Computing and Informatics, Vol. 25, 2006, 127–152

A LOGICAL FRAMEWORK FOR IDENTIFYING AND EXPLAINING UNEXPECTED NEWS

Emma Byrne

Department of Computer Science, University of Wales
Aberystwyth, Ceredigion, SY23 3DB, Wales, UK
e-mail: elb@aber.ac.uk

Revised manuscript received 3 February 2006

Abstract. The number of news reports published online is too large for any person to read all of them. Not all of these reports are equally interesting. Automating the identification and evaluation of interest in news is therefore a valuable goal. This paper presents a framework that permits the identification of interesting news by means of violated expectations. Facts derived from news reports, expectations and related background knowledge can be used to (i) justify the decision to rate news as interesting, (ii) explain why the information in the report is unexpected and, (iii) explain the context in which the report appears. Explanations supported by this framework are general purpose explanations based on the data in the system. The explanations are natural language renditions of first order logic facts and rules.

Keywords: Logical inconsistency, interest, news reports, explanations

1 INTRODUCTION

Information is usually considered interesting because it is *relevant* to the reader or because it is *unexpected* by the reader [21, 17, 23]. News reports in particular owe much of their interest to unexpectedness. It is therefore a worthwhile goal to automate the identification of *unexpected* news. The Explanation Violation Analysis (EVA) framework, presented in [4], supports the development of tools that are able to identify unexpected news. The framework consists of a number of sets of information, together with a means of generating useful expectations from sets of news reports, and is presented in full and evaluated in [3]. The sets of information that form part of the framework are data on which explanations can be based.

In this paper I will discuss the types of explanation an EVA system could give, based on these sets of information and explain these sets in some detail. I will also discuss ways in which the explanation-giving abilities of an EVA system could be used and improved.

The explanation giving based on the EVA framework is general. That is to say, the explanations given are not based on an understanding of the needs and knowledge of the user. The explanations are created by the generation of natural language fragments that relate the knowledge in the set of first order logic facts and rules that are used to identify interesting news.

The paper begins with an overview of news recommender systems. This section explains why the EVA framework is novel and particularly well suited to the recommendation of interesting news reports. This section also discusses the need for explanation giving abilities in such systems. There then follows a section that details the explanatory abilities of an EVA system. The EVA framework is then introduced in a section that gives brief definitions and examples of the various elements of the EVA framework. The section also describes how expectations, a key element of the EVA framework, may be generated. These elements are the basis on which it is possible to build explanations in an EVA system. The paper then goes on to explain how interesting news is identified, with recourse to the EVA framework. Finally, there follows a discussion of how the explanation giving abilities of an EVA system could be extended.

2 THE NEED FOR NEWS RECOMMENDER SYSTEMS

Clearly, the EVA framework is not the first attempt to make large volumes of information more manageable to users. Information retrieval, information filtering and collaborative filtering are three popular methods of information management. However, the EVA framework is more than 'just another information management method': there are clear and significant differences between all of these methods and the EVA framework that represent a step change away from traditional information management systems.

The particular challenges posed by news for a recommender system are twofold: Firstly, news is predominantly of interest because it is unexpected rather than because it is relevant to a query. Secondly, the interest value of news is short-lived. This is particularly true for those users such as stock market analysts, whose ability to respond quickly to events is crucial. It follows that any approach that aims to recommend news must be able to identify news that is unexpected and be able to do so in a timely manner. The EVA framework supports a system to accomplish this.

Existing recommender systems fall broadly into two camps: information retrieval and filtering (IR/IF) on the one hand and collaborative filtering (CF) on the other. IR/IF are two similar approaches in which information is determined to be relevant to a query (IR) or a user profile (IF). Belkin and Croft [2] present an overview of

IR and IF systems. In CF methods, users are clustered into groups with similar "tastes" [25]. Users make quality judgments about items, and these judgments are then used as the basis for recommendations. CF systems have been used to recommend, among other things, films [19] and purchasing decisions [20].

The strengths of the IR/IF, collaborative filtering and EVA approaches are summarised in Table 1.

	Unexpectedness	**Relevance**	**Timeliness**
EVA	Yes	No	Yes
IR/IF	No	Yes	Yes
CF	Yes	Yes	No

Table 1. The features of recommender systems based on the EVA framework, information retrieval and filtering technology and collaborative filtering

IR/IF methods do not attempt to identify unexpected information and so are not suited to recommending news reports. Collaborative filtering could be used to solicit judgments regarding the unexpectedness of news, but the lead time between gathering the users' judgments and providing recommendations (known as the "cold start problem", [22]) makes a collaborative filtering approached unsuitable for providing timely recommendations of news reports.

In conclusion, an EVA system is particularly well suited to the recommendation of news reports, where unexpectedness of the report and timeliness of the recommendation are of the essence. An EVA system could also be used to pre- or post-filter news reports that are assessed for relevance by some IR/IF system.

2.1 The Need for Explanation in Recommender Systems

Explanation may mean one of a number of things. It may consist of giving reasons for some decision, or of making clear an idea that was previously not well understood by the user of the system. Explanation may consist of a set of statements or a dialogue in which the user's and the system's information needs are explored in order to generate some common understanding [6].

For the sake of this paper, I will take a relatively open view of what constitutes an explanation. My definition will be as follows: an explanation is any set of statements that gives a possible reason for a hypothesis[1]. Explanations are natural language statements derived from logical formulae that represent facts derived from a news report, a violated expectation and facts in the system's knowledgbase that are relevant to that report.

There are essentially two types of explanation aware systems: those that are explanation providing and those that are explanation using. Providing explanations as a way of justifying decisions is a technique that has been developed in expert

[1] The term hypothesis is used here as a general term that encompasses decisions, events, actions and so on.

systems for example, as far back as the development of MYCIN (as reported in [6]). In MYCIN, explanations are given in order to justify the system's conclusions and requests for further information. Explanation presenting systems seek to present an explanation for a particular hypothesis to the user. For example, explanation is given to justify the choice of cases to retrieve in a case based reasoning system [9]. Explanation can also be given in order to increase the users' knowledge of the process used to make some decision. Explanation of this type is used in the system presented in [8]. An empirical study by Gregor [11] found that such explanations were most likely to be of benefit to users who had an interest in the explanations the system gave, and who were engaged in a problem solving exercise, in collaboration with the system.

Many explanation-*using* systems, on the other hand, use a process of inference to the best explanation in order to make decisions. Explanatory techniques have also been used as a method of reasoning, in case-based recommender systems for example. Explanation oriented retrieval of cases is used as a method of determining the cases that are appropriate as exemplars given their "explanatory utility" [9]. An example of an explanation-using system is given in [16]. This system is a plan recognition system to enable the understanding of utterances by system users, by using abduction to the best explanation to decide what the user's utterance actually means.

There are two important questions that must be addressed when developing an explanatory system. Firstly, what are the information needs of the user in an information-giving system, or the needs of the system, in an explanation-using system? Secondly, how can the best explanation be given with the information at the system's disposal?

There are a number of points of view regarding what constitutes a good explanation. These range from the formal and abstract [15], through to the user centred [5] and informal [16]. Many of these points of view share the common features that an explanation should

1. explain the relevant event,
2. be suited to the user's needs,
3. be based on facts that are readily available to the user and the system and
4. be the most compact, informative explanation possible.

I will use these four characteristics as the yardstick against which the explanatory abilities of an EVA system are judged.

Current applications of explanation include Case Based Reasoning systems in which explanation utility is used as the measure that drives the selection of cases to be presented to the user, and the work by Belanger and Martel [1], which generates explanations for choices in military planning systems.

