Semantic Data Mining of Financial News Articles

Anže Vavpetič[1,2], Petra Kralj Novak[1], Miha Grčar[1], Igor Mozetič[1], and Nada Lavrač[1,2,3]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[3] University of Nova Gorica, Nova Gorica, Slovenia
anze.vavpetic@ijs.si

Abstract. Subgroup discovery aims at constructing symbolic rules that describe statistically interesting subsets of instances with a chosen property of interest. Semantic subgroup discovery extends standard subgroup discovery approaches by exploiting ontological concepts in rule construction. Compared to previously developed semantic data mining systems SDM-SEGS and SDM-Aleph, this paper presents a general purpose semantic subgroup discovery system Hedwig that takes as input the training examples encoded in RDF, and constructs relational rules by effective top-down search of ontologies, also encoded as RDF triples. The effectiveness of the system is demonstrated through an application in a financial domain with the goal to analyze financial news in search for interesting vocabulary patterns that reflect credit default swap (CDS) trend reversal for financially troubled countries. The approach is showcased by analyzing over 8 million news articles collected in the period of eighteen months. The paper exemplifies the results by showing rules reflecting interesting news topics characterizing Portugal CDS trend reversal in terms of conjunctions of terms describing concepts at different levels of the concept hierarchy.

Keywords: semantic data mining, subgroup discovery, ontology, credit default swap, financial crisis.

1 Introduction

This paper addresses the task of subgroup discovery, first defined by Klösgen [1] and Wrobel [2]. The goal of SD is to find subgroups of instances that are statistically interesting according to some property of interest for a given population of instances. SD is commonly described as being in the intersection of predictive and descriptive data mining as it is used for descriptive rule learning although the rules are induced from class-labeled data. Patterns discovered by subgroup discovery methods (called subgroup descriptions) are rules of the form `Class` ← `Conditions`, where the condition part of the rule is a logical conjunction of features (items, attribute values) or a conjunction of logical literals that are characteristic for a selected class of instances.

J. Fürnkranz, E. Hüllermeier, and T. Higuchi (Eds.): DS 2013, LNAI 8140, pp. 294–307, 2013.
© Springer-Verlag Berlin Heidelberg 2013

It is well known from the literature on inductive logic programming (ILP) [3, 4] and relational data mining (RDM) [5] that the performance of data mining methods can be significantly improved if additional relations among the data objects are taken into account. In other words, the knowledge discovery process can significantly benefit from the domain (background) knowledge.

A special form of background knowledge, which has not been exploited in the original ILP and RDM literature, are ontologies. Ontologies are consensually developed domain models that formally define the semantic descriptors and can act as means of providing additional information to machine learning (data mining) algorithms by attaching semantic descriptors to the data. Such domain knowledge is usually represented in a standard format which encourages knowledge reuse. Two popular formats are the Web Ontology Language (OWL) for ontologies and the Resource Description Framework (RDF) triplets for other structured data. The RDF data model is simple, yet powerful. A representation of the form *subject-predicate-object* ensures the flexibility of the data structures, and enables the integration of heterogeneous data sources. Data can be directly represented in RDF or (semi-)automatically translated from propositional representations to RDF as graph data. Consequently, more and more data from public relational databases are now being translated into RDF as linked data. In this way, data items from various databases can be easily linked and queried over multiple data repositories through the use of semantic descriptors provided by the supporting ontologies encoding the domain models and knowledge.

The process of exploiting formal ontologies within the process of data mining, called Semantic Data Mining (SDM), was formalized by Vavpetič and Lavrač [6]. Early work in using ontologies in machine learning and data mining is due to Kietz [7] who extended the standard learning bias used in ILP with description logic (DL) in his CLARIN-DL system. More recently, Lehmann and Haase [8] defined a refinement operator in a variant of DL, but considered only the construction of consistent and complete hypotheses. Lawrynowicz and Potoniec [9] introduced an algorithm for frequent concept mining in another variant of DL. Combining web mining and the semantic web was proposed by Berendt et al. [10]. Early work on this topic is due to Lisi et al. [11, 12], proposing an approach to mining the semantic web by using a hybrid language AL-log, used for mining multi-level association rules.

