Electronic Commerce Research and Applications 28 (2018) 244–260

Contents lists available at ScienceDirect

Electronic Commerce Research and Applications

journal homepage: www.elsevier.com/locate/ecra

An event-extraction approach for business analysis from online Chinese news

Songqiao Han [1], Xiaoling Hao [1,*], Hailiang Huang [1]

School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, China
Shanghai Key Laboratory of Financial Information Technology, Shanghai, China

ARTICLE INFO

Article history:
Received 31 August 2017
Received in revised form 20 February 2018
Accepted 20 February 2018
Available online 23 February 2018

Keywords:
Business events
Business intelligence
Chinese text analytics
Event extraction
Explanatory econometrics
Machine learning models
Natural language processing
Online news
Patterns
Word embedding

ABSTRACT

Extracting events from business news aids users to perceive market trends, be aware of competitors' strategies, and to make valuable investment decisions. Prior research lacks event extraction in the area of business and event based business analysis, especially in Chinese language. We propose a novel business event-extraction approach integrating patterns, machine learning models and word embedding technology in deep learning, which is applied to extract events from online Chinese news. Word embedding and a semantic lexicon are utilized to extend an event trigger dictionary with high accuracy. Then the trigger features in the dictionary are introduced into a machine learning classification algorithm to implement more refined event-type recognition. Based on a scalable pattern tree, the event type that is discovered is used to find the best-suited pattern for extracting event elements from online news. Experimental results show the effectiveness of the proposed approach. In addition, empirical studies demonstrate the practical value of extracted events, especially in finding the relationships between news events and excess returns for stock, and analyzing industry trends based on events in China.

1. Introduction

E-commerce has become a rapidly expanding area in China, and has led to a significant increase in Internet-based commercial transactions. As a result, firms are pursuing more valuable development and investment opportunities. Internet news is a very important information resource for perceiving the ever-changing market trends and making far sighted business decisions. This genre of text has been one of the best resources for studying business trends with large potential commercial value. News of large investments and cross-border e-commerce trade often result in news that may impact e-commerce trends and the future directions of the related enterprises. We focus on business events, such as mergers & acquisitions, investment and product development, directly related to firms that may affect corporation decision-making or social policy-making. However, extracting event information from a large amount of free text is still a challenge (Ahn, 2006; Hogenboom et al., 2016; Nguyen et al., 2016).

Event extraction is a key technique in *nature language processing* (NLP), which aims to discover event triggers with specific types and their arguments from unstructured text, and save them in a structured format. Extracte- event analysis has been effectively applied in personalized news recommendations (Frasincar et al., 2009), risk analysis applications (Capet et al., 2008), especially in the medical domain, for example, to find protein–protein interactions and pathways (Kilicoglu and Bergler, 2009; Ananiadou et al., 2010). But in business domains, this is still a lack of effective event-extraction approaches (Arendarenko and Kakkonen, 2012; Wang et al., 2016) due to some problems that still have to be solved.

First, previous event-extraction techniques have only been only aimed at general events rather than business events. For example, *automatic content extraction* (ACE), has been studied in a research program for developing advanced information extraction technologies. An annual conference has been convened by the U.S. National Institute of Standards (NIST) on this. The ACE Consortium (2009) has defined eight event types (e.g., Life, Movement, Transaction, Business, etc.), with 33 event subtypes (e.g., the event type Business has 4 event subtypes, including Start-org, Merge-org, Declare-bankruptcy and End-org), but the 4 event subtypes of Business are too coarse for business analysis.

* Corresponding author.
E-mail address: hxling@mail.shufe.edu.cn (X. Hao).
[1] The three authors make equal contribution to this paper and are listed in alphabetical order.

https://doi.org/10.1016/j.elerap.2018.02.006
1567-4223/

Second, other existing event-extraction approaches, including *machine learning* (ML) and pattern-based approaches, have hardly been able to achieve acceptable performance for business analysis applications. The ML approaches have not achieved very high-precision performance estimates, due to the lack of annotated corpus content for news and documents. Meanwhile, the pattern-based approaches have not achieved very high-recall performance. Recall indicates the coverage of the approach, and has been limited due to incomplete event patterns and incomplete dictionaries. And third, earlier approaches have been limited to evaluating the accuracy performances of the algorithms or models, but were not evaluated and applied in real business domains.

The present research offers two main contributions. From the viewpoint of event extraction, we present a fine-grained, domain-oriented event-extraction approach based on patterns, ML algorithms and word embedding technology. We first define a *business event taxonomy*, including more types and subtypes of business events compared to prior research. It can describe firm-level behavior at a more-detailed level. Then word embedding technology in deep learning is used to extend the trigger dictionary, which is the most important component in the pattern method. We combine ML classification algorithms and additional knowledge via an extended trigger dictionary, to improve the performance of event-type recognition.

Next, after obtaining the event type of a news text item, the corresponding event pattern is selected to extract the relevant elements of events from online news, such as event triggers and company names. Compared to most previous approaches that used only one of the three methods, our approach integrates all three into a unified framework, which can improve both precision and recall performance.

For event analysis, we make another contribution by exploiting the extracted events to analyze business applications, such as stock market investment strategies and industry analysis. Specifically, for stock markets, we combine stock technical indicators with business event indicators to analyze the excess rate of return. We will report on how the types of events have different and significant impacts on the excess rate of return. In addition, through industry analysis, news events can be applied to reflect the status of different developments in each industry, such as profitability, acquisitions, corporate restructuring, and product transformation.

2. Literature review

We study two main research issues, event-extraction approaches and business event analysis. We review the major literature in the two areas. The event-extraction approaches can be mainly classified into ML methods, knowledge-based methods and hybrid methods (Hogenboom et al., 2016).

ML methods in event extraction mainly have used classification algorithms, such as decision trees, the *naïve Bayes* (NB) technique, *support vector machines* (SVM), Random Forest (RF), AdaBoost and neural networks (Hall et al., 2009; Han et al., 2011), to recognize event elements, such as subjects, triggers and objects, in an event sentence. In doing so, the most important task has been to select appropriate classification features. Then, a diverse set of strategies have been exploited to convert classification clues, such as sequences and sparse trees, into feature vectors. For example, Ahn (2006) used the lexical features (e.g., part-of-speech tags), syntactic features (e.g., dependency features), and external knowledge features (e.g., WordNet) to extract event elements from event sentences. Further, Ji and Grishman (2008), Liao and Grishman (2010) and Hong et al. (2011) respectively tried to use discourses, documents, and cross-document features to improve the performance of information extraction. But the main challenge of the ML approaches has been the lack of annotated event corpus content, which has resulted in the low accuracy of event extraction.

Recently, deep learning has become a very popular ML method which has be widely applied to a variety of signal and information processing tasks (LeCun et al., 2015). The basis of applying *deep learning* to solve natural language processing tasks is to obtain distributed representations of words, through word embedding, from large amounts of unlabeled text data. Word2Vec (Mikolov et al., 2013), a well-known word embedding tool, uses deep-learning technology to represent a term as a vector, thereby offering support for the calculation of semantic relatedness between terms. Theoretically, Word2Vec can find a set of words with high semantic relatedness for a specific seed word, but our experiments show that such a set of words contains many noise words, including irrelevant words or antonyms. Our goal in this article, as a result, is to present a *synonym recognition algorithm* to improve the performance of Word2Vec.

Based on word embedding, two typical deep-learning models, *convolutional neural networks* (CNN) (Chen et al., 2015) and *recurrent neural networks* (RNN) (Nguyen et al., 2016), were used to capture more sentence-level semantic features for event identification. However, compared to ML models, deep-learning models need a larger amount of annotated event text to train, which is not easy to acquire.

Knowledge-driven event-extraction methods often use predefined patterns to express expert knowledge rules. There are two types of patterns that can be applied to natural language corpora for event extraction, especially *lexicon-syntactic patterns* (Nishihara et al., 2009; Hung et al., 2010) and *lexicon-semantic patterns* (Xu et al., 2006; Hogenboom et al., 2013). The former patterns are a combination of lexical representations and syntactic information, and are easy to use and maintain, but lack rule flexibility and adaptability. The latter patterns are more expressive, and combine lexical representations with both syntactic and semantic information, but require more expertise due to their complexity.

Compared to ML methods, pattern-based methods are able to obtain higher precision. This is one of reasons why they are popular in industry, but low recall implies that there are a lot of events that get missed. To improve recall performance, there are two major research directions: how to construct relatively complete pattern repositories, and also to build trigger dictionaries using automatic or semi-automatic approaches rather than manual approaches. In pattern learning, distant supervision (Mintz et al., 2009) has been shown to be effective to automatically learn more relation expressions or patterns from the corpus according to the known knowledge, relations or events (Gupta and Manning, 2014). In trigger learning, existing trigger word extension approaches find the synonyms and hyponyms of the seed triggers, according to synonym word lists and semantic lexicons (Liu and Strzalkowski, 2012), such as Wordnet (Miller, 1995). For Chinese language, the Chinese trigger words were extended using a thesaurus (Qin et al., 2010), a Chinese-English parallel corpus (Ji, 2009), or a semantic-driven joint model (Li et al., 2016). But the trigger-extension approaches often lead to lots of unrelated words or antonyms added to trigger dictionaries, which decreases the performance of processes, such as event-type recognition and event-argument extraction.

Hybrid event-extraction methods combine ML methods and pattern-based methods together. *Knowledge-based event-extraction patterns* can be learned by applying ML techniques, such as *conditional random fields* (CRF) and SVM (Jungermann and Morik, 2008; Björne et al., 2010). On the other hand, the ML approaches are improved using results from knowledge based approaches. For example, part-of speech tagging can be improved by adding domain knowledge in form of ontology (Lee et al., 2003). We utilize the basic idea of hybrid methods. Compared to previous

approaches, we integrate three methods, including word embedding technology, ML methods and pattern methods, to obtain better performances of event extraction in the conditions of a large amount of unannotated text and a small amount of annotated event text. Such data requirements are easy to meet in reality.

In business event analysis, most previous studies focused on news event influences in stock markets. For example, Bali et al. (2009) found that the unusual news events increase the level of investor disagreement about firms' fundamental values. Nuij et al. (2014) combined news events and stock technical indicators to research changes in stock excess returns. Feuerriegel et al. (2016) analyzed the impact of event topics in the news on the stock price. Also, Tafti et al. (2016) studied real-time relationships between chatter on Twitter and the stock trading volume of 96 firms listed in the NASDAQ 100 Index. In addition, Kim et al. (2016) used text processing to study social sentiment and its links to mobile phone-based trading in the KOSDAQ and KOSPI stock markets of the Korean Stock Exchange (KRX). The above research mainly studied the U.S. and Korean stock markets, while we focus on event-based business analysis in the Chinese stock market. The two markets obviously have different characteristics, microstructures, and mechanisms. Moreover, we utilized extracted business events to analyze industry trends which, to our best knowledge, has not been reported in the literature till now.

3. Event-extraction approach

3.1. Event-extraction framework

Our research presents a fine-grained, domain-oriented event-extraction framework, which can be used to extract more types and elements of business events from massive news documents. See Fig. 1. It consists of four phases in pipeline form, including trigger dictionary construction, event-type recognition, pattern-based event extraction, and event analysis and applications.

In the first phase of event extraction, triggers should be discovered from an event sentence according to a *trigger dictionary*. An *event trigger* is the key word that most clearly expresses an event's occurrence. It is generally a verb or a noun, such as "invest" or "investment." We propose a semi-automatic approach to construct the trigger dictionary, where domain experts manually find seed triggers for different types of events, and then automatically extend the original trigger dictionary through a *trigger-extension algorithm* based on word embedding. In the second phase, the event type of a news document can be recognized using ML classification algorithms and the trigger dictionary.