3 EXPLANATION AND THE EVA FRAMEWORK

The EVA framework supports the development of a system that can recommend news reports because of their unexpectedness. In order to do so, an EVA system must be aware of several sets of knowledge: news reports, background facts, an event model and a set of expectations. These sets are defined more fully below. Whilst this information permits the identification of unexpected news, it can also be exploited to present the user with an explanation of why a given report is considered interesting. Such an explanation serves two purposes: firstly it increases the reader's confidence in the recommendations given by the system and secondly, it presents the reader with context relevant to the decision that they may not have previously known, thus increasing the users' knowledge.

The EVA framework is a "knowledge intensive" approach: Recommendations are made on the basis of deductions from sets of first-order logical formulae that have clear semantics. As such, the EVA framework is ideally placed to explain *why* a news story may be of interest to the user. This is in contrast with black box approaches, such as neural networks, in which no readily accessible rationale is available.

Explanation giving systems lie along a spectrum of approaches from user-driven to data-driven. Data-driven explanations predominantly use a wide range of detailed, domain specific information that is available to the system. Little, if anything, is done to tailor the explanations to the needs of individual users. In contrast, user-driven explanations rely on a detailed model of individual users' needs and knowledge and tailor explanations accordingly. The range and level detail of domain knowledge may be less than in a data-driven system, but the knowledge is more specifically useful to the user.

Explanations may come from a variety of sources. Case Based Reasoning systems harness cases as a way of providing explanations by analogy. From the cases chosen, users can identify the similarities and differences between the current situation and other, similar cases. The EVA system, in contrast, presents natural language explanations based directly on the facts and rules in the knowledge base.

Figure 1 is a prototype of a graphical user interface (GUI) for a "naive" explanation presenting system, based on the knowledgebases in the EVA framework. In such a system, all the knowledge used by the system in determining whether a report is interesting is presented to the user.

The panes that support navigation through the recommended reports are the Report pane and the Latest Reports panes. The Report pane shows either the original, free text news report, or else a version created from the structured text version by a simple natural language generator. The Latest Reports pane shows the list of interesting reports given as a headline, a star rating, a one sentence summary and a short phrase that summarises the expectation that has been violated.

The explanation of the decision to rate a news report as interesting, and the strength of that rating, is provided in the Latest Reports pane. The star rating is reflective of the strength of the expectation that has been violated. The stronger

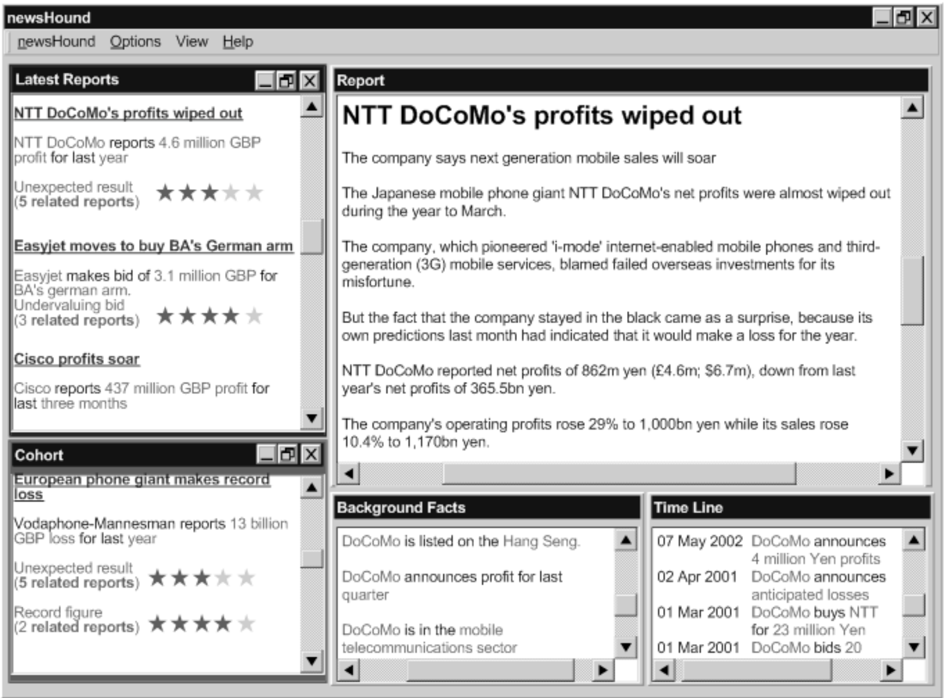

Fig. 1. An example Graphical User Interface for an EVA system

the expectation, the higher the star rating. All the user needs to be aware of is that the star rating is indicative of *how unexpected* the report is assessed as being. In addition, the pane gives a one phrase description of the expectation that has been violated. This description is created by a natural language generator, applied to the expectation violated by the report. For example the first story is considered interesting because it is an "Unexpected result". The description of the violated expectation explains to the reader *why* the news report is unexpected. The way in which expectations are generated and violations of expectations identified is described in the next section.

Contextual explanation is presented to the user by the other panes in the interface. The Background Facts pane contains general contextual information that is relevant to the report. This information contributes to the explanation of why the news story is unexpected. The provenance of this information is discussed in the next section.

An event model is another set of contextual information, also described in the next section. The event model records, and supports reasoning about, sequences of events. The information from the event model is presented in the Time Line pane, which presents a set of events that are relevant to the selected report. As with the set of background facts, the event model provides the reader with the contextual information that was used in order to determine whether the report was unexpected or not: it explains *on what basis* the decision to rate the story as interesting was taken. The Cohort pane presents similar behaviour by related entities in previous representative sets, thus allowing the reader to see *trends* developing. The details of a method by which such cohorts may be identified is given in Chapter 3 of [3] and is not considered further in this paper.

These various elements of explanation help the reader understand why the reported event is considered to be unexpected, which in turn helps them understand how the system has made its recommendation.

The datasets in the EVA framework are held as first-order logic formulae. There are several methods of text generation that could be used to transform data held as first-order logical formulae into natural language text [13]. A tool such as YAG (Yet Another Generator, [24]) for example, is ideal for the handling of such formulae. Text generation is beyond the scope of this paper and will not be considered further. Firtst order logic is ideal for reasoning with sets of facts, but it does have certain shortcomings. Information that is uncertain or ambiguous cannot be easily reasoned with. Nevertheless, it is possible to draw conclusions from the facts and rules in the EVA framework.

How well does a system of the type shown above fulfil the requirements for good explanations that were introduced above? That is, do the explanations given:

1. explain the relevant event,
2. suit the user's needs,
3. rest on facts that are readily available to the user and to the system and
4. give the most compact, informative explanation possible?

The prototype interface given in Figure 1 addresses the first of these requirements. Each decision to rank a news report as being interesting is explained. The system as shown above is able to give both *background explanations* (from the event model and background knowledge) and *procedural explanations* (the expectations that were used to determine that a report is interesting) as described in [11]. Expectations provide some procedural explanation, that is, they inform the user as to why a report has been considered interesting. The contextual knowledge in the background facts, time line and cohort panes explains how the report was determined to be inconsistent with an expectation or expectations.

Regarding the second requirement, there is no real understanding of the users' needs at this stage, so a system as described above has no ability to tailor its explanations in any way. The system shown above may perhaps "over explain" in that all the information used by the system is given to the user. Or indeed the system may "under explain" in some ways, in that the user may require more or different information to that which is given. Further work is required to determine the needs of users in order to allow the system to provide the most appropriate level of explanation.