In this paper, we present a new semantic subgroup discovery system named Hedwig, which searches for subgroups with descriptions constructed from the given ontological vocabulary (including any provided binary relations). The traversal of the search space is effectively guided by the hierarchical structure of the ontology. The most relevant related work in exploiting ontologies in real-life data mining tasks is by Trajkovski et al. [13] who used the gene ontology to find enriched gene sets from microarray data, and by akova et al. [14] who used an ontology of Computer Aided Design elements and structures to find frequent design patterns.

In this paper, we present the results of applying the Hedwig system to get insight into a vast amount of news articles collected in last two years as part

of the 7FP EU projects FIRST and FOC. We seek for insight in the financial domain; more specifically we investigate the vocabulary related to the European sovereign debt crisis used in news articles and financial blogs. We investigate the relationship between the financial market perception of a financial entity and the articles mentioning the financial entity. As a measure of market perception, we use the credit default swap (CDS) price. In essence, CDS is insurance for country bonds and reflects the market expectation that the issuer will default. The higher the CDS price, the more likely it is that that country will be unable to repay its debt [15]. Portugal is the focus of our investigation as an example of a financially troubled country.

Gamberger et al. [16] employed SD techniques on a related problem. They have induced indicators of systemic banking crises by looking at past crises in the period 1976-2007. Rather than looking at news articles and relating them to the CDS prices, they used 105 publicly available financial indicators. Their main result is that demographic indicators are the most important: the percentage of the active population in connection to the annual percentage of money growth and the male life expectancy are especially crucial.

The main contributions of this paper are the new semantic data mining system named Hedwig, which is presented with its premiere application in understanding financial news, and the extensive data acquisition pipeline that was used for collecting the data. Another contribution is the first insights into the relationship between the European sovereign debt crisis vocabulary and the CDS price trends.

The paper is structured as follows. Section 2 describes the developed Hedwig semantic SD system. Section 3 describes the data acquisition and cleaning pipeline, while Section 4 describes the data preparation stage, the experimental setup and the results. Section 5 gives directions for further work and concludes the paper.

2 Methodology

This section describes the newly developed semantic subgroup discovery system Hedwig. Compared to standard subgroup discovery algorithms, Hedwig uses domain ontologies to guide the search space and formulate generalized hypothesis. Existing semantic subgroup discovery algorithms are either specialized for a specific domain [13] or adapted from systems that do not take into the account the hierarchical structure of background knowledge [6]. Hedwig overcomes these limitations as it is designed to be a general purpose semantic subgroup discovery system.

Semantic subgroup discovery, as addressed by the Hedwig system, results in relational descriptive rules, using training examples in RDF triples form and using several ontologies as background knowledge used. As an illustration, take three simplified ontologies illustrated in Figure 1, as sample ontologies which could be used in mining financial data.

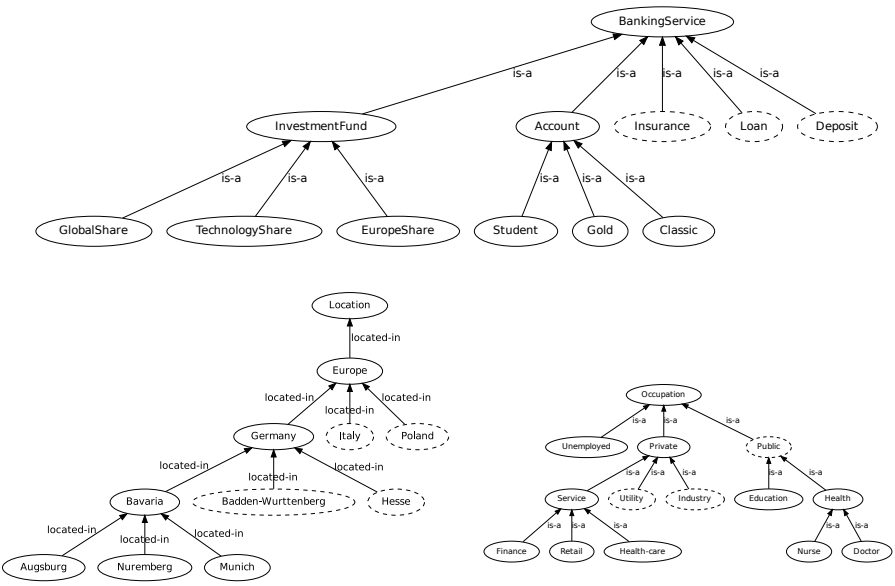

Fig. 1. The ontologies of banking services, locations and occupations. Concepts with omitted sub-concepts are drawn with a dashed line.