In the third phase, according to the recognized event type, the best suitable pattern is selected from an event pattern repository to extract the corresponding event elements with the aid of dictionaries for entities, tenses and results. The entity dictionary can be constructed by imported existing entity names, such as firm names. For the unseen entity names, a named entity recognition algorithm is used to discover them. The extracted events are stored in an event repository. Finally, a large number of events can be analyzed analysis for stock and bond markets, industry and enterprise development trends, and supply chain processes by using the well-known statistical and ML models.

To explain the complicated processes of the event-extraction framework, we give an example shown in Fig. 2. From a list of business news, an event related to "*entering a new field*" can be found in a news title. For example, "*enter Internet healthcare, but failed to accomplish cross-border mergers & acquisitions*," The analyst can obtain the trigger "*enter*" from the "*entering a new field*" event. Then a synonym recognition algorithm can be used to find more synonyms (e.g., "*participate in*") of "*enter*" and add them to the corresponding trigger dictionary.

Next, the system will scan more news titles to find those that include the words within the trigger dictionary. For example, news of "*Blue whale No.1 of CIMC participated in the trial mining of combustible ice*" will be discovered since it includes the trigger "*participate in*." But it is not clear whether this news belongs to the "*entering a new field*" event. So the content of this news will be input to an ML model to determine its event type for "*entering a new field*." Thus a suitable event pattern will be selected from a pattern repository, and applied to extract other elements using entity dictionaries or a named entity recognition algorithm, including the firm name "*CIMC*," a new field name "*combustible ice*," "*present* tense," the result of "*success*," and the date of "*May 8, 2017*." The extracted event is then stored in the event repository. Finally, the event repository can be easily leveraged for the analysis of business and industry settings.

3.2. Trigger-dictionary construction

3.2.1. Trigger-word semantic representation based on word embedding

In event extraction, a trigger word is the most important element of an event, and it denotes event type. But an event can be expressed using different triggers in different styles, which means that there various triggers will exist for one event. Given the same number of event mentions, there are 30% more triggers in Chinese than in English (Li et al., 2014). This leads to lower recall performance for event extraction in Chinese compared to English. So we aim to construct a relatively complete Chinese trigger dictionary to improve recall performance.

Given a business-event type, its triggers usually will have similar semantics. This motivates us to use word embedding technology in deep learning to discover the synonyms of seed triggers for building a trigger dictionary. In natural language processing, the first important task of deep learning is how to represent a word in semantics. We utilize Word2Vec, a well-known tool that implements a word-embedding algorithm. It describes words' meaning

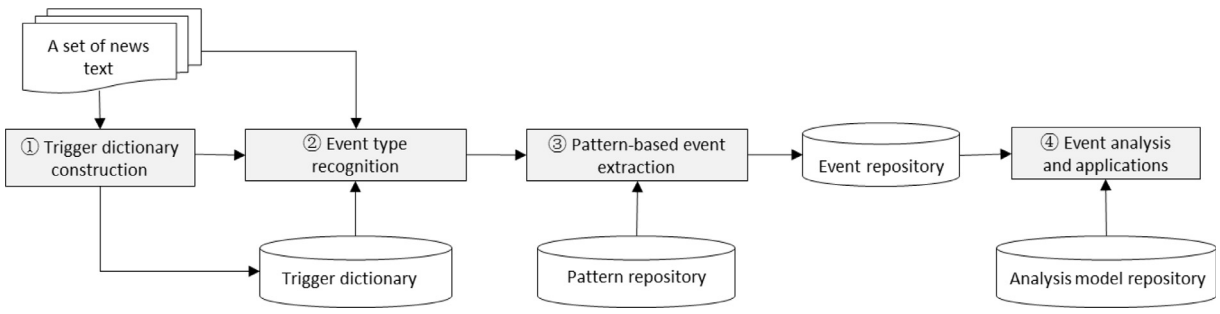

Fig. 1. A fine-grained, domain-oriented event-extraction framework.

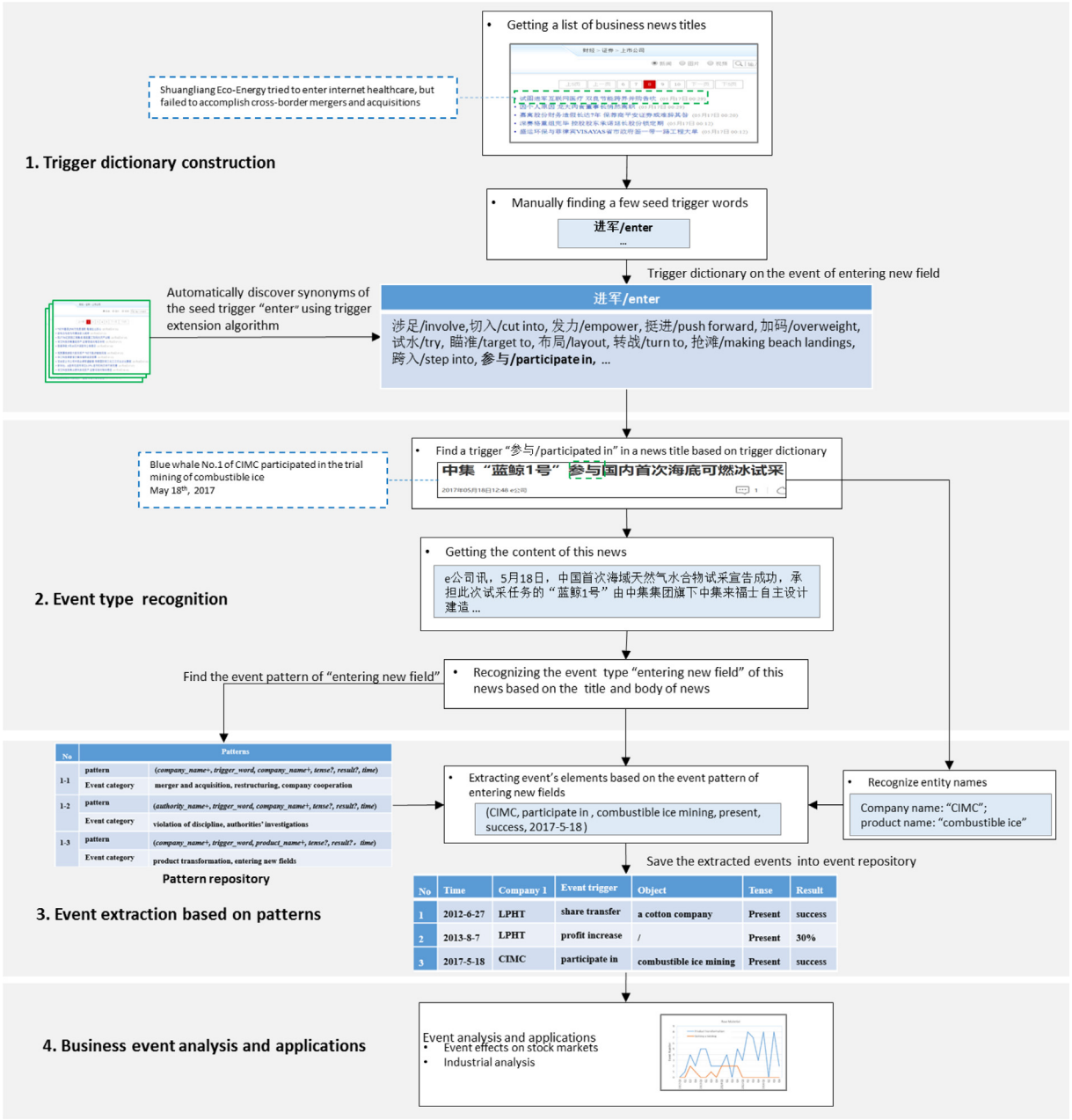

Fig. 2. An example of business event extraction and event analysis.

based on the hypothesis that the words in similar contexts have similar meanings. Word2Vec can adopt two model architectures, a *continuous bag-of-words* (CBOW) *model* and *continuous Skip-Gram model*, to learn a vector representation of a word. The CBOW model predicts a word based on its context in sentences, and the skip-gram model uses the center word to predict the surrounding words (see Fig. 3). In our experimental environments, we find that the skip-gram model performs better than CBOW. So we will focus on the Skip-Gram model.

In the skip-gram model, each word corresponds to a unique vector used as features to predict the surrounding words. Given a word sequence $D = \{w_1, \ldots, w_N\}$, the objective of Skip-Gram is to maximize the average log probability:

$$\mathcal{L}(D) = \frac{1}{M}\sum_{i=1}^{M}\sum_{-k\leqslant c\leqslant k, c\neq 0} \log \Pr(w_{i+c}|w_i) \tag{1}$$

where k is the context size of a target word. Skip-Gram formulates the probability $Pr(w_c|w_i)$ using a softmax function as follows:

$$\Pr(\mathrm{w}_c|\mathrm{w}_i) = \frac{\exp(\mathrm{w}_c \cdot \mathrm{w}_i)}{\sum_{w_i \in W} \exp(\mathrm{w}_c \cdot \mathrm{w}_i)} \tag{2}$$

where w_i and w_c are respectively the vector representations of target word w_i and context word w_c, with W representing the word vocabulary.

Then a word can be represented as a vector with a specified number of dimensions, such as 50 or 100. Then the relatedness of the two words can be obtained by calculating the similarity between the two corresponding vectors. The similarity of two words w_i and w_j can be measured with the inner product of their word vectors.

$$S(w_i, w_j) = w_i \cdot w_j \tag{3}$$

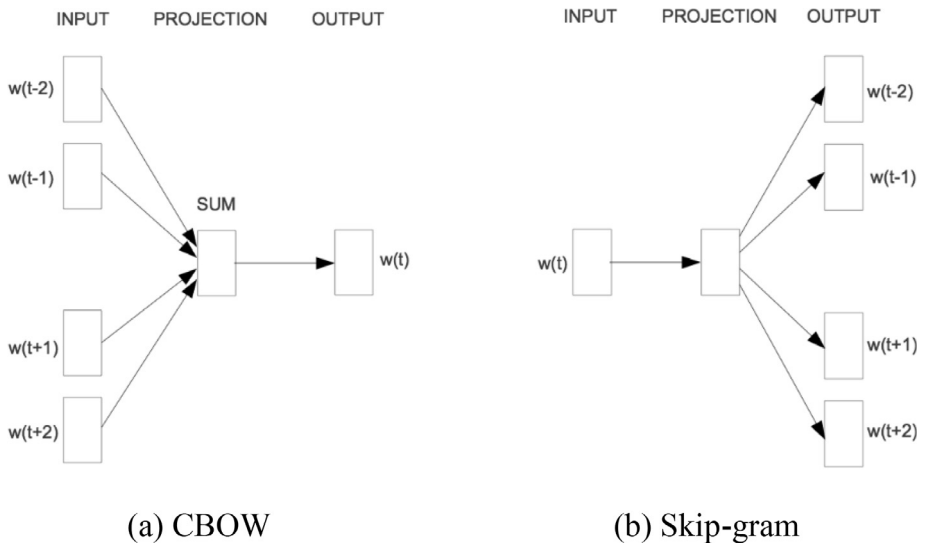

Fig. 3. Two methods of word embedding.

3.2.2. Trigger word extension algorithm

Theoretically, the Word2Vec tool can obtain a set of high-relatedness words of a seed word. However, in our experiments, we observed that most of words in the discovered word set have similar semantics – as synonyms or near-synonyms, but there still are some unrelated words and even antonyms of the word. For example, given the seed word "*increase the shares*" in Chinese, Word2Vec discovers some dissimilar words, such as the codes of firm shares, person names, and even antonyms like "*reduce the shares*," because these words have similar contexts to the seed word in the training corpus. Thus, we should try to discard the noise words from the discovered similar word set. The non-synonym words can be classified into two types: irrelative words and antonyms in semantics. For example, in the product research event, given a seed word "*place bets on*," the Word2Vec algorithm finds a word set, such as "*target at*, *stack*, *withdraw* and *superman*," as shown in Fig. 4, where the distance between two nodes indicate the degree of relatedness between them.

