand.

2. Credibility analysis

Performing credibility analysis for a the insights generated by a CI solution is an often mentioned problem in CI would close to no technical implementation details, therefore this area is a good investment of research effort.

3. UI/UX & Data visualisation

Data visualisation as well as UI/UX (user interface / user experience) has been out of the scope of the thesis since the very beginning, yet is it often mentioned as an integral part of data dissemination (personal communication, Alain Wille, 27 feb 2014). Each insight is better presented using different tools or diagrams and an insightful CI solution must be able to select the optimal way. Easily visualizing the traceability of these insights needs to also be treated, as it is important to be able to quickly identify and understand the reasoning behind them. This area remains as well part of further research.

4. Security & Compliance

While competitive intelligence is generally considered information which should be known throughout the company (personal communication, Company C1, 8 jan 2014), there are cases of some security policies must be enforced (personal communication, Arent van 't Spijker, 18 feb 2014). Security & compliance models were also out of the scope of this thesis and subject to future research.

5. CI solutions vendor analysis

The thesis aims to analyse what techniques a competitive solution should use and what features should each component possess, but does not take into account any commercial solutions already existing on the market. Although at the moment the CI solutions market is immature, a thorough vendor analysis of existing implementations and how they map to our architecture is an interesting possibility for future research.

Chapter 8

Conclusions

To summarise, the main deliverables of the thesis are:

- D1 the list of *Key Intelligence Topics* for CI solution as identified in literature (Figure 3.1)
- D2 the list of *functional requirements* for CI solution as identified in literature and confirmed by expert interviews (Figure 3.2)
- D3 the reference architecture (Figure 5.6) along with the list of binding time decisions which need to be taken when during architecture instantiation (Section 5.4)
- D4 overview of implementation details for some most common analysis modules and postprocessors (Subsection 5.3.3)

As a secondary deliverables, we can mention the contributions we made to the *Topic Exploration* module during the use-case analysis, namely:

- D5 process-deliverable diagram of the module (Figure 6.4)
- D6 modifications we made to the formulas used by Liu et al. (2009), summarised in subsection 6.1.2. Due to space considerations, we haven't reproduced the entire set of formulas used in the original paper, just the changes we made.

Reviewing the research questions we set out to answer in this thesis, we summarise how our deliverables answer them.

- RQ1 *What are the common requirements for a CI solution (the features of the solution and the types of insights to be produced)?*

The types of insights are the Key Intelligence Topics presented in D1 and the list of solution requirements are enumerated in D2, thus answering this research question.

RQ2 *What components should be part of a competitive intelligence system and what is their role?*

The main components and their interactions can be better seen in the reference architecture (D3), but we enumerate them here as well:

- *Crawlers*, which collect data from unstructured data sources
- *ETL adapters*, which collect data from structured data sources
- *Postprocessors*, a type of analysis modules which run on the post level prior to storage
- *Filters*, a component responsible for deciding if the post should be stored or discarded
- *Storage Middleware*, component responsible for storing new posts or updating existing posts
- *Storage*, the component responsible for data persistence
- *Analysis Modules*, component responsible for different types of analysis running after storage
- *Scheduler*, component scheduling the running of inter-dependent analysis modules
- *Query Tool*, responsible with querying the dataset and returning results to the user
- *Personalised Router*, responsible with routing different kind of automated reports to different users, depending on personal preferences
- *User Interface*, the user's main point of exploration

RQ3 *What techniques can be used to achieve the objectives for each of the components identified?*

This question is answered by the list of binding time decisions (D3) as well as the analysis module implementation suggestions (D4).

We find the deliverables to be useful for any company wishing to implement a CI solution as it combines the knowledge found in the literature with insights from industry experts. The architecture has been validated by being instantiated in a prototype, which delivered results evaluated by the customer to be accurate. We also suggested an implementation process which helps in integrating the deliverables in section 5.5.

References

- Agrawal, R. & Srikant, R. (1994). Fast algorithms for mining association rules. *Proc 20th int conf very large data bases*.
- Anica-Popa, I. & Cucui, G. (1841). A framework for enhancing competitive intelligence capabilities using decision support system based on web mining techniques. *International Journal of Computers, Communications & Control*.
- Bass, L., Clements, P. & Kazman, R. (2012). *Software architecture in practice*. Addison-Wesley.
- Bernhardt, D. C. (1994, February). ‘I want it fast, factual, actionable’—tailoring competitive intelligence to executives’ needs. *Long Range Planning*, 27(1), 12–24.
- Bollacker, K. D., Lawrence, S. & Giles, C. L. (1999). A system for automatic personalized tracking of scientific literature on the Web. In *the fourth acm conference* (pp. 105–113). New York, New York, USA: ACM Press.
- Bose, R. (2008). Competitive intelligence process and tools for intelligence analysis. *Industrial Management & Data Systems*, 108(4), 510–528.
- Calof, J. & Smith, J. (2009, December). The integrative domain of foresight and competitive intelligence and its impact on R&D management. *R&D Management*, 40(1), 31–39.
- Can, F. (1993, April). Incremental clustering for dynamic information processing. *ACM Transactions on Information Systems (TOIS)*, 11(2), 143–164.
- Cattell, R. (2011, May). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, 39(4), 12–27.
- Chen, H., Chau, M. & Zeng, D. (2002). CI Spider: a tool for competitive intelligence on the Web. *Decision Support Systems*, 34(1), 1–17.
- Cobb, P. (2003). Competitive Intelligence through Data Mining. *Journal of competitive intelligence and management*.
- Dai, Y. (2013, September). Designing Text Mining-Based Competitive Intelligence Systems. *epublications.uef.fi*.
- Dai, Y., Kakkonen, T. & Sutinen, E. (2011). MinEDec: a decision-support model that combines text-mining technologies with two competitive intelligence analysis methods. *International Journal of Computer Information Systems and Industrial Management Applications*.
- de Oliveira, J. P. M., Loh, S., Wives, L. K., Scarinci, R. G., Musa, D., Silva, L. & Zambenedetti, C. (2004). Applying text mining on electronic messages for competitive intelligence. , 277–286.
- Dey, L., Haque, S. M., Khurdiya, A. & Shroff, G. (2011). Acquiring competitive intelligence from social media. In *the 2011 joint workshop* (p. 1). New York, New York, USA: ACM Press.