Requirements three and four are partially met: the explanations are based on information that is readily available to the system, although there is no conception of the users' levels of knowledge available to the system. Likewise, the information given is partly restrained by use of the functions that determine the representative set for a report, and that determine which expectation(s) are violated by that representative set. However, there is no attempt to understand which elements of the explanation have the greatest explanatory power.

User involvement may lead to explanations that are more compact and more suited to the user's level of knowledge. Some form of dialogue, in which the user can respond to the explanations given with requests for more, less or different information may allow the system to develop more suitable explanations. Potential improvements are proposed in the Discussion section at the end of this paper.

We now turn our attention to sources of the data used in these explanations.

4 COMPONENTS OF THE FRAMEWORK

The Expectation Violation Analysis framework supports the identification of unexpected information in news. As such, it includes several datasets that are used to reason about news, events, expectations and contextual information. These sets are fully defined in [3] and summarised here.

The input to an EVA system is a stream of news reports in some structured text format, such as XML. An EVA system then translates each incoming news report into first-order logic; identifies events featured in each report; identifies inconsistencies between each report, the relevant background knowledge, event model and a repository of expectations and finally presents interesting reports to the reader. The sequence of activities is shown in Figure 2. The activities will be the same for a wide variety of domains but the background knowledge and set of expectations need to be tailored for each domain. The process is explained in detail in [4] and summarized below.

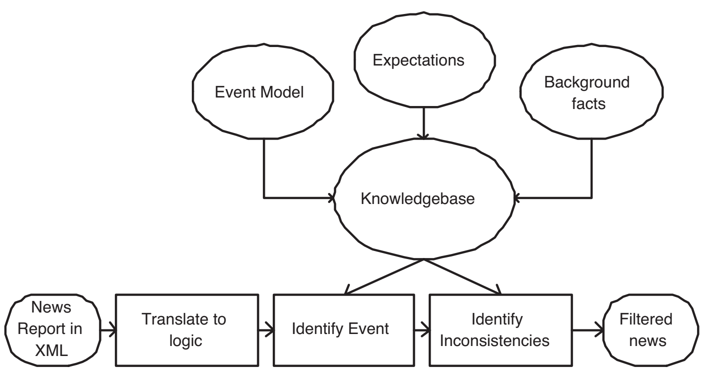

Fig. 2. The EVA process

The framework contains several datasets that will be described below. These datasets are used to identify unexpected news but also form the basis of the explanations shown in Figure 1.

4.1 Structured News Reports

Structured text, such as XML is used to publish semi-structured data. It is possible to apply integrity constraints to structured text data [10]. Such constraints can be used to validate the data input to an EVA system. Because of the structured nature of XML, data can easily be transformed into logic [12]. In general, structured news reports will be in a format such as the following:

Definition 1. Let τ be a tag name (i.e. an element name), ψ be a text entry (i.e. a phrase represented by a string) and $\nu_1, \ldots, \nu_n$ be structured news reports. R is a *structured news report* iff:

$$\begin{array}{l} R = \langle \tau \rangle \psi \langle / \tau \rangle \text{ or} \\ R = \langle \tau \rangle \nu_1, \ldots, \nu_n \langle / \tau \rangle \end{array}$$

Example: A structured news report

$$\begin{array}{l}
\langle report \rangle \\
\quad\quad \langle date \rangle \; 23 \; april \; 2006 \; \langle /date \rangle \\
\quad\quad \langle location \rangle \\
\quad\quad\quad\quad \langle city \rangle \; london \; \langle /city \rangle \\
\quad\quad\quad\quad \langle country \rangle \; uk \; \langle /country \rangle \\
\quad\quad \langle /location \rangle \\
\quad\quad \langle event \rangle \\
\quad\quad\quad\quad \langle action \rangle \; takeoverBid \; \langle /action \rangle \\
\quad\quad\quad\quad \langle buyer \rangle \; megaCorp \; \langle /buyer \rangle \\
\quad\quad\quad\quad \langle target \rangle minnowtech \langle /target \rangle \\
\quad\quad \langle /event \rangle \\
\langle /report \rangle
\end{array}$$

Each structured news report can be considered as a tree, with each tag name being represented by a non-leaf node and each text entry being represented by a leaf node. Furthermore, each subtree in a structured news report is isomorphic to a ground term (that is, a term that contains no variable symbols) where each tag name is represented by a function symbol and each text entry is represented by a constant symbol. Hence, each structured news report can be represented by a ground logical atom in classical logic as follows.

Definition 2. A *report atom* is a ground atom that is the representation of a structured news report in first order logic. Let $\langle \tau \rangle \psi_1, \ldots, \psi_n \langle / \tau \rangle$ be a structured news report, t_1 be a ground term that represents ψ_1 and ... and t_n be a ground term that represents ψ_n. $\tau(t_1, \ldots, t_n)$ is the report atom for $\langle\tau\rangle\psi_1, \ldots, \psi_n\langle/\tau\rangle$.

Example: Consider the example after Definition 1. This report can be represented by the following report atom:

$$\begin{aligned}&report(date(23\ april\ 2006),\\ &\qquad location(city(london), country(uk)),\\ &\qquad event(action(takeoverBid), buyer(megaCorp),\\ &\qquad target(minnowtech)))\end{aligned}$$

Each report atom conserves the structure and content of the original structured news report. However, the report atom is a logical formula and as such can be used in inferences, combined with other logical formulae, tested for consistency and so on.

4.2 News Atoms and Access Rules

We now define news atoms and access rules and give some illustrative examples. Structured news reports are converted into sets of first order ground predicates, *news atoms*, that represent the news story in an EVA system. News atoms are logical formulae extracted from report atoms. News atoms are *ground predicates*, that is, they contain no variables, only predicate, function and constant symbols. Access rules extract ground predicates nested within report atoms. These predicates can then be directly used with other information in the background knowledge.

Definition 3. Let $\rho(t_1, \ldots, t_n)$ be a report atom. A *news atom* for this report is a literal that represents information in the report atom. A news atom is of the form $p(s_1, \ldots, s_m)$ where p is a predicate symbol and $s_1, \ldots, s_m$ are subterms of $t_1, \ldots, t_n$.

The news atoms that are obtained for a given report atom are defined by a set of access rules as follows.

Definition 4. Let α, $\beta_1 \ldots \beta_n$ be unconditional formulae, $\bar{x}$ be the variables in α, and let the variables in $\beta_1 \ldots \beta_n$ be a subtuple or equal to $\bar{x}$. A rule that extracts news atoms from a report atom, denoted an *access rule*, is a first order formula of the form

$$\forall \bar{x}\ \alpha \rightarrow \beta_1 \wedge \cdots \wedge \beta_n.$$

The set of news atoms from a report is extracted using the set of access rules. In order to define the set of news atoms derived this way we need to consider a grounding function. Recall that ground terms are those in which there are no variable symbols. Let Φ be a set of pairs of variable and constant symbols, such as $\{x/a, y/b\}$ where x and y are variable symbols and a and b are constant symbols. Let α be a universally quantified, unground formula. For example, let $\alpha = \forall x, y\ p(x) \wedge q(y)$.

$\mathsf{ground}(\alpha, \Phi)$ is simply the result of substituting the variables in α with the paired constants from Φ. For example, $\mathsf{ground}(\alpha, \Phi) = p(a) \wedge q(b)$.

