Expert Systems with Applications 41 (2014) 4494–4504

Contents lists available at ScienceDirect

Expert Systems with Applications

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

CFinder: An intelligent key concept finder from text for ontology development

Yong-Bin Kang, Pari Delir Haghighi, Frada Burstein *

Faculty of Information Technology, Monash University, 900 Dandenong Rd, Caulfield East 3145, Victoria, Australia

ARTICLE INFO

Keywords:
Key concept extraction
Keyphrase extraction
Domain-specific concept extraction
Ontology development
Ontology learning

ABSTRACT

Key concept extraction is a major step for ontology learning that aims to build an ontology by identifying relevant domain concepts and their semantic relationships from a text corpus. The success of ontology development using key concept extraction strongly relies on the degree of relevance of the key concepts identified. If the identified key concepts are not closely relevant to the domain, the constructed ontology will not be able to correctly and fully represent the domain knowledge. In this paper, we propose a novel method, named *CFinder*, for key concept extraction. Given a text corpus in the target domain, CFinder first extracts noun phrases using their linguistic patterns based on Part-Of-Speech (POS) tags as candidates for key concepts. To calculate the weights (or importance) of these candidates within the domain, CFinder combines their statistical knowledge and domain-specific knowledge indicating their relative importance within the domain. The calculated weights are further enhanced by considering an inner structural pattern of the candidates. The effectiveness of CFinder is evaluated with a recently developed ontology for the domain of 'emergency management for mass gatherings' against the state-of-the-art methods for key concept extraction including—Text2Onto, KP-Miner and Moki. The comparative evaluation results show that CFinder statistically significantly outperforms all the three methods in terms of F-measure and average precision.

1. Introduction

Due to an exponential growth of available information and knowledge, ontologies have become widely exploited in many different domains. Ontologies are typically built to formally conceptualize knowledge in a domain of interest. Their main aim is to provide a shared and common understanding of domain knowledge and promote interoperability between people and many application systems (Chandrasekaran, Josephson, & Benjamins, 1999).

The formal and explicit specifications of ontologies are mainly defined in the form of *concepts* and their *relations* that need to be shared and conceptualized (Wong, Liu, & Bennamoun, 2012). Although there is an increasing demand for building ontologies in different domains, this task is very tedious and complex, and requires a huge amount of effort and domain knowledge from domain experts. To facilitate this task, *ontology learning* has been widely studied and used to build an ontology semi-automatically or automatically. Its aim is to extract concepts and their relations including occasional axioms about the concepts from documents to build an ontology (Wong et al., 2012).

In an ontology, concepts typically represent a set of classes of entities or things within a domain (Noy & mcguinness, 2001). According to prior studies (Jiang & Tan, 2010; Li & Wu, 2006), concepts can be often described by *noun phrases* that are suitable for representing the key information within text documents. A noun phrase means a single noun or a group of words that function together as a noun. However, the main problem under this scheme is that not all the noun phrases can be considered as *domain-specific* concepts and useful for accurately conceptualizing domain knowledge. The reason is that such concepts may contain noise terms or include terms that are too general and common.

Therefore, in ontology learning, a key challenge is how to automatically extract domain-specific *key concepts* that can correctly represent the key information of a corpus of document (s) in a domain of interest. Thus, *key concept extraction* is a primary and the most basic step for ontology learning from text documents (Jiang & Tan, 2010; Wong et al., 2012). If extracted key concepts are non-relevant, an ontology may not fully and correctly represent domain knowledge as such irrelevant concepts can also lead to generating non-relevant relations and axioms.

* Corresponding author. Tel.: +61 3 990 32011; fax: +61 3 990 31077.
E-mail addresses: yongbin.kang@monash.edu (Y.-B. Kang), pari.delirhaghighi@monash.edu (P. Delir Haghighi), frada.burstein@monash.edu (F. Burstein).

http://dx.doi.org/10.1016/j.eswa.2014.01.006
0957-4174/

For key concept extraction, many existing approaches have focused on extracting *keyphrases* (Cimiano & Volker, 2005; Jiang & Tan, 2010; Li & Wu, 2006; Tonelli, Rospocher, Pianta, & Serafini, 2011) from a corpus of documents.[1] Keyphrases are a set of terms, each comprising one or more words, and describe the document with which they are associated conveying the primary information of the document (El-Beltagy & Rafea, 2009; Li & Wu, 2006). These approaches can be categorized as either (1) *machine learning approaches* that require a training corpus of documents (i.e. supervised approaches) (Frank, Paynter, Witten, Gutwin, & Nevill-Manning, 1999; Medelyan & Witten, 2006), (2) *multiple corpus-based approaches* that require corpora of multiple document collections from multiple domains (Jiang & Tan, 2010; Missikoff, Velardi, & Fabriani, 2003; Xu, Kurz, Piskorski, & Schmeier, 2002), (3) *glossary-based approaches* that use author-provided keyphrases or glossary terms (Diederich & Balke, 2007; Novalija, Mladenic, & Bradesko, 2011; Wang, Mamaani Barnaghi, & Bargiela, 2010), or (4) *heuristic-based approaches* that often mix both natural language processing (NLP) techniques and statistical information of phrases extracted from the corpus (Li & Wu, 2006).

In this paper, we propose a novel key concept finder named CFinder that uses a heuristic that combines NLP techniques, statistical knowledge, domain-specific knowledge and an inner structural pattern of terms extracted from a corpus of documents in the target domain. More specifically, CFinder makes use of linguistic patterns to extract key concept candidates considering their POS tags. To calculate the *weights* (or importance) of the candidates within the target domain, CFinder combines statistical and domain-specific knowledge of the candidates. Such weights are further enhanced using an inner structural (i.e. *word-occurrence*) pattern existed within the candidates. CFinder can be categorized into the heuristic-based approaches as it incorporates the above heuristic to identify relevant key concepts.

In our evaluation, we compare CFinder with three state-of-the-art heuristic-based methods for key concept extraction which are Text2Onto[2] (Cimiano & Volker, 2005), KP-Miner[3] (El-Beltagy & Rafea, 2009) and Moki[4] (Tonelli et al., 2011) that are publicly available. The evaluation results show that CFinder significantly outperforms all the three methods at the 99% confidence level in terms of widely used evaluation metrics in Information Retrieval, which are *F-measure* and average precision (Manning, Raghavan, & Schütze, 2008; Wong et al., 2012). For evaluation purposes, we particularly focus on the domain of '*emergency management for mass gatherings*', simply referred to as *DMG* in this paper. In DMG, we evaluate the effectiveness of CFinder using actual concepts in a recently developed ontology, *Domain Ontology for Mass Gatherings* (*DO4MG*) (Delir Haghighi, Burstein, Zaslavsky, & Arbon, 2013). This ontology is publicly available and the source materials it was built upon are fully accessible.

This article is organized as follows. Section 2 presents a comprehensive review of current studies on key concept extraction and highlights its advantageous features of CFinder. Section 3 provides the details of the CFinder approach and its main steps. Section 4 describes the comparative evaluation of CFinder in terms of F-measure and average precision. Section 5 summarizes the findings and results and concludes the paper.

2. Related work

Most existing approaches to key concept extraction can be divided into four categories: (1) machine learning approaches, (2) multiple corpus-based approaches, (3) glossary-based approaches, and (4) heuristic-based approaches. Below, we provide the related work for each of these categories, and highlight how CFinder is distinguished from the existing approaches with its main features.