- Fahey, L. (2007). Connecting strategy and competitive intelligence: re-focusing intelligence to produce critical strategy inputs. *Strategy & Leadership*, 35(1), 4–12.
- Fan, W., Gordon, M. D. & Pathak, P. (2005). Effective profiling of consumer information retrieval needs: a unified framework and empirical comparison. *Decision Support Systems*.
- Fan, W., Gordon, M. D. & Pathak, P. (2006, October). An integrated two-stage model for intelligent information routing. *Decision Support Systems*, 42(1), 362–374.
- Fan, W., Wallace, L., Rich, S. & Zhang, Z. (2006, September). Tapping the power of text mining. *Communications of the ACM*, 49(9), 76–82.
- Fogg, B. J., Marshall, J., Laraki, O., Osipovich, A., Varma, C., Fang, N., ... Treinen, M. (2001). *What makes Web sites credible?: a report on a large quantitative study*. ACM.
- Fricke, E. & Schulz, A. P. (2005). Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. *Systems Engineering*, 8(4).
- Glance, N., Hurst, M., Nigam, K., Siegler, M., Stockton, R. & Tomokiyo, T. (2005). *Deriving marketing intelligence from online discussion*. New York, New York, USA: ACM.
- Gordon, M., Lindsay, R. K. & Fan, W. (2002). Literature-based discovery on the World Wide Web. *ACM Transactions on Internet Technology*.
- Herring, J. P. (1999). Key intelligence topics: a process to identify and define intelligence needs. *Competitive Intelligence Review*, 10(2), 4–14.
- Hevner, A. R., March, S. T., Park, J. & Ram, S. (2004, March). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.
- Hipp, J., Güntzer, U. & Nakhaeizadeh, G. (2000, June). Algorithms for association rule mining — a general survey and comparison. *ACM SIGKDD Explorations Newsletter*, 2(1), 58–64.
- Hu, M. & Liu, B. (2004a). Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.
- Hu, M. & Liu, B. (2004b). Mining opinion features in customer reviews. *AAAI*.
- Huggins, R. (2010). Regional competitive intelligence: benchmarking and policy-making. *Regional Studies*.
- Jindal, N. & Liu, B. (2006). Mining comparative sentences and relations. *AAAI*.
- Khoury, I., El-Mawas, R. M., El-Rawas, O., Mounayar, E. F. & Artail, H. (2007, May). An Efficient Web Page Change Detection System Based on an Optimized Hungarian Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 19(5), 599–613.
- Kiczales, G., Lamping, J., Lopes, C. V., Maeda, C., Mendhekar, A. & Murphy, G. (1997). *Open implementation design guidelines*. New York,

- New York, USA: ACM.
- Kim, Y., Jung, Y. & Myaeng, S.-H. (2007). Identifying Opinion Holders in Opinion Text from Online Newspapers. In *2007 IEEE International Conference on Granular Computing (grc 2007)* (pp. 699–699). IEEE.
- Kong, L., Fu, Y., Zhou, X., Liu, Q. & Cui, Z. (2007). Study on Competitive Intelligence System based on Web. In *Workshop on intelligent information technology application (iita 2007)* (pp. 339–342). IEEE.
- Kwok, Y.-K. & Ahmad, I. (1998). Benchmarking the task graph scheduling algorithms. *Journal of Parallel and Distributed Computing*, 531–537.
- Liu, D. R., Shih, M. J., Liao, C. J. & Lai, C. H. (2009). Mining the change of event trends for decision support in environmental scanning. *Expert Systems with Applications*.
- Mani, I., House, D., Klein, G., Hirschman, L., Firmin, T. & Sundheim, B. (1999). The TIPSTER SUMMAC Text Summarization Evaluation. In *the ninth conference* (pp. 77–85). Morristown, NJ, USA: Association for Computational Linguistics.
- McGonagle, J. J. & Vella, C. M. (2012). *Proactive Intelligence*. Springer.
- Mellish, C., Scott, D., Cahill, L., Paiva, D., Evans, R. & Reape, M. (2006, March). A Reference Architecture for Natural Language Generation Systems. *Natural Language Engineering*, 12(01), 1–34.
- Mikroyannidis, A., Theodoulidis, B. & Persidis, A. (2006). PARMENIDES: towards business intelligence discovery from web data. , 1057–1060.
- Nadeau, D. & Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1), 3–26.
- Norling, P. M., Herring, J. P., Rosenkrans, W. A., Stellpflug, M. & Kaufman, S. B. (2000). Putting Competitive Technology Intelligence To Work. *Research-Technology Management*.
- Oder, N. (2001, February). The Competitive Intelligence Opportunity. , 1–3.
- Polo, J., Carrera, D., Becerra, Y., Torres, J., Ayguadé, E., Steinder, M. & Whalley, I. (2010). Performance-driven task co-scheduling for MapReduce environments. *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, 373–380.
- Prescott, J. E. (1995). The evolution of competitive intelligence. *International Review of Strategic Management*, 6, 71–90.
- Priporas, C.-V., Gatsoris, L. & Zacharis, V. (2005). Competitive intelligence activity: evidence from Greece. *Marketing Intelligence & Planning*, 23(7), 659–669.
- Radev, D. R., Libner, K. & Fan, W. (2002). Getting answers to natural language questions on the Web. *Journal of the American Society for Information Science and Technology*.
- Rao, R. (2003, November). From unstructured data to actionable intelligence. *IT Professional*, 5(6), 29–35.

- Riloff, E., Wiebe, J. & Phillips, W. (2005). Exploiting subjectivity classification to improve information extraction. In *Proceedings of the national conference on artificial intelligence*.
- Rouach, D. & Santi, P. (2001, October). Competitive Intelligence Adds Value:. *European Management Journal*, 19(5), 552–559.
- Safarnia, H., Akbari, Z. & Abbasi, A. (2011, May). Review of Competitive Intelligence & Competitive Advantage in the Industrial Estates Companies in the Kerman City: Appraisal and Testing of Model by Amos Graphics. *International Business and Management*, 2(2), 47–61.
- Shih, M.-J., Liu, D.-R. & Hsu, M.-L. (2010, April). Discovering competitive intelligence by mining changes in patent trends. *Expert Systems with Applications*, 37(4), 2882–2890.
- Smith, J. R., Wright, S. & Pickton, D. (2010). Competitive intelligence programmes for SMEs in France: Evidence of changing attitudes. *Journal of Strategic Marketing*.
- Song, H. S., Kim, J. k. & Kim, S. H. (2001, October). Mining the change of customer behavior in an internet shopping mall. *Expert Systems with Applications*, 21(3), 157–168.
- Teo, T. S. & Choo, W. Y. (2001). Assessing the impact of using the Internet for competitive intelligence. *Information & management*, 39(1), 67–83.
- Tsubiks, O. (2012). MINING CONSUMER TRENDS FROM ONLINE REVIEWS: AN APPROACH FOR MARKET RESEARCH.
- Tuan, L. T. (2013). Leading to learning and competitive intelligence. *Learning Organization, The*, 20(3), 216–239.
- van de Weerd, I. & Brinkkemper, S. (2008). Meta-Modeling for Situational Analysis and Design Methods. In *Handbook of research on modern systems analysis and design technologies and applications* (pp. 35–54). IGI Global.
- van de Weerd, I., de Weerd, S. & Brinkkemper, S. (2007). Developing a Reference Method for Game Production by Method Comparison. *Situational Method Engineering: Fundamentals and Experiences*, 313–327.
- Vaughan, L., Yang, R., Chen, C., Liang, W. & Li, B. (2010). Extending web co-link analysis to web co-word analysis for competitive intelligence. In *Proceedings of the annual conference of the canadian association for information science*.
- Vedder, R. G., Vanecek, M. T., Guynes, C. S. & Cappel, J. J. (1999). CEO and CIO perspectives on competitive intelligence. *Communications of the ACM*, 42(8), 108–116.
- Walter, C. (2005, August). Kryder’s Law. *Scientific American*, 293(2), 32–33.
- Wei, C.-P. & Lee, Y.-H. (2004, March). Event detection from online news documents for supporting environmental scanning. *Decision Support Systems*, 36(4), 385–401.