Definition 5. Let Γ be a set of access rules and let ρ be a report atom. $\mathsf{access}(\rho, \Gamma)$ is the smallest set of news atoms obtained by exhaustively applying the report atom ρ for a structured news report to the access rules in Γ as defined below.

$$\begin{aligned}\mathsf{access}(\rho, \Gamma) = \{&\mathsf{ground}(\beta_1, \Phi), \ldots, \mathsf{ground}(\beta_n, \Phi) \mid \\ &\forall x_1, \ldots, x_k\ \alpha \rightarrow \beta_1 \wedge \cdots \wedge \beta_n \in \Gamma \text{ and} \\ &\mathsf{ground}(\alpha, \Phi) \text{ is } \rho\}\end{aligned}$$

Example: Let ρ be the following report:

$$\begin{aligned}&report((date(070905), company(M\&S)\\ &\qquad result(loss)).\end{aligned}$$

Suppose the set of access rules, Γ, contains the following:

$$\begin{aligned}&\forall x, y, z\ \ report((date(x), company(y), result(z)) \rightarrow\\ &\quad date(x) \wedge company(y) \wedge result(z).\end{aligned}$$

Γ applied to ρ results in:

$$\begin{aligned}&date(070905), company(M\&S), result(loss)\\ &\qquad \in \mathsf{access}(\rho, \Gamma).\end{aligned}$$

Suppose Γ also contains the following access rule:

$$\begin{aligned}&\forall x, y\ \ report(date(x), company(y), result(loss))\\ &\qquad \rightarrow profitWarning(x, y).\end{aligned}$$

As a result,

$$profitWarning(070905, M\&S) \in \mathsf{Access}(\rho, \Gamma).$$

Text entries in structured news reports are usually heterogeneous in format. For example, the format of date values is unconstrained (12/12/1974; 31st Dec 96; 12 Nov 2001 etc.) as is the format of numbers and currency values (3 million; 3 000 000 GBP; \$ 4, ¥500 K etc.). To simplify matters, it is assumed that a preprocessor will convert these text entries into a standard format, that is, the input to an EVA system is homogenous. It is also assumed that the application of access rules to news reports always results in a consistent set of news atoms, that is, for any ρ and Γ, there are no contradictions in the set $\mathsf{Access}(\rho, \Gamma)$.

4.3 Background Knowledge

Background knowledge consists of rules and facts that can be considered alongside information from news reports in order to provide context. Background knowledge includes domain facts, domain rules and may include an event model. It is assumed that the set of facts derived from domain knowledge is internally consistent, and that it also consistent with the set of news atoms for any report ρ and set of access rules, Γ.

The set of *background facts* is a set of first order formulae that are derived from databases, previously received news reports or from domain experts. The facts are stored as ground predicates or obtained by the application of *domain rules*. These rules allow the derivation of further facts. For example, such a rule might be $\forall x\ ftse100(x) \rightarrow plc(x)$, which in plain English states that any company listed in the FTSE100 must also be a public limited company (plc). The set of background facts is the closure of the set of facts derived from background knowledge, the set of facts derived from news atoms and the domain rules.

For many domains, background knowledge is readily available. Facts may be extracted from databases, such as the Kompass database of businesses, the ABI/Inform database of companies and the Economist Intelligence Unit countries database. Domain rules are readily available in the form of regulations, laws, and ontologies. Event models can be automatically constructed from reports previously received by an EVA system.

4.4 Domain Facts

Domain facts may come from a set of domain specific databases. These hold data concerning key entities. In the mergers and acquisitions domain these would include companies, subsidiaries, key personnel, turnover, business activities and so on.

Definition 6. The *domain facts* are a set of ground literals (i.e. atoms and negated atoms).

Example: For the mergers and acquisitions domain, domain facts may include the following:

$$memberOf(United\,Kingdom,\ EU)$$
$$sector(Pirelli, tyreAndCable)$$
$$\neg sector(Pirelli, food).$$

In many domains such as mergers and acquisitions news, there is much information available in existing relational databases that can be used as domain facts. Domain facts may include negative literals. These may be listed explicitly as negative literals or they may have been obtained by the closed world assumption [18]. There may be restrictions on which types of facts may be subjected to the closed world assumption. The implementation details are not considered further here.

4.5 Domain Rules

Domain rules may come from several sources including machine learning and domain experts. Domain rules differ from expectations in that they are "hard rules" that are considered to be inviolable. Therefore it is possible to make inferences from domain rules, domain facts and news atoms in order to generate new facts.

Definition 7. Let α and β be unconditional formulae, $\bar{x}$ be the variables in α, and let the variables in β be a subtuple or equal to $\bar{x}$. *A domain rule* is a formula of the form

$$\forall \bar{x}\ \alpha \rightarrow \beta.$$

Example: The domain rule that all companies in the FTSE 100 share index are public limited companies (plcs) could be expressed as follows:

$$\forall x\ ftse100(x) \rightarrow plc(x).$$

The domain rule that companies do not launch takeover bids for companies they already own could be expressed as:

$$\forall x, y\ takeover(x, y) \rightarrow \neg owns(x, y).$$

Domain rules are also ideal for representing ontologies. Special predicate symbols define relations between entities, such as $partOf$, $typeOf$. In several domains there are already standard ontologies such as the Standard Industry Classification (SIC) codes, used to classify sectors of industrial activity.

Example: The following are domain rules that can be derived from the SIC codes:

$$\begin{array}{l} \forall x\ softwareCompany(x) \rightarrow \\ \qquad businessServicesCompany(x) \\ \forall x\ businessServicesCompany(x) \rightarrow \\ \qquad serviceCompany(x) \end{array}$$

Example: The following are domain rules that could apply to the mergers and acquisitions domain:

$$\begin{array}{l} 1)\ \forall x\ ftse100(x) \rightarrow listed(x) \\ 2)\ \forall x, y\ buyer(x) \wedge target(y) \rightarrow \neg owns(x, y) \end{array}$$

Rule 1 states that if company x is one of the FTSE100 (Financial Times Stock Exchange top 100) companies then company x must be a "listed" company, that is, listed on the stock exchange. Rule 2 states that if company x wishes to buy company y then company x does not currently own company y.

4.6 Expectations

Expectations are "soft rules": they may be violated at least some of the time. This is in contrast to the domain rules that are assumed to be inviolable. Expectations are not used as the basis for inferences to new sets of facts. While they could be used to make uncertain inferences, with a confidence level that is based on the accuracy of the expectation in question, this is not part of the EVA framework. The identification of unexpected news is an act of reasoning under uncertainty, and to add another layer of uncertainty is not considered desirable.

Expectations are unground, first order rules, quantified universally outermost. For example, the expectation $\forall x, y\ takeover(x, y) \to larger(x, y)$ states that if company x takes over company y we expect company x to be larger than company y.

Expectations represent the expected state of the world. Where a representative set violates an expectation this suggests that the reported event is unexpected and therefore interesting. [3] presents a method by which expectations can be derived from the stream of news reports. Some expectations are *stronger*, that is, they are a more accurate representation of the world, than others. The more strongly held an expectation, the more interesting it is when that expectation is violated. A news report violates an expectation if any subset of the facts in that report imply a ground version of the antecedent and the negation of the consequent of an expectation. Further details and concrete examples are given below.

Expectations are formulae that capture general patterns concerning news reports and background knowledge. The set of expectations is at the heart of an EVA framework as it is central to the identification of unexpected information. Unlike hard rules, captured in the domain rules, it is assumed expectations will be *violated* some of the time. An expectation is a defeasible rule: violations of expectations allow the system to identify information which is unusual but not necessarily incorrect.