2.1. Machine learning approaches

KEA (Frank et al., 1999) and KEA++ (Medelyan & Witten, 2006) extract keyphrase candidates from a corpus of documents using NLP techniques, and then use a *model* to identify key concepts by determining which of the candidates are most likely to be key concepts. The model is learned using a *classifier* (i.e. Naive Bayes) from training documents where the author-provided keyphrases are already known. However, the effectiveness of these approaches relies strongly on the quality and amount of the underlying training documents.

2.2. Multiple corpus-based approaches

The multiple corpus-based approaches rely mainly on the exploitation of corpora of *multiple* document collections from *multiple* domains to assign higher weights to more domain-specific terms (Jiang & Tan, 2010; Missikoff et al., 2003; Xu et al., 2002). In common, these approaches first extract noun phrases from a target document using NLP techniques. Then, they use the frequency information of these phrases, which is computed from a corpus in the target domain and another corpus from the contrasting domain (Jiang & Tan, 2010). Alternatively, Missikoff et al. (2003) and Xu et al. (2002) proposed to use corpora of multiple document collections from different domains instead of a corpus from the contrasting domain.

However, a common problem of these approaches is that they require a substantial number of documents from different domains to accurately identify key concepts. Thus, in a relatively less matured domain (or a newly-born domain) that may have a small corpus of relevant documents, these approaches may suffer from a large difference between the size of the corpus and corpora of different domains, thereby leading to a high skewness in terms of the frequency information of the extracted noun phrases among the domains. Eventually, this may often lead to making the overall performance poor.

2.3. Glossary-based approaches

The glossary-based approaches are characterized by the use of author-provided glossary terms in a corpus of documents. For example, Novalija et al. (2011) used a set of glossary terms provided from the corpus as key concepts to extend an existing ontology. However, it is hard to see that all these terms can be always regarded as key concepts, because these terms may also include newly introduced or uncommon but trivial terms as well as too general terms. Thus, sometimes, such terms are seen as non domain-specific, and thus convey not important information of the corpus. Also, if the corpus does not provide glossary terms, this approach cannot extract key concepts.

As another trend, author-provided keyphrases that more frequently appear in the corpus are also adopted as key concepts (Diederich & Balke, 2007; Wang et al., 2010). A problem of this scheme however is that such keyphrases cannot be always seen as key concepts, because these terms can be sometimes too general to be considered as key concepts, thus hardly making them useful as key concepts.

2.4. Heuristic-based approaches

According to our survey, many approaches to key concept extraction can be categorized into the heuristic-based approaches.

[1] Due to their relatedness, we do not distinguish between terms 'key concept' and 'keyphrase', and use them interchangeably.

[2] https://code.google.com/p/text2onto/.

[3] http://www.claes.sci.e.g./coe_wm/kpminer/.

[4] https://moki.fbk.eu/website/index.php.

These approaches depend largely on various heuristics to identify key concepts.

For example, KIP (Li & Wu, 2006) first extracts keyphrase candidates from a corpus of documents using NLP techniques. Then, it uses a heuristic to measure the weights of the candidates through two steps. First, for each candidate, it assigns a weight to each keyword (i.e. a single word parsed from the candidate) based on its frequency in a glossary database that contains pre-defined domain-specific terms. Then, it computes the weight of the candidate by combining its frequency in the corpus and the weights of keywords within the candidate. Finally, the candidates with higher weights are selected as key concepts. However, two limitations of KIP lie in that (1) determining a weight for each keyword is done using training documents (i.e. it requires a training process and thus may be computationally expensive), and (2) the weighting scheme for keyphrase candidates is based on the assumption that a higher weight must be assigned to a candidate whose length in words (i.e. word count) is longer; however this assumption cannot be always correct, since some keywords (i.e. single-word keyphrase) must be also adopted as key concepts (e.g. 'weather' is a concept in the DO4MG ontology).

Text2Onto (Cimiano & Volker, 2005) determines key concepts after extracting keyphrase candidates based on different weighting criteria (i.e. entropy, RTF (Relative Term Frequency), and TF-IDF (Term Frequency-Inverse Document Frequency)) as heuristics for estimating their domain relevance. However, in the entropy and RTF criteria, only using the frequency information may lead to extracting more single-word terms than multi-word terms as key concepts. This is because multi-word terms are less likely to appear than single-word terms in general. To illustrate this, consider two terms 'service' and 'emergency medical service'. In this example, it is highly possible for these weighting criteria (i.e. entropy and RTF) to recognize the term 'service' which can be largely used in many domains as a key concept, even though it is a general term and the latter term is closer to a key concept in the DMG domain. This approach will result in missing many of multi-word terms that need to be considered as key concepts. Also, in the TF-IDF criterion, if a key concept appears frequently due to its importance, it may not be selected since its IDF value tends to be a value of 0. However, in this work, we have observed that very often key concepts are frequently appearing in most of documents in the underlying corpus. The TF-IDF scheme is also adopted to identify key concepts in Chen, Liang, and Pan (2008), Rezgui (2007) and Villaverde, Persson, Godoy, and Amandi (2009).

TextOntoEx (Dahab, Hassan, & Rafea, 2008) uses semantic patterns to identify key concepts and their relations to construct an ontology. A semantic pattern is defined using a user-defined linguistic format to represent a specific meaning. For example, a semantic pattern '<Plant Part><Becomes.Verb><Color>' indicates that any plant parts in a given text and any colors connected to different synonyms of the verb 'becomes' can be considered key concepts. However, a main problem of this approach is that it is very hard and time-consuming to define all semantic patterns manually to identify key concepts for the target domain. Also, it is hard to generalize this approach without human interventions in different domains, as semantic patterns must be differently defined for a given domain by domain experts.

A graph-based key concept extraction approach was proposed in Hou, Ong, Nee, Zhang, and Liu (2011). It attempts to extract key concepts from a corpus using a graph structure, where each node represents a single-word term extracted from the corpus, and each edge represents an associative strength between two nodes where the strength is measured by a total number of occurrences of the nodes together adjacently. For each node, its weight is calculated using its frequency information considering its adjacent nodes. Finally, based on this weighting heuristic, the approach generates a number of key concepts, where each of them is represented as a cluster of adjacent terms. However, this approach prefers to identify single-word terms to represent nodes in a graph, thus tending to miss many important multi-word terms. Also, since each concept is represented as a cluster of adjacent terms, it may be hard to interpret the meaning of each concept clearly and precisely.

Shih, Chen, Chu, and Chen (2011) proposed a key concept extraction technique by considering synonyms of each term extracted from a given corpus. It first determines whether two terms are synonymous with each other by examining the similarity between their respective co-occurring terms, linguistic similarity and semantic similarity. Then, a concept is described by its synonyms, and its weight is measured by its frequency in the corpus. However, since this techniques is based only on frequency information of terms, it leads to the same problems as Text2Onto with the entropy and RTF criteria.

KP-Miner (El-Beltagy & Rafea, 2009) selects keyphrase candidates using statistical and NLP techniques with heuristics (e.g. the first occurrence position of the candidates in each of the corpus). Then, it measures the weights of the candidates using a variant form of the TF-IDF weight, and finally determines keyphrases based on these weights. A recent ontology building system, Moki (Tonelli et al., 2011), uses a keyphrase extractor KX (Pianta & Tonelli, 2010) for key concept extraction. It first extracts n-grams from a corpus of documents, and then finds most important phrases based on different kinds of heuristics (e.g. the first occurrence position of terms, keyphrase length and longer concept boosting) with NLP techniques.