- Weiss, A. & Naylor, E. (2010, October). Part I: Competitive intelligence: How independent information professionals contribute to organizational success. *Bulletin of the American Society for Information Science and Technology*, 37(1), 30–34.
- Wright, S., Bisson, C. & Duffy, A. P. (2012). Applying a behavioural and operational diagnostic typology of competitive intelligence practice: empirical evidence from the SME sector in Turkey. *Journal of Strategic Marketing*.
- Wright, S., Pickton, D. W. & Callow, J. (2002). Competitive intelligence in UK firms: a typology. *Marketing Intelligence & Planning*, 20(6), 349–360.
- Xu, K., Liao, S. S., Li, J. & Song, Y. (2011). Mining comparative opinions from customer reviews for Competitive Intelligence. *Decision Support Systems*.
- Yang, Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *ICML*.
- Yu, H. & Hatzivassiloglou, V. (2003). Towards answering opinion questions. In *the 2003 conference* (pp. 129–136). Morristown, NJ, USA: Association for Computational Linguistics.
- Zanasi, A. (2001). Competitive intelligence through data mining public sources. *Competitive Intelligence Review*, 9(1), 44–54.
- Zelenko, D., Aone, C. & Richardella, A. (2003, March). Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3, 1083–1106.
- Zhai, Y. & Liu, B. (2005). Extracting Web Data Using Instance-Based Learning. *Web Information Systems Engineering–WISE 2005*, 318–331.
- Zhao, J. & Jin, P. (2009). Towards the Extraction of Intelligence about Competitor from the Web. , 118–127.
- Zhao, J. & Jin, P. (2010a, April). Conceptual Modeling for Competitive Intelligence Hiding in the Internet. *Journal of Software*, 5(4).
- Zhao, J. & Jin, P. (2010b). A Framework for Credibility Evaluation of Web-Based Competitive Intelligence. *Proc Of IITSI'10*.
- Zhao, J. & Jin, P. (2011, January). Extraction and Credibility Evaluation of Web-based Competitive Intelligence. *Journal of Software*, 6(8), 1513–1520.
- Zhao, J. & Jin, P. (2013). Design Consideration on a Real-time System for Collecting Intelligence from the Web.
- Zhao, J., Jin, P. & Liu, Y. (2010). Business Relations in the Web: Semantics and a Case Study. *Journal of Software*, 5(8), 826–833.
- Ziegler, C.-N. (2012). Competitive Intelligence Capturing Systems. *Mining for Strategic Competitive Intelligence*, 51–62.
- Ziegler, C. N. & Jung, S. (2009). Leveraging sources of collective wisdom on the web for discovering technology synergies. In *Proceedings of the*

- 32nd international acm sigir conference on research and development in information retrieval.*
- Ziegler, C. N., Simon, K. & Lausen, G. (2006). Automatic computation of semantic proximity using taxonomic knowledge. *Proceedings of the 15th ACM international conference on Information and knowledge management.*
- Ziegler, C.-N. & Skubacz, M. (2006). Towards Automated Reputation and Brand Monitoring on the Web. In *2006 iee/wic/acm international conference on web intelligence (wi 2006 main conference proceedings)(wi'06)* (pp. 1066–1072). IEEE.
- Ziegler, C.-N. & Skubacz, M. (2012). Content Extraction from News Pages Using Particle Swarm Optimization. , 135–149.
- Ziegler, C. N., Skubacz, M. & Viermetz, M. (2012). Mining and Exploring Customer Feedback Using Language Models and Treemaps - Springer. *Proceedings of the 15th ACM international conference on Information and knowledge management.*

Appendix A

Papers Reviewed

This presents the papers selected for analysis in the literature review phase of the research and are the ones coded into NVivo. Note that both main results and papers citing and cited by are included (up to third level) in each round.

R1: Competitive Intelligence

1. Anica-Popa and Cucui (1841)
2. Bernhardt (1994)
3. Prescott (1995)
4. Yang and Pedersen (1997)
5. Bollacker, Lawrence and Giles (1999)
6. Vedder, Vanecek, Guynes and Cappel (1999)
7. Norling, Herring, Rosenkrans, Stellpflug and Kaufman (2000)
8. Teo and Choo (2001)
9. Zanasi (2001)
10. Rouach and Santi (2001)
11. Chen, Chau and Zeng (2002)
12. Wright, Pickton and Callow (2002)
13. Radev, Libner and Fan (2002)
14. Gordon, Lindsay and Fan (2002)

15. Cobb (2003)
16. Rao (2003)
17. de Oliveira et al. (2004)
18. Hu and Liu (2004b)
19. Wei and Lee (2004)
20. Fan, Gordon and Pathak (2005)
21. Mikroyannidis, Theodoulidis and Persidis (2006)
22. Fan, Wallace, Rich and Zhang (2006)
23. Fan, Gordon and Pathak (2006)
24. Khoury, El-Mawas, El-Rawas, Mounayar and Artail (2007)
25. Bose (2008)
26. Zhao and Jin (2009)
27. Zhao and Jin (2010b)
28. Zhao, Jin and Liu (2010)
29. Zhao and Jin (2010a)
30. Dey, Haque, Khurdiya and Shroff (2011)
31. Dai, Kakkonen and Sutinen (2011)
32. Zhao and Jin (2011)
33. Tsubiks (2012)
34. Zhao and Jin (2013)
35. Dai (2013)

R2: Competitive Intelligence (> 2009)

1. Yu and Hatzivassiloglou (2003)
2. Zelenko, Aone and Richardella (2003)
3. Hu and Liu (2004a)
4. Riloff, Wiebe and Phillips (2005)
5. Ziegler, Simon and Lausen (2006)
6. Jindal and Liu (2006)

7. Ziegler and Skubacz (2006)
8. Kim, Jung and Myaeng (2007)
9. Kong, Fu, Zhou, Liu and Cui (2007)
10. Ziegler and Jung (2009)
11. Liu, Shih, Liao and Lai (2009)
12. Calof and Smith (2009)
13. Smith, Wright and Pickton (2010)
14. Vaughan, Yang, Chen, Liang and Li (2010)
15. Huggins (2010)
16. Shih, Liu and Hsu (2010)
17. Weiss and Naylor (2010)
18. Xu, Liao, Li and Song (2011)
19. Wright, Bisson and Duffy (2012)
20. Ziegler (2012)
21. Ziegler and Skubacz (2012)
22. Ziegler, Skubacz and Viermetz (2012)
23. Tuan (2013)

Appendix B

Interview Protocol

- ★ *Thank you for agreeing to this meeting.*
 - ★ *(Needless to say) you have the right not to answer questions or stop the interview at any time. Interview is semi-structured, so feel free to speak freely.*
 - ★ *Short description of the research and how information from this interview is going to be used. . . Ask for consent.*
 - ★ *Would you like this interview to be confidential (for you/the company)?*
 - ★ *Is it OK to record this interview?*
1. Some general questions about organization.
 - a. What is the size of the IT/IS department?
 - b. What are major IS projects implemented or in development?
 - c. What is your role in these projects?
 2. Competitive Intelligence Key Intelligence Topics
 - a. Competitors (activity, products, jobs)
 - b. Customers (segmentation, feedback, trends)
 - c. Markets (regulations, research & patents trends)
 - d. Suppliers (activities)
 3. Competitive Intelligence Systems
- ↪ *probe for attitude: immune/task-driven/operational/strategic*
- ↪ *probe for existing systems: standard/tailored/bespoke*

4. (if applicable) Architectural Decisions

a. What are the core functionalities needed/planned/implemented?

↪ *probe for ontology mapping, credibility analysis, intelligent distribution, security policies, subjectivity detection*

b. What technological stack have you decided for?

↪ *probe for commercial/open-source*

c. What are the techniques you used in your implementation? How would you rate their sophistication?

d. Is your architecture extensible? Are there plans for continuous improvement or was it a one-off effort?

5. Lesson learnt

a. Is such a project feasible? Is ROI high enough?

b. How successful were you in your endeavours? How was the solution received by management/users?

↪ *probe for usage and impact*

c. What would you do differently if had the chance?