To remove the irrelevant words from the candidate synonyms discovered by the word-embedding algorithm (Step 1 in Fig. 4), we present a triangle relation algorithm (Step 2 in Fig. 4). In general, if there are two words that are similar to the seed word, they are also similar in semantics to each other. That is, in word vector space, three nodes denoting similar meanings form a triangle, where the length of each edge is not bigger than a threshold. Otherwise, at least one edge between two nodes has a longer length than the threshold. Then the irrelevant words can be removed according to the triangle relations for words. In Fig. 4, the node of the seed word "*place bets on*" has four neighbor nodes with short distances. Each edge of both triangles, ΔABC and ΔBCD, is short, but both edges AE of ΔACE and DE of ΔCDE are longer than the other edge, which means that the semantics of "*superman*" are different from other words in the set. The distance between "*place bets on*" and "*superman*" is short only because the two words appear in the similar contexts in the corpus. So "*superman*" is considered as a noise word and removed from the candidate synonyms.

The triangle relation algorithm can filter irrelevant words, but cannot effectively distinguish antonyms from synonyms for a word because. This is because both antonyms and synonyms have simi-

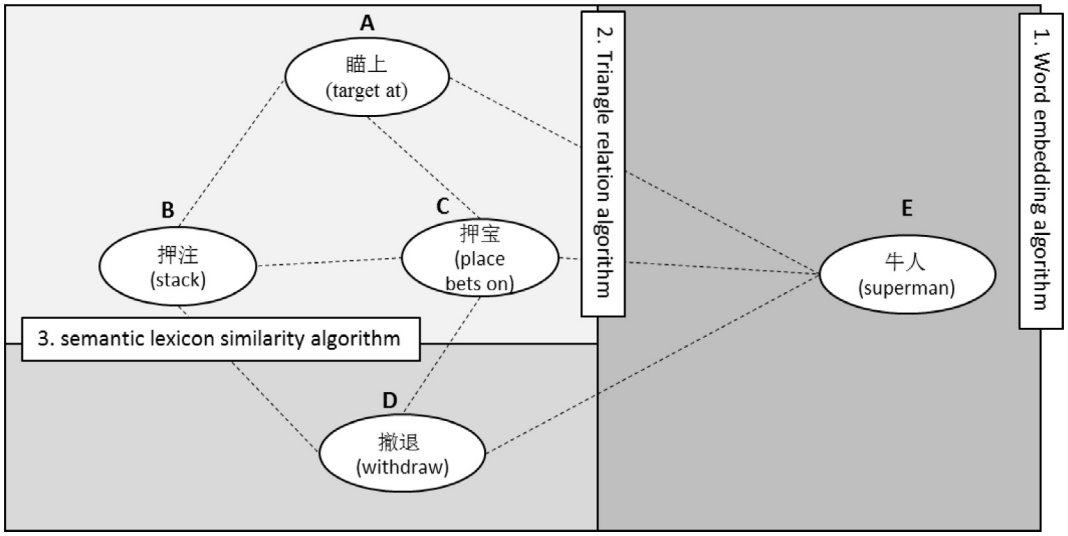

Fig. 4. Illustration of a synonym refinement process of a seed trigger.

lar contexts in most cases. For example, in two news sentences, "*the shareholders buy the shares* and *the shareholders sell the shares*," since the antonym pair, "*buy*" and "*sell*," have similar contexts, Word2Vec obtains high relatedness between them. Thus, to filter the antonyms, we exploited a semantic lexicon to calculate the similarity between two words according to their shortest distances in a concept tree. Hownet (Dong et al., 2006) is a well-known Chinese semantic knowledge base, and it describes inter-conceptual and inter-attribute relationships between concepts. We use the semantic similarity calculation methods (Zhu et al., 2006) of Hownet to obtain the similarity between any two words (Step 3 in Fig. 4). The antonyms pairs usually have low similarity scores according to which the antonyms can be filter from the candidate synonyms.

The overall synonym recognition process is shown in Fig. 5. First, news text is divided into words using a Chinese word segmentation tool, such as Jieba. Then Word2Vec is utilized to obtain a word vector of each word. In the word vector space, we can calculate the Euclidean distance between each two words to indicate their semantic relatedness and similarity to the seed triggers, and then obtain the related words with noise corresponding to the trigger words. The triangle relation algorithm works to remove the irrelevant words, and the semantic lexicon similarity algorithm removes the antonyms. This allows us to obtain a set of synonyms for the trigger words. Through the synonym recognition process, we can implement an automatic trigger dictionary extension.

3.3. Fine-grained event-type recognition

3.3.1. Business event taxonomy

Automatic content extraction (ACE) define four subtypes of business events which cannot satisfy the requirements for business analysis. Based on Hogenboom et al. (2013), we define a taxonomy of business events for business analysis in this research containing 8 event types and 16 event subtypes (see Table 1). The business event taxonomy can be modified and extended according to future needs.

3.3.2. Business event-type recognition algorithm

Before extracting the event elements, we first identify the event types involved in a news article. In most cases, there is a one-to-one correspondence between event types and trigger words. Then, the event type can be recognized according to the trigger word. For example, the news item "*Alibaba Group Holding Limited, the largest e-commerce company in China, purchases AutoNavi Holdings Ltd*" means that there is a mergers & acquisitions event due to the trigger word "*purchase*."

Table 1
Business event taxonomy.

No	Event types	Event subtypes
1	Product transformation	Product transformation, Win bidding
2	Equity change	Shareholding increase, Shareholding decrease
3	Share price movement	Share price rise, Share price fall, Stock resumption, Stock suspension
4	Acquisition & restructuring	Merger & acquisitions, Restructuring
5	Personnel changes	Resignation, Take office
6	Violation of discipline	Violation of discipline
7	Financial status	Profit, Debt
8	Refinance	Refinance

However, the trigger dictionary extension tends to createå two problems. The first problem is that more noise trigger words may be included in the dictionary, which may make false decisions when recognizing event type only according to trigger words. The second problem is that one trigger word may be corresponding to multiple event types or there are multiple trigger words in a sentence. For instance, both the news "*Blue whale No.1 of CIMC participated in the trial mining of combustible ice* and the news "*CIMC participated in debt restructuring*" contain the trigger "*participate in*," but they denote different types of events, "*entering a new field*" and "*restructuring*," respectively. Thus we need use a text classification algorithm to identify event types according to trigger words, titles and the contents of the news.

The *text classification problem* is defined as follows. There are a set of training documents $\mathcal{D} = \{X_1, \ldots, X_N\}$, and each document is labeled with a class value drawn from a set of k different discrete values indexed by $\{1,\ldots,k\}$. In this research, *class value* is an event type and k is the number of event types. The training data are used to train a classification model, which relates the features in the underlying document to one of the class labels. For a given test document for which the class is unknown, the classification model is used to predict a class label for it.

In text classification, a major task is feature selection. A type for documents refers to the documents belonging to the same event type. The *term frequency-inverse document frequency* (TF-IDF) (Salton, 1989) is often used to extract distinguishable words as classification features among the documents. In general, the problem of event-type recognition involves multiple classifications (k-classification) based on feature words discovered by TF-IDF. That is, the k-classification model should classify news documents in a test set into k candidate event types. The performance of the k-classification model is often less than that of the binary classifica-

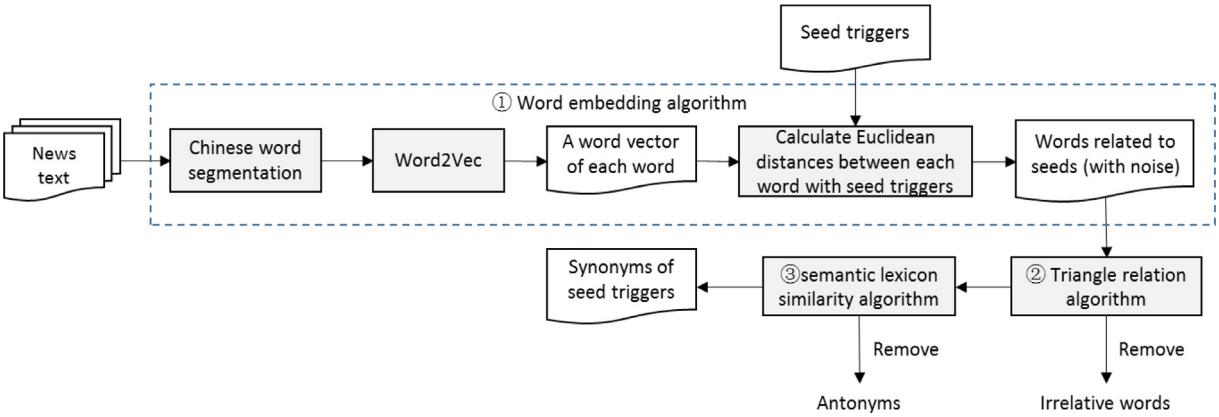

Fig. 5. Synonym recognition process for the seed triggers.

tion model, which only predicts two candidate classes. So if a news item involves multiple events, we can only extract the major event from the news document.

In event-type recognition, we find that the trigger word is a vital indicator of an event type. Traditional k-classification model only considers the feature words discovered by TF-IDF, but does not exploit the information associated with the trigger words. So we present a novel event-type recognition algorithm that considers both trigger words and TF-IDF feature words. The algorithm includes these steps:

(1) From a given news title, one or more trigger words are found according to the trigger dictionary. If the title does not contain any trigger words, it is removed.
(2) The TF-IDF value of each term in the news corpus is calculated using:

$$tfidf_{i,j} = tf_{i,j} \times idf_i \tag{4}$$

where $tf_{i,j}$ denotes the frequency of term t_i in a type of documents d_j, and idf_i represents how important term t_i is. The term frequency $tf_{i,j}$ can be calculated with:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{5}$$

where $n_{i,j}$ denotes the number of times that term t_i appears in the type of documents d_j, and $\sum_k n_{k,j}$ denotes the total number of terms in documents d_j, so the inverse document frequency idf_i can be calculated via:

$$idf_i = log \frac{|D|}{|\{j : t_i \in d_j\}|} \tag{6}$$

where $|D|$ is the number of document types in the new corpus, and $|\{j : t_i \in d_j\}|$ is the number of document types with the term t_i.
(3) The TF-IDF feature words can be obtained by selecting the top k terms in terms of TF-IDF values, where the value of parameter k can be assigned in the experiments. The reason is that the top k terms are words that are more likely to be able to classify event types than other terms, which reduces the dimensionality of the features and improve classification performance.
(4) The event features can be obtained by selecting the corresponding table of event types and trigger words according to the trigger words in the news article titles. Suppose there are m trigger words which correspond to n types of events. Then n event features are denoted by the identifications of these events.
(5) The classification models includes event features and TF-IDF features. In the training set, these features are utilized to train N event classification models, where N equals the number of event types, using well-known classification algorithms, including naive-Bayes, SVM, AdaBoost and RF. Each event classifier is a binary classification model for which the training set is divided into two types according to whether the news text belongs to a particular type of event.
(6) In the test set, a news document containing n types of event triggers can be predicted using n binary classifiers corresponding to n types of candidate events. Then we can use a one-vs.-the-rest algorithm to classify the n types of events, to obtain an event type.

The proposed algorithm has two advantages. The first is to transform the problem of N event-type classification to the problem of n classification, where $n \ll N$, and usually $n = 1$ (for binary classification), which makes the problem easier to solve. The second is to add event types as event features to the TF-IDF features, which introduces more distinguishable features. These advantages can improve the performance of event-type recognition.