In this paper, we compare CFinder with three heuristic-based approaches—Text2Onto, KP-Miner and Moki—using the ontology, DO4MG, in the domain DMG. In our evaluation, we show that CFinder significantly outperforms these methods in terms of F-measure and average precision.

2.5. CFinder's distinctive features

In the following, we present how CFinder is distinguished from the above approaches and what its advantageous features are:

- CFinder can be considered as an unsupervised approach for key concept extraction in the sense that it does not require training documents prior to learning in contrast to the supervised approaches.
- CFinder does not require a set of corpus documents from multiple domains to identify key concepts compared to the multiple corpus-based approaches. This leads to avoiding the human effort to collect relevant corpora and also being computationally less expensive to compute the weights of key concept candidates. Further, the common problems in the multiple corpus-based approaches such as the high skewness in terms of the frequency information of terms do not occur in CFinder.
- CFinder identifies key concepts by combining various techniques and knowledge sources—NLP techniques, statistical knowledge, domain-specific knowledge and an inner structural pattern of extracted terms. Therefore, unlike the glossary-based approaches, it does not rely only on author-provided keyphrases, and thus being able to find more implicitly important key concepts even if these are not explicitly specified by the authors of the target corpus. Also, CFinder is able to filter out too general or trivial terms through the use of a combination of statistical and domain-specific knowledge even if these were specified as keyphrases by the authors.
- Unlike many of the heuristic-based approaches that are mainly based on both NLP techniques and statistical knowledge, CFinder further leverages domain-specific knowledge and an inner

structural pattern of key concept candidates. This helps to overcome the problems that have been found in the heuristic-based approaches where only the frequency or TF-IDF weights of the candidates are used to measure the weights of the candidates. Also, unlike the approaches that require user-specified semantic patterns (Dahab et al., 2008), CFinder does not rely on any user-specified input. This makes CFinder simpler to use and easily applicable to other domains.

3. CFinder: a novel key concept finder

CFinder consists of three steps to discover key concepts. Its overall procedural steps are outlined in Fig. 1.

First, CFinder identifies key concept candidates using POS tags, linguistic patterns and a synonym table from a corpus of documents. A POS tagger and linguistic patterns are used to extract noun phrases. A synonym table is used to define abbreviations used in the corpus. Second, as the major step, CFinder calculates the weights of the candidates using a combination of statistical and domain-specific knowledge. The weight of each candidate is represented as a real number and indicates its degree of relevance in the target domain. The higher a weight of a candidate is, the more relevant the candidate is in the target domain. Statistical knowledge is obtained using statistical analysis of the candidates from the corpus, while domain-specific knowledge is obtained through a domain-specific glossary list. The weight of each candidate is further enhanced using an inner structural (i.e. word-occurrence) pattern within the candidate. Finally, CFinder generates a ranked list of key concepts according to their weights. In the following, we describe each of these steps in more details.

3.1. Key concept candidate extraction

Key concept candidate extraction is a preliminary step for identifying key concepts. Its objective is to extract all possible candidates for key concepts using NLP techniques. Broadly, there are two main schemes that focus on extracting concept candidates in ontology learning systems.

The first scheme is to initially find domain-specific single-word terms, and then derive compound phrases by mixing them using statistical measures (Hou et al., 2011; Xu et al., 2002). An example of such a statistical measure is the term co-occurrence criterion (Manning & Schütze, 1999). However, as reported by Jiang and Tan (2010), the first scheme tends to lead to generating more single-word terms as key concepts, thereby missing many important multi-word terms that constitute the majority of domain-specific concepts. According to Nakagawa and Mori (2002), 85% of key concepts are actually comprised of multi-word terms.

As an effort to overcome this problem, most recent approaches follow the second scheme. This scheme initially extracts noun phrases, and then generates possible combinations of those terms belonging to each of the extracted noun phrases (Cimiano & Volker, 2005; Jiang & Tan, 2010; Li & Wu, 2006). For example, Jiang and Tan (2010) used all possible combinations of such terms within each phrase and a single-word term that is the head of the phrase where the head means a word that determines the syntactic type of the phrase (see an example in Table 1). On the other hand, Li and Wu (2006) used all single-word terms and all adjacent terms within each phrase (see also an example in Table 1).

In our approach, we follow the second scheme, as it is more promising than the first scheme for generating key concepts made up with multi-word terms. The detailed steps are as follows:

1. **Noun phrase extraction using POS tags:** CFinder first extracts key concept candidates using their linguistic patterns based on POS tags. POS tagging is the process for assigning a part of speech to each word in a text corpus. We use the Stanford POS tagger (Toutanova, Klein, Manning, & Singer, 2003) to parse a given corpus of documents in the target domain by splitting the corpus into sentences and assigning a POS tag to each word extracted from each sentence. Then, we extract noun phrases from each sentence using the following linguistic pattern used in Li and Wu (2006): $(JJ)^*(N)^+$, where 'JJ' means an adjective and 'N' denote nouns, and this pattern is interpreted as phrases starting with (1) one or more nouns or (2) one or more adjectives followed by one or more nouns ('$*$': zero or more time occurrences, '+': one or more time occurrences).
2. **Synonym finding and stopword removing:** We use a synonym table to identify abbreviations and their original forms in the extracted phrases in the first step, for example, the original form of 'EMS' is 'Emergency Medical Service'. This table is manually built by looking up abbreviations provided by the authors in the corpus if exist. Also, we remove stopwords used by the MySQL FullText feature from these phrases.
3. **Candidate enrichment:** We enrich key concept candidates by finding more nouns within each of the phrases obtained through the above steps. However, we separate our approach from the approaches (Jiang & Tan, 2010; Li & Wu, 2006) described above. Specifically, within a target phrase, we additionally consider and extract (1) single nouns and (2) all combinations of adjacent words belonging to the target phrase where each combination is only a noun phrase. We refer to these

Table 1
Key concept candidate extraction for a phrase: 'medical care system'.

Approach	Single-word terms	Multi-word terms
(Jiang & Tan, 2010)	system (the phrase head)	medical care, care system, medical system
(Li & Wu, 2006)	medical, care, system	medical care, care system
CFinder	care, system	medical care, care system

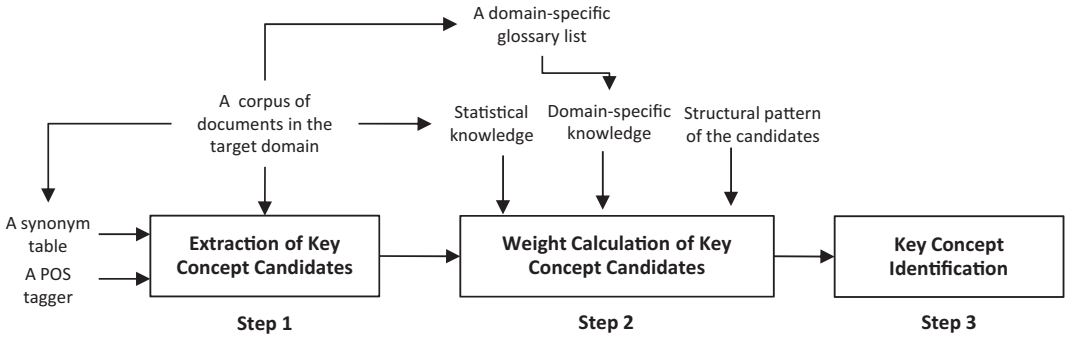

Fig. 1. The overview of CFinder.

additionally extracted candidates as *dependent-phrases*. An example is given in Table 1. As shown in the table, compared to Jiang and Tan (2010), our approach not only increases the number of more single-word terms as key concept candidates (i.e. 'care') but also decreases the number of multi-word terms by ignoring non-adjacent noun phrases (i.e. 'medical system') that do not co-occur adjacently together. Also, unlike Li and Wu (2006), our approach reduces the number of single-word terms that are non-nouns (i.e. 'medical') that may rarely become key concepts.