★ *If appropriate, ask for permission for follow-up and for taking part in an expert panel assessment.*

★ *Thank you again for your time.*

Interview Consent Form

Title of research project:

A Reference Architecture for a Dynamic Competitive Intelligence Solution

Name and position of researcher:

Alex Cepoi, Master student in Business Informatics, Utrecht University

1. I confirm that I have been given information for the above study and have had the opportunity to ask questions.
2. I understand that my participation is voluntary and that I am free to withdraw at any time without giving a reason.
3. I agree to take part in the study.
4. I agree to the interview being audio recorded. Yes No
5. I agree to the use of anonymised quotes in publications. Yes No

_____	_____	_____
Name of Participant	Date	Signature
_____	_____	_____
Researcher	Date	Signature

Appendix C

Diagram Symbols

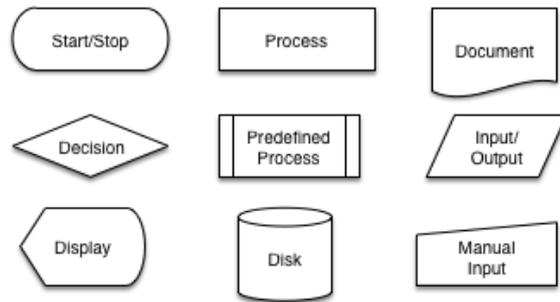


Figure C.1: Flowchart symbols

Appendix D

Topic Exploration – Sample Report

Below is a sample Topic Exploration Report on “Patents & R&D & Innovation” run for the last 2 years:

```
$ python bin/run_ecd.py 8 4 --src newsitems_nofeeds --
  support 0.1 --filter "patents_&_r&d_&_innovation" --
  ft_types orgs people areas markets --top 20
```

```
Excluded posts: {'no-date': 56, 'different-lang': 2}
Posts: 846 | Features: 2898 | Promotions: 6
```

```
Classified:
```

```
2013 Q4 153
```

```
2013 Q3 101
```

```
2013 Q2 83
```

```
2013 Q1 66
```

```
2012 Q4 77
```

```
2012 Q3 51
```

```
2012 Q2 52
```

```
2012 Q1 54
```

```
Total items: 637
```

```
==> 2013 Q4
```

```
Items: 153 | Features: 758 (+ 2 promoted)
```

```
153 ["markets", "Patents & R&D & Innovation"]
```

```
58 ["areas", "Asia > Korea, South"]
```

```
47 ["markets", "Offshore Oil & Gas > Research Vessel"]
```

```
47 ["markets", "Defence & Security > Research Vessel"]
```

```
41 ["markets", "Offshore Oil & Gas"]
```

```
33 ["areas", "Asia > China"]
```

```
17 ["markets", "Harbour & Terminal"]
```

```
17 ["markets", "Ship Repair"]
```

```

16 ["areas", "Europe"]
16 ["markets", "Defence & Security"]
15 ["areas", "Europe > Netherlands"]
14 ["orgs", "Daewoo"]
13 ["areas", "Asia > Russia"]
12 ["areas", "Arctic Region"]
11 ["areas", "Europe > France"]
11 ["markets", "Environmental Safety & Control"]
11 ["markets", "Offshore Wind"]
11 ["markets", "Fishing"]
11 ["areas", "Europe > Germany"]
11 ["areas", "Asia"]

```

Rules: 24 (+74 useless):

```

conf lift rule
0.468 3.255 ["markets", "Patents & R&D & Innovation"]
["markets", "Offshore Oil & Gas > Research Vessel"]
-> ["markets", "Defence & Security > Research Vessel"]
["areas", "Asia > Korea, South"]
0.468 3.255 ["markets", "Defence & Security >
Research Vessel"] ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Asia > Korea, South"] ["
markets", "Offshore Oil & Gas > Research Vessel"]
0.404 1.509 ["markets", "Defence & Security >
Research Vessel"] ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Offshore Oil & Gas"]
0.463 1.509 ["markets", "Patents & R&D & Innovation"]
["markets", "Offshore Oil & Gas"] -> ["markets", "
Offshore Oil & Gas > Research Vessel"]
0.463 1.509 ["markets", "Patents & R&D & Innovation"]
["markets", "Offshore Oil & Gas"] -> ["markets", "
Defence & Security > Research Vessel"]
0.468 1.235 ["markets", "Defence & Security >
Research Vessel"] ["markets", "Patents & R&D &
Innovation"] ["markets", "Offshore Oil & Gas >
Research Vessel"] -> ["areas", "Asia > Korea, South"]
0.468 1.235 ["markets", "Defence & Security >
Research Vessel"] ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Asia > Korea, South"]
0.468 1.235 ["markets", "Patents & R&D & Innovation"]
["markets", "Offshore Oil & Gas > Research Vessel"]
-> ["areas", "Asia > Korea, South"]
0.379 1.235 ["areas", "Asia > Korea, South"] ["
markets", "Patents & R&D & Innovation"] -> ["markets
", "Defence & Security > Research Vessel"]
0.379 1.235 ["areas", "Asia > Korea, South"] ["
markets", "Patents & R&D & Innovation"] -> ["markets
", "Offshore Oil & Gas > Research Vessel"]

```

```

0.379 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Korea, South"]
0.307 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
"]
0.307 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas > Research Vessel
"]
0.268 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas"]
0.216 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > China"]
0.144 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
"] ["areas", "Asia > Korea, South"] ["markets", "
Offshore Oil & Gas > Research Vessel"]
0.144 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Korea, South"] ["markets", "
Offshore Oil & Gas > Research Vessel"]
0.144 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
"] ["areas", "Asia > Korea, South"]
0.124 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
"] ["markets", "Offshore Oil & Gas"]
0.124 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas"] ["markets", "
Offshore Oil & Gas > Research Vessel"]

```

==> 2013 Q3

Items: 101 | Features: 560 (+ 0 promoted)

```

101 ["markets", "Patents & R&D & Innovation"]
29 ["markets", "Defence & Security > Research Vessel"]
28 ["markets", "Offshore Oil & Gas > Research Vessel"]
27 ["areas", "Asia > Korea, South"]
20 ["markets", "Harbour & Terminal"]
16 ["areas", "North America > United States"]
13 ["areas", "Europe"]
13 ["markets", "Offshore Oil & Gas"]
13 ["markets", "Offshore Wind"]
13 ["markets", "Ship Repair"]
12 ["areas", "Asia > China"]
11 ["orgs", "Hyundai"]
10 ["markets", "Environmental Safety & Control"]
10 ["markets", "Environmental Safety & Control >
Pollution Control Vessel"]
10 ["areas", "Europe > Germany"]
7 ["areas", "Europe > Netherlands"]
7 ["orgs", "Samsung"]

```

```

7 ["areas", "Arctic Region"]
7 ["orgs", "EU"]
6 ["markets", "Offshore Wind > Construction Vessel"]

Rules: 11 (+19 useless):
conf lift rule
0.287 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel"]
]
0.277 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas > Research Vessel"]
]
0.267 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Korea, South"]
0.198 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Harbour & Terminal"]
0.158 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "North America > United States"]
0.129 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas"]
0.129 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Ship Repair"]
0.129 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Wind"]
0.129 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe"]
0.119 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > China"]
0.109 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["orgs", "Hyundai"]

==> 2013 Q2
Items: 83 | Features: 518 (+ 3 promoted)
83 ["markets", "Patents & R&D & Innovation"]
22 ["areas", "Asia > Korea, South"]
16 ["areas", "North America > United States"]
13 ["markets", "Offshore Oil & Gas"]
13 ["markets", "Harbour & Terminal"]
12 ["areas", "Europe > Norway"]
11 ["markets", "Defence & Security > Research Vessel"]
10 ["markets", "Offshore Wind"]
10 ["markets", "Offshore Oil & Gas > Research Vessel"]
8 ["areas", "Europe > France"]
7 ["areas", "Europe"]
7 ["areas", "Europe > Germany"]
7 ["markets", "Ship Repair"]
7 ["areas", "Asia > China"]
6 ["areas", "Europe > Netherlands"]
6 ["areas", "Oceania > Australia"]