Formally, the semantic data mining task addressed in this paper is defined as follows.

Given:

- The empirical data in the form of a set of training examples expressed as RDF triples,
- Domain knowledge in the form of ontologies (one or more), and
- An object-to-ontology mapping which associates each object from the RDF triplets with appropriate ontological concepts.

Find:

- A hypothesis (a predictive model or a set of descriptive patterns), expressed by domain ontology terms, explaining the given empirical data.

Subgroup describing rules are first-order logical expressions. Take the following rule, used to explain the format of induced subgroup describing rules.

```
Max(X) ← Country(X), Before(X,Y), comp_NESTLE_S_A(Y). [50, 10]
```

where variables `X`, `Y` represent sets of input instances. Note the convention that lowercase predicates (e.g., `comp_NESTLE_S_A`) represent specific instances (appearing in the leaves of the ontology), while capitalized predicates represent classes (appearing at higher hierarchy levels of the ontology), i.e., sets of specific instances (e.g., predicate `Country` subsumes instances like `cou_Portugal` or

```
function induce():
    rules = [default_rule]
    while improvement(rules):
        foreach rule in rules:
            rules.extend(specialize(rule))
        rules = best(rules, N)
    return rules

function specialize(rule):
    specializations = []
    foreach predicate in eligible(rule.predicates):
        # Specialize by traversing the subClassOf hierarchy
        for subclass in subclasses(predicate):
            new_rule = rule.swap(predicate, subclass)
            if can_specialize(new_rule):
                specializations = specializations.add(new_rule)
    if rule != default_rule:
        # Specialize by adding a new unary predicate to the rule
        new_predicate = next_non_ancestor(eligible(rule.predicates))
        new_rule = rule.append(new_predicate)
        if can_specialize(new_rule):
            specializations.add(new_rule)
    if rule.predicates.last().arity == 1:
        # Specialize by adding new binary predicates
        specializations.extend(add_binary_predicate(rule))
    return specializations
```

Fig. 2. Pseudo code of the Hedwig semantic SD algorithm

`cou_Slovenia`). The above rule is interpreted as follows. Let `Max(X)` denote a local maximum of credit default swap (CDS), which needs to be related with the information available in the extracted features of news articles at time point `X`. The countries `Country(X)`, which were frequently mentioned in articles on day `X` that is followed by `Y` in which the Nestle company was frequently mentioned. This rule condition is true for 50 input instances, 10 of which are of target class `Max`. The two numbers refer to coverage (the number of instances for which the rule body is true) and support (the number of instances for which both the rule head and body are true), respectively.

In order to search for interesting subgroups, we employed the algorithm described in Figure 2. The Hedwig system, which implements this algorithm, supports ontologies and examples to be loaded as a collection of RDF triples (a graph). The system automatically parses the RDF graph for the `subClassOf` hierarchy, as well as any other user-defined binary relations. Hedwig also defines a namespace of classes and relations for specifying the training examples to which the input must adhere.

The algorithm uses beam search, where the beam contains the best N rules found so far. The search starts with the default rule which covers all input examples. In every iteration of the search, each rule from the beam is specialized via one of the three operations:

1. Replace the rules predicate with a predicate that is a sub-class of the previous one, e.g., `City(X)` is specialized to `Capital(X)`.
2. Append a new unary predicate to the rule, e.g., `Max(X)` ← `City(X)` is specialized to `Max(X)` ← `City(X), Company(X)`.
3. Append a new binary predicate, thus introducing a new existentially quantified variable, e.g.: `Max(X)` ← `City(X)` is specialized to `Max(X)` ← `City(X), Before(X,Y)`.[1]

Rule induction via specializations is a well-established way of inducing rules, since every specialization either maintains or reduces the current number of covered examples. A rule will not be specialized once its coverage is zero or falls below some predetermined threshold. After the specialization step is applied to each rule in the beam, a new selection of the best scoring N rules is made. If no improvement is made to the collection of rules, the search is stopped. In principle, our procedure supports any rule scoring function. Currently we implemented the popular SD scoring functions WRAcc [17], χ^2 for discrete target classes [18], and Z-score for ranked examples [19].