Once key concept candidates are identified based on the above steps, we calculate their weights using a combination of statistical and domain-specific knowledge, which will be presented in Section 3.2.

3.2. Weight calculation of key concept candidates

CFinder calculates the weights of key concept candidates in the target domain using a combination of the following two types of knowledge:

- *Statistical knowledge* is formulated as the frequency information of key concept candidates extracted from a corpus $\mathcal{D}$ in the target domain. The intuition behind the use of this knowledge is that the more frequently a term occurs, the higher its weight is.
- *Domain-specific knowledge* is formulated by measuring the relative importance of the candidates in the target domain. For this, we build a glossary list $\mathcal{T}$ that consists of domain-specific terms. These terms correspond to author-provided keywords or glossary terms if exist. The aim is to use this knowledge to assign higher weights to more domain-specific terms.

Another key feature of the weight calculation in CFinder includes the way of calculating the weight of a candidate whose *phrase length* (i.e. word count) is more than 1. The weight of such a candidate is calculated by aggregating the weights of its dependent-phrases. We refer to such a candidate as a *composite candidate*.

Let $\mathcal{KC}$ be the set of all key concept candidates. Formally, the weight of a candidate $c \in \mathcal{KC}$ with respect to a document $d \in \mathcal{D}$, denoted as $w(c,d)$, is computed as

$$w(c,d) = \begin{cases} tf(c,d) * w_d(c), & \text{if } len(c) = 1, \\ \sum_{i=1}^{n} w(c_i,d), & \text{otherwise}, \end{cases} \quad (1)$$

where $len(c)$ is the phrase length of the candidate c, and $tf(c,d)$ represents the frequency ratio of c in the document d, i.e.,

$$tf(c,d) = \frac{f(c,d)}{\max_t f(t,d)}, \quad (2)$$

where $f(c,d)$ is the number of times that the candidate c appears in the document d and the maximum is computed over the frequencies $f(t,d)$ for all candidates $t \in \mathcal{KC}$ that appear in d. Thus, $tf(c,d)$ is calculated through the use of statistical knowledge.

In Eq. (1), $w_d(c)$ represents the weight of the candidate c calculated using domain-specific knowledge. Formally, it is defined as

$$w_d(c) = 1 + \frac{\log(df(c))}{\log(\max_t df(t))}, \quad (3)$$

where $df(c)$ is the *domain-specific frequency* of the candidate c, computed as the number of times that c appears as part of a term in the glossary list $\mathcal{T}$. Also, $\max_t df(t)$ is the maximum number of times that any candidate in $\mathcal{KC}$ appears as part of a term in $\mathcal{T}$. The logarithm is used to mitigate the difference between $df(c)$ and $\max_t df(t)$.

Another key principle observed in Eq. (1) is that the weight of a composite candidate c is calculated by summing the weights of each dependent-phrase c_i of c, i.e. $\sum_{i=1}^{n} w(c_i,d)$, based on the *divide-and-conquer* paradigm (Cormen, Stein, Rivest, & Leiserson, 2001). The idea is to exploit an inner structural (i.e. word-occurrence) pattern observed in the candidate. More specifically, this weight calculation problem is achieved by recursively breaking down this problem into smaller calculation problems of its dependent-phrases. This procedure is continued until these problems are solved directly by using the dependent-phrases whose phrase length is 1. Finally, partially calculated weights are combined together into the weight of the composite candidate c. An example is given in Table 2.

The idea underlying this weight calculation principle is to simplify the weight calculation of a composite candidate c using c's dependent-phrases. It also enables us to formulate a well-structured weight calculation even if the phrase length of c is longer. Also, this principle naturally leverages the *inherent* weights of the dependent-phrases of c. This aspect separates it from approaches in Pianta and Tonelli (2010) that impose particular heuristics on 'phrase length' of terms (e.g. longer phrases are more weighted) that are not always correct in various domains as reported in Jiang and Tan (2010).

As the final step, we refine the weight $w(c)$ of a candidate $c \in \mathcal{KC}$ by eliminating the weights of those dependent-phrases that are repeatedly considered to compute the weight $w(c)$. For example, referring to Table 2, w('care', d) is employed to calculate three functions—w('medical care system', d), w('medical care', d) and w('care system', d)—to calculate the weight of 'medical care system'. This may not be reasonable, since it can make a high skewness between the weights of two candidates whose phrase length is shorter and longer, respectively. To avoid this problem, we only consider and use the weights of the dependent-phrases of c that are the *maximal subsets* of a composite candidate c to measure the weight of c. In our context, a maximal subset is a phrase that is not a subset of any other independent subset in the set of dependent-phrases of a composite candidate. The premise here is that maximal subsets of a composite candidate c can sufficiently approximate the information content of c. Finally, using the concept of maximal subsets, w('medical care system', d) is calculated as seen in Table 3.

Table 2
The weight calculation for a composite candidate 'medical care system'.

```
w('medical care system', d) =
    w('medical care', d) +
    w('care system', d) +
    w('care', d) +
    w('system', d), where
            w('medical care', d) =
                w('care', d); and
            w('care system', d) =
                w('care', d) +
                w('system', d).
```

Table 3
The refined weight calculation for a composite candidate 'medical care system' using the maximal subsets of the dependent-phrases of the candidate.

```
w('medical care system', d) =
        w('medical care', d) +
        w('care system', d), where
                w('medical care', d) =
                    w('care', d); and
                w('care system', d) =
                    w('care', d) +
                    w('system', d).
```

As can be seen in Table 3, two weights of two dependent-phrases 'care' and 'system', used to calculate w('medical care system', d), are eliminated compared to Table 2, since (1) 'care' is a subset of both 'medical care' and 'care system' and (2) 'system' is a subset of 'care system'.