```

```

6 ["orgs", "Hyundai"]
6 ["areas", "Arctic Region"]
5 ["markets", "Offshore Wind > Construction Vessel"]
5 ["markets", "Offshore Oil & Gas > FPSO"]

Rules: 8 (+16 useless):
conf lift rule
0.265 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Korea, South"]
0.193 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "North America > United States"]
0.157 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas"]
0.157 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Harbour & Terminal"]
0.145 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe > Norway"]
0.133 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
"]
0.120 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas > Research Vessel
"]
0.120 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Wind"]

==> 2013 Q1
Items: 66 | Features: 350 (+ 0 promoted)
66 ["markets", "Patents & R&D & Innovation"]
23 ["areas", "Asia > Korea, South"]
17 ["markets", "Offshore Oil & Gas > Research Vessel"]
17 ["markets", "Defence & Security > Research Vessel"]
15 ["areas", "Asia > China"]
12 ["markets", "Offshore Oil & Gas"]
12 ["markets", "Harbour & Terminal"]
12 ["markets", "Offshore Wind"]
10 ["markets", "Ship Repair"]
8 ["areas", "Europe > Netherlands"]
6 ["markets", "Offshore Wind > Construction Vessel"]
6 ["orgs", "Daewoo"]
6 ["areas", "Europe > Norway"]
5 ["orgs", "Hyundai"]
5 ["areas", "Europe > Germany"]
5 ["areas", "Asia > Japan"]
4 ["areas", "Europe > France"]
4 ["markets", "Environmental Safety & Control"]
4 ["markets", "Defence & Security"]
4 ["markets", "Seagoing Transport > LNG Tanker"]

```

```

Rules: 9 (+17 useless):
conf lift rule
0.348 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Korea, South"]
0.258 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas > Research Vessel
"]
0.258 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
"]
0.227 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > China"]
0.182 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Wind"]
0.182 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Harbour & Terminal"]
0.182 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas"]
0.152 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Ship Repair"]
0.121 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe > Netherlands"]

==> 2012 Q4
Items: 77 | Features: 405 (+ 0 promoted)
77 ["markets", "Patents & R&D & Innovation"]
29 ["areas", "Asia > Korea, South"]
17 ["markets", "Offshore Oil & Gas > Research Vessel"]
17 ["markets", "Defence & Security > Research Vessel"]
13 ["markets", "Harbour & Terminal"]
12 ["markets", "Offshore Oil & Gas"]
11 ["markets", "Offshore Wind"]
9 ["markets", "Ship Repair"]
9 ["areas", "Asia > China"]
8 ["orgs", "Daewoo"]
8 ["areas", "Europe"]
8 ["areas", "Europe > Norway"]
8 ["areas", "North America > United States"]
7 ["areas", "Europe > Netherlands"]
7 ["areas", "Europe > Germany"]
6 ["markets", "Offshore Oil & Gas > FPSO"]
6 ["orgs", "Hyundai"]
5 ["markets", "Environmental Safety & Control"]
5 ["orgs", "Samsung"]
5 ["areas", "Europe > Denmark"]

Rules: 22 (+46 useless):
conf lift rule

```

```

0.529 4.529 ["markets", "Defence & Security >
  Research Vessel"] ["markets", "Patents & R&D &
  Innovation"] -> ["markets", "Offshore Oil & Gas >
  Research Vessel"] ["areas", "Asia > Korea, South"]
0.529 4.529 ["markets", "Offshore Oil & Gas >
  Research Vessel"] ["markets", "Patents & R&D &
  Innovation"] -> ["markets", "Defence & Security >
  Research Vessel"] ["areas", "Asia > Korea, South"]
0.529 1.406 ["markets", "Defence & Security >
  Research Vessel"] ["markets", "Patents & R&D &
  Innovation"] -> ["areas", "Asia > Korea, South"]
0.529 1.406 ["markets", "Offshore Oil & Gas >
  Research Vessel"] ["markets", "Patents & R&D &
  Innovation"] -> ["areas", "Asia > Korea, South"]
0.529 1.406 ["markets", "Defence & Security >
  Research Vessel"] ["markets", "Offshore Oil & Gas >
  Research Vessel"] ["markets", "Patents & R&D &
  Innovation"] -> ["areas", "Asia > Korea, South"]
0.310 1.406 ["areas", "Asia > Korea, South"] ["
  markets", "Patents & R&D & Innovation"] -> ["markets
  ", "Offshore Oil & Gas > Research Vessel"]
0.310 1.406 ["areas", "Asia > Korea, South"] ["
  markets", "Patents & R&D & Innovation"] -> ["markets
  ", "Defence & Security > Research Vessel"]
0.377 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Korea, South"]
0.221 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas > Research Vessel
  "]
0.221 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
  "]
0.169 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Harbour & Terminal"]
0.156 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas"]
0.143 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Wind"]
0.117 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas > Research Vessel
  "] ["areas", "Asia > Korea, South"]
0.117 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Ship Repair"]
0.117 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > China"]
0.117 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
  "] ["areas", "Asia > Korea, South"]

```

```

0.117  1.000  ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
   "] ["markets", "Offshore Oil & Gas > Research Vessel
   "] ["areas", "Asia > Korea, South"]
0.104  1.000  ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe > Norway"]
0.104  1.000  ["markets", "Patents & R&D & Innovation"]
-> ["areas", "North America > United States"]

==> 2012 Q3
Items: 51 | Features: 264 (+ 1 promoted)
  51 ["markets", "Patents & R&D & Innovation"]
  15 ["areas", "Asia > Korea, South"]
  14 ["areas", "Asia > China"]
  10 ["markets", "Defence & Security > Research Vessel"]
  10 ["markets", "Offshore Oil & Gas"]
  10 ["markets", "Offshore Oil & Gas > Research Vessel"]
   9 ["areas", "Europe"]
   8 ["markets", "Harbour & Terminal"]
   8 ["areas", "Europe > Germany"]
   7 ["orgs", "Daewoo"]
   7 ["orgs", "Hyundai"]
   6 ["areas", "Asia > Japan"]
   6 ["markets", "Offshore Wind"]
   6 ["markets", "Ship Repair"]
   5 ["areas", "Europe > Norway"]
   5 ["areas", "Asia"]
   4 ["markets", "Seagoing Transport > Bulk Carrier"]
   4 ["people", "Lloyd"]
   4 ["orgs", "Ministry of Knowledge Economy"]
   4 ["markets", "Environmental Safety & Control"]

Rules: 19 (+31 useless):
conf  lift  rule
1.000  3.400  ["markets", "Patents & R&D & Innovation"]
         ["orgs", "Daewoo"] -> ["areas", "Asia > Korea, South
         "]
0.467  3.400  ["markets", "Patents & R&D & Innovation"]
         ["areas", "Asia > Korea, South"] -> ["orgs", "Daewoo
         "]
0.857  2.914  ["markets", "Patents & R&D & Innovation"]
         ["orgs", "Hyundai"] -> ["areas", "Asia > Korea,
         South"]
0.400  2.914  ["markets", "Patents & R&D & Innovation"]
         ["areas", "Asia > Korea, South"] -> ["orgs", "
         Hyundai"]
0.294  1.000  ["markets", "Patents & R&D & Innovation"]
         -> ["areas", "Asia > Korea, South"]