3 Data Acquisition and Cleaning

In this section, we present the data acquisition pipeline by describing each of its components.

The pipeline consists of several technologies that interoperate to achieve the desired goal, i.e., preparing the data for further analysis. It is responsible for acquiring unstructured data from several data sources, preparing it for the analysis, and brokering it to the appropriate analytical components. Our data acquisition pipeline is running continuously (since October 24, 2011), polling the Web and proprietary APIs for recent content, turning it into a stream of preprocessed text documents.

The news articles and web blogs are collected from 175 web sites and 2,600 RSS feeds, intentionally selected to have a strong bias for finance. We collect data from the main news providers and aggregators (like yahoo.com, dailymail.co.uk, nytimes.com, bbc.co.uk, wsj.com) and also from the main financial blogs (like zerohedge.com). The hundred most productive web sites account for 85% of collected documents. The fifty most productive domains with their average document production per day are displayed in Figure 3.

In the period from October 24, 2011 to March 31, 2013, 8,703,895 documents were collected and processed. On an average work day, about 18,000 articles are

[1] Note that variable `Y` needs to be 'consumed' by a literal to be conjunctively added to this clause in the next step of rule refinement.

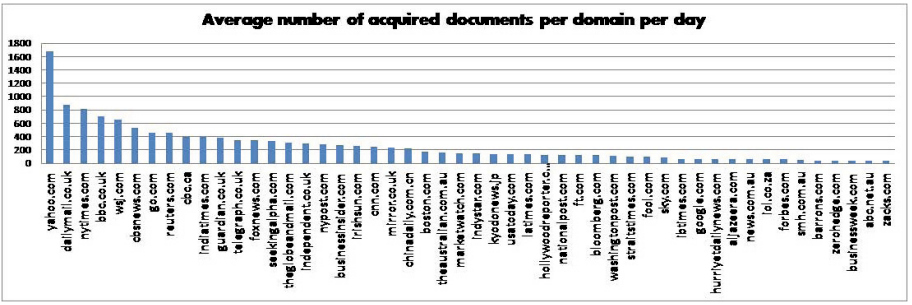

Fig. 3. The average number of acquired documents per domain per day for the fifty most productive domains. The hundred most productive web sites account for 85% of our acquired documents.

collected. The number of collected articles is substantially lower during weekends; around 10,000 per weekend day. Holidays are also characterized by a lower number of documents. The number of collected documents per day is presented in Figure 4.

When dealing with official news streams, some pre-processing steps can be avoid-ed. Official news is provided in a semi-structured fashion such that titles, publication dates, and other metadata are clearly indicated. Furthermore, named entities (i.e., company names and stock symbols) are identified in texts and article bodies are provided in a raw textual format without any boilerplate (i.e., undesired content such as advertisements, copyright notices, navigation elements, and recommendations).

Content from blogs, forums, and other Web content, however, is not immediately ready to be processed by the text analysis methods. Web pages contain a lot of noise that needs to be identified and removed before the content can be analyzed. For this reason, we have developed DacqPipe (or Dacq), a data acquisition and pre-processing pipeline. Dacq consists of (i) data acquisition components, (ii) data clean-ing components, (iii) natural-language preprocessing components, (iv) semantic anno-tation components, and (v) ZeroMQ emitter components.

The data acquisition components are mainly RSS readers that poll for data in parallel. One RSS reader is instantiated for each Web site of interest. The RSS sources, corresponding to a particular Web site, are polled one after another by the same RSS reader to prevent the servers from rejecting requests due to concurrency. An RSS reader, after it has collected a new set of documents from an RSS source, dispatches the data to one of several processing pipelines. The pipeline is chosen according to its current load size (load balancing). A processing pipeline consists of a boilerplate remover, duplicate detector, language detector, sentence splitter, tokenizer, part-of-speech tagger, lemmatizer, stop-word detector and a semantic annotator. Some of the components are custom-made while other use the functionality available from the OpenNLP library . Each pipeline component is described in more detail below.