In summary, we highlight the following distinctive features of our approach for calculating the weights of key concept candidates in $\mathcal{KC}$:

- Our approach assigns higher weights to more frequently occurring candidates in the corpus $\mathcal{D}$ and the domain-specific glossary list $\mathcal{T}$. Therefore, although a candidate whose phrase length is 1, it tends to become highly weighted if its frequency is high in $\mathcal{D}$ and/or $\mathcal{T}$.
- Candidates in $\mathcal{KC}$ will be highly weighted, if their phrase length is longer and also their dependent-phrases are frequently occurring in $\mathcal{D}$ and/or $\mathcal{T}$. Thus, although a composite candidate in $\mathcal{KC}$ is not frequently occurring in $\mathcal{D}$, its weight will be high if its dependent-phrases are frequently occurring in $\mathcal{D}$ and/or $\mathcal{T}$.
- As seen in Eq. (1), we do not consider including the IDF factor used in TF-IDF: $tf(c,d) * idf(c)$, where $idf(c)$ is the inverse document frequency that measures whether c is common or rare across in the corpus $\mathcal{D}$, assigning higher weights to more rare candidates in $\mathcal{KC}$. Formally, it is defined as $idf(c) = \log \frac{|\mathcal{D}|}{(1+|d_j|)}$, where d_j is the set of documents in $\mathcal{D}$ that c appears. In TF-IDF, if a candidate c, which needs to be considered as a key concept, frequently appears in many documents in $\mathcal{D}$, it cannot be selected as a key concept since $idf(c)$ is closer to 0. That is, the TF-IDF is sensitively affected by the number of documents in $\mathcal{D}$. As mentioned in Section 2.4, however, very often domain-specific terms (i.e. key concepts) are frequently appearing in the underlying corpus. Thus, the TF-IDF measure may perform poorly in the context. Thus, we do not incorporate the IDF factor into the calculation of $w(c)$.

3.3. Key concept extraction

Having identified key concept candidates $\mathcal{KC}$ with their weights through Sections 3.1 and 3.2 for each document in the corpus $\mathcal{D}$, we finally aggregate their weights across all documents in $\mathcal{D}$. Formally, given a candidate $c \in \mathcal{KC}$, its weight with respect to $\mathcal{D}$ is defined as follows:

$$w(c,\mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} w(c,d_i), \tag{4}$$

where d_i is a document in $\mathcal{D}$. Finally, CFinder generates a ranked list of key concept candidates $\{c_1, c_2, \ldots, c_{|\mathcal{KC}|}\} \in \mathcal{KC}$ according to their weights in the target domain based on the corpus $\mathcal{D}$, where $w(c_i,\mathcal{D}) \geqslant w(c_j,\mathcal{D})$ for all $i < j$. Thus, we can finally select a user-specified number of the candidates as key concepts from the list.

4. Evaluation

To evaluate CFinder in terms of effectiveness, we apply it to an existing ontology named the DO4MG ontology (Delir Haghighi et al., 2013), which is developed for the DMG domain. The concepts in the ontology were defined and annotated by rigorous text analysis on the corpora (Delir Haghighi et al., 2013): (1) the main 'Compendium of Mass Gatherings' corpus that is a collection of 27 scientific papers for emergency management for mass gatherings mainly from the Prehospital and Disaster Medicine journal (Arbon, 2009), (2) major journal and conference papers for emergency and crisis management and (3) a public report/government manual (e.g. the Emergency Management Australia (EMA)). We choose the DO4MG ontology and the 'Compendium of Mass Gatherings' corpus due to their both availability for public access, and also this ontology has been thoroughly evaluated based on a structured approach and domain expert feedback (Delir Haghighi et al., 2013). We select this ontology in our evaluation as it allows us to validate CFinder more effectively and accurately. Thus, the key concept candidates extracted and ranked by CFinder are evaluated in comparison to actual concepts in the DO4MG ontology.

The competitiveness of CFinder is measured by comparing its performance with three state-of-the-art methods for key concept extraction, which are Text2Onto (Cimiano & Volker, 2005), KP-Miner (El-Beltagy & Rafea, 2009) and Moki (Tonelli et al., 2011). The criteria for choosing them are summarized as follows: (1) these are all publicly available to use; (2) these all take unsupervised approaches to key concept extraction so that we can make a fair comparison between CFinder and them; (3) we aim to choose specific methods that have been widely compared for evaluation purposes for key concept extraction, so we select two widely known methods—Text2Onto (also compared in Jiang & Tan (2005), Jiang & Tan (2010) and Novalija et al., 2011) and KP-Miner (also compared in Sarkar (2013) and Lim, Wong, & Lim (2013)). Moki is chosen because it is mainly based on a combination of NLP techniques and statistical methods for key concept extraction. Thus, we believe that a comparison between CFinder and Moki will provide us with an important insight into the advantages of our proposed combination scheme and will enable us to see how CFinder can be competitive with Moki.

In the following, we provide detailed descriptions on the dataset (i.e. DO4MG), evaluation process and metrics, and evaluation results of CFinder and compared methods.

4.1. The DO4MG ontology

Organizing a successful *mass gathering* is complex and includes a variety of tasks. These tasks and activities can be grouped under three main phases of pre-event, during-the-event and post-event phases (Delir Haghighi et al., 2013). It is highly important to maintain the consistency and standardization of operations through all these phases to improve the overall results and effectiveness of medical provision. This can be mainly achieved by utilizing a common and unified knowledge structure that can be shared through all the phases. Delir Haghighi et al. (2013) introduced a domain ontology, named DO4MG, to provide a unified and comprehensive view on the problem domain that can be used by all concerned stakeholders and can be applied to all the phases and tasks of mass gatherings.

DO4MG has been developed by first identifying the scope and objectives, and then knowledge acquisition from the corpus of key documents in this domain as described earlier in Section 4. Then, the ontology has been thoroughly evaluated and refined using a criteria-based ontology evaluation approach and based on the feedback provided by domain experts.

In total, DO4MG contains 234 concepts. Fig. 2 shows top-level concepts in DO4MG with their 'is-a' relationships. The root concept is 'mass gathering', and the second level of DO4MG includes four concepts: (1) 'crowd features' conceptualizing various crowd characteristics of mass gatherings, (2) 'environmental factors' covering various environmental factors, (3) 'event venue' including concepts that relate to the internal and external characteristics of different venues where crowds are gathered, and (4) 'mass gathering plan' having the largest subsumed concepts in DO4MG and conceptualizing various aspects of emergency management for mass gatherings. The third level of the ontology includes 38 subclasses, i.e. 'children' or 'leaf classes', which are broken into further subclasses.

Among all concepts in DO4MG, we choose to use 200 concepts identified from the main corpus (i.e. 'Compendium of Mass

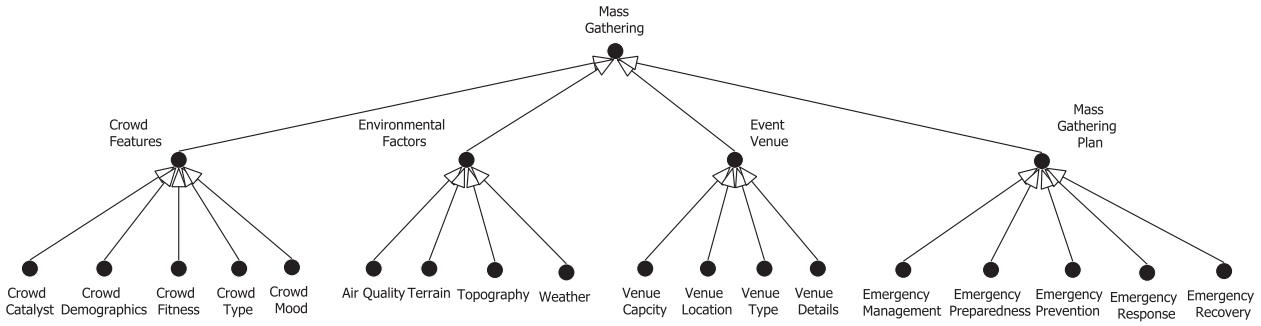

Fig. 2. A top-level view of concepts in DO4MG.