```

```

0.275 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > China"]
0.196 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
"]
0.196 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas > Research Vessel
"]
0.196 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas"]
0.176 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe"]
0.157 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Harbour & Terminal"]
0.157 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe > Germany"]
0.137 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["orgs", "Daewoo"] ["areas", "Asia > Korea, South
"]
0.137 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["orgs", "Daewoo"]
0.137 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["orgs", "Hyundai"]
0.118 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Ship Repair"]
0.118 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["orgs", "Hyundai"] ["areas", "Asia > Korea,
South"]
0.118 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Japan"]
0.118 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Wind"]

```

==> 2012 Q2

Items: 52 | Features: 225 (+ 1 promoted)

```

52 ["markets", "Patents & R&D & Innovation"]
16 ["areas", "Asia > Korea, South"]
16 ["areas", "Asia > China"]
9 ["markets", "Offshore Oil & Gas"]
8 ["markets", "Defence & Security > Research Vessel"]
8 ["markets", "Offshore Oil & Gas > Research Vessel"]
7 ["orgs", "Hyundai"]
6 ["markets", "Environmental Safety & Control"]
6 ["areas", "Asia > Japan"]
6 ["markets", "Offshore Wind"]
6 ["markets", "Ship Repair"]
5 ["areas", "Europe > Norway"]
5 ["areas", "Europe > Netherlands"]
5 ["markets", "Harbour & Terminal"]

```

```

5 ["orgs", "Daewoo"]
4 ["markets", "Seagoing Transport > Bulk Carrier"]
4 ["areas", "Asia"]
4 ["areas", "Europe > Germany"]
3 ["areas", "Southeast Asia > Singapore"]
3 ["markets", "Seagoing Transport > LNG Tanker"]

Rules: 13 (+23 useless):
conf lift rule
0.857 2.786 ["markets", "Patents & R&D & Innovation"]
["orgs", "Hyundai"] -> ["areas", "Asia > Korea,
South"]
0.375 2.786 ["markets", "Patents & R&D & Innovation"]
["areas", "Asia > Korea, South"] -> ["orgs", "
Hyundai"]
0.308 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Korea, South"]
0.308 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > China"]
0.173 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas"]
0.154 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research Vessel
"]
0.154 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas > Research Vessel
"]
0.135 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["orgs", "Hyundai"]
0.115 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["orgs", "Hyundai"] ["areas", "Asia > Korea,
South"]
0.115 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Wind"]
0.115 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Japan"]
0.115 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Environmental Safety & Control"]
0.115 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Ship Repair"]

==> 2012 Q1
Items: 54 | Features: 292 (+ 0 promoted)
54 ["markets", "Patents & R&D & Innovation"]
23 ["areas", "Asia > Korea, South"]
19 ["areas", "Asia > China"]
10 ["areas", "Asia > Japan"]
10 ["markets", "Defence & Security > Research Vessel"]
10 ["markets", "Offshore Oil & Gas > Research Vessel"]

```

```

9 ["markets", "Seagoing Transport > LNG Tanker"]
8 ["markets", "Seagoing Transport > Bulk Carrier"]
8 ["areas", "Europe"]
7 ["markets", "Offshore Oil & Gas"]
6 ["markets", "Ship Repair"]
6 ["areas", "Europe > Germany"]
5 ["areas", "Europe > Netherlands"]
5 ["orgs", "Hyundai"]
5 ["markets", "Offshore Wind"]
4 ["orgs", "D Center"]
4 ["orgs", "Mitsui"]
4 ["areas", "Europe > France"]
4 ["orgs", "Daewoo"]
3 ["areas", "Southeast Asia > Singapore"]

```

Rules: 23 (+39 useless):

```

conf lift rule
0.368 2.211 ["areas", "Asia > China"] ["markets", "
  Patents & R&D & Innovation"] -> ["markets", "Seagoing
  Transport > LNG Tanker"]
0.778 2.211 ["markets", "Seagoing Transport > LNG
  Tanker"] ["markets", "Patents & R&D & Innovation"] ->
  ["areas", "Asia > China"]
0.750 2.132 ["markets", "Patents & R&D & Innovation"]
  ["markets", "Seagoing Transport > Bulk Carrier"] ->
  ["areas", "Asia > China"]
0.316 2.132 ["areas", "Asia > China"] ["markets", "
  Patents & R&D & Innovation"] -> ["markets", "Seagoing
  Transport > Bulk Carrier"]
0.600 1.409 ["markets", "Patents & R&D & Innovation"]
  ["areas", "Asia > Japan"] -> ["areas", "Asia > Korea
  , South"]
0.261 1.409 ["markets", "Patents & R&D & Innovation"]
  ["areas", "Asia > Korea, South"] -> ["areas", "Asia
  > Japan"]
0.426 1.000 ["markets", "Patents & R&D & Innovation"]
  -> ["areas", "Asia > Korea, South"]
0.352 1.000 ["markets", "Patents & R&D & Innovation"]
  -> ["areas", "Asia > China"]
0.185 1.000 ["markets", "Patents & R&D & Innovation"]
  -> ["markets", "Defence & Security > Research Vessel
  "]
0.185 1.000 ["markets", "Patents & R&D & Innovation"]
  -> ["areas", "Asia > Japan"]
0.185 1.000 ["markets", "Patents & R&D & Innovation"]
  -> ["markets", "Offshore Oil & Gas > Research Vessel
  "]
0.167 1.000 ["markets", "Patents & R&D & Innovation"]
  -> ["markets", "Seagoing Transport > LNG Tanker"]

```

```

0.148 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Seagoing Transport > Bulk Carrier"]
0.148 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > China"] ["areas", "Asia > Korea
, South"]
0.148 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe"]
0.130 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > China"] ["markets", "Seagoing
Transport > LNG Tanker"]
0.130 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Offshore Oil & Gas"]
0.111 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > Japan"] ["areas", "Asia > Korea
, South"]
0.111 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Asia > China"] ["markets", "Seagoing
Transport > Bulk Carrier"]
0.111 1.000 ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe > Germany"]

```

Change Detection

```
*** 2013 Q3 => 2013 Q4
```

```
> emerging
```

```
deg +1.08 (0.27) ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Offshore Oil & Gas"]
deg +0.82 (0.22) ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Asia > China"]
deg +0.42 (0.38) ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Asia > Korea, South"]

```

```
> cons_change
```

```
deg +1.23 (0.13) ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Offshore Wind"] ==>
(0.10) ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security"]
deg +1.10 (0.16) ["markets", "Patents & R&D &
Innovation"] -> ["areas", "North America > United
States"] ==> (0.47) ["markets", "Defence &
Security > Research Vessel"] ["markets", "Patents
& R&D & Innovation"] -> ["areas", "Asia > Korea,
South"]
deg +1.10 (0.16) ["markets", "Patents & R&D &
Innovation"] -> ["areas", "North America > United
States"] ==> (0.47) ["markets", "Patents & R&D &
Innovation"] ["markets", "Offshore Oil & Gas >
Research Vessel"] -> ["areas", "Asia > Korea,
South"]