The flow diagram of the algorithm is shown in Fig. 6, where an event-trigger table stores all of the triggers for each type of event.

The ML models use the triggers found in the titles of the news articles and the TF-IDF features in their contents to classify an event type. We note that, compared with traditional ML event-extraction approaches, the proposed algorithm only needs a largely smaller amount of annotated text data to train an ML model: first, our model only recognizes event types while the traditional approaches need to recognize more event elements, such as event types, entities, tenses and results; second, we only need to annotate the event types in a corpus, while traditional approaches need to annotate all of the event elements; third, we introduce new knowledge – event triggers – to help with training the ML models, so less training data are needed compared with the traditional approaches.

3.4. Pattern-based event extraction

3.4.1. Pattern construction

In event extraction, the patterns are used to extract event elements from a sentence. With an increase in the event types, more patterns may be need to be constructed. To unify these patterns, we present a hierarchical pattern structure to manage the patterns in a scalable way. This structure is a pattern tree with three layers, including root patterns, abstract patterns, and event patterns. They can be modified to adapt to different applications. The first layer is the root of the pattern tree. The second layer consists of abstract patterns, which means groups of children nodes, for the event nodes. The last layer includes the event patterns used to extract various types of events. With a pattern tree, a pattern can be easily found using a traversal algorithm We use three abstract and six event patterns (see Fig. 7).

This pattern tree introduces the idea of inheritance from object-oriented programming, which adds stability to the upper-level patterns and flexibility to the lower-level patterns. Thus, a new pattern added in the tree not only reuses the abilities of its parent node, but also creates new abilities. This can support the management, maintenance and update of patterns.

The form of the root pattern is:

- Pattern 0 (*entity_name*, trigger_word, entity_name*, tense?, result?, time*)

where the first *entity_name* denotes entity name as subject, and the second one represents entity name as object; the asterisk "*" indicates zero or more entity names; *trigger_word* indicates a type of event; and *tense* is the event's state, such as the past, present, or future state. In addition, *result* represents the result of the event, such as success or failure; the question mark "?" indicates zero or one occurrence of tense and the result of the event; and *time* represents the time when the event occurred.

The abstract patterns aim to classify the event patterns, but cannot be directly used to extract events. Table 2 shows the abstract patterns that we use in this research. In the abstract patterns, the entity names include company names (e.g., JD.Com, Longping High-tech), authority names (e.g., State-Owned Assets Supervision & Administration Commission), product names (e.g., cross-border electronic business platform), person names (e.g., "Wang Yawei"), and position titles (e.g., CEO), etc. The entity names can be divided into known and unknown entities. The *known entities*, such as the

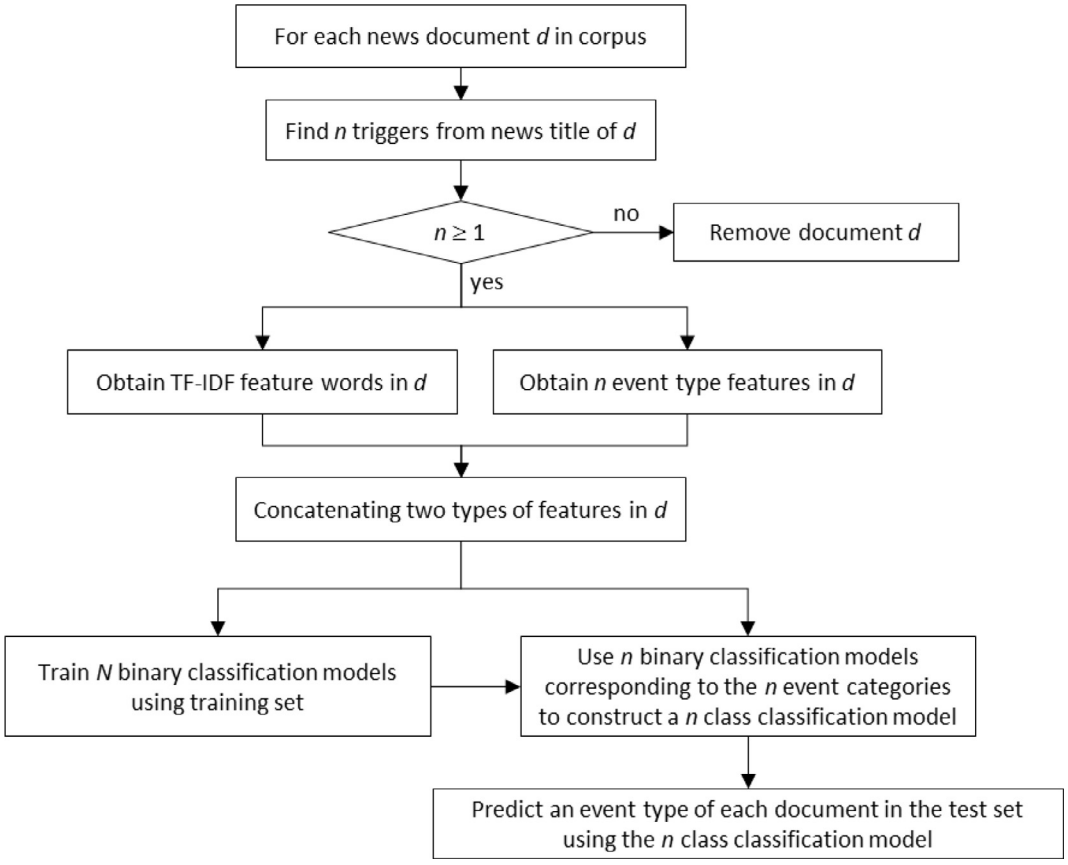

Fig. 6. Event-type recognition algorithm.

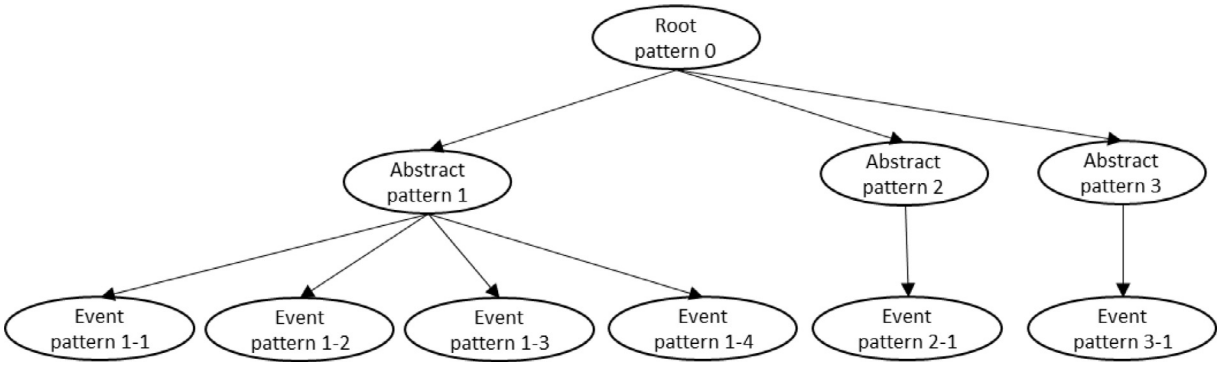

Fig. 7. A pattern tree.

Table 2
Abstract patterns.

No	Abstract Patterns	Basic Forms
1	(*entity_name+, trigger_word, entity_name+, tense?, result?,time*)	subject + T + object
2	(*entity_name+, trigger_word, tense?, result?, time*)	subject + T
3	(*trigger_word, entity_name+, tense?, result?, time*)	T + object

Note: ("+" denotes one or more entity names, "T" denotes trigger).

names of listed companies, authority names, some product names, are stored in the corresponding entity dictionaries. The unknown or new entities, such as the names of unlisted enterprises and new products, are recognized by named-entity recognition algorithm. There has been much research on English and Chinese in this area (Nadeau and Sekine, 2007; dos Santos and Guimaraes, 2015; Shen et al., 2007).

Event patterns are concrete, and can be utilized to extract the corresponding event arguments. The named entities are also classified into various types, such as companies, authorities, and products, which lead to different event patterns. Six business-event patterns related to the companies cover the event types used in this research (see Table 3). They are stored in an event pattern repository.

When the event type of a document is obtained by using an event-type recognition algorithm, we should select one or more patterns suitable for the event type using a correspondence table between event types and event patterns. This table can be constructed by experts according to domain knowledge. Table 4 shows the correspondence table that we used.

Table 3
Event patterns.

No	Event Patterns	Basic Forms
1-1	(*company_name+, trigger_word, company_name+, tense?, result?, time*)	company + T + company
1-2	(*authority_name+, trigger_word, company_name+, tense?, result?, time*)	authority + T + company
1-3	(*company_name+, trigger_word, product_name+, tense?, result?, time*)	company + T + product
1-4	(*person_name+, trigger_word, company_name+, tense?, result?, time*)	person + T + company
2-1	(*company_name+, trigger_word, tense?, result?, time*)	company + T
3-1	(*trigger_word, company_name+, tense?, result?, time*)	T + company

Table 4
Correspondence table between event types and event patterns.

Pattern No.	Event Types
1-1	Mergers & acquisitions, restructuring, company cooperation
1-2	Violation of discipline, authorities' investigations
1-3	Product transformation, entering new fields
1-4	Resignation, take office
2-1	Profit, debt, share price rise, share price fall, shareholding increase, shareholding decrease, violation of discipline, refinance
3-1	Mergers & acquisitionr, restructuring, violation of discipline, investigations

3.4.2. Event extraction based on patterns

Based on the proposed algorithms and dictionaries, we used event patterns to extract the event triggers and arguments, according to the process shown in Fig. 8.

(1) Given a news document, the candidate trigger words are extracted from the news title based on the trigger dictionary.
(2) According to the titles, triggers and contents of the news, the event type is recognized by the event-type recognition algorithm. Then the trigger word corresponding to the type of event is selected from the candidate trigger words.
(3) The candidate patterns are selected from the correspondence table between the event types and patterns according to the recognized event type. For a given event, there may be more than one corresponding pattern. For example, the event type, "mergers & acquisitions" corresponds to patterns 1-1 and 3-1 in Table 3.
(4) The arguments of the event are extracted based on the patterns. Besides the trigger words, the event arguments, such as entities, tenses and results, can be extracted from the news title based on the corresponding dictionaries, such as entity dictionary, tense dictionary and result dictionary. A tense dictionary contains tense words, such as past, present and future. A result dictionary contains result words, such as success, failure and rumor. Since the two dictionaries are relatively simple, they can be easily built manually.
(5) In the case of more than one pattern corresponding to the event, each candidate pattern is applied to the news title to extract the elements of the event. The pattern that extracts the most event elements is selected as best, and its event-extraction result is considered as the final result. For a new entity that is not in the entity dictionaries, we use a named-entity recognition algorithm (Nadeau and Sekine, 2007; Santos and Guimaraes, 2015) to recognize