Definition 8. Let $\alpha_1, \ldots, \alpha_n, \beta$ be unground literals and $\bar{x}$ be free variables occurring in the literals. An *expectation* is a formula of the following form

$$\forall \bar{x}\ \alpha_1 \wedge \cdots \wedge \alpha_n \to \beta$$

where for all Φ, if $\mathsf{ground}(\alpha_1 \wedge \cdots \wedge \alpha_n, \Phi)$ is a ground formula, then $\mathsf{ground}(\beta, \Phi)$ is a ground formula.

Example: The expectation "A company is expected to have sufficient available capital to be able to cover a bid" can be represented as follows:

$$\begin{aligned} \forall c \in companies, v \in bidValue, m \in monetaryValue \\ bidTendered(c, v) \wedge availableCapital(c, m) \to \\ greaterThan(m, v). \end{aligned}$$

Example: The expectation "A company is expected to bid for a target in a sector which supplies or is supplied by that company's sector or that is compatible with the company's sector" can be represented as follows:

$$\begin{aligned}&\forall x, y \in \mathit{companies}\ \mathit{bidFor}(x, y) \rightarrow \\ &\quad (\mathit{sector}(x) = \mathit{sector}(y) \vee \\ &\quad \mathit{supplier}(\mathit{sector}(x), \mathit{sector}(y)) \vee \\ &\quad \mathit{supplier}(\mathit{sector}(y), \mathit{sector}(x)) \vee \\ &\quad \mathit{compatible}(\mathit{sector}(y), \mathit{sector}(x))).\end{aligned}$$

Explanations have strength depending on how well they represent the real world, as seen through the news reports and background knowledge available to the system. The accuracy and coverage of an expectation are values that are determined by the number of reports that, together with a relevant subset of the background knowledge, fire, attack and violate that expectation. The facts from a report and the relevant subset of background knowledge are called a representative set. For a representative set to fire an expectation it must imply a ground version of that expectation's antecedent. For a representative set to attack and expectation it must imply a ground version of the *negation* of that expectations consequent. For an representative set to violate an expectation it must imply both the antecedent and the negation of the consequent of that expectation, as grounded by the same grounding.

Formal definitions for firing, attacking a violating an expectation are given in a later section (Finding unexpected reports). However, in order to understand how expectations are generated it is necessary to know that each expectation has an accuracy value. Informally put, that accuracy value is:

$$1 - \frac{\text{The number of times an expectation has been violated}}{\text{The number of times the expectation has been fired}}.$$

4.7 Generating Expectations

Good expectations are those that are rarely violated by news reports. The more accurately an expectation reflects the behaviour of entities in the real world, the more interesting the news is that violates it.

The set of all possible expectations for a language of predicate, constant and variable symbols grows exponentially with the size of that language. The set of working expectations is defined as the subset of the set of all possible expectations that will be used to identify interesting news. In order to generate the set of working expectations, it is necessary to know what the qualities are that make a "good" expectation, and how such expectations are distributed though the set of expectations.

The set of working expectations is a subset of the set of expectations. The set of working expectations must have the following properties:

1. set of working expectations is small enough to search on receipt of each report,
2. each member of the set of working expectations has a fired value that is greater than a given threshold and

3. each member of the set of working expectations has an accuracy value that is greater than a given threshold.

There are two ways a set of working expectations can be obtained: firstly, by having a set of expectations that is generated by a domain expert; secondly, by using some learning technique to generate those expectations. The first of these methods has several drawbacks: the process is time consuming, relies on the subjective judgment of one or more individuals and does not allow automatic updating.

The method of generating working expectations for the EVA framework uses confirmation theory to determine which are the fittest expectations in terms of their accuracy and coverage. However, analysing the entire space of expectations is unfeasible for all but the most compact of languages. Therefore some method of "navigating" the expectation space is required. Chapter 5 of [3] demonstrates that the specificity relation between the antecedents of two expectations determines their relative coverage values. Likewise, the ordering of the antecedents and consequents of two expectations may determine the relative accuracy of those expectations.

The ordering over expectations can then be used as a way of limiting the search for fit expectations: if we know that expectation ϵ has too low a coverage value to be included in the set of working expectations then any expectation ϵ' whose antecedent is more specific than that of ϵ need not be considered for inclusion in the set of working expectations. This use of a specificity relation as a way of describing a search space has also been applied to the problem of searching for good templates for information extraction and is presented in [7].

4.8 Event Models

Reports do not exist in isolation. There is an underlying narrative which concerns a number of entities that are related in some way over a period of time. In most domains, reports form narratives such that each report tells part of an ongoing story. In the mergers and acquisitions domain for example, a narrative may begin with rumours of an impending bid, continue with news of a bid being made, then go on through the negotiations until the bid is finally agreed on or rejected.

All reports are part of at least one narrative and all narratives contain at least one report. Some approach that captures and reasons with these narratives is a necessary part of the background knowledge.

The event model is a set of facts and rules based on the event calculus (EC) proposed in [14] and modified to infer missing states in [3]. The EC defines rules and metapredicates that can be used to reason about series of events and their effects on the states of entities. Knowing the state in which an entity is allows us to determine what its expected behaviour should be.

There are certain words and phrases in news reports that indicate events that change the state of entities. In the domain of mergers and acquisitions for example, phrases include "agree", "complete", and "approve". These words or their synonyms are used to indicate when a state changes. Additional information about narratives is usually found in close proximity to these phrases in news reports, such as dates, entity names and the tense of the phrases (for example, "shareholders will approve" indicates a different state to "shareholders have approved").

Whilst some expectations are applicable to all entities in all states, others are only applicable to entities in certain states. For example, companies in the state of bankruptcy are expected not to launch takeover bids. In order to apply this expectation it is necessary to know whether an entity is in the state of being bankrupt. This requires an event model, or some similar way of representing the order of events and states in first order logic.

An event model, based on the event calculus, is used to reason over narratives. The event model is modular and as such can be replaced by any other approach that results in a model that can be interrogated by event queries. The event model will not be presented in detail here, but some examples are given below, to give a sense of how it is used.

From an event model it must be possible to derive facts that record which states hold at which timepoints. The example below demonstrates how this could be achieved using the event calculus. It is sufficient at this stage to assume that states can be derived from news reports. Full details are given in Chapter 7 of [3]. Informally, states are descriptions of the position in which an entity may be: bankrupt, subject to a takeover bid, profitable and so on. Entities are the elements of the domain that can perform some action or be in some state.

Definition 9. Let a *state* be a ground predicate that denotes the state of one or more entities and a timepoint be a constant that represents a time point of any granularity, such as a date or a time. *holdsAt* is a meta level predicate that relates states to timeperiods. For some state, s, and some timepoint of any granularity, t, $holdsAt(s, t)$ means that state s holds at timepoint t.

Example: Let $t = 15/12/03$ and let $s = takingOver(morrisons, safeway)$. The ground predicate $holdsAt(takingOver(morrisons, safeway), 15/12/03)$ indicates that on the 15th December 2003, Morrisons was in the process of mounting a takeover bid for Safeway.

The *holdsAt* predicates can be incorporated in expectations to restrict that expectation's applicability to only those entities in a given state as in the following example:

Example: The following expectation states that a company that has launched a takeover bid is expected to be profitable:

$$\begin{aligned}&\forall c_1, c_2 \in companies,\ t \in times\\ &\qquad holdsAt(biddingFor(c_1, c_2), t) \rightarrow\\ &\qquad\qquad holdsAt(profitable(c_1), t).\end{aligned}$$

The expectation given in the above example demonstrates clearly the way in which the event calculus meta-predicates can be used in expectations.