```

```

deg +1.04 (0.13) ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Offshore Wind"] ==>
(0.12) ["markets", "Patents & R&D & Innovation"]
-> ["markets", "Defence & Security > Research
Vessel"] ["markets", "Offshore Oil & Gas"]
deg +1.04 (0.13) ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Offshore Wind"] ==>
(0.40) ["markets", "Defence & Security > Research
Vessel"] ["markets", "Patents & R&D & Innovation
"] -> ["markets", "Offshore Oil & Gas"]
deg +1.04 (0.13) ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Offshore Wind"] ==>
(0.46) ["markets", "Patents & R&D & Innovation"]
["markets", "Offshore Oil & Gas"] -> ["markets",
"Offshore Oil & Gas > Research Vessel"]
deg +1.04 (0.13) ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Offshore Wind"] ==>
(0.46) ["markets", "Patents & R&D & Innovation"]
["markets", "Offshore Oil & Gas"] -> ["markets",
"Defence & Security > Research Vessel"]

*** 2013 Q2 => 2013 Q3
> emerging
deg +1.30 (0.28) ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Offshore Oil & Gas >
Research Vessel"]
deg +1.17 (0.29) ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Defence & Security >
Research Vessel"]
deg +0.26 (0.20) ["markets", "Patents & R&D &
Innovation"] -> ["markets", "Harbour & Terminal"]

*** 2013 Q1 => 2013 Q2
> cons_change
deg +1.57 (0.23) ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Asia > China"] ==>
(0.14) ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe > Norway"]
deg +1.18 (0.23) ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Asia > China"] ==>
(0.19) ["markets", "Patents & R&D & Innovation"]
-> ["areas", "North America > United States"]
deg +0.84 (0.12) ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Europe > Netherlands"]
==> (0.14) ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Europe > Norway"]
deg +0.63 (0.12) ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Europe > Netherlands"]
==> (0.19) ["markets", "Patents & R&D &

```

```

    Innovation"] -> ["areas", "North America > United
    States"]

*** 2012 Q4 => 2013 Q1
> emerging
deg +0.94 (0.23) ["markets", "Patents & R&D &
    Innovation"] -> ["areas", "Asia > China"]
deg +0.30 (0.15) ["markets", "Patents & R&D &
    Innovation"] -> ["markets", "Ship Repair"]
deg +0.27 (0.18) ["markets", "Patents & R&D &
    Innovation"] -> ["markets", "Offshore Wind"]

> cons_change
deg +0.86 (0.10) ["markets", "Patents & R&D &
    Innovation"] -> ["areas", "Europe > Norway"] ==>
    (0.12) ["markets", "Patents & R&D & Innovation"]
-> ["areas", "Europe > Netherlands"]
deg +0.86 (0.10) ["markets", "Patents & R&D &
    Innovation"] -> ["areas", "North America > United
    States"] ==> (0.12) ["markets", "Patents & R&D &
    Innovation"] -> ["areas", "Europe > Netherlands
    "]

Longest running features:
213 |+++++++| ["areas", "Asia > Korea, South"]
125 |+++++++| ["areas", "Asia > China"]
66  |+++++++| ["areas", "Europe"]
62  |+++++++| ["areas", "North America > United States
    "]
58  |+++++++| ["areas", "Europe > Germany"]
56  |+++++++| ["areas", "Europe > Netherlands"]
55  |+++++++| ["areas", "Europe > Norway"]
47  |+++++++| ["areas", "Asia > Japan"]
39  |+++++++| ["areas", "Europe > France"]
37  |+++++++| ["areas", "Asia"]
37  |+++++++| ["areas", "Arctic Region"]
33  |+++++++| ["areas", "Southeast Asia > Singapore"]
32  |+ +++++| ["areas", "Asia > Russia"]
24  |+++++++| ["areas", "Europe > Denmark"]
20  |++ + ++| ["areas", "South America > Brazil"]
19  |+ +++++| ["areas", "Oceania > Australia"]
17  | + ++ +| ["areas", "Europe > Finland"]
16  |++ +++++| ["areas", "North America > Canada"]
15  |++++ ++| ["areas", "Africa"]
14  | + ++++| ["areas", "Europe > Italy"]

637 |+++++++| ["markets", "Patents & R&D & Innovation
    "]

```

```

149 |+++++++| ["markets", "Defence & Security >
      Research Vessel"]
147 |+++++++| ["markets", "Offshore Oil & Gas >
      Research Vessel"]
117 |+++++++| ["markets", "Offshore Oil & Gas"]
  91 |+++++++| ["markets", "Harbour & Terminal"]
  74 |+++++++| ["markets", "Offshore Wind"]
  74 |+++++++| ["markets", "Ship Repair"]
  46 |+++++++| ["markets", "Environmental Safety &
      Control"]
  35 |+++++++| ["markets", "Seagoing Transport > Bulk
      Carrier"]
  34 |+++++++| ["markets", "Seagoing Transport > LNG
      Tanker"]
  32 |+++++++| ["markets", "Environmental Safety &
      Control > Pollution Control Vessel"]
  32 |+ +++++| ["markets", "Defence & Security"]
  30 |+ +++++| ["markets", "Offshore Wind >
      Construction Vessel"]
  28 |+++++++| ["markets", "Offshore Oil & Gas > FPSO"]
  23 | +++++| ["markets", "Fishing"]
  20 |+++++++| ["markets", "Seagoing Transport >
      Container Vessel"]
  19 |+++++++| ["markets", "Harbour & Terminal >
      Harbour Tug"]
  19 | + +++| ["markets", "Fishing > Research Vessel"]
  18 |+++++++| ["markets", "Public Transport"]
  17 |++++++ +| ["markets", "Seagoing Transport >
      Chemical/Products Tanker"]

56 |+++++++| ["orgs", "Hyundai"]
54 |+++++++| ["orgs", "Daewoo"]
32 |+++++++| ["orgs", "Samsung"]
20 |++ + +++| ["orgs", "independent"]
18 | + +++| ["orgs", "Ministry of Trade"]
16 |+ +++ ++| ["orgs", "Offshore Plant"]
16 |+ ++ +++| ["orgs", "D Center"]
15 |+ +++++| ["orgs", "EU"]
12 | + +++| ["orgs", "GE"]
12 | + +++| ["orgs", "Ministry of Oceans"]
11 |+ + ++++| ["orgs", "Government"]
11 |+ ++ + +| ["orgs", "Information Technology"]
11 | ++ +++| ["orgs", "Korea Institute of Ocean
      Science"]
11 | ++ +++| ["orgs", "NYSE"]
11 | + ++| ["orgs", "International Maritime
      Organization"]
10 |++ +++++| ["orgs", "European Commission"]
10 |+ ++ ++| ["orgs", "Equipment Research Institute"]

```

```

10 |+ + ++| ["orgs", "Mitsui"]
9 | +++ +++| ["orgs", "Business Development"]
8 |+ ++++ +| ["orgs", "Ministry of Industry"]

23 | +++ +++| ["people", "Lloyd"]
12 |+++++++| ["people", "Executive Vice"]
9 |+ + +++| ["people", "Hyun"]
6 |+ + +++| ["people", "Senior Executive Vice"]
6 | + ++++| ["people", "Jung"]
6 | + ++ +| ["people", "RINA"]
5 | + ++| ["people", "Henrik O. Madsen"]
5 | ++ | ["people", "President Kim"]
4 |+ ++ + | ["people", "Executive Director"]
4 | + ++ | ["people", "CEO Christopher J"]
4 |+ ++ | ["people", "Bernard Meyer"]
4 | + ++ | ["people", "Marin"]
4 | + | ["people", "Minister Yoon"]
3 | + + +| ["people", "President Ko"]
3 | + ++ | ["people", "E.ON"]
3 | + ++| ["people", "Chief Operating Officer
Subsea"]
3 |++ + | ["people", "President Roh"]
3 | ++ +| ["people", "Cho"]
3 | +++ | ["people", "Jan"]
3 |+ ++ | ["people", "Meyer Werft"]