Gatherings') for our evaluation. The remaining 34 concepts out of the 234 concepts are excluded, since these were not extracted from the main corpus but from a small number of sources of documents (including journal articles, conference papers, and a public report/government manual), and also because they were provided explicitly by domain experts during the evaluation of the DO4MG ontology.

Each paper in the corpus provides author-provided keyphrases and abbreviations. The keyphrases were used to build a domain-specific glossary list used for building domain-specific knowledge for measuring the weights of key concept candidates (see also Section 3.2). Also, the abbreviations were used to build a synonym table applied at the first step for key concept extraction (see also Section 3.1).

4.2. Evaluation metrics

To determine whether extracted key concepts from CFinder, Text2Onto, KP-Minder and Moki are relevant or not in the DMG domain, the simplest possible way is to verify whether such concept labels are actually included in DO4MG. However, there are some concepts that have the same meaning but labeled differently in DO4MG and the corpus respectively. Examples of such concepts in DO4MG are those that are labeled by domain experts based on their experience and knowledge of common terminologies used by emergency management services. As reported in El-Beltagy and Rafea (2009), this situation can be occurred in many ontologies in which concepts are manually or semi-automatically identified and annotated. Table 4 shows an example of this occasion, where we see actual key concept candidates extracted from CFinder are different with the matched concept labels in DO4MG. Therefore, given each of the key concept candidates extracted by each method, we determine manually whether the candidate is matched with one of the concepts chosen for evaluation purposes (i.e. 200 concepts) in DO4MG. If matched, we refer to it as a *relevant* key concept.

To evaluate the effectiveness of each method, we used the following metrics: *precision*, *recall* and their harmonic mean (i.e. *F-measure*):

$$\mathrm{F-measure} = \frac{2 * \mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}. \quad (5)$$

Table 4
An example of key concept candidates and the corresponding concepts in DO4MG.

Extracted key concepts	The matched concepts in DO4MG
Mass gathering health	Public health
Crowd	Crowd features
Event type	Gathering type
Mass gathering medicine	Medicine
Emergency care	Emergency management
Field hospital	Local hospital

These metrics are widely used to evaluate information retrieval systems and also ontology concept extraction methods (Jiang & Tan, 2010; Li & Wu, 2006; Pianta & Tonelli, 2010). In our context, given a method, precision means the proportion of relevant key concepts among all those retrieved by the method. Recall is the proportion of the number of relevant key concepts among all the 200 concepts in DO4MG.

Furthermore, we also analyze the ranking performance of each method, i.e., determining how each method can produce relevant key concepts within *a ranking list* that is a ranked sequence of the top 200 key concept candidates ordered by their weights. For this purpose, we extend precision and recall to evaluate the quality of the ranking list generated by each method. In this context, we calculate the precision and recall scores at every position in the ranking list, and these scores are plotted to provide a *precision-recall curve* (Manning et al., 2008).

In our experimental context, the precision-recall curve has a distinctive saw-toothed shape, if the $(k+1)$ th key concept candidate in the ranking list is non-relevant. This is because the recall remains the same as k, while the precision drops where $1 \leqslant k \leqslant 200$. If it is relevant, the precision and recall scores increase, and the curve jags up and to the right. Often, it is useful to remove these jiggles. A standard way is to replace precision with the *interpolated* precision (Manning et al., 2008): the interpolated precision p_{interp} at a certain recall score r is defined as the highest precision found for any recall score $r' \geqslant r$:

$$p_{interp}(r) = \max_{r' \geqslant r} p(r')$$

Thus, in our evaluation, we used the interpolated precision to draw the precision-recall curve for each method (depicted in Fig. 4). The area under a precision-recall curve is called *average precision* (AP). AP provides a single-figure measure of quality across recall scores, more specifically, the average of the precision scores after each relevant key concept is retrieved. Formally, in our context, given a method, its AP is calculated as follows (Turpin & Scholer, 2006):

$$\mathrm{AP} = \frac{1}{R}\sum_{i=1}^{200} r_i \left(\frac{\sum_{j=1}^{i} r_j}{i} \right), \quad (6)$$

where R is the number of relevant key concepts extracted by the method, r_i is 1 if the ith key concept candidate is a relevant key concept, and 0, otherwise. In AP, the relevant key concepts ranked higher contribute more to the AP than those ranked lower. Thus, AP is widely used to evaluate methods that are more interested in returning more relevant items (e.g. key concepts) earlier.

Based on the above metrics, to determine whether there is a significant difference between the results of two methods, we carried out statistical tests using Wilcoxon signed rank test (Wilcoxon, 1945), which is a non-parametric version of a paired t-test and popularly used where the assumption of a normal distribution of the differences is not justified.

4.3. Results

Table 5 shows the top 20 and last 20 ranked key concept candidates identified by CFinder and compared methods. The symbol '*' attached to a term indicates that the term is a relevant key concept. As can be shown, referring to the top 20 key concept candidates, CFinder outperforms all the three methods by finding the largest relevant key concepts (i.e. 17). Looking at the last 20 key concept candidates, CFinder also identifies the largest number of relevant key concepts (i.e. 8).

Also, we observed that both Text2Onto and KP-Miner identified multi-word terms as key concept candidates. However, as the ranked candidates are lower, these methods tended to prefer to generate single-word terms as key concept candidates that are mostly non-relevant key concepts. For example, looking at the last 20 key concept candidates identified by these two methods, all terms are single-word terms where Text2Onto identified only 2 relevant key concepts while KP-Miner found nothing. On the other hand, we observed that CFinder and Moki were able to generate multi-word terms that are relevant key concepts even within the last 20 key concept candidates (i.e. CFinder and Moki identified 8 and 5 relevant key concepts, respectively).

We now present the number of relevant key concepts identified by each methods in Table 6 with the interval of 10 ranks. This table aims to provide a better understanding of how each method identified relevant key concepts as a rank increases. For each rank, the largest number of relevant key concepts across the 4 methods is highlighted. As can be seen, considering all top 200 ranked key concept candidates, CFinder discovered the largest number of relevant key concepts (i.e. 105) and Moki (i.e. 83) is the second best, while KP-Miner discovered the lowest number of relevant key concepts (i.e. 28). It is also shown that Moki identified the largest number of relevant key concepts during the ranks between 30 and 50, but gradually identified the lower number of relevant key concepts compared to CFinder as ranks increase.

Fig. 3 shows the F-measure curve, connecting each F-measure score at each position in the ranking list for each method. A high F-measure score indicates that both precision and recall are reasonably high. We observe that as the position increases, the improvements of CFinder become larger over the compared methods. Thus, overall, we discover that CFinder shows the best results and Moki the second best while KP-Miner turns out to be the lowest, in terms of F-measure.

Considering all the top 200 ranked key concept candidates, the F-measure score for each method is finally obtained as seen in Table 7. As observed, CFinder highly outperforms all compared methods reaching its F-measure score at 0.53. The improvement ratio of CFinder against each method ranges from 20% (with Moki) and to 278% (with KP-Miner) as seen in Table 8.

In order to determine whether the improvement of CFinder against each method is statistically significant, we used Wilcoxon signed rank test. The input of this statistical test was two lists for

Table 5
The top 20 and the last 20 ranked key concepts.