5 FINDING UNEXPECTED REPORTS

When assessing the potential violation of an expectation by a report it is necessary to ensure that only the subset of the background knowledge that is somehow *relevant* to that report is considered. Let Δ be the set of background knowledge. The function $\mathsf{match}(\rho, \Gamma, \Delta)$ returns the subset of Δ whose members are all the facts in Δ that concern the entities in ρ. Informally, entities are things with a distinct, not necessarily concrete, existence. Marks and Spencer plc, the FTSE 100 share index and Britain are examples of entities. Distinct from entities are things such as attributes, such as monetary values for example, and actions or events, such as the making of a bid.

Definition 10. Let ρ be a report, let Γ be a set of access rules and let Δ be a set of background knowledge. The set $\mathsf{match}(\rho, \Gamma, \Delta)$ is a set of literals that can be derived from Δ and ρ such that there are one or more constants in each literal that appear as constants in one or more literals in $\mathsf{access}(\rho, \Gamma)$:

$$\begin{aligned}&\mathsf{match}(\rho, \Gamma, \Delta) = \\ &\qquad \{\alpha(p_1, \ldots, p_j) | \mathsf{access}(\rho, \Gamma) \cup \Delta \vdash \alpha(p_1, \ldots, p_j) \\ &\qquad \text{ and there exists } \beta(q_1, \ldots, q_k) \in \mathsf{access}(\rho, \Gamma) \\ &\qquad \text{such that there exists } p_h \in p_1, \ldots, p_j \text{ and} \\ &\qquad q_i \in q_1, \ldots, q_k \text{ where } p_h = q_i\}.\end{aligned}$$

Example: Let ρ be a report and let Γ be a set of access rules. Let $\mathsf{access}(\rho, \Gamma) = \{profitable(ba)\}$. Let Δ be a set of background knowledge. Let

$$\begin{aligned}\Delta = \{&airline(ba), airline(ryanair), \\ &british(ba), british(marksandspencer), \\ &irish(ryanair), retailer(marksandspencer), \\ &\forall x\ profitable(x) \rightarrow \neg bankrupt(x)\}.\end{aligned}$$

$\mathsf{match}(\rho, \Gamma, \Delta)$ contains all literals in Δ which share a constant symbol with the literals in $\mathsf{access}(\rho, \Gamma)$ and the facts derivable from the domain rules in Δ and the literals in $\mathsf{match}(\rho, \Gamma, \Delta)$. Therefore

$$\mathsf{match}(\rho, \Gamma, \Delta) = \{airline(ba), british(ba), \neg bankrupt(ba)\}.$$

The assumption is made that "spurious" matches (for example, matches based on date values or monetary amounts, where these are not of interest) can be avoided by such means as ascribing types to the constant symbols. The implementation details are not considered further in this paper.

It is now possible to define a set that contains all the facts in a report ρ and the set of background knowledge that is relevant to ρ. A *representative set* is set of ground facts derived from the news atoms and relevant facts from the event model and background facts. The representative set for a report is a representation of that report and its context. The set $\mathsf{representatives}(\rho, \Gamma, \Delta)$ is the union of the literals extracted from a report and the subset of relevant literals from Δ. $\mathsf{representatives}(\rho, \Gamma, \Delta)$ is a first order representation of the situation that is described in the report.

Definition 11. Let ρ be a report, Γ be a set of access rules and Δ be a set of background knowledge. The set of all facts relevant to ρ, the *representative set* is a set such that:

$$\mathsf{representatives}(\rho, \Gamma, \Delta) = \mathsf{access}(\rho, \Gamma) \cup \mathsf{match}(\rho, \Gamma, \Delta).$$

We assume that $\mathsf{representatives}(\rho, \Gamma, \Delta)$ is a consistent set.

Example: Let ρ be a report and let Γ be a set of access rules. Let

$$\begin{aligned}\mathsf{access}(\rho, \Gamma) = \{&airline(ryanair),\\ &takeover(ryanair, buzz)\}\end{aligned}$$

and let

$$\begin{aligned}\mathsf{match}(\rho, \Gamma, \Delta) = \{&irish(ryanair),\\ &profitable(ryanair),\\ &\neg profitable(buzz)\}.\end{aligned}$$

The set

$$\begin{aligned}\mathsf{representatives}(\rho, \Gamma, \Delta) = \{&airline(ryanair),\\ &takeover(ryanair, buzz), irish(ryanair),\\ &profitable(ryanair), \neg profitable(buzz)\}.\end{aligned}$$

One of the requirements of a good explanation is that the information given must be the most compact, informative explanation possible. The representative set, by means of the $\mathsf{match}(\rho, \Gamma, \Delta)$, restricts the set of background knowledge that is considered to only that subset of Δ that is relevant to the report ρ. This results in a limit on the information that is given to the user as an explanation.

It is possible to restrict this set still further to a minimally inconsistent subset. In this way, the user is only presented with information that is "implicated" in the decision to rate the news story as interesting. To illustrate with an example, consider a representative set that includes the following facts about high street store, Marks and Spencer (M & S): $headOffice(london, M\,\&\,S)$, $chairman(terryBurns, M\,\&\,S)$, $ftse100(M\,\&\,S)$. Now consider the news atom $\neg profitable(M\,\&\,S)$ and the expectation $\forall x\ ftse100(x) \rightarrow profitable(x)$.

The facts $headOffice(london, M\,\&\,S)$ and $chairman(terryBurns, M\,\&\,S)$ are not relevant to the conclusion that Marks and Spencer's lack of profitability is unexpected. However, the fact that Marks and Spencer is listed in the FTSE100 share index is related to the unexpectedness of the report. Therefore the fact $ftse100(M\&S)$ would be presented to the user as part of an explanation.

It is now possible to verify expectations against representative sets. Assume two functions, $\mathsf{antecedent}(\epsilon)$ and $\mathsf{consequent}(\epsilon)$ such that for any expectation ϵ, they return the antecedent and the consequent of that expectation respectively.

If $\mathsf{representatives}(\rho, \Gamma, \Delta)$ implies a ground version of $\mathsf{antecedent}(\epsilon)$ then we say that ρ *fires* that expectation.

Definition 12. Let ϵ be an expectation, Γ be a set of access rules, Δ be a set of background knowledge and ρ be a report. Let Φ be a grounding set. ϵ is *fired* by ρ iff there exists a grounding set Φ such that $\mathsf{ground}(\epsilon, \Phi) = \epsilon_\Phi$ and

$$\mathsf{representatives}(\rho, \Gamma, \Delta) \vdash \mathsf{antecedent}(\epsilon_\Phi).$$

Note that neither the consequent of ϵ nor its negation need be implied by $\mathsf{representatives}(\rho, \Gamma, \Delta)$ in order for ϵ to be fired.

If $\mathsf{representatives}(\rho, \Gamma, \Delta)$ implies a ground version of $\neg\mathsf{consequent}(\epsilon)$ then we say that ρ *attacks* that expectation.