```

Longest running rules:

```

0.333 |+++++++| ["markets", "Patents & R&D & Innovation
"] -> ["areas", "Asia > Korea, South"]
0.217 |+++++++| ["markets", "Patents & R&D & Innovation
"] -> ["markets", "Defence & Security > Research
Vessel"]
0.214 |+++++++| ["markets", "Patents & R&D & Innovation
"] -> ["markets", "Offshore Oil & Gas > Research
Vessel"]
0.214 | ++ | ["markets", "Patents & R&D & Innovation
"] ["orgs", "Hyundai"] -> ["areas", "Asia > Korea,
South"]
0.201 |+++++ ++| ["markets", "Patents & R&D & Innovation
"] -> ["areas", "Asia > China"]
0.173 |+++++++| ["markets", "Patents & R&D & Innovation
"] -> ["markets", "Offshore Oil & Gas"]
0.125 | + | ["markets", "Patents & R&D & Innovation
"] ["orgs", "Daewoo"] -> ["areas", "Asia > Korea,
South"]
0.124 | + +| ["markets", "Defence & Security >
Research Vessel"] ["markets", "Patents & R&D &
Innovation"] -> ["areas", "Asia > Korea, South"]

```

```
0.121 | ++++++| ["markets", "Patents & R&D & Innovation  
    "] -> ["markets", "Harbour & Terminal"]  
0.106 |+++++ ++| ["markets", "Patents & R&D & Innovation  
    "] -> ["markets", "Ship Repair"]  
  
** errors: {}  
** timers:  
  > rule learning: 0.54 (clock: 0.53)  
  > overall: 3.89 (clock: 3.83)  
  > change detection: 0.1 (clock: 0.1)
```

Glossary

F_1 commonly used measure to determine a test's accuracy; defined as the harmonic mean of precision and recall.

allocation a representation of how the system will relate to non-software structures in its environment (e.g CPUs, file systems, networks, development teams, etc.).

API application programming interface.

application programming interface a specification on how some software components should interact with each other.

ASR Architecturally Significant Requirements.

association rule learning a popular and well researched method for discovering interesting relations between variables in large databases.

B2B Business-to-Business: commerce transactions between businesses, such as between a manufacturer and a wholesaler, or between a wholesaler and a retailer.

B2C Business-to-Customer: commerce transactions transaction that occur between a company and a consumer, as opposed to transactions between companies (B2B).

binding time decision allowable ranges of variation in the implementation of an artefact. This variation can be bound at different times in the software life cycle by different entities – from design or implementation time to runtime by an end-user.

Business Intelligence an umbrella term that refers to a variety of software applications used to analyse an organization's data warehouse to perform enterprise reporting, OLAP, querying, and predictive analytics.

cache miss the operation of accessing a memory location in the cache and it is not found.

CI Competitive Intelligence.

cohesion property which measures how strongly the responsibilities of a module are related; it measures a module's "unity of purpose". High cohesion is desirable as changes in one module would only affect one or similar responsibilities (Bass et al., 2012).

Competitive Intelligence a field which focuses on monitoring the competitive environment with the aim of providing actionable intelligence that will provide a competitive edge to an organization (Safarnia et al., 2011).

component an architectural element with a well defined runtime behaviour; it is a principal unit of computation (e.g. services, peers, clients, servers, filters, etc.).

conditional random field a class of statistical modelling methods often applied in machine learning, where they are used for structured prediction.

connector an architectural element which ensures interaction between components (e.g. call-return, process synchronization operators, pipes).

corpus a large and structured set of texts (used in computational linguistics).

coupling property which measures the overlap of responsibilities in the modules of an architecture; measured by the probability of changes in one module to propagate in another. Loose coupling is desirable for modifiability (Bass et al., 2012).

CRF conditional random field.

decision support system a computer-based information system that supports business or organizational decision-making activities; they serve the management, operations, and planning levels of an organization (usually mid and higher management) and help to make decisions, which may be rapidly changing and not easily specified in advance.

decision tree a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

Document Object Model a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents.

DOM Document Object Model.

Entity Relation Detection a subtask of information extraction that seeks to identify the relations between entities identified using Named Entity Recognition (also called *Relationship Extraction*).

ERD Entity Relation Detection.

ETL Extract, Transform, Load.

feature a type of tag with which a post is classified.

flowchart a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows.

functional requirement property which states a particular function or qualification of a system (Bass et al., 2012).

grounded theory a systematic methodology in the social sciences involving the discovery of theory through the analysis of data. Rather than beginning with a hypothesis, the first step is data collection, through a variety of methods. From the data collected, the key points are marked with a series of codes, which are extracted from the text. The codes are grouped into similar concepts in order to make the data more workable.

hash a function used to map data of arbitrary size to data of fixed size, with slight differences in input data producing very big differences in output data.

inverse document frequency weight for a term that depends on the percentage of documents from a document set in which that term occurs.

Key Intelligence Topic type of CI insight requirements.

KIT Key Intelligence Topic.

latent Dirichlet allocation a generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar; for example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics.

machine learning a subfield of computer science and artificial intelligence that deals with the construction and study of systems that can learn from data, rather than follow only explicitly programmed instructions.

- module** a collection of architectural elements with a well defined functional responsibility.
- Named Entity Recognition** a subtask of information extraction that seeks to locate and classify elements in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.
- natural language processing** a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages.
- NER** Named Entity Recognition.
- neural network** a computational model inspired by the central nervous systems (in particular the brain) which is capable of machine learning.
- NLP** natural language processing.
- ontology** formal representation of knowledge as a hierarchy of concepts within a domain, using a shared vocabulary to denote the types, properties and interrelationships of those concepts.
- open coding** first phase of grounded theory, where everything is coded down, and concepts are constantly merged, renamed and modified for the purpose of building a list of core concepts.
- POS** part-of-speech.
- post** abstract article or unitary snippet of text together with all its metadata (e.g. date, author, source, user comments).
- post item** representation of a post in a structured format (i.e. JSON), containing its text, metadata and possibly other annotations (for the sake of simplicity we use post to represent a post item where there is no risk of confusion).
- quality attribute** measurable or testable property of a system that is used to indicate how well the system satisfies the needs of its stakeholders; effectively a qualification of a functional requirement or sometimes called a *non-functional requirement* (Bass et al., 2012).
- regional benchmarking** interregional comparisons of performance, processes, practices, policies and resources with the aim of improving regional development.
- R&D** Research & Development.

selective coding second phase of grounded theory, where coding is performed more or less around an existing core set of concepts, for the purpose of speeding up the desk research.

self-organising maps a type of neural network that is trained using unsupervised learning to produce a low-dimensional, discretized representation of the input space of the training samples, called a map.

SI Strategic Intelligence.

snapshot the state of a storage system at a particular point in time.

support vector machine supervised learning models with associated learning algorithms that analyse data and recognize patterns, used for classification and regression analysis (machine learning).

SVM support vector machine.

tag a feature-value pair with which a post can be associated.

term frequency weight for a term that depends on the number of occurrences of that term in a document.

term frequency-inverse document frequency a numerical statistic that is intended to reflect how important a word is to a document in a corpus; composed of term frequency and inverse document frequency, it increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others.

tf-idf term frequency-inverse document frequency.

timestamp a sequence of characters or encoded information identifying when a certain event occurred, usually giving date and time of day, sometimes accurate to a small fraction of a second.