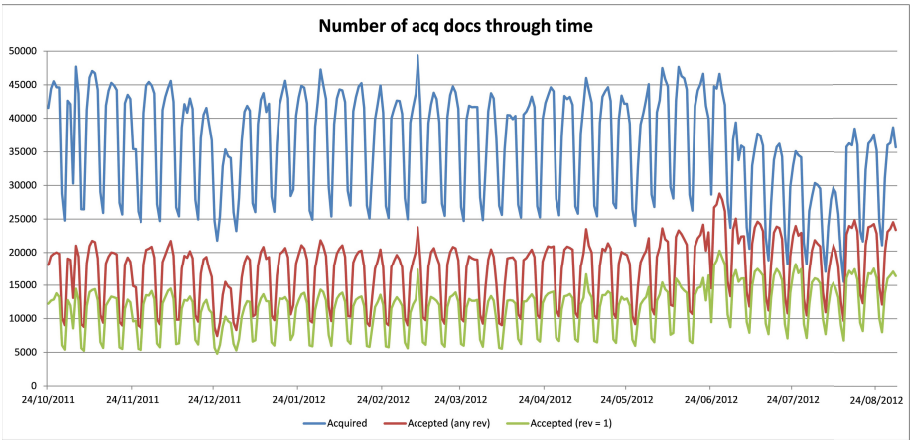

Fig. 4. The number of acquired documents per day. The top line represents the number of all acquired documents. The bottom line represents the documents that our system sees for the first time and the middle line represents the revisions of already acquired documents.

- *Boilerplate Remover*. Extracting meaningful content from Web pages presents a challenging problem. Our setting focuses on content extraction from streams of HTML documents. The developed infrastructure converts continuously acquired HTML documents into a stream of plain text documents. Our novel content extraction algorithm is efficient, unsupervised, and language-independent. The information extraction approach is based on the observation that HTML documents from the same source normally share a common template. The core of the proposed content extraction algorithm is a simple data structure called URL Tree. The performance of the algorithm was evaluated in a stream setting on a time-stamped semi-automatically annotated dataset which was made publicly available.
- *Duplicate Detector*. News aggregators are websites that aggregate web content such as news articles in one location for easy viewing. They cause articles to appear on the web with many different URLs pointing to it. To have a concise dataset of unique articles, we developed a duplicate detector that is able to see if the document was already acquired or not.
- *Language Detector*. By using a machine learning model, it detects the language and discards all the documents that are detected to be non-English. The model is trained on a large multilingual set of documents. The basic features for the model are frequencies of two consecutive words.
- *Sentence Splitter*. Splits the text into sentences. The result is the input to the part-of-speech tagger. We use the OpenNLP implementation of the Sentence splitter.

- *Tokenizer.* Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. In our pipeline, we use our own implementation of the tokenizer, which supports the Unicode character set and is based on rules.
- *Part-of-speech Tagger.* The Part of Speech (POS) Tagger marks tokens with their corresponding word type (e.g., noun, verb, proposition) based on the token itself and the context of the token. A token might have multiple POS tags depending on the token and the context. The part-of-speech tagger from the OpenNLP library is used.
- *Lemmatizer.* Lemmatization is the process of finding the normalized forms of words appearing in text. It is a useful preprocessing step for a number of language engineering and text mining tasks, and especially important for languages with rich inflectional morphology. In our data acquisition pipeline, we use LemmaGen [20] for lemmatization, which is the most efficient publicly available lemmatizer trained on large lexicons of multiple languages, whose learning engine can be retrained to effectively generate lemmatizers of other languages. We lemmatize to English.
- *Stop-word detector.* In automated text processing, stop words are words that do not carry semantic meaning. In our data acquisition pipeline, stop words are detected and annotated.
- *Semantic annotator.* Each entity has associated gazetteers; gazetteers are rules describing the entity in text. For example, "The United States of America" can appear in text as "USA", "US", "The United States", and so on. The rules include capitalization, lemmatization, POS tag constraints, must-contain constraints (another gazetteer must be detected in the document or in the sentence) and followed-by constraints.