Each firing of an expectation also has an effect on that expectations strength. The set of reports that leads to the firing of an expectation is denoted $\mathsf{fset}(\epsilon, \Pi, \Gamma, \Delta)$. The cardinality of that set is informative of the number of times that expectation has been fired given a set of reports, access rule and background knowledge. The set itself is a record of all of those representative sets that have implied a ground version of the antecedent of that expectation:

Definition 13. Let Γ be a set of access rules, Δ be a set of background knowledge, ϵ be an expectation and Π be a set of reports. The subset of Π whose members fire ϵ, denoted $\mathsf{fset}(\epsilon, \Pi, \Gamma, \Delta)$ is as follows:

$$\begin{aligned}\mathsf{fset}(\epsilon, \Pi, \Gamma, \Delta) = \{\rho \in \Pi \mid & \text{ there exists a grounding set } \Phi \\ & \text{such that } \mathsf{ground}(\epsilon, \Phi) = \epsilon_\Phi \text{ and} \\ & \mathsf{representatives}(\rho, \Gamma, \Delta) \vdash \mathsf{antecedent}(\epsilon_\Phi)\}.\end{aligned}$$

Definition 14. Let ϵ be an expectation, Γ be a set of access rules, Δ be a set of background knowledge, ρ be a report and Φ be a grounding set. $\mathsf{ground}(\epsilon, \Phi)$ is *attacked* by a report ρ iff there exists a grounding set Φ such that $\mathsf{ground}(\epsilon, \Phi) = \epsilon_\Phi$ and

$$\mathsf{representatives}(\rho, \Gamma, \Delta) \vdash \neg\mathsf{consequent}(\epsilon_\Phi).$$

Note that the antecedent of ϵ need not be fired in order for $\mathsf{consequent}(\epsilon)$ to be attacked. Also, $\mathsf{representatives}(\rho, \Gamma, \Delta)$ is consistent but if ρ attacks ϵ then the union of $\mathsf{representatives}(\rho, \Gamma, \Delta)$ and $\{\mathsf{consequent}(\epsilon_\Phi)\}$ will be inconsistent. The inconsistency arises from the inclusion of the ground consequent, $\mathsf{consequent}(\epsilon_\Phi)$.

As with the firing of an expectation, we wish to keep track of the number of times a report has been attacked, and the representative sets that have caused it to be so. The set of reports that attack an expectation is that expectation's aset. The attacked value for an expectation is equal to the size of the aset:

Definition 15. Let Γ be a set of access rules, Δ be a set of background knowledge, ϵ be an expectation and Π be a set of reports. The subset of Π whose members attack ϵ, denoted $\mathsf{aset}(\epsilon, \Pi, \Gamma, \Delta)$ is as follows:

$$\begin{aligned}\mathsf{aset}(\epsilon, \Pi, \Gamma, \Delta) = \{\rho \in \Pi \mid\ & \text{there exists a grounding set } \Phi \\ & \text{such that } \mathsf{ground}(\epsilon, \Phi) = \epsilon_\Phi \text{ and} \\ & \mathsf{representatives}(\rho, \Gamma, \Delta) \vdash \neg\mathsf{consequent}(\epsilon_\Phi)\}.\end{aligned}$$

An EVA system identifies as interesting information that which violates an expectation. In order for an expectation to be violated, it must be both fired and attacked by the same representative set.

Definition 16. Let ϵ be an expectation, let Γ be a set of access rules, Δ be a set of background knowledge and ρ be a report. Let Φ be a grounding set. ϵ is *violated* by a report ρ iff there is a grounding set Φ such that $\mathsf{ground}(\epsilon, \Phi) = \epsilon_\Phi$ and

$$\begin{aligned}&\mathsf{representatives}(\rho, \Gamma, \Delta) \vdash \\ &\qquad\qquad \mathsf{antecedent}(\epsilon_\Phi) \wedge \neg\mathsf{consequent}(\epsilon_\Phi).\end{aligned}$$

That is, ρ both fires and attacks a given grounded expectation.

Example: Let ϵ be an expectation $\forall x\ airline(x) \vee bank(x) \rightarrow profitable(x)$, Γ be a set of access rules, Δ be a set of background knowledge and ρ be a report. Let Φ be a grounding set,$\{x = buzz\}$. Let

$$\begin{aligned}\mathsf{ground}(\epsilon, \Phi) =& airline(buzz) \vee bank(buzz) \rightarrow \\ & profitable(buzz)\end{aligned}$$

$$\begin{aligned}\mathsf{access}(\rho, \Gamma) &= \{\neg profitable(buzz)\} \\ \mathsf{match}(\rho, \Gamma, \Delta) &= \{airline(buzz)\}\end{aligned}$$

$$\begin{aligned}\mathsf{representatives}(\rho, \Gamma, \Delta) =& \{\neg profitable(buzz)\} \cup \\ & \{airline(buzz)\}\end{aligned}$$

$$\begin{aligned}&\mathsf{ground}(\mathsf{antecedent}(\epsilon) = airline(buzz) \vee bank(buzz)) \\ &\mathsf{ground}(\mathsf{consequent}(\epsilon), \Phi) = profitable(buzz).\end{aligned}$$

It then follows that

$$\{\neg profitable(buzz), airline(buzz)\} \vdash \\ airline(buzz) \vee bank(buzz)$$

therefore ρ fires $\mathsf{ground}(\epsilon, \Phi)$. Also,

$$\{\neg profitable(buzz), airline(buzz)\} \vdash \\ \neg profitable(buzz)$$

therefore ρ attacks $\mathsf{ground}(\epsilon, \Phi)$. Consequently,

$$\{\neg profitable(buzz), airline(buzz)\} \vdash \\ airline(buzz) \vee bank(buzz) \wedge \neg profitable(buzz)$$

therefore ρ violates $\mathsf{ground}(\epsilon, \Phi)$.

When an expectation is violated, we know some unexpected, and therefore interesting, event has taken place. An explanation of why news has been rated as interesting can be given by presenting the user with a natural language version of the expectation that was violated.

The accuracy of an expectation is based, in part, on the number of times it has been violated. The set of reports that violate an expectation is that expectation's vset. The cardinality of that vset is a record of the number of times the expectation has been violated, and the members of that vset are a record of the representative sets that have lead to the expectation being violated.

Definition 17. Let Γ be a set of access rules, Δ be a set of background knowledge, ϵ be an expectation and Π be a set of reports. The subset of Π whose members violate ϵ, denoted $\mathsf{vset}(\epsilon, \Pi, \Gamma, \Delta)$ is as follows:

$$\mathsf{vset}(\epsilon, \Pi, \Gamma, \Delta) = \{\rho \in \Pi \mid \text{there exists a grounding set } \Phi \\ \text{such that } \mathsf{ground}(\epsilon, \Phi) = \epsilon_\Phi \text{ and} \\ \mathsf{representatives}(\rho, \Gamma, \Delta) \vdash \mathsf{antecedent}(\epsilon_\Phi) \wedge \\ \neg\mathsf{consequent}(\epsilon_\Phi)\}.$$

The *accuracy* value measures the strength of an expectation with respect to the number of times it has been fired and not attacked. This value has already been informally defined. Let us now consider a formal definition:

Definition 18. Let Γ be a set of access rules, Δ be a set of background knowledge, ϵ be an expectation and Π be a set of reports.

If $|\mathsf{fset}(\epsilon, \Pi, \Gamma, \Delta)| > 0$ then

$$\mathsf{accuracy}(\epsilon, \Pi, \Gamma, \Delta) = 1 - \frac{|\mathsf{vset}(\epsilon, \Pi, \Gamma, \Delta)|}{|\mathsf{fset}(\epsilon, \Pi, \Gamma, \Delta)|}.$$

If $\mathsf{fset}(\epsilon, \Pi, \Gamma, \Delta) = \emptyset$ then $\mathsf{accuracy}(\epsilon, \Pi, \Gamma, \Delta)$ is *undetermined.*

The sets and values introduced in this section give us a number of other sources of evidence on which explanations may be based. The prototype in Figure 1 included explanation of the *degree* to which a news report is unexpected, based on the strength of the expectation that has been violated. However, no explanation is given as to why the expectation has the *strength* that it does. The set of representative sets that have lead to that expectation being violated and fired, the vset and fset, are available to an EVA system. These sets of representative sets are, in essence, the evidence on which the assessment of the strength of the expectation is made. This evidence, or some summary thereof, could be made available to the users if they wished to know why a given expectation has a certain strength value.

