Detecting Economic Events Using a Semantics-Based Pipeline

Alexander Hogenboom, Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, Otto van der Meer, and Kim Schouten

Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, The Netherlands

{hogenboom, fhogenboom, frasincar, kaymak}@ese.eur.nl,
{276933rm, 288054ks}@student.eur.nl

Abstract. In today's information-driven global economy, breaking news on economic events such as acquisitions and stock splits has a substantial impact on the financial markets. Therefore, it is important to be able to automatically identify events in news items accurately and in a timely manner. For this purpose, one has to be able to mine a wide variety of heterogeneous sources of unstructured data to extract knowledge that is useful for guiding decision making processes. We propose a Semantics-based Pipeline for Economic Event Detection (SPEED), which aims at extracting financial events from news articles and annotating these events with meta-data, while retaining a speed that is high enough to make real-time use possible. In our pipeline implementation, we have reused some of the components of an existing framework and developed new ones, such as an Ontology Gazetteer and a Word Sense Disambiguator.

1 Introduction

In today's information-driven society, machines that can perform Natural Language Processing (NLP) tasks can be of great importance. Decision makers are expected to process a continuous, overwhelming flow of (news) messages by extracting information and understanding their meaning. Knowledge can subsequently be acquired by applying reasoning to the gathered information. In today's global economy, it is of paramount importance for decision makers to have a sensible intuition on the state of their market, which is often extremely sensitive to breaking news on economic events like acquisitions, stock splits, dividend announcements, etc. In this context, identification of events can guide decision making processes, as these events provide means of structuring information using concepts, with which knowledge can be generated by applying inference. Automating information extraction and knowledge acquisition processes can facilitate or support decision makers in fulfilling their cumbersome tasks, as one can make better informed decisions due to faster processing of more data.

Therefore, we aim to have a fully automated framework for processing financial news messages gathered from Really Simple Syndication (RSS) feeds. These events are represented in a machine-understandable way. Extracted events can be made accessible for other applications through the use of Semantic Web technologies. Furthermore, we aim for the framework to be able to handle news messages at a speed useful for real-time use, as new events can occur any time and require decision makers to respond in a timely and adequate manner.

We propose a pipeline that identifies the concepts related to economic events, which are defined in a domain ontology and are associated to synsets from a semantic lexicon such as WordNet [3]. For concept identification, we employ lexico-semantic patterns based on ontology concepts in order to match lexical representations of concepts retrieved from the text with event-related concepts that are available in the semantic lexicon, and thus aim to maximize recall. The identified lexical representations of relevant concepts are subject to a Word Sense Disambiguation (WSD) procedure for determining the corresponding sense, in order to maximize precision. To enable real-time use, we also aim to minimize the latency, i.e., the time it takes for the pipeline to process a news message.

The remainder of this paper is structured as follows. First, Sect. 2 discusses related work. Subsequently, Sect. 3 elaborates on the proposed framework. Then, the framework is evaluated in Sect. 4, after which Sect. 5 concludes the paper.

2 Related Work

Several tools have already been proposed for our desired Information Extraction (IE) purposes, most of which have their own IE framework. However, the General Architecture for Text Engineering (GATE) [2], a freely available general purpose framework for IE purposes, has become increasingly popular. The tool is highly flexible in that the user can construct processing pipelines from components that perform specific tasks. One can distinguish between linguistic analysis applications such as tokenization (e.g., distinguishing words), syntactic analysis jobs like Part-Of-Speech (POS) tagging, and semantic analysis tasks such as understanding. By default, GATE loads the A Nearly-New Information Extraction (ANNIE) system, which consists of several key components, i.e., the *English Tokenizer*, *Sentence Splitter*, *Part-Of-Speech (POS) Tagger*, *Gazetteer*, *Named Entity (NE) Transducer*, and *OrthoMatcher*.