4 Financial Use Case

First, this section presents the data and the data preparation stage needed to apply the proposed methodology. Three sources of data were used: texts from news and blogs, CDS prices and a domain ontology. Finally, this section present the experimental results achieved by applying subgroup discovery on the prepared data.

We started from a large database of annotated news articles (over 8 million), which were acquired using the data acquisition pipeline presented in the previous section. We considered articles collected over eighteen months period from October 24, 2011 to January 13, 2013. Among other properties of each article (e.g., title and URL), the most important ones for our task are the information about which entities from a pre-defined European Sovereign Debt vocabulary appear in the given article (e.g., entities like "Portugal" or "Angela Merkel" or "austerity"). These entities (counting over 6,000) are part of a larger domain ontology which consists of several class hierarchies, e.g., the Euro crisis vocabulary, companies and banks, and geographical data.

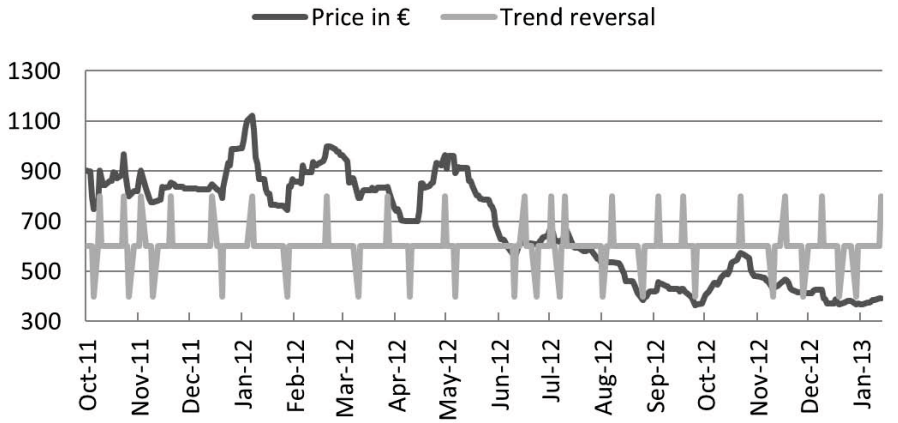

Fig. 5. Portugal CDS prices and trend reversals between October 2011 and January 2013. Upward spikes indicate local maxima, while downward spikes indicate local minima.

We decided to focus our experiments on Portugal, as it is representative and was a financially troubled country in the analyzed period. Therefore the news articles were filtered to include only the articles mentioning Portugal. The preparation stage consisted of two steps. The first step involved counting the number of times Portugal occurs together with every other entity of interest for each day of the collected history of articles. The second step involved selecting only the significant co-occurrences as example features. Each day represents one learning example and each example is described by the presence or absence of a certain entity that co-occurred with Portugal on that day. To filter out uninformative entities, we kept only the entities with a co-occurrence frequency at least 1.5 times greater than the average co-occurrence frequency over all days.

The target attribute for each example (day) was computed from the CDS prices of Portugal and has three possible values that indicate the significant local extremes in the CDS price timelines: 'max' or 'min' if the local extreme was reached, respectively, or 'steady' if there was no change in the trend (Figure 5). These steps yielded a dataset of 337 examples, each with an average of 282 features (ranging between 35 and 761).

The processed news and blogs articles, the CDS local extremes and the domain ontology were encoded as a set of RDF triples which were input to the Hedwig system.

The financial ontology which we actually used in the experiments is illustrated in Figure 6. The ontology has three main branches: financial entities, geographical entities and a specialized vocabulary of the European sovereign debt crisis. Some parts of the ontology were automatically induced by reusing various data sources, while other parts, like the vocabulary, were constructed manually.

Each entity in the ontology is equipped with a gazetteer. The gazetteer contains lexical knowledge about the possible forms in which the entity occurs in texts. This knowledge is used by the entity recognition engine which is attached to the data acquisition pipeline. Note that the gazetteers are initially built automatically in the ontology construction process. This approach to entity recognition is prone to errors due to homographs, i.e., words that are spelled the same but have different meanings. This is especially prominent for acronyms and stock symbols. To improve the entity recognition process and to reduce the noise in the stream of discovered entities, we have performed several semi-automated ontology refinement iterations.