Rank	CFinder	Text2Onto	KP-Miner	Moki
1	Mass gathering*	Event*	Crowd*	Mass gathering event*
2	Event*	Patient*	Event*	Patient presentation
3	Patients*	Number	Mass gatherings*	Mass gathering*
4	Emergency medical services*	Mass gathering*	Spectator*	Patient presentation rates*
5	Patient presentation rates*	Care	Injuries*	Spectator*
6	Medical usage rate*	Injury*	Public event*	Crowd*
7	Patient presentations	Crowd*	Medical care*	Medical care*
8	World health organization	Hospital*	Venues*	Patient*
9	Crowd*	Datum	Illness*	Accessing
10	Injury*	Study	Planning	Hospital*
11	Games	Physician*	Care	On-site
12	Physician*	Day	Health	Rock concert*
13	Event type*	People	Number	Emergency medical services*
14	Medical care*	Spectator*	Attendance*	Recording
15	Emergency department*	Emergency	Factors	Injuries*
16	Rock concert*	Factor	Types	Physician*
17	Patients per 10,000*	Type	Rock concerts*	Venue*
18	Mass gathering health*	Year	Occur	Game
19	Injury surveillance system*	Concert*	Mass gathering events*	First-aid station*
20	Mass casualty incident*	Illness*	Safety	Concert*
No. of relevant key concepts	17	10	12	15
181	Persons	Meeting	Readiness	Mass gathering medicine*
182	Features	Planner	Document	Alcohol use*
183	Mass gathering situations*	Prediction	Scope	Injured patient
183	Mass gathering guidelines*	Preparedness*	Consulted	Heat index*
185	Weather*	Chart review	View	Established
186	Large public events*	Clinic	Sources	Security
187	Event organizers	Death	Later	Key
188	Public health*	End	Intensive	American red cross
189	Onbashira festival	Environment	Project	Evacuation
190	Planning	Evidence	Guide	Mass site
191	Chemicals	Goal	Quite	Loading
192	Medical care system*	Leader	Party	Model
193	Training*	Paper	Inherent	Health
194	Provision	Place	Awareness	Workload*
195	Aid station	Researcher	Initiative	Conducted
196	Patient management*	Scene	Sound	Relative humidity
197	Fans	Surveillance*	March	Field hospital*
198	Patient charts	Term	Granted	Rc
199	Apparent temperature	Action	Topic	Assist
200	Demonstrations	Athlete	Unit	Disaster medicine
No. of relevant key concepts	8	2	0	5

Table 6
The number of relevant key concepts.

Rank	CFinder	Text2Onto	KP-Miner	Moki
1	1	1	1	1
10	8	6	9	8
20	17	10	12	15
30	21	14	16	24
40	27	18	17	31
50	32	22	18	35
60	39	23	20	37
70	44	27	20	42
80	49	31	21	47
90	56	36	22	52
100	62	39	22	57
110	67	41	23	59
120	73	41	25	63
130	81	42	26	65
140	86	44	28	66
150	87	46	28	68
160	91	49	28	72
170	94	52	28	74
180	97	53	28	78
190	102	54	28	81
200	105	55	28	83

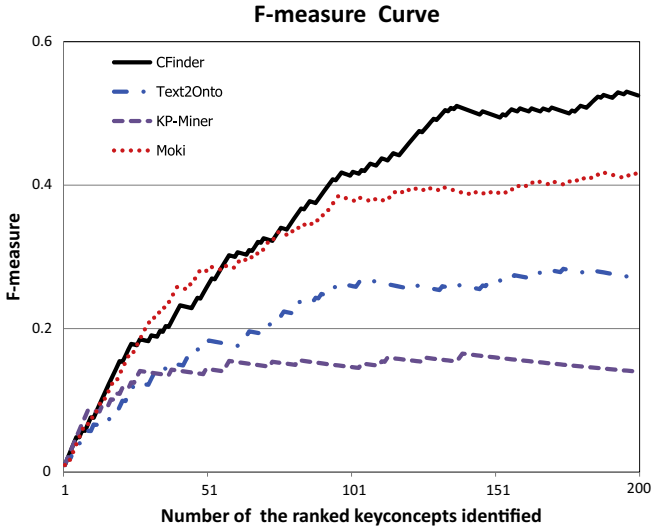

Fig. 3. The F-measure curves.

CFinder and a compared method, each being the set of the 200 F-measure scores generated by each method, where each score is calculated at each position in the ranking list. As seen in Table 8, the improvement of CFinder over all the three methods turns out to be statistically significant at the 99.9% confidence level (i.e. p-value < 0.001).

We now evaluate the ranking performance for each method using the AP metric. As mentioned earlier, for each method, AP is approximated by the area under a precision-recall curve, where the (interpolated) precision and recall scores are plotted with different positions in the ranking list generated by the method. Given each method, its precision-recall curve is depicted in Fig. 4 that represents a natural way of looking at its performance at every position in the ranking list in terms of precision and recall.

The curve explains how the interpolated precision and recall change as a value of k changes. A good method here ranks actual relevant key concepts near the top of the ranking list, while a poor method takes a higher score for precision to reach a higher score for recall. Rather than comparing the curves, as described earlier,

Table 7
The comparison in terms of F-measure.

Method	F-measure
CFinder	0.53
Text2Onto	0.28
KP-Miner	0.14
Toki	0.43

Table 8
Improvement of CFinder over the compared methods in terms of F-measure.

Method	Improvement of CFinder	Significance
Text2Onto	92%	Sig at 99.9% confidence
KP-Miner	278%	Sig at 99.9% confidence
Toki	20%	Sig at 99.9% confidence

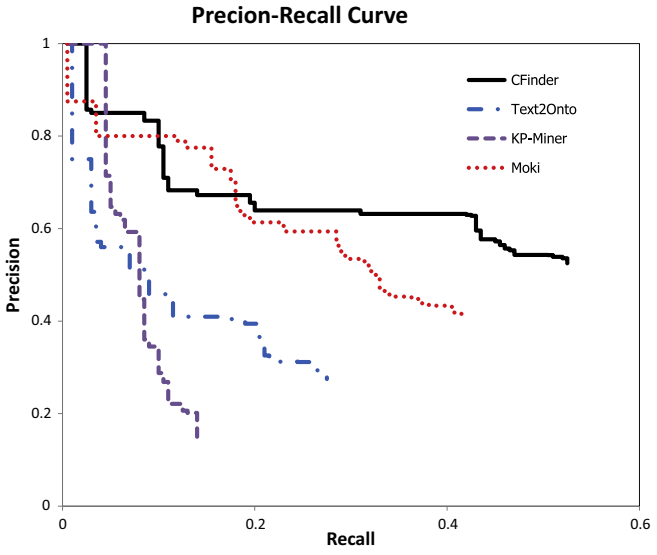

Fig. 4. Precision-recall curves.

Table 9
The comparison in terms of average precision (AP).

Method	AP
CFinder	0.662
Text2Onto	0.454
KP-Miner	0.595
Toki	0.607

Table 10
Improvement of CFinder over the compared methods in terms of AP.

Method	Improvement of CFinder	Significance
Text2Onto	45%	Sig at 99.9% confidence
KP-Miner	11%	Sig at 99.9% confidence
Toki	9%	Sig at 99% confidence

we use a single-figure measure that characterizes the performance of each method, i.e., AP presented in Eq. (6). The computed AP score for each method is finally given in Table 9. We discover that CFinder also outperforms all compared methods as large as 45% in terms of AP as seen in Table 10.