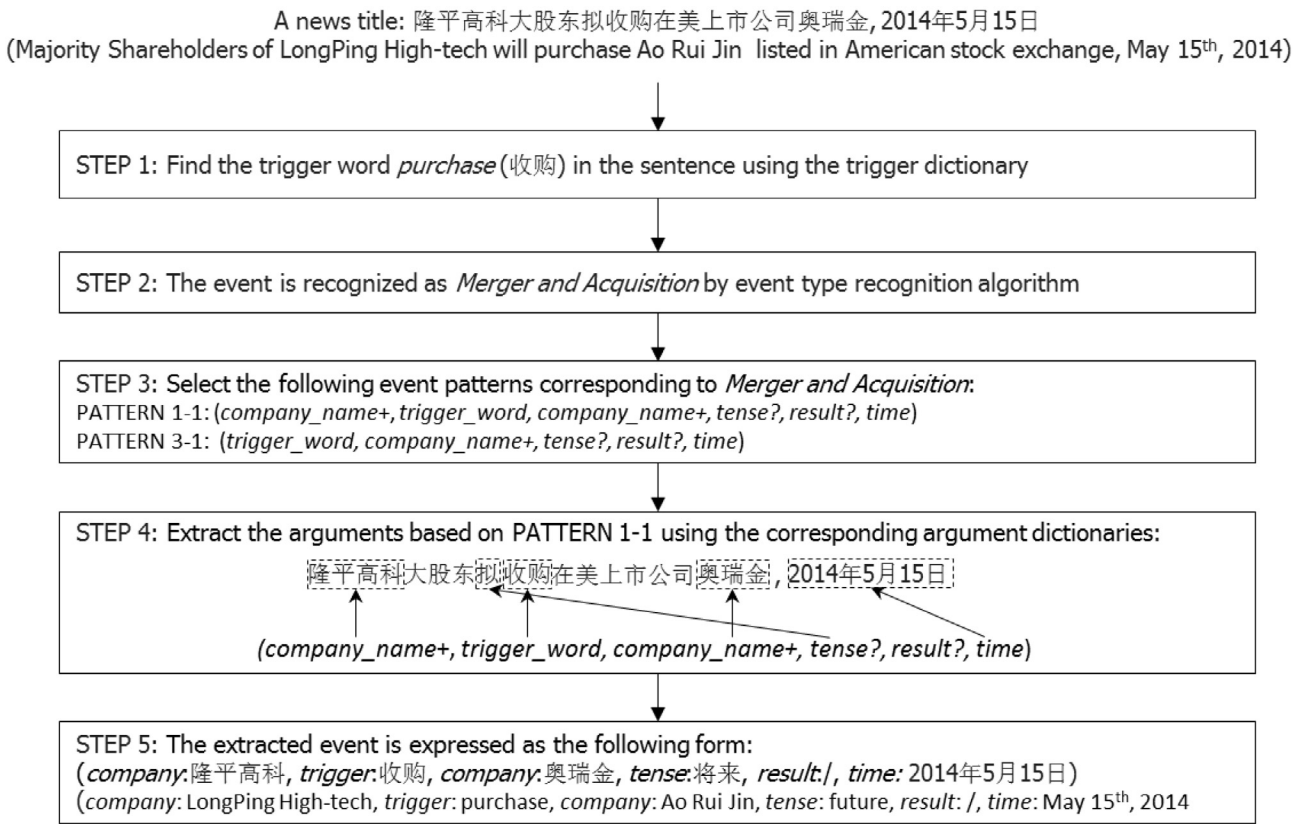

Fig. 8. An example of event-extraction process.

the entity and its type. In addition, new organization names in Chinese can be recognized using the named-entity recognition tool presented by Shen et al. (2007).

(6) Finally, the trigger, type and arguments of the discovered event are stored in an event repository.

Evaluations of the event-extraction approach. We have conducted a series of experiments to evaluate the effectiveness of our proposed approach. The experiments are classified into three groups which correspond to three phases in the event-extraction framework, as shown in Fig. 1. The first group in the series evaluates the performance of the trigger-extension algorithm in the first phase. The second one evaluates the performance of the event-type recognition algorithm in the second phase. The last aims assesses the proposed event-argument extraction approach in the third phase, and the event-extraction approach as a whole.

4.1. Dataset and evaluation metrics

In our experiments, we used three datasets to make performance evaluations. The first dataset contains business news text from Sina.com during the period of 5 years (January 1, 2012 to December 31, 2016). This dataset does not need to be annotated since it aims to train the Word2Vec model with the trigger-extension algorithm. The second dataset is news text from Sina.com during one month (December 1, 2016 to December 31, 2016). In that dataset, 1500 news items are annotated with event triggers, event types and argument types. This dataset was used to train and evaluate the classifiers in the event-type recognition algorithm, and to the evaluate performance of the event-extraction approaches. The third dataset is the ACE 2005 Chinese corpus, which was annotated by ACE and is used compare the performance of the proposed approach and other event-extraction approaches.

The standard evaluation metrics for ML models, especially in the natural language processing, information retrieval, and event-extraction fields (Reeve and Han, 2005; Kim et al., 2009), are *precision* (*P*), *recall* (*R*) and *F-measure* (*F*), hereafter referred to only by Zhuang et al. (2006), Ananiadou et al. (2006). *P* measures the proportion of true events identified out of all extracted events. *R* measures the proportion of true events identified out of all true events out there. It measures the coverage of the method. And *F* represents the harmonic means of precision and recall. It balances them. The technical definitions of the three metrics are:

$$P = \frac{Number\ of\ correctly\ extracted\ events}{Number\ of\ all\ extracted\ events} \tag{7}$$

$$R = \frac{Number\ of\ correctly\ extracted\ events}{Total\ number\ of\ true\ events} \tag{8}$$

$$F = \frac{2 \times P \times R}{P + R} \tag{9}$$

4.2. Evaluations of the trigger-extension algorithm

We used the first dataset to train a Word2Vec model. Based on the relatedness of the words calculated by Word2Vec, we applied the proposed synonym-recognition algorithm to obtain the synonyms of the seed triggers. The parameters and their values from the training based on Word2Vec are shown in Table 5.

For each type of event, we first manually found six seed triggers. Our trigger-extension algorithm was applied to obtain synonyms of the seed triggers with the first dataset. Fig. 9 shows that the count of the discovered synonyms falls dramatically with an increase in similarity. We also found that words with similarity equal to or greater than 0.5 have relatively similar meanings, although there are different synonym counts for different event types. Thus, we choose 0.5 as the threshold value for word similarity.

Through refining the results of Word2Vec, we implemented the proposed trigger dictionary extension algorithm and compared its performance with the other two approaches, a well-known Chinese synonym thesaurus (Tongyici CiLin Extension), and the original Word2Vec tool. Table 6 shows the synonyms of the two seed triggers, "*transform*" and "*place bets on*," discovered by the three approaches. The table notes that the real synonyms were found manually, denoted by the bold font. From the experimental results, we observe that most of words discovered by the Chinese synonym thesaurus had no similar meanings to the seed words in the business domain.

Moreover, the original Word2Vec tool discovered a lot of related or similar words of the seed words, but there also were some noise words. For example, the seed word "*place bets on*" is totally different from "*Wang YaWei*," a person's name. As we stated, our approach can obtain more accurate synonyms than Word2Vec due to the proposed noise filtering method using the triangle relation algorithm and the semantic-lexicon similarity algorithm. As result, given 96 seed trigger words in 16 event subtypes, the Chinese synonym thesaurus approach obtained 1514 candidate synonyms with an accuracy of 12%. For the same word set, the original Word2Vec tool obtained 1835 candidate synonyms with an accuracy of 63%. Our approach, in contrast, yielded 1359 candidate synonyms with an accuracy of 82%. Our approach also can be easily adapted to other domains, so long as it is trained using text data in the corresponding domains.

4.3. Evaluations of event-type recognition algorithm

In this group of experiments, we use the second data set containing 1500 Chinese news articles. The data set is divided into

Table 5
The parameters of training command using Word2Vec.

Parameters	Comments	Values
-train	Name of input file	train_text.txt
-output	Name of output file	vectors.bin
-cbow	Choice of training model, 0: Skip-gram model, 1: CBOW model	0
-size	Dimension of vectors	200
-window	Size of training window	8
-negative	Choice of training method, 0: hierarchical softmax method, >1: the number for negative specifying how many "noise words" should be drawn in negative sampling method.	25
-hs	Choice of using hierarchical softmax, 0: not used, 1: used	0
-sample	Threshold of sampling	1e−4
-threads	Number of running threads	20
-binaray	Mode of storage 1: Common format, 1: Binary format	1
-iter	Number of iterations over the corpus	15

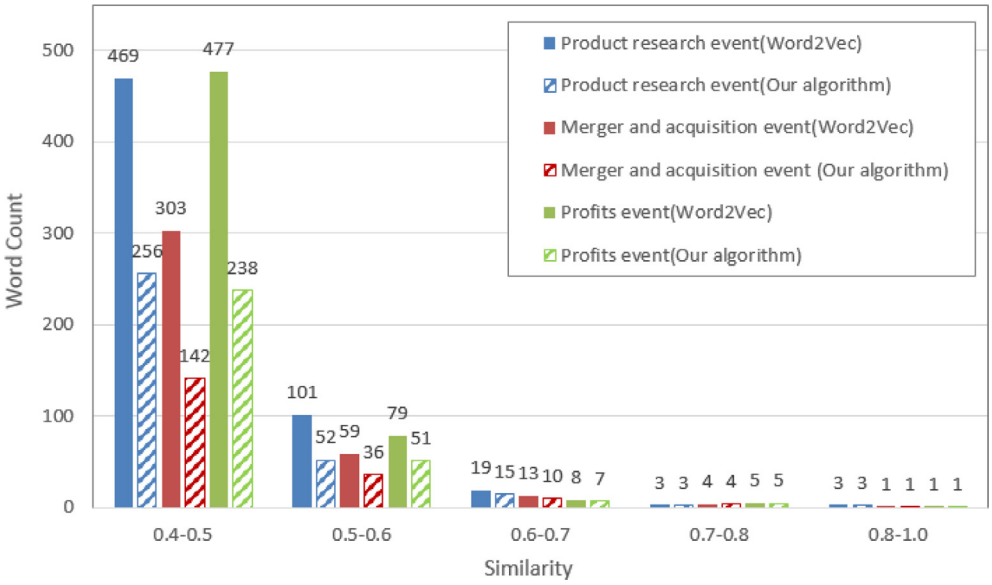

Fig. 9. The relationships between word count and word similarity.

Table 6
Illustration of performance comparisons for three trigger-extension approaches.

Seed Word #1	转型/transform
Chinese synonym thesaurus	换季/shifting of seasons, 改装/repack, 改期/change time, 改道/change route, 改嫁/remarry, 改判/change judgment, 改写/rewrite, 改编/reedit, 改组/reorganize, 改制/restructuring, 改称/rename, 改种/replant, 改型/transform, 改用/reuse, 换气/change air, 换人/change person, 换向/change direction, 农转非/change from agricultural to non-agricultural, 扭亏增盈/ turn, . . .
Word2Vec	多元化/diversification, 变革/reform, 转变/change, 传统产业/traditional industry, 传统/tradition, 升级/upgrade, 大力发展/rapidly develop, 发展/develop, 突围/break through, 跨界/cross border, 布局/layout, 主业/major work, 方向/direction, 服务化/service oriented, 触网/touch network, 国际化/internationalize, 结构调整/structure adjust, 尝试/try, 拥抱/embrace, 转向/transfer
Our approach	变革/reform, 转变/change, 升级/upgrade, 大力发展/rapidly develop,发展/develop, 突围/break through, 跨界/cross border, 布局/layout, 服务化/service oriented, 国际化/internationalize, 结构调整/structure adjust, 转向/transfer to, 整合/integrate, 网络化/networking, 迈进/enter, 高端化/develop high-end, 进军/enter, 融合/reconcile, 谋求/seek
Seed Word #2	押宝/place bets on
Chinese synonym thesaurus	/
Word2Vec	押注/stack, 豪赌/unrestrained gambling, 潜伏/hide, 潜入/dive into, 下注/chip in, 杀入/force to enter, 梦碎/break dream, 伏击/ambush, 折戟/break the spear, 瞄上/target at, 掘金/dig for gold, 牛人/superman, 火中取栗/pull chestnuts out of the fire, 埋伏/ambush, 重组股/reorganization share, 失利/loss, 画饼/draw cakes, 王亚伟/Wang Yawei, 玩转/play, 淘金/wash gold, 扎堆/flock together, 相中/be satisfied in, 撤退/withdraw, 最爱/the favorite, 牛散/super retail investor, 抄底/buy at bottom, 围猎/hunt, 出击/attack, 钟情/love, 翻身/turn over
Our approach	押注/stack, 豪赌/unrestrained gambling, 潜伏/hide, 潜入/dive into, 下注/chip in, 杀入/force to enter, 伏击/ambush, 瞄上/target at, 掘金/dig for gold, 埋伏/ambush, 淘金/wash gold, 相中/be satisfied in, 抄底/buy at bottom, 围猎/hunt, 出击/attack

the training set with 1200 news items and a test set with 300 items. The 10-fold cross-validation method was used to train and evaluate the performance of these models, as shown in the first two columns in Table 7. To determine the value of parameter *k*, meaning that the top *k* terms for TF-IDF are selected as TF-IDF features, in the event recognition algorithm, we conducted a series of experiments where *k* was assigned 100, 200, . . ., 1000, respectively. We found that when *k* is equal to 500, the best performance was obtained. So *k* is assigned as 500 in the following experiments.