Although the ANNIE pipeline has proven to be useful in various information extraction jobs, its functionality does not suffice when applied to discovering economic events in news messages. For instance, ANNIE lacks important features such as a WSD component, although some disambiguation can be done using JAPE rules in the *NE Transducer*. This is however a cumbersome and ineffective approach where rules have to be created manually for each term, which is prone to errors. Furthermore, ANNIE lacks the ability to individually look up concepts from a large ontology within a limited amount of time. Nevertheless, GATE is highly flexible and customizable, and therefore ANNIE's components are either usable, extendible, or replaceable in order to suit our needs.

An example of a tool utilizing ANNIE components is Hermes [4], which extracts a set of news items related to specific concepts of interest. ANNIE components are used that make use of concepts and relations stored in ontologies. Another example of an adapted ANNIE pipeline is the Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RElations (CAFETIERE) relation extraction pipeline [1], which contains an ontology look-up process and a rule engine. CAFETIERE employs extraction rules defined at lexico-semantic level which are more easy to express, yet less flexible than JAPE rules. CAFETIERE stores knowledge in a type of ontology which has no formal semantics and lacks reasoning support, rendering this an unattractive approach for identifying, e.g., financial events. Furthermore, gazetteering is a slow process when going through large ontologies. Finally, the pipeline also misses a WSD component.

The Knowledge and Information Management (KIM) platform [8] provides an infrastructure for IE purposes, by combining the GATE architecture with semantic annotation techniques. KIM focuses on automatic annotation of news articles, where entities, inter-entity relations, and attributes are discovered. For this, a pre-populated Web Ontology Language (OWL) upper ontology is employed, i.e., a minimal but sufficient ontology that is suitable for open domain and general purpose annotation tasks. The semantic annotations in articles allow for applications such as semantic querying and exploring the semantic repository. The differences between KIM and our approach are in that we aim for a financial event-focused information extraction pipeline, which is in contrast to KIM's general purpose framework. Hence, we employ a domain-specific ontology instead of an upper ontology. Also, rather than just annotating corpora with event concepts, we extract additional information by utilizing lexico-semantic patterns for linking identified concepts, thus realizing a rich knowledge base. Furthermore, no mention has been made regarding WSD within the KIM platform, whereas we consider WSD to be an essential component in an IE pipeline.

3 Economic Event Detection based on Semantics

Where current approaches to automated IE from news messages are more focused on annotation of documents, we strive to actually extract information – i.e., specific economic events – from documents, with which for instance a knowledge base can be updated. The analysis of texts needs to be driven by semantics, as the domain-specific information captured in these semantics facilitates detection of relevant concepts. Therefore, we propose a Semantics-Based Pipeline for Economic Event Detection (SPEED), consisting of several components which sequentially process documents. This approach is driven by an ontology containing information on the NASDAQ-100 companies, extracted from Yahoo! Finance. This domain ontology has been developed by domain experts through an incremental middle-out approach, validated using the OntoClean methodology [5]. The ontology captures concepts and events from the financial domain, e.g., companies, competitors, products, etc. Many concepts in this ontology stem from a semantic lexicon (e.g., WordNet) or represent named entities.

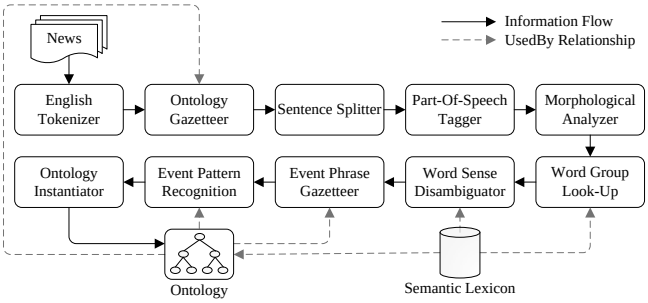

Fig. 1. SPEED design.

Our proposed pipeline, depicted in Fig. 1, is designed to identify relevant concepts and their relations in a document. To this end, individual components of the text are first identified as such by means of the *English Tokenizer*, which splits text into tokens (e.g., words or numbers) while taking into account rules specific to the English language. These tokens are then linked to ontology concepts by an *Ontology Gazetteer*. Matching tokens in the text are thus annotated with a reference to their associated concepts defined in the ontology.

Then, the *Sentence Splitter* groups the tokens in the text into sentences, based on tokens indicating a separation between sentences, e.g., (a combination of) punctuation symbols or new line characters. These sentences are used for discovering the grammatical structure in text by determining the part-of-speech of each word token by means of the *Part-Of-Speech Tagger*. As words can have many forms that have a similar meaning, the *Morphological Analyzer* subsequently reduces the tagged words to their lemma and an affix.

Words and meanings, denoted often as synsets (set of synonyms) have a many-to-many relationship. Hence, the next step in interpreting a text is disambiguation of its words' meaning, given their POS tags, lemmas, etc. To this end, a *Word Group Look-Up* component first combines words into word groups containing as many words per group as possible for representing some concept in the semantic lexicon. The *Word Sense Disambiguator* then determines the word sense of each word group by exploring the mutual relations between senses (as defined in the semantic lexicon and the ontology) of word groups; the stronger the relation with surrounding senses, the more likely a sense matches the context.

To this end, we propose an adaptation of the Structural Semantic Interconnections (SSI) [7] algorithm. The SSI approach uses graphs to describe word groups and their context (word senses), as derived from a semantic lexicon. The senses are determined based on the number and type of detected semantic interconnections in a labeled directed graph representation of all senses of the considered word groups. We differ from SSI in that our algorithm, shown in Algorithm 1, considers the two most likely senses for each word group and iteratively disambiguates the word group with the highest confidence (i.e., weighted difference of the similarity of both senses to already disambiguated senses), rather

```
a = ∅; // Lists ambiguous word groups yet to be disambiguated
d = ∅; // Lists disambiguated word groups
s = ∅; // Lists senses of disambiguated word groups
c = ∅; // Lists context (i.e., possible senses of all considered word groups)
l = ∅; // Lists similarity of context to disambiguated senses
// Initialize disambiguation
w = getWordGroups();
foreach g in w do
    senses = getSenses(g);
    // Add word group g with one sense to d and its sense to s
    if |senses| == 1 then
        add(d,g);
        add(s,senses);
    // Add ambiguous word group g to a and its senses to c
    else
        add(a,g);
        foreach sense in senses do if sense ∉ c then add(c,sense);
    end
end
// Determine similarity of all senses in c to all disambiguated senses in s
foreach sense in c do
    simToS = 0;
    foreach knownSense in s do simToS += 1/shortestPathLength(sense,knownSense);
    add(l,simToS);
end
// Disambiguate word groups in a
lastAddedSense = ∅;
while a ≠ ∅ do
    bestPick,bestPickSense = ∅;
    bestPickConf = −∞;
    foreach g in a do
        bestSense1,bestSense2 = ∅;
        bestSim1,bestSim2 = −∞;
        senses = getSenses(g);
        foreach sense in senses do
            // Update similarity of sense to s with similarity to lastAddedSense
            indexSense = indexOf(c,sense);
            simToS = get(l,indexSense);
            simToS += 1/shortestPathLength(sense,lastAddedSense);
            set(l,indexSense,simToS);
            // Update best senses
            if simToS > bestSim2 then
                if simToS > bestSim1 then
                    bestSense2 = bestSense1; bestSense1 = sense;
                    bestSim2 = bestSim1; bestSim1 = simToS;
                else
                    bestSense2 = sense;
                    bestSim2 = simToS;
                end
            end
        end
        // Update best pick
        confidence = ((bestSim1-bestSim2)*bestSim1);
        if confidence > bestPickConf then
            bestPick = g;
            bestPickSense = bestSense1;
            bestPickConf = confidence;
        end
    end
    // Disambiguate best pick, move it from a to d, and add its sense to s
    rem(a,indexOf(a,bestPick));
    add(d,bestPick);
    add(s,bestPickSense);
    lastAddedSense = bestPickSense;
end
```

Algorithm 1: Word Sense Disambiguation for an arbitrary news item.

than the word group with the greatest similarity for its best sense. Furthermore, in case an arbitrary word cannot be disambiguated, we default to the statistically most likely sense in our semantic lexicon, whereas the original SSI algorithm fails to provide a word sense. In our algorithm, we compute the similarity of a sense to already disambiguated senses as the sum of the inverse of the shortest path length between this sense and the disambiguated senses in the semantic graph.

When the meaning of word groups has been disambiguated, the text can be interpreted by introducing semantics linking word groups to an ontology, thus capturing their essence in a meaningful and machine-understandable way. As we are interested in specific economic events, the *Event Phrase Gazetteer* scans the text for those events. It uses a list of phrases or concepts that are likely to represent some part of a relevant event. Events thus identified are then supplied with available additional information (e.g., time stamps) by the *Event Pattern Recognition* component, which matches identified events with predefined domain-specific lexico-semantic patterns. Finally, the knowledge base can be updated by inserting the identified events and their extracted associated information into the ontology using the *Ontology Instantiator*, as detailed in our previous work [9].

4 Evaluation

The modularity of an architecture like GATE can facilitate the implementation and subsequent evaluation of our proposed semantics-based pipeline for economic event detection. Therefore, we have made a Java-based implementation of the proposed framework, partially using default GATE components which suit our needs, i.e., the *English Tokenizer*, *Sentence Splitter*, *Part-Of-Speech Tagger*, and the *Morphological Analyzer*. Additionally, we have extended the functionality of other GATE components (e.g., for ontology gazetteering), and also implemented new components to tackle the WSD and economic event detection processes.

The implementations of both our *Ontology Gazetteer* and *Word Group Look-Up* components match concepts (i.e., ontology concepts and WordNet word groups, respectively) with lexical representations stored in a look-up tree, where nodes represent individual tokens and a path from the root node to an arbitrary leaf node represents a concept's lexical representation. For each token, the look-up tree is consulted, starting from the root node. If the token is not in the root, the next token in the text is again looked up in the root. Else, the next token in the text is looked up in the root node of the subtree belonging to the former token. This process is iterated until either a leaf node is reached, or the current node does not have a reference to the next token in the text. The word group associated with the followed path is then annotated with the associated concept. Our trees for ontology concepts and word groups have been implemented using hash maps, in order to reduce the time needed to traverse the trees.

In order to evaluate SPEED's performance, we assess statistics that describe the cumulative error, i.e., precision and recall, and latency. We define precision as the part of the identified concepts (e.g., word senses or events) that have been identified correctly, and recall represents the number of identified concepts as

a fraction of the number of concepts that should have been identified. When we compare the performance of different approaches, we assess the statistical relevance of differences in performance by means of a paired t-test.

We evaluate our *Word Sense Disambiguator* on a large, publicly available WSD corpus – SemCor [6]. On this corpus, the original SSI algorithm exhibits an average precision of 53% with a standard deviation of 5 percentage points and a recall of 31% with a standard deviation of 9 percentage points. Conversely, our proposed adaptation of SSI exhibits an average precision and recall of 59% with a standard deviation of 5 percentage points. This implies an overall improvement in precision and recall with 12% and 90%, respectively, compared to the original SSI algorithm, at a significance level of 0.001.

In the evaluation of our framework as a whole, we focus on a data set consisting of 200 news messages extracted from the Yahoo! Business and Technology newsfeeds. Three domain experts have manually annotated these for our considered economic events and relations, while ensuring an inter-annotator agreement of at least 66% (i.e., at least two out of three annotators agree). We distinguish between ten different financial events, i.e., announcements regarding CEOs (60), presidents (22), products (136), competitors (50), partners (23), subsidiaries (46), share values (45), revenues (22), profits (33), and losses (27).