We have examined the sets of facts and rules that exist in the EVA framework. We have considered how these may be used in order to generate explanations. In the next section, let us consider how those explanations could be developed.

6 DISCUSSION

Explanations in the EVA framework are based on facts from a knowledgebase that includes domain facts and rules and an event model or models. The violated expectation is the reason why a news report may be condidered interesting, but the knowledgebase supplies the basis for contextual explanations. Explanations are natural language statements that are derived from the expectations and knowledgebase, as shown in Figure 1.

The EVA framework, as it stands, provides a foundation for a system that not only identifies unexpected news but that can provide data-driven explanations as to *why* that news is unexpected. However, the expectations given are "needs blind": they are constructed solely on the basis of whatever information an EVA system has at its disposal. Therefore there is scope for future work to improve the degree of sophistication of the explanations given.

First and foremost, work is required to determine what the needs of users are. This may be done *a priori*, by undertaking user studies and a process of requirements gathering. This would result in a standardised pattern of explanation giving for all users. Alternatively, user needs could be understood *in situ*, by some learning mechanism that prompts and analyses user feedback, perhaps by initially providing very basic explanations and allowing the user to request more detailed information. Such feedback would then allow the system to tailor future explanations accordingly.

Such feedback could be gathered directly, via a dialogue with, or by soliciting ratings from users. Alternatively, it could be gathered passively, through the detection of the attention given to certain items, or a measure of the user's click through rate. The direct methods are more intrusive but are less likely to result in misleading data than the passive methods.

An EVA system also has awareness of the user's state of knowledge, via the historical context provided by the set of reports previously shown to the user. Here there is also scope to tailor explanations, perhaps simply by suppressing information that the user has seen recently, or perhaps by some use of analogical reasoning ("This is like the time when…"). More sophisticated selection of previous news events could be achieved in similar manner to that developed in (BC05).

Furthermore, there is scope for the EVA framework to support an explanation *using* system. Currently, an EVA system looks for reports that are inconsistent with expectations in order to identify unexpected news. However, it may be possible to explain away some of this unexpectedness. For example, the news of a large, previously profitable company going bankrupt could be explained away by the hypothesis that the CEO is about to be indicted for embezzlement. This would change the focus of an EVA system from the identification of unexpected news to the generation of predictions and hypotheses based on that unexpected event.

Acknowledgements

This work was in part supported by EPSRC grant GR/R22551/01 Logic-based Technology for Measuring Inconsistency of News Reports

REFERENCES

[1] Belanger, M.—Martel, J.-M.: An Automated Explanation Approach for a Decision Support System Based on Mcda. In International Symposium on Explanation-aware Computing (ExaCt 2005). American Association for Artificial Intelligence, 2005.

[2] Belkin, N. J.—Croft, B. W.: Information Filtering and Retrieval: Two Sides of the Same Coin? Artificial Intelligence, Vol. 35, 1992, No. 12, pp. 29–38.

[3] Byrne, E.: Expectation Violation Analysis of News Reports. Ph. D. thesis, University College London, Department of Computer Science, 2005.

[4] Byrne, E.—Hunter, A.: Man Bites Dog: Looking for Interesting Inconsistencies in Structured News Reports. Data and Knowledge Engineering, Vol. 48, 2004, No. 3, pp. 265–285.

[5] Cawsey, A. J.: User Modelling in Interactive Explanations. Journal of User Modelling and User Adapted Interaction, Vol. 3, 1993, No. 3, pp. 221–247.

[6] Cawsey, A. J.: Developing an Explanation Component for a Knowledge-Based System: Discussion. Expert Systems with Applications, Vol. 8, 1995, No. 4, pp. 527–531.

[7] Corney, D.—Byrne, E.—Buxton, B.—Jones, D.: A Logical Framework for Template Creationand Information Extraction. In ICDM05. IEEE, 2005.

[8] Daniel, B. H.—Callaway, C. B.—Bares, W. H.—Lester, J. C.: Student-Sensitive Multimodal Explanation Generation for 3D Learning Environments. In AAAI'99/IAAI'99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference, pp. 114–120, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[9] Doyle, D.—Cunningham, P.—Bridge, D.—Rahman, Y.: Explanation Oriented Retrieval. In P. Funk and P. Calero, editors, Seventh European Conference on Case-Based Reasoning, pp. 157–168. Springer, 2004.

[10] Fan, W.—Simeon, J.: Integrity Constraints for XML. In Symposium on Principles of Database Systems, pp. 23–34, 2000.

[11] Gregor, S.: Explanations from Knowledge-Based Systems and Co-Operative Problem Solving: An Empirical Study. Int J. Human-Computer Studies, pp. 81–105, 2001.

[12] Hunter, A.: Merging Potentially Inconsistent Items of Structured Text. Data and Knowledge Engineering, Vol. 3, 2000, pp. 305–332.

[13] van Deemter, E. K.—Theune, M.: Real vs. Template-Based Natural Language Generation: A False Opposition? Computational Linguistics, Vol. 31, 2005, No. 1, pp. 15–23.

[14] Kowalski, R. A.—Sergot, M.: A Logic-Based Calculus of Events. In New Generation Computing, Vol. 4, pp. 67–95. Springer-Verlag, 1986.

[15] Lozinskii, E.: Explaining by Evidence. Journal of Experimental and Theoretical Artificial Intelligence, Vol. 12, 2000, pp. 69–89.

[16] Mayfield, J.: Evaluating Plan Recognition Systems: Three Properties of a Good Explanation. Artif. Intell. Rev., Vol. 14, 2000, Nos. 4–5, pp. 351–376.

[17] Ortony, A.—Partridge, D.: Surprisingness and Expectation Failure, What's the Difference? In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, pp. 106–108, Milan, Italy, 1987.

[18] Reiter, R.: On Closed World Databases. In H. Gallaire and J. Minker, editors, Logic and Databases, pp. 55–76. Plenum Press, 1978.

[19] Resnick, P.—Iacovou, N.—Suchak, M.—Bergstorm, P.—Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work, pp. 175–186, Chapel Hill, North Carolina, 1994. ACM.

[20] Schafer, J. B.—Konstan, J. A.—Riedl, J.: E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery, Vol. 5, 2001, Nos. 1–2, pp. 115–153.

[21] Schank, R.: Interestingness: Controlling inferences. Artificial Intelligence, Vol. 12, 1979, No. 3, pp. 273–297.

[22] Schein, A. I.—Popescul, A.—Ungar, L. H.: Methods and Metrics for Cold-Start Recommendations. In Proceedings of the 25th annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002.

[23] Silberschatz, A.—Tuzhilin, A.: What Makes Patterns Interesting in Knowledge Discovery Systems. IEEE Trans. On Knowledge And Data Engineering, Vol. 8, 1996, pp. 970–974.

[24] Susan, S. S. A.—McRoy, W.—Channarukul, S.: An Augmented Template-Based Approach to Text Realization. Natural Language Engineering, Vol. 9, 2003, No. 4, pp. 381–420.

[25] Terveen, L.—Hill, W.: Beyond Recommender Systems: Helping People Help Each Other. In J. Carroll, editor, HCI in the New Millennium, pp. 487–509. Addison Wesley, 2001.

Emma Byrne recently completed her Ph. D. in computer science at University College London. She is currently working on the Robot Scientist project at the University of Wales, Aberystwyth. Her research interests include applications of symbolic reasoning and machine learning.