We exploited three well-known ML classification models, including SVM, RF and AdaBoost, to recognize 16 sub-types of events. From the experimental results, the AdaBoost model obtained the best performance for *P*, *R* and *F*.

In addition, the proposed event-type recognition algorithm was applied to the same data set. It uses the ML models – SVM, RF and AdaBoost &Triggers – to recognize the different event types. Its results are shown in the last two columns in Table 7. Our approach obtained better performance compared than its counterparts. The AdaBoost & Triggers model achieved the best performance: precision, *P*, of 92.7%; recall, *R*, of 87.0%; and a combination of precision and recall, *F*, of 89.8%. The AdaBoost & Triggers model was used in our experiments.

4.4. Evaluations of event-extraction approaches

The proposed event-extraction approach is evaluated with the second and third datasets. The third dataset, the ACE dataset, is a well-known open event-extraction dataset, and it allows our approach to be compared with state-of-the-art event-extraction approaches. The second dataset is a Chinese business-text dataset, which allows our approach to be evaluated to that it guarantees good performance in real business news-analysis environments.

4.4.1. Evaluations of event-extraction approaches for ACE dataset

We compare our approach with five representative event-extraction approaches. They obtained good performance using different technologies on the ACE dataset. The five approaches are follows: (1) a *cross-event extract approach* (Liao and Grishman, 2010)

Table 7
Performance comparisons of event-type recognition algorithms.

Models	*P*	*R*	*F*	Models	*P*	*R*	*F*
SVM	74.0	67.4	70.5	SVM & Triggers	89.8	81.2	85.3
RF	76.2	69.2	72.5	RF & Triggers	91.5	84.1	87.6
AdaBoost	77.3	74.3	75.8	AdaBoost & Triggers	**92.7**	**87.0**	**89.8**

that uses document-level information to improve the performance of a sentence-level event-extraction system; (2) a *cross-entity extract approach* (Hong et al., 2011), which extracts events by using cross-entity inference; (3) a *joint event extraction approach* (Li et al., 2013) that extracts events based on structure prediction and extract the triggers and arguments jointly; (4) the *DMCNN approach* (Chen et al., 2015) that uses a word representation model to capture meaningful semantic regularities for words and adopts a method based on a dynamic multi-pooling convolutional neural network; and (5) the *S-CNN approach* (Zhang et al., 2016) is a joint event-extraction method which uses skip-window convolutional neural networks to extract global structured features, and uses RNNs models to extract triggers and arguments simultaneously.

The performance of our approach versus the performance of the other five approaches are shown in Table 8. The column on trigger identification means the performance of identifying the triggers, such as "acquire," while the column on event-type recognition indicates the performance of recognizing event types, such as "mergers & acquisitions." The columns on event-argument identification and argument-role identification represent the performance of identifying event arguments, such as "Longping High-tech," and their roles, such as "company name as a subject."

In trigger identification, our approach is superior to all the other approaches in terms of precision. This is because our approach uses a pattern-based method to improve precision compared to the ML classification methods in the other five approaches. Recall for our approach is lower than DMsCNN and S-CNN due to our trigger dictionary being incomplete in other domains, except for the business domain. But the overall result of our approach based on combined precision and recall, *F*, achieved the best performance.

For event-type recognition, our approach significantly outperformed the other approaches on all three of the performance metrics. Our approach obtained precision, *P*, of 82.7%, recall, *R*, of 73.1%, and a related *F*1-measure of 77.6%; both deep-learning approaches, DMCNN and S-CNN, only obtained an *F*1-measure of 69.1%. This is the reason our approach integrates an ML method, a pattern-based method and a deep-learning model. The first two improve the precision and the last improves the recall. But these five approaches only use ML or deep-learning models, and these do not use human knowledge from the pattern-based methods. Only when a large amount of annotated corpus is used for training with ML or deep-learning models can they achieve better performance. In reality, however, the lacking availability of annotated text corpus has limited the performance of ML and deep-learning models.

In event-argument recognition and argument-role identification, since our approach focuses on the arguments extraction of business events, we use the proposed approach to identify arguments and their roles mainly for business events in the ACE dataset. From the experimental results, we observe that our approach leads to recall *R* comparable to the other five approaches, but it obtains the best precision, *P*, and combined precision and recall performance, *F*, among all of the approaches.

Overall, since our approach integrates pattern-based, ML method and word-embedding methods, it can fully exploit the advantages of each. The high-precision performance originates from the pattern-base and ML methods, and its recall performance comes from the word-embedding method.

4.4.2. Evaluations of event-extraction approaches for our dataset

This group of experiments evaluate event-extraction approaches for the second Chinese business news dataset containing 1500 Chinese news stories. In the experiments, we use three event-extraction approaches, including the synonym thesaurus approach, Word2Vec and the proposed approach.

Like most research on event extraction, we evaluate each step of the event extraction process, including trigger identification, event-type recognition, event-argument identification, and argument-role identification. Most importantly, we evaluate the overall performance of event-extraction approaches, which is very important for event-based applications in real environments, but is neglected by previous research. Table 9 shows the phase performance for event extraction in the 2nd to the 5th columns, and the overall performance of event extraction in the last column.

For the synonym thesaurus approach, its precision and recall are the lowest among the three approaches. This is because a synonym thesaurus introduces a lot of noise words, which decrease its performance. The Word2Vec approach can obtain more relevant triggers for the business domain compared to the synonym thesaurus approach, so it achieves better performances than the synonym thesaurus approach. The Word2Vec approach also obtains the highest recall value among these approaches since it adds a large number of new trigger words into the dictionary. Our approach can effectively filter most of irrelevant words or antonyms from the trigger dictionary constructed by Word2Vec. This leads to a significant increase in precision and a slight decrease in recall. If the approach on the ACE dataset and our dataset are compared, the performance on our dataset is better than on the ACE dataset. This is because there are more complete dictionaries

Table 8
Performance comparisons of event-extraction approaches on ACE dataset.

Approaches	Trigger identification			Event-type recognition			Event-argument identification			Argument-role identification		
Metrics	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Cross-event extraction	N/A	N/A	N/A	68.7	68.9	68.8	50.9	49.7	50.3	45.1	44.1	44.6
Cross-entity extraction	N/A	N/A	N/A	72.9	64.3	68.3	53.4	**52.9**	53.1	51.6	45.5	48.3
Joint event extraction	76.9	65.0	70.4	73.7	62.3	67.5	69.8	47.9	56.8	64.7	44.4	52.7
DMCNN	80.4	67.7	73.5	75.6	63.6	69.1	68.8	51.9	59.1	62.2	**46.9**	53.5
S-CNN	78.1	**71.8**	74.8	74.1	64.8	69.1	69.2	50.8	58.6	63.3	45.8	53.1
Our approach	**89.5**	65.9	**75.9**	**82.7**	**73.1**	**77.6**	**78.9**	51.5	**62.3**	**71.3**	45.1	**55.3**

Table 9
Performance comparisons of event-extraction approaches on our dataset.

Approach	Trigger identification			Event-type recognition			Event-argument identification			Argument-role identification			Overall event extraction		
Metrics	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
Synonym_thesaurus	84.0	70.3	76.5	82.7	65.4	73.0	82.0	76.7	79.2	79.8	70.9	75.1	78.1	56.9	65.8
Word2Vec	87.2	**91.5**	89.3	85.3	**88.2**	86.7	88.5	**82.1**	85.2	87.6	**78.8**	83.0	81.2	**74.1**	77.5
Our approach	**94.1**	90.8	**92.4**	**92.7**	87.0	**89.8**	**91.2**	80.9	**85.7**	**90.5**	78.6	**84.1**	**87.9**	73.7	**80.2**

for business news than other domains' news. This also indicates that the dictionary extension based on word embedding plays an important role in improving its final performance for event extraction. Overall, our approach achieved a good balance between precision and recall, and the best *F*-measure among all approaches.

5. Business analysis based on events

In this section, we study the effects of the extracted events on the stock markets, and analyze various industry development trends also.

5.1. Econometric analysis

Using our proposed approach, we extracted a sample of 14,522 events from the news text data, and assessed the empirical relationship study between events and stock markets, to answer whether there is a relationship between the excess rate of return and event type in the stock market.

Empirical model and variables. The excess rate of return (or abnormal rate of return) is a one of most widely used measures of risk-adjusted performance in stock markets, which is also a commonly used indicator in event study. The *excess rate of return* is defined as the difference between the actual and required rate of return on equity, derived from the *capital asset pricing model* (CAPM):

$$\Delta r_i = r_i - E(r_i) \tag{10}$$

The *actual rate of return on equity*, r_i, is calculated using the following formula.

$$r_i = \frac{P_{i,1} - P_{i,0} + div_{i,0}}{P_i} \tag{11}$$

where $P_{i,0}$ is the price of the share at the beginning of the time period, $P_{i,1}$ is the price of the share at the end of the time period, $div_{i,0}$ are dividends received for the period, all for company *i*. The *required rate of return*, $E(r_i)$, is determined by CAPM also:

$$E(r_i) = r_f + \beta_i \times (r_m - r_f) \tag{12}$$

where r_f is the *risk-free rate of return*, r_m is the *market rate of return and* β_i is the *standardized measure for equity risk company i.* β_i is calculated as follows.

$$\beta_i = \frac{\mathrm{cov}(r_i, r_m)}{Var(r_m)} \tag{13}$$

In the experiments, we selected the excess rate of return on the day when the event occurred as the dependent variable and selected event type as an independent variable. Since there are 16 event subtypes, we also constructed 15 dummy variables to avoid multicollinearity, with one of them as the base case. Also, we introduced some technical indices, including opening price, closing price, an exponential moving average of 5 days (EMA_5), and a simple moving average of 20 days (SMA_20), as control variables since such indices may cause differences in the excess rate of returns.

The *exponential moving average* (EMA) identifies trends by using a short and long-term average. The short-term average is assigned at 5 days and the long-term average at 20 days. Then EMA can be computed with:

$$E_i = \frac{2}{N+1} \times (P_i - E_{i-1}) + E_{i-1} \tag{14}$$

where P_i represents the price on day 5, and N is the number of days. When the short-term average moves upwards and crosses the long-term average, a buy signal is created. When the short-term average moves downwards, and crosses the long-term average, a sell signal is created.

The *simple moving average* (SMA) is an average of the last 20 days of the price movement of a stock, based on:

$$M_i = \frac{\sum_{i=1}^{N} P_i}{N} \tag{15}$$

where P_i represents the price on day *i*. When the price crosses the moving average upwards, a buy signal is created, while when the price crosses the moving average downwards, a sell signal is created.