To determine whether the improvement of CFinder is statistically significant over each method, we also applied Wilcoxon signed rank test. For this test, the input used was the two lists for CFinder and each of the compared methods, in which each list

is the precision scores across all the recall scores used in building the precision-recall curves seen in Fig. 4. According to Table 10, CFinder shows a significant improvement over Text2Onto and KP-Miner at the 99.9% confidence level and over Moki at the 99% confidence level.

Through the evaluation results, we conclusively find that CFinder significantly outperforms the three methods in terms of both F-measure and AP. The evaluation results provide strong evidence that CFinder can substantially enhance effectiveness of key concept extraction.

5. Discussion and conclusion

In this paper, we presented a novel method named CFinder that can be effectively used to extract key concepts for ontology learning from a text corpus in a domain of interest. The main contribution of this paper is to propose CFinder for key concept extraction that is based on a heuristic that combines NLP techniques, statistical knowledge, domain-specific knowledge and inner structural pattern of terms extracted from the corpus. More specifically, we proposed what NLP techniques are used to extract key concept candidates, how statistical and domain-specific knowledge can be built and combined to estimate their degrees of relevance in the target domain, and how the inner structural patterns of the candidates were further enhanced to identify key concepts within the knowledge sources.

As demonstrated through the evaluation, CFinder has a strong ability to improve the effectiveness of key concept extraction. The real strength of CFinder lies in that it performs in a unsupervised manner which means it does not rely on training documents to build a model. Also, it does not require many corpora resources to perform compared to the corpus-based approaches that exploit multiple documents collections in multiple domains. Further, CFinder is designed to work with a corpus consisting of a small number of documents even a single document. This aspect is also an advantage of CFinder compared to the TF-IDF based approaches that typically require a large number of documents in the corpus to perform effectively.

A practical application of CFinder is that it could be effectively used for keyphrase (or key concept) extraction in various domains, where keyphrases play an important role such as text categorization, text summarization and information retrieval. More importantly, it is originally designed to extract key concepts for ontology learning. Therefore, it could be valuably leveraged for automatically building an ontology from a text corpus in the stage of key concept extraction. Furthermore, in practice, CFinder is towards a generalizable approach to key concept extraction, meaning that it is not restricted to any particular domains or applications. Although it does need a glossary list that is the fundamental resource for building domain-specific knowledge, but it is not compulsory. If it is not provided, CFinder performs using only statistical knowledge.

We demonstrated the strengths and significance of CFinder over three state-of-the-art methods for key concept extraction (i.e. Text2Onto, KP-Miner, and Toki) based on a recently developed ontology (i.e. DO4MG) for the domain of 'emergency management for mass gatherings', in terms of F-measure and AP. In our evaluation, we showed that CFinder achieved a F-measure score of 0.53 and AP score of 0.66. In comparison to these methods, we observed that these performance figures are statistically significant improvements as large as 278% in F-measure and 45% in AP (p-value $<$ 0.001). Our evaluation showed that CFinder is an effective method and provides a promising potential in making a practical impact on key concept extraction.

As future work, CFinder could be attempted to integrate with an existing ontology framework (e.g. Text2Onto, Toki). Also, we are planning to apply and measure the performance of CFinder for ontology extension (Novalija et al., 2011). Moreover, identifying properties of key concepts and estimating semantic relations between key concepts identified by CFinder could be interesting research topics for ontology learning.

Acknowledgments

This research is funded by Australian Research Council funding (LP0453745). This research is also partly supported by the University of Ballarat 'Self-sustaining Regions Research and Innovation Initiative', an Australian Government Collaborative Research Network (CRN).

References

Arbon, P. (2009). Prehospital and disaster medicine – Compendium of mass gatheirngs. *Journal of the World Association for Disaster and Emergency Medicine*, 24.

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems, 14*, 20–26.

Chen, R.-C., Liang, J.-Y., & Pan, R.-H. (2008). Using recursive ART network to construction domain ontology based on term frequency and inverse document frequency. *Expert Systems with Applications, 34*, 488–501.

Cimiano, P., & Volker, J. (2005). Text2Onto – A framework for ontology learning and data-driven change discovery. *Proceedings of the 10th international conference on applications of natural language to information systems (NLDB)* (Vol. 3513, pp. 227–238). Springer.

Cormen, T. H., Stein, C., Rivest, R. L., & Leiserson, C. E. (2001). *Introduction to algorithms* (2nd ed.). McGraw-Hill Higher Education.

Dahab, M. Y., Hassan, H. A., & Rafea, A. (2008). TextOntoEx: Automatic ontology construction from natural English text. *Expert Systems with Applications, 34*, 1474–1480.

Delir Haghighi, P., Burstein, F., Zaslavsky, A., & Arbon, P. (2013). Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings. *Decision Support Systems, 54*, 1192–1204.

Diederich, J., & Balke, W. -T. (2007). The semantic GrowBag algorithm: Automatically deriving categorization systems. In Proceedings of the 11th European conference on research and advanced technology for digital libraries ECDL'07 (pp. 1–13).

El-Beltagy, S. R., & Rafea, A. (2009). KP-Miner: A keyphrase extraction system for English and Arabic documents. *Information Systems, 34*, 132–144.

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction. In Proceedings of the 16th international joint conference on artificial intelligence – Vol. 2 IJCAI'99 (pp. 668–673).

Hou, X., Ong, S., Nee, A., Zhang, X., & Liu, W. (2011). GRAONTO: A graph-based approach for automatic construction of domain ontology. *Expert Systems with Applications, 38*, 11958–11975.

Jiang, X., & Tan, A. -H. (2005). Mining ontological knowledge from domain-specific text documents. In Proceedings of the fifth IEEE international conference on data mining ICDM '05 (pp. 665–668).

Jiang, X., & Tan, A.-H. (2010). CRCTOL: A semantic-based domain ontology learning system. *Journal of the American Society for Information Science and Technology, 61*, 150–168.

Li, Q., & Wu, Y.-F. B. (2006). Identifying important concepts from medical documents. *Journal of Biomedical Informatics, 39*, 668–679.

Lim, V. -H., Wong, S. F., & Lim, T. M. (2013). Automatic keyphrase extraction techniques: A review. In 2013 IEEE symposium on computers informatics (ISCI) (pp. 196–200).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. New York, NY, USA: Cambridge University Press.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA, USA: MIT Press.

Medelyan, O., & Witten, I. H. (2006). Thesaurus based automatic keyphrase indexing. In Proceedings of the sixth ACM/IEEE-CS joint conference on digital libraries JCDL '06 (pp. 296–297).

Missikoff, M., Velardi, P., & Fabriani, P. (2003). Text mining techniques to automatically enrich a domain ontology. *Applied Intelligence, 18*, 323–340.

Nakagawa, H., & Mori, T. (2002). A simple but powerful automatic term extraction method. In COLING-02 on COMPUTERM 2002: second international workshop on computational terminology – Vol. 14 COMPUTERM '02 (pp. 1–7).

Novalija, I., Mladenic, D., & Bradesko, L. (2011). OntoPlus: Text-driven ontology extension using ontology content, structure and co-occurrence information. *Knowledge-Based Systems, 24*, 1261–1276.

Noy, N. F., & mcguinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. Online.

Pianta, E., & Tonelli, S. (2010). KX: A flexible system for keyphrase extraction. In Proceedings of the fifth international workshop on semantic evaluation SemEval '10 (pp. 170–173).

Rezgui, Y. (2007). Text-based domain ontology building using TF-IDF and metric clