SPEED: A Semantics-Based Pipeline for Economic Event Detection

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

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

{fhogenboom, hogenboom, frasincar, kaymak}@ese.eur.nl,
{276933rm, 288054ks, 305415dv}@student.eur.nl

Abstract. Nowadays, emerging news on economic events such as acquisitions has a substantial impact on the financial markets. Therefore, it is important to be able to automatically and accurately identify events in news items in a timely manner. For this, one has to be able to process a large amount of heterogeneous sources of unstructured data in order to extract knowledge useful for guiding decision making processes. We propose a Semantics-based Pipeline for Economic Event Detection (SPEED), aiming to extract financial events from emerging news and to annotate these with meta-data, while retaining a speed that is high enough to make real-time use possible. In our implementation of the SPEED pipeline, we reuse some of components of an existing framework and develop new ones, e.g., a high-performance Ontology Gazetteer and a Word Sense Disambiguator. Initial results drive the expectation of a good performance on emerging news.

1 Introduction

In today's information-driven society, machines that can process natural language can be of great importance. Decision makers are expected to be able to extract information from an ever increasing amount of data such as emerging news, and subsequently to be able to acquire knowledge by applying reasoning to the gathered information. In today's global economy, it is of utmost importance to have a complete overview of the business environment to enable effective, well-informed decision making. Financial decision makers thus need to be aware of events on their financial market, which is often extremely sensitive to economic events like stock splits and dividend announcements. Proper and timely event identification can aid decision making processes, as these events provide means of structuring information using concepts, with which knowledge can be generated by applying inference. Hence, automating information extraction and knowledge acquisition processes can be a valuable contribution.

This paper proposes a fully automated framework for processing financial news messages gathered from RSS feeds. These events are represented in a

machine-understandable way. Extracted events can be made accessible for other applications as well through the use of Semantic Web technologies. Furthermore, it is aimed that the framework is 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.

Our proposed framework (pipeline) identifies the concepts related to economic events, which are defined in a domain ontology and are associated to synsets from a semantic lexicon (e.g., WordNet [1]). For concept identification, lexico-semantic patterns based on concepts from the ontology are employed 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. In order for our pipeline to be real-time applicable, we also aim to minimize the latency, i.e., the time it takes for a new news message to be processed by the pipeline.

The remainder of this paper is structured as follows. Firstly, Sect. 2 discusses related work. Subsequently, Sects. 3 and 4 elaborate on the proposed framework and its implementation, respectively. Finally, Sect. 5 wraps up this paper.

2 Related Work

This section discusses tools that can be used for Information Extraction (IE) purposes. Firstly, Sect. 2.1 discusses the ANNIE pipeline and subsequently, Sects. 2.2 and 2.3 elaborate on the CAFETIERE and KIM frameworks, respectively. Finally, Sect. 2.4 wraps up this section.

2.1 The ANNIE Pipeline

The General Architecture for Text Engineering (GATE) [2] is a freely available general purpose framework for IE tasks, which provides the possibility to construct processing pipelines from components that perform specific tasks, e.g., linguistic, syntactic, and semantic analysis tasks. 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. An important lacking component is one that can be employed for WSD, 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. Despite its drawbacks, GATE is highly flexible and customizable, and therefore ANNIE's components are either usable, or extendible and replaceable in order to suit our needs.

2.2 The CAFETIERE Pipeline

An example of an adapted ANNIE pipeline is the Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RElations (CAFETIERE) relation extraction pipeline [3], which consists of an ontology lookup process and a rule engine. Within CAFETIERE, the Common Annotation Scheme (CAS) DTD is applied, which allows for three layers of annotation, i.e., structural, lexical, and semantic annotation. CAFETIERE employs extraction rules defined at lexico-semantic level which are similar to JAPE rules. Nevertheless, the syntax is at a higher level than is the case with JAPE, resulting in more easy to express, but less flexible rules.

As knowledge is stored in an ontology using Narrative Knowledge Representation Language (NKRL), Semantic Web ontologies are not employed. NKRL has no formal semantics and there is no reasoning support, which is desired when identifying for instance economic events. Furthermore, gazetteering is a slow process when going through large ontologies. Finally, the pipeline also misses a WSD component.

2.3 The KIM Platform

The Knowledge and Information Management (KIM) platform [4] combines GATE components with semantic annotation techniques in order to provide an infrastructure for IE purposes. The framework focuses on automatic annotation of news articles, where entities, inter-entity relations, and attributes are discovered. For this, the authors employ a pre-populated OWL upper ontology. In the back-end, a semantically enabled GATE pipeline, which utilizes semantic gazetteers and pattern-matching grammars, is invoked for named entity recognition using the KIM ontology. Furthermore, GATE is used for managing the content and annotations within the back-end of KIM's architecture. The middle layer of the KIM architecture provides services that can be used by the topmost layer, e.g., semantic repository navigation, semantic indexing and retrieval, etcetera. The front-end layer of KIM embodies front-end applications, such as the *Annotation Server* and the *News Collector*.

The differences between KIM and our envisaged 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 rather than an upper ontology. Furthermore, we focus on event extraction from corpora, in contrast to mere (semantic) annotation. Finally, the authors do not mention the use of WSD, whereas we consider WSD to be an essential component in an IE pipeline.

2.4 Conclusions

The IE frameworks discussed in this section have their merits, yet each framework fails to fully address the issues we aim to alleviate. The frameworks incorporate semantics only to a limited extent, e.g., they make use of gazetteers or

knowledge bases that are either not ontologies or ontologies that are not based on OWL. Being able to use a standard language as OWL fosters application interoperability and the reuse of existing reasoning tools. Also, existing frameworks lack a feed-back loop, i.e., there is no knowledge base updating. Furthermore, WSD appears not to be sufficiently tackled in most cases. Finally, most existing approaches focus on annotation, rather than event recognition. Therefore, we aim for a framework that combines the insights gained from the approaches that are previously discussed, targeted at financial event discovery in news articles.

3 Economic Event Detection based on Semantics

The analysis presented in Sect. 2 demonstrates several approaches to automated information extraction from news messages, which are often applied for annotation purposes and are not semantics-driven. Because we hypothesize that domain-specific information captured in semantics facilitates detection of relevant concepts, we propose a Semantics-Based Pipeline for Economic Event Detection (SPEED). The framework is modeled as a pipeline and is driven by a financial ontology developed by domain experts, containing information on the NASDAQ-100 companies that is extracted from Yahoo! Finance. Many concepts in this ontology stem from a semantic lexicon (e.g., WordNet), but another significant part of the ontology consists of concepts representing named entities (i.e., proper names).

Figure 1 depicts the architecture of the pipeline. In order to identify relevant concepts and their relations, the *English Tokenizer* is employed, which splits text into tokens (which can be for instance words or numbers) and subsequently applies linguistic rules in order to split or merge identified tokens. These tokens are linked to ontology concepts by means of the *Ontology Gazetteer*, in contrast to a regular gazetteer, which uses lists of words as input. Matching tokens in the text are annotated with a reference to their associated concepts defined in the ontology.

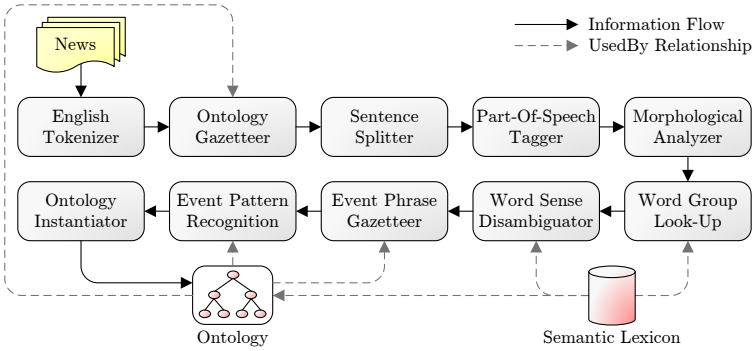

Fig. 1. SPEED design.

Subsequently, the *Sentence Splitter* groups the tokens in the text into sentences, based on tokens indicating a separation between sentences. These sentences are used for discovering the grammatical structures in a corpus by determining the type 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 as well as a suffix and/or affix.

A word can have multiple meanings and a meaning can be represented by multiple words. Hence, the framework needs to tackle WSD tasks, given POS tags, lemmas, etcetera. To this end, first of all, the *Word Group Look-Up* component combines words into maximal word groups, i.e., it aims for as many words per group as possible for representing some concept in a semantic lexicon (such as WordNet). It is important to keep in mind groupings of words, as combinations of words may have very specific meanings compared to the individual words. Subsequently, the *Word Sense Disambiguator* determines the word sense of each word group by exploring the mutual relations between senses of word groups using graphs. 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 [5].

After disambiguating word group senses, the text can be interpreted by introducing semantics, which links word groups to an ontology, thus capturing their essence in a meaningful and machine-understandable way. Therefore, the *Event Phrase Gazetteer* scans the text for specific (financial) events, by utilizing 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 by the *Event Pattern Recognition* component, which matches events to lexico-semantic patterns that are subsequently used for extracting additional information. Finally, the knowledge base is updated by inserting the identified events and their extracted associated information into the ontology by means of the *Ontology Instantiator*.

4 SPEED Implementation

The analysis presented in Sect. 2 exhibits the potential of a general architecture for text engineering: GATE. The modularity of such an architecture can be of use in the implementation of our semantics-based pipeline for economic event detection, as proposed in Sect. 3. Therefore, we made a Java-based implementation of the proposed framework by using default GATE components, such as the *English Tokenizer*, *Sentence Splitter*, *Part-Of-Speech Tagger*, and the *Morphological Analyzer*, which generally suit our needs. Furthermore, we extended the functionality of other GATE components (e.g., ontology gazetteering), and also implemented additional components to tackle the disambiguation process.

Initial results on a test corpus of 200 news messages fetched from the Yahoo! Business and Technology RSS feeds show fast gazetteering of about 1 second and a precision and recall for concept identification in news items of 86% and 81%,

respectively, which is comparable with existing systems. Precision and recall of fully decorated events result in lower values of approximately 62% and 53%, as they rely on multiple concepts that have to be identified correctly.

5 Conclusions and Future Work

In this paper, we have proposed a semantics-based framework for economic event detection: SPEED. The framework 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. We discussed the main components of the framework, which introduce some novelties, as they are semantically enabled, i.e., they make 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. Finally, we briefly touched upon the implementation of the framework and initial test results on the basis of emerging news. The established fast processing time and high precision and recall provide a good basis for future work. The merit of our pipeline is in the use of semantics, enabling broader application interoperability.

For future work, we do not only aim to perform thorough testing and evaluation, but also to implement the proposed feedback of newly obtained knowledge (derived from identified events) to the knowledge base. Also, it would be worthwhile to investigate further possibilities for implementation in algorithmic trading environments, as well as a principal way of linking sentiment to discovered events, in order to assign more meaning to these events.

Acknowledgments The authors are partially sponsored by the NWO EW Free Competition project FERNAT: Financial Events Recognition in News for Algorithmic Trading.

References

1. Fellbaum, C.: WordNet an Electronic Lexical Database. Computational Linguistics 25(2), 292–296 (1998)
2. Cunningham, H.: GATE, a General Architecture for Text Engineering. Computers and the Humanities 36(2), 223–254 (2002)
3. 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)
4. 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 (2004)
5. Navigli, R., Velardi, P.: Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(7), 1075–1086 (2005)