The regression model for the causal relationship between event type and the excess rate of returns is:

$$Exs\_Rtn = \alpha_0 + \sum_{i=1}^{} \sum_{j=1} \alpha_{i-j} Ctgy_{i-j} + \beta_1 OpnPrc + \beta_2 ClsPrc + \beta_3 EMA\_5 + \beta_4 SMA\_20 + \varepsilon \tag{16}$$

The variables are defined in Table 10.

In addition, escriptive statistics are useful for describing basic features of data. The summary statistics for the variables and measures of the data is shown in Table 11.

Table 10
Definition of variables.

Variable type	Variable Name	Variable Construction/ Definition Data
Dependent Variables	*Exs_Rtn*	The excess rate of return of on the day of the event
Independent Variables	$Ctgy_{1\text{-}1}$	Product transformation
	$Ctgy_{1\text{-}2}$	Win the bidding
	$Ctgy_{2\text{-}1}$	Shareholding decrease
	$Ctgy_{2\text{-}2}$	Shareholding increase
	$Ctgy_{3\text{-}1}$	Share price rise
	$Ctgy_{3\text{-}2}$	Share price fall
	$Ctgy_{3\text{-}3}$	Trade resumption
	$Ctgy_{3\text{-}4}$	Trade suspension
	$Ctgy_{4\text{-}1}$	Mergers & acquisitions
	$Ctgy_{4\text{-}2}$	Restructuring
	$Ctgy_{5\text{-}1}$	Resignation
	$Ctgy_{5\text{-}2}$	Take office
	$Ctgy_{6\text{-}1}$	Violation of discipline
	$Ctgy_{7\text{-}1}$	Debt
	$Ctgy_{7\text{-}2}$	Profit
Control Variables	*OpnPrc*	Opening price on the day of event occurrence
	ClsPrc	Closing price on the day of event occurrence
	SMA_20	A simple moving average of 20 days
	EMA_5	Exponential moving average of 5 days

Table 11
Summary Statistics.

Variables	Min.	Max.	Mean.	Std. Dev.
$Ctgy_{1\text{-}1}$	0	1	0.040	0.184
$Ctgy_{1\text{-}2}$	0	1	0.020	0.125
$Ctgy_{2\text{-}1}$	0	1	0.060	0.242
$Ctgy_{2\text{-}2}$	0	1	0.040	0.207
$Ctgy_{3\text{-}1}$	0	1	0.010	0.074
$Ctgy_{3\text{-}2}$	0	1	0.010	0.106
$Ctgy_{3\text{-}3}$	0	1	0.010	0.102
$Ctgy_{3\text{-}4}$	0	1	0.020	0.122
$Ctgy_{4\text{-}1}$	0	1	0.100	0.307
$Ctgy_{4\text{-}2}$	0	1	0.060	0.234
$Ctgy_{5\text{-}1}$	0	1	0.010	0.115
$Ctgy_{5\text{-}2}$	0	1	0.000	0.039
$Ctgy_{6\text{-}1}$	0	1	0.120	0.321
$Ctgy_{7\text{-}1}$	0	1	0.010	0.093
$Ctgy_{7\text{-}2}$	0	1	0.050	0.220
OpnPrc	1.52	337.20	16.94	21.75
ClsPrc	1.53	337.76	16.98	21.82
SMA_20	1.56	323.18	16.79	21.39
EMA_5	1.53	331.37	16.94	21.68
Exs_Rtn	−0.77	6.92	0.003	0.093
#of Obs.	14,521			

Table 12
Results of regression analysis.

Variables	Model 1 Coef (*p*-values)	Model 2 Coef (*p*-values)
$Ctgy_{2\text{-}1}$	-0.007^{**} (0.038)	
$Ctgy_{3\text{-}1}$	-0.042^{***} (0.000)	
$Ctgy_{3\text{-}2}$	0.002^{***} (0.000)	
$Ctgy_{3\text{-}3}$	0.045^{***} (0.000)	0.046^{***} (0.000)
$Ctgy_{3\text{-}4}$	0.031^{***} (0.000)	0.030^{***} (0.000)
$Ctgy_{4\text{-}1}$	0.006^{**} (0.012)	0.006^{***} (0.003)
$Ctgy_{6\text{-}1}$	-0.009^{***} (0.000)	-0.006^{**} (0.022)
OpnPrc		-0.004^{***} (0.000)
ClsPrc		0.032^{***} (0.000)
EMA_5		-0.035^{***} (0.000)
SMA_20		0.007^{***} (0.000)
_cons	0.003^{***} (0.001)	0.003^{***} (0.001)
Adj R^2	0.007	0.149
Obs.	14,521	

Notes: $^{***}p < 0.01$; $^{**}p < 0.05$; $^{*}p < 0.1$.

Main findings. We created two models for explanatory regression analysis, building on our machine-based methods for data analytics, as we have seen in research related to this special issue (Kauffman et al., 2017), and other big data analytics settings, including music track popularity analytics (Ren and Kauffman, 2017) and video-on-demand sampling and purchasing policy analytics (Hoang and Kauffman, 2018). In our research, Model 1 uses only event types as explanatory variables, and Model 2 uses event types and technical indices as explanatory variables. The results are shown in Table 12.

We can see that 0.69% of uncertainty in the excess rate of return can be explained by the event types in Model 1, while 14.9% of uncertainty in the excess rate of return can be explained in Model 2. So adding four technical indices makes the adjusted R^2 increase from 0.69% to 14.9%. It is worth noting that, although the event types have a low R^2 value in Model 1, the technical indices worked better and the events contributed to a higher R^2 in Model 2.

In addition, in Model 1 of Table 12, the variables $Ctgy_{2\text{-}1}$, $Ctgy_{3\text{-}1}$, $Ctgy_{3\text{-}2}$, $Ctgy_{3\text{-}3}$, $Ctgy_{3\text{-}4}$, $Ctgy_{4\text{-}1}$, and $Ctgy_{6\text{-}1}$ all achieved a significance level of 0.05, based on the *t*-tests. In Model 2, the independent variables $Ctgy_{2\text{-}1}$, $Ctgy_{3\text{-}1}$, $Ctgy_{3\text{-}2}$ were no longer significant, while the variables $Ctgy_{3\text{-}3}$, $Ctgy_{3\text{-}4}$, $Ctgy_{4\text{-}1}$, $Ctgy_{6\text{-}1}$ still had significant *p*-values; and they remained significant even after controlling for the effects of the technical indices.

We also found that the estimated coefficients for $Ctgy_{3\text{-}2}$, $Ctgy_{3\text{-}3}$, $Ctgy_{3\text{-}4}$, $Ctgy_{4\text{-}1}$ were positive and significant ($p < 0.01$), which means that such events as "*share price fall*," "*stock resumption*," "*stock suspension*," "*merger & acquisition*" have real impacts on the excess rate of return. The coefficients on the variables of $Ctgy_{2\text{-}1}$, $Ctgy_{3\text{-}1}$, *and* $Ctgy_{6\text{-}1}$ were negative and significant ($p < 0.01$), so such events as "*shareholding decrease*," "*share price rise*," and *violation of discipline*" appear to have been able to reverse the declining excess rate of returns. Meanwhile, the coefficients of the control variables, *EMA_5* and *OpnPrc*, were negative, so the higher the opening price or the greater the exponential moving average of 5 days, the lower the excess rate of return was. (Note: Due to limited space, we only demonstrate our causal analysis between event types and stock markets. More complete causal experiments, such as sentiments for events, are beyond the scope of this research.)

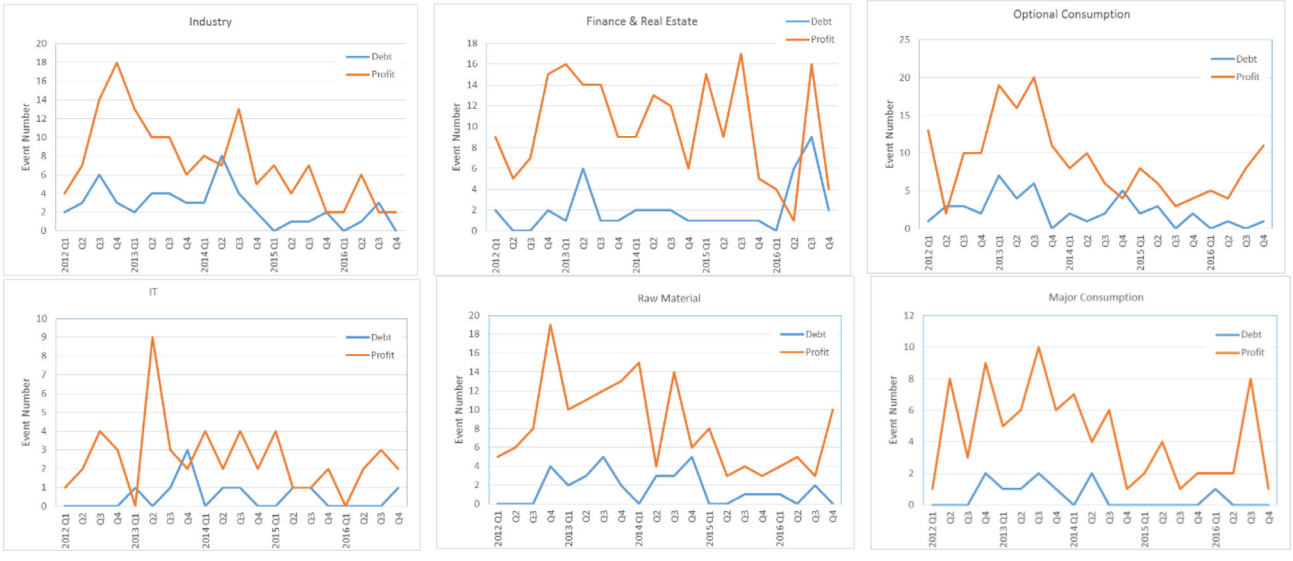

Fig. 10. Profitability of various industries.

5.2. Industry analysis

The extracted events were analyzed over time according to event types and industry types in order to find industry development trends. When company-level events were extracted from the news, the events were classified by the corresponding industries. If an event belonged to a company, the event was classified into the industry which the company belonged to, according to the industry classification for the Chinese stock markets. To illustrate, we selected six industries from the Chinese stock markets, including Industry, Finance & Real Estate, Optional Consumption, IT, Raw Materials, and Major Consumption. We used them to represent the development trends of these industries for: profitability, increasing and decreasing shareholding, acquisition and restructuring, product transformation, and obtaining bids.

Fig. 10 shows that Industry, IT and Energy have had lower profitability from 2012 to now. Meanwhile, Finance & Real Estate has maintained higher profitability during this period, although it has fluctuated. Also the profits of Optional Consumption, Major Consumption and Raw Materials decreased during 2012 and 2nd quarter in 2016, but increased after the 3rd quarter in 2016.

Fig. 11 shows that the number of shareholding-related events in Finance & Real Estate increased recently. The increases in the number of shareholding-related in the Industry and Major Consumption plots are roughly equal to their decreases. And shareholders in the IT and Raw Material sectors have tended to sell their shares.

Fig. 12 indicates the trends of "mergers & acquisitions," and "restructuring" events. The number of acquisition events obviously increased recently in Industry, Finance & Real Estate and Optional Consumption, while other industries did not have more of them. There are more "restructuring" events in the Finance & Real Estate industries.

From Fig. 13, we can see that most industries tried to enter other new fields in 2014, while the number of this type of events

Fig. 11. Increasing and decreasing shareholding.

Fig. 12. Acquisitions and restructuring events.

Fig. 13. Product transformation and getting bidding.

decreased in 2016. Also, more events involving large-scale bidding, especially in industry, occurred in 2012 than in other periods.