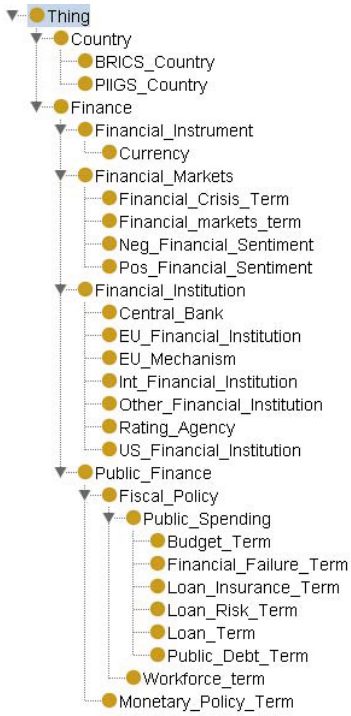

Fig. 6. The ontology that conceptualizes the European financial crisis vocabulary

We used the IDMS database and MSN Money[2] to grow the ontology from a list of seed stock indices to its constituents (stocks) and further on to the companies that issue these stocks. This resulted in to 2019 financial entities (like banks, companies, investment funds, stocks and stock indexes). The geographical part of the ontology was generated from GeoNames[3] (countries, cities, regions, etc). We selected 598 most important geographical entities and included them into the ontology. The specialized vocabulary of the European financial crisis (166 terms) was developed manually by using expert knowledge (Figure 6). The main protagonists of the crisis were taken from Wikipedia[4].

In our experiment, we focused on finding subgroups for two target classes which represent trend reversals: the local maximum ('max') represents the date when the CDS price started to decrease and the local minimum ('min') the opposite. In both cases, we used the WRAcc subgroup discovery rule score, a beam width of 100, minimum coverage of 5 examples and the maximum number of predicates per rule of 6.

[2] http://money.msn.com/
[3] http://www.geonames.org/
[4] http://en.wikipedia.org/wiki/List_of_protagonists:_European_sovereign-debt_crisis

For the case of CDS price reaching the maximum (target class 'max'), the best scoring subgroup was:

```
Max(X) ← reg_Western_Europe(X), Angela_Merkel(X),
   glo_austerity(X), glo_recession(X). [28, 7]
```

For the case of CDS price reaching the minimum (target class 'min'), the best scoring subgroup was:

```
Min(X) ← Index(X), comp_GALP_ENERGIA(X), Loan_Term(X),
            glo_fiscal_stimulus(X). [43, 8]
```

The first rule indicates that Portugal CDS prices reaching a local maximum are characterized by increased frequency of the following entities co-occurring with Portugal: the Western Europe region, Angela Merkel, and the terms 'austerity' and 'recession'. We should point out that a local maximum in a country's CDS price indicates that from that day on, the market expectation that the country will default decreased. Conversely, the second rule tells us that when the CDS price reach a local minimum, we can notice an increased frequency of (stock) index terms, Portugal's corporation of natural and renewable energy companies (Galp Energia), loan terms and 'fiscal stimulus'. These results show that the higher the CDS prices, the more the sovereign debt vocabulary is used. When CDS prices are low, a more general financial terminology is used.

5 Conclusions

The newly developed semantic subgroup discovery system Hedwig was presented, which overcomes the limitations of existing semantic subgroup discovery systems. Compared to standard subgroup discovery, novelties of this paper are the exploitation of the ontology to generalize over the entities, while also using of the user-provided binary relations and using the `subClassOf` relation to guide the search procedure. We are currently performing a comprehensive study which should result in a comparison of the new system with the related work.

We employed Hedwig for analyzing news articles about Portugal during the last year and a half. Using co-occurrence frequencies of entities appearing together with Portugal, a domain ontology linking the entities into a formal hierarchy, and a history of Credit Default Swap (CDS) prices, we induced subgroups describing prominent entities appearing at times of CDS trend reversals (either upward or downward). The extracted subgroup descriptions give us a clear indication that news articles content indeed reflects the CDS prices. Having this information, we are encouraged to proceed with building a model for CDS trend reversal prediction. For this purpose, we plan to include additional information about the entities (e.g., TF-IDF weights) and extra-textual information (not only the pre-defined ontological entities) into the input data. Additionally, we will employ several classification algorithms and compare them.