We observe a precision for the concept identification in news items of 86% and a recall of 81%. It should however be noted that precision and recall of fully decorated events result in lower values of approximately 62% and 53% respectively, as they rely on multiple concepts that have to be identified correctly. Errors in concept identification result from missing lexical representations of the knowledge base concepts, and missing concepts in general. Despite using only WordNet as a semantic lexicon, we obtain high precision as many of our concepts' lexical representations are named entities, which often are monosemous. High recall can be explained by SPEED's focus on detecting ontology concepts in the text, rather than on identifying all concepts in the text.

On our data set, our pipeline exhibits a latency of on average 632 milliseconds per document, with a standard deviation of 398 milliseconds. Of this execution time, roughly 30% is allocated to the first part of the pipeline, performing linguistic and syntactic analysis tasks. The subsequent WSD task on average takes up about 60% of the execution time, whereas the remaining tasks are typically performed in about 10% of the execution time.

5 Conclusions and Future Work

We have proposed a semantics-based framework for economic event detection (SPEED), which aims to extract financial events from news articles (announced through RSS feeds) and to annotate these with meta-data, while maintaining a speed that is high enough to enable real-time use. For implementing the SPEED pipeline, we have reused existing components and developed new ones such as gazetteers and word sense disambiguator. Our framework is semantically enabled, i.e., it makes use of semantic lexicons and ontologies. Furthermore,

pipeline outputs also make use of semantics, which introduces a potential feedback loop, making event identification a more adaptive process. The merit of our pipeline is in the use of semantics, enabling broader application interoperability. Although we focus on the financial domain, SPEED is generalizable to other domains, as we separate the domain-specific aspects from the domain-independent ones. The established fast processing time and high precision and recall provide a good basis for future work.

For future work, we aim to investigate further possibilities for implementation in algorithmic trading environments. We aim to find a way of utilizing discovered events in this field. To this end, we also envision another addition, i.e., a way of associating sentiment with discovered events. As sentiment of actors with respect to events may be the driving force behind their reactions to these events, this information could be exploited in an algorithmic trading setup.

Acknowledgment

The authors are partially sponsored by the NWO Physical Sciences Free Competition project 612.001.009: Financial Events Recognition in News for Algorithmic Trading (FERNAT).

References

[1] Black, W.J., M^{c}Naught, J., Vasilakopoulos, A., Zervanou, K., Theodoulidis, B., Rinaldi, F.: CAFETIERE: Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RElations. Technical Report TR–U4.3.1, Department of Computation, UMIST, Manchester (2005)

[2] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: 40th Anniversary Meeting of the ACL (ACL 2002). pp. 168–175. ACL (2002)

[3] Fellbaum, C.: WordNet an Electronic Lexical Database. Computational Linguistics 25(2), 292–296 (1998)

[4] Frasincar, F., Borsje, J., Levering, L.: A Semantic Web-Based Approach for Building Personalized News Services. International Journal of E-Business Research 5(3), 35–53 (2009)

[5] Guarino, N., Welty, C.A.: Evaluating Ontological Decisions with OntoClean. Communications of the ACM 45(2), 61–65 (2002)

[6] Miller, G., Chodorow, M., Landes, S., Leacock, C., Thomas, R.: Using a Semantic Concordance for Sense Identification. In: Proceedings of the Human Language Technology Workshop (HLT 1994). pp. 240–243. ACL (1994)

[7] Navigli, R., Velardi, P.: Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. Trans. Pattern Anal. Machine Intell. 27(7), 1075–1086 (July 2005)

[8] Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A.: KIM - A Semantic Platform For Information Extraction and Retrieval. Journal of Natural Language Engineering 10(3–4), 375–392 (September 2004)

[9] Schouten, K., Ruijgrok, P., Borsje, J., Frasincar, F., Levering, L., Hogenboom, F.: A Semantic Web-Based Approach for Personalizing News. In: 25th Symposium On Applied Computing (SAC 2010). pp. 854–861. ACM (2010)