With the help of the structured form of the extracted events, we can do interesting business analysis, make decisions, and create applications in a new way. For instance, from a series of business events over time, a firm can rapidly discover and predict the future business trends in an industry. Moreover, we can find the relationships between competition or cooperation between enterprises, and understand the development strategies of other partners or competitors. Furthermore, we can exploit business events to make suitable marketing and advertisements, which is known as event marketing.

6. Conclusions

We have proposed a business-oriented, fine-grained event-extraction approach for Chinese business news. To identify fine-grained event types in business fields, we integrated word embedding technology in a deep-learning, classification algorithm in ML and pattern base approaches together to improve the performance of event extraction. The proposed approach has two main advantages. One is that it only needs a small amount of annotated text corpus, which is in line with real situations. The other is that it can achieve significantly improved performance, which makes it possible for use with event-based business analysis. Our experimental results illustrate the effectiveness of the proposed approach. Extracted events are exploited to support interesting business analysis, which offers a new perspective to make business intelligence by mining massive free text in the age of big data and AI. Also, the application results show great practical value of the proposed approach.

There is a large amount of valuable information hidden in the text data, especially for business development. In the future, we will further improve the proposed event-extraction approach in the following two aspects. One is to extract more types of events, such as industry policies, technology innovations and economic situations, to cover more fields rather than events merely related to companies. The other is to introduce distant supervision technologies to the proposed approach for automatically constructing patterns to replace the manual patterns analyzed in this research.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant No 71401096), Shanghai Science and Technology Innovation Project, China, (Grant No. 16511102900), Humanity and Social Science Foundation of Ministry of Education of China (Grant No. 17YJA630029), and Key Program Grant of State Language Commission of China (Grant No. ZDI135-18). We would like to thank the reviewers for valuable comments and advice.

References

Ahn, D., 2006. The stages of event extraction. Proceedings of the Workshop on Annotating and Reasoning about Time and Events, Association for Computational Linguistics. Springer Verlag, Berlin Heidelberg, Germany.

Ananiadou, S., Kell, D.B., Tsujii, J.I., 2006. Text mining and its potential applications in systems biology. Trends Biotechnol. 24 (12), 571–579.

Ananiadou, S., Pyysalo, S., Tsujii, J.I., Kell, D.B., 2010. Event extraction for systems biology by text mining the literature. Trends Biotechnol. 28 (7), 381–390.

Arendarenko, E., Kakkonen, T., 2012. Ontology-based information and event extraction for business intelligence. In: Proceedings of the 15th Annual Conference on Artificial Intelligence: Methodology, Systems, and Applications. Springer-Verlag, Berlin, Heidelberg, pp. 89–102.

Bali, T.G., Scherbina, A., Tang, Y., 2009. Unusual news events and the cross-section of stock returns. Soc. Sci. Electr. Publ. 62 (4), 1623–1661.

Björne, J., Ginter, F., Pyysalo, S., Tsujii, J.I., Salakoski, T., 2010. Complex event extraction at PubMed scale. Bioinformatics 26 (12), i382–i390.

Capet, P., Delavallade, T., Nakamura, T., Sandor, A., Tarsitano, C., Voyatzi, S., 2008. A risk assessment system with automatic extraction of event types. Intell. Inf. Process. IV, 220–229.

Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 1, Beijing, China, pp. 167–176.

Consortium, L.D., 2009. ACE (Automatic Content Extraction) Chinese annotation guidelines for events.

Dong, Z., Dong, Q., Hao, C., 2006. How Net and the Computation of Meaning. World Scientific, Singapore.

dos Santos, C.N., Guimaraes, V., 2015. Boosting named entity recognition with neural character embeddings. arXiv:1505.05008.

Feuerriegel, S., Ratku, A., Neumann, D., 2016. Analysis of how underlying topics in financial news affect stock prices using latent Dirichlet allocation. In: Sprague, R., Bui, T. (Eds.), Proceedings of the 49th Hawaii International Conference on System Sciences. IEEE Computer Society Press, Washington, DC1081, p. 1072.

Frasincar, F., Borsje, J., Levering, L., 2009. A semantic web-based approach for building personalized news services. Int. J. E-Business Res. 5 (3), 35–53.

Gupta, S., Manning, C.D., 2014. Improved pattern learning for bootstrapped entity extraction. In: Proceedings of the 2014 Conference on Natural Language Learning, Baltimore, MD, pp. 98–108.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18.
Han, J., Pei, J., Kamber, M., 2011. Data Mining: Concepts and Techniques. Morgan Kauffmann, San Francisco, CA.
Hoang, A.P., Kauffman, R.J., 2018. Content sampling, household informedness and the consumption of digital information goods. J. Manage. Inf. Syst. in press.
Hogenboom, F., Frasincar, F., Kaymak, U., De Jong, F., Caron, E., 2016. A survey of event extraction methods from text for decision support systems. Decis. Support Syst. 85, 12–22.
Hogenboom, A., Hogenboom, F., Frasincar, F., Schouten, K., van der Meer, O., 2013. Semantics-based information extraction for detecting economic events. Multimedia Tools Appl. 64 (1), 27–52.
Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., Zhu, Q., 2011. Using cross-entity inference to improve event extraction. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 1, Stroudsburg, PA, pp. 1127–1136.
Hung, S.H., Lin, C.H., Hong, J.S., 2010. Web mining for event-based commonsense knowledge using lexico-syntactic pattern matching and semantic role labeling. Expert Syst. Appl. 37 (1), 341–347.
Ji, H., Grishman, R., 2008. Refining event extraction through cross-document inference. In: Proceedings of the Annual Conference of the Association for Computer Linguistics, Stroudsburg, PA, pp. 254–262.
Ji, H., 2009. Cross-lingual predicate cluster acquisition to improve bilingual event extraction by inductive learning. In: Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics, Association for Computational Linguistics, Stroudsburg, PA, pp. 27–35.
Jungermann, F., Morik, K., 2008. Enhanced services for targeted information retrieval by event extraction and data mining. In: International Conference on Application of Natural Language to Information Systems. Springer, Berlin Heidelberg, Germany, pp. 335–336.
Kauffman, R.J., Kim, K., Lee, S.Y.T., Hoang, A.P., Ren, J., 2017. Combining machine-based and econometrics methods for policy analytics insights. Electr. Res. Appl. 25, 115–140.
Kilicoglu, H., Bergler, S., 2009. Syntactic dependency based heuristics for biological event extraction. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task, Association for Computational Linguistics, Stroudsburg, PA, 119–127.
Kim, J.D., Ohta, T., Pyysalo, S., Kano, Y., Tsujii, J.I., 2009. Overview of biomedical natural language processing: shared task on event extraction. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task, Association for Computational Linguistics, Stroudsburg, PA, pp. 1–9.
Kim, K., Lee, S.T.Y., Kauffman, R.J. 2016. Social sentiment and stock trading via Mobile phones. In: Proceedings of the American Conference on Information Systems, Association for Information Systems, Atlanta, GA.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.
Lee, C.S., Chen, Y.J., Jian, Z.W., 2003. Ontology-based fuzzy event extraction agent for Chinese e-news summarization. Expert Syst. Appl. 25 (3), 431–447.
Li, Q., Ji, H., Huang, L. 2013. Joint event extraction via structured prediction with global features. In: Proceedings of the 51st Annual Meeting of the Association for Computer Linguistics, 1, Sofia, Bulgaria, pp. 73–82.
Li, P., Zhou, G., Zhu, Q., 2016. Semantic based Chinese event triggered word extraction joint model, J. Software, 27(2), 280–294. (李培峰, 周国栋, 朱巧明, 2016. 基于语义的中文事件触发词抽取联合模型.软件学报, 27(2), 280-294.)
Li, P.F., Zhu, Q.M., Zhou, G.D., 2014. Using compositional semantics and discourse consistency to improve Chinese trigger identification. Inf. Process. Manage. 50 (2), 399–415.
Liao S, Grishman R., 2010. Using document level cross-event inference to improve event extraction. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden, pp. 789–797.
Liu, T., Strzalkowski, T., 2012. Bootstrapping events and relations from text. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, pp. 296–305.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinburger (eds.), Proceedings of the Conference on Advances in Neural Information Processing Systems, Lake Tahoe, CA, pp. 3111–3119.
Miller, G.A., 1995. WordNet: a lexical database for English. Commun. ACM 38 (11), 39–41.
Mintz, M., Bills, S., Snow, R., Jurafsky, D., 2009. Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computer Linguistics and the 4th International Joint Conference on Natural Language Processing, Stroudsburg, PA, 2(2), pp. 1003–1011.
Nadeau, D., Sekine, S., 2007. A survey of named entity recognition and classification. Lingvisticae Investigationes 30 (1), 3–26.
Nguyen, T.H., Cho, K., Grishman, R., 2016. Joint event extraction via recurrent neural networks. In: Proceedings of the 16th North American Chapter of the Association of Computer Linguistics Conference, Stroudsburg, PA, pp. 300–309.
Nishihara, Y., Sato, K., Sunayama, W., 2009. Event extraction and visualization for obtaining personal experiences from blogs. In: Salvendy, G., Smith, M.J. (Eds.), Human Interface and the Management of Information: Information and Interaction, Lecture Notes in Computer Science, 5618. Springer, Berlin Heidelberg, Germany, pp. 315–324.
Nuij, W., Milea, V., Hogenboom, F., Frasincar, F., Kaymak, U., 2014. An automated framework for incorporating news into stock trading strategies. IEEE Trans. Knowl. Data Eng. 26 (4), 823–835.
Qin, B., Zhao, Y., Ding, X., Liu, T., Zhai, G., 2010. Event type recognition based on trigger extension. Tsinghua Sci. Technol. 15 (3), 251–258.
Reeve, L., Han, H., 2005. Survey of semantic annotation platforms. ACM Press, New York, pp. 1634–1638.
Ren, J., Kauffman, R.J., 2017. Understanding music track popularity in a social network. In: Proceedings of the 25th European Conference on Information Systems, Association for Information Systems, Atlanta, GA, pp. 374–388.
Salton, G., 1989. Automatic text processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading, MA.
Shen, J.Y., Li, F., Xu, F.Y., Uszkoreit, H., 2007. Recognition of Chinese organization names and abbreviations. Journal of Chinese Information Processing, 21(6), 17–21. (沈嘉懿, 李芳,徐飞玉, Hans Uszkoreit, 2007. 中文组织机构名称与简称的识别. 中文信息学报, 21(6), 17–21.)
Tafti, A., Zotti, R., Jank, W., 2016. Real-time diffusion of information on twitter and the financial markets. PLoS ONE 11 (8), 1–16.
Wang, Y., Ma, H., Lowe, N., Feldman, M., Schmitt, C., 2016. Business event curation: Merging human and automated approaches. In: Proceedings of the AAAI, Workshop on Event Extraction and Synthesis. AAAI Press, Menlo Park, CA, pp. 4272–4273.
Xu, F., Uszkoreit, H., Li, H., 2006. Automatic event and relation detection with seeds of varying complexity. In: Proceedings of the AAAI. AAAI Press, Menlo Park, CA, pp. 12–17.
Zhang, Z.K., Xu W.R., Chen Q.Q., 2016. Joint event extraction based on skip-window convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language, 1, Long Papers, Stroudsburg, PA, pp. 167–176.
Zhu, Y.L., Min, J., Zhou, Y.Q., Huang, X.J., Wu, L.D., 2006. Semantic orientation computing based on HowNet. J. Chin. Inf. Process. 20 (1), 14–20. (朱嫣岚, 闵锦, 周雅倚, 2006. 基于HowNet的词汇语义倾向计算. 中文信息学报, 20(1), 14–20.)
Zhuang, L., Jing, F., Zhu, X.Y., 2006. Movie review mining and summarization. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 43–50. (赵妍妍, 秦兵, 车万翔, 刘挺, 2008. 中文事件抽取技术研究. 中文信息学报, 22(1), 3–8.