Acknowledgments. This work was supported by the Slovenian Research Agency [grants P-103 and P-04431] and the EU projects "Large scale information extraction and integration infrastructure for supporting financial decision making" (FIRST, grant agreement 257928) and "Forecasting Financial Crises" (FOC, grant agreement 255987).

References

[1] Klösgen, W.: Explora: a multipattern and multistrategy discovery assistant. In: Advances in Knowledge Discovery and Data Mining, pp. 249–271. American Association for Artificial Intelligence, Menlo Park (1996)
[2] Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Komorowski, J., Żytkow, J.M. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 78–87. Springer, Heidelberg (1997)
[3] Muggleton, S. (ed.): Inductive Logic Programming. The APIC Series, vol. 38. Academic Press (1992)
[4] De Raedt, L.: Logical and Relational Learning. Springer, Heidelberg (2008)
[5] Džeroski, S., Lavrač, N. (eds.): Relational Data Mining. Springer, Berlin (2001)
[6] Vavpetič, A., Lavrač, N.: Semantic subgroup discovery systems and workflows in the SDM-Toolkit. Comput. J. 56(3), 304–320 (2013)
[7] Kietz, J.-U.: Learnability of description logic programs. In: Matwin, S., Sammut, C. (eds.) ILP 2002. LNCS (LNAI), vol. 2583, pp. 117–132. Springer, Heidelberg (2003)
[8] Lehmann, J., Haase, C.: Ideal downward refinement in the $\mathcal{EL}$ description logic. In: De Raedt, L. (ed.) ILP 2009. LNCS, vol. 5989, pp. 73–87. Springer, Heidelberg (2010)
[9] Ławrynowicz, A., Potoniec, J.: Fr-ONT: An algorithm for frequent concept mining with formal ontologies. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 428–437. Springer, Heidelberg (2011)
[10] Berendt, B., Hotho, A., Stumme, G.: Towards semantic web mining. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 264–278. Springer, Heidelberg (2002)
[11] Lisi, F.A., Malerba, D.: Inducing multi-level association rules from multiple relations. Machine Learning 55, 175–210 (2004), 10.1023/B:MACH.0000023151.65011.a3
[12] Lisi, F.A., Esposito, F.: Mining the semantic web: A logic-based methodology. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 102–111. Springer, Heidelberg (2005)
[13] Trajkovski, I., Železný, F., Lavrač, N., Tolar, J.: Learning relational descriptions of differentially expressed gene groups. IEEE Transactions on Systems, Man, and Cybernetics, Part C 38(1), 16–25 (2008)
[14] Žáková, M., Železný, F., Garcia-Sedano, J.A., Tissot, C.M., Lavrač, N., Křemen, P., Molina, J.: Relational data mining applied to virtual engineering of product designs. In: Muggleton, S.H., Otero, R., Tamaddoni-Nezhad, A. (eds.) ILP 2006. LNCS (LNAI), vol. 4455, pp. 439–453. Springer, Heidelberg (2007)
[15] Hull, J., Predescu-Vasvari, M., White, A., Rotman, J.L.: The relationship between credit default swap spreads, bond yields, and credit rating announcements (2002)

[16] Gamberger, D., Lučanin, D., Šmuc, T.: Descriptive modeling of systemic banking crises. In: Ganascia, J.-G., Lenca, P., Petit, J.-M. (eds.) DS 2012. LNCS, vol. 7569, pp. 67–80. Springer, Heidelberg (2012)
[17] Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup discovery with CN2-SD. Journal of Machine Learning Research 5, 153–188 (2004)
[18] Shimada, K., Hirasawa, K., Hu, J.: Class association rule mining with chi-squared test using genetic network programming. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2006, vol. 6, pp. 5338–5344 (2006)
[19] DeGroot, M.H., Schervish, M.J.: Probability and Statistics, ch. 8, 9. Addison-Wesley (2002)
[20] Juršič, M., Mozetič, I., Erjavec, T., Lavrač, N.: Lemmagen: Multilingual lemmatisation with induced ripple-down rules. J. UCS 16(9), 1190–1214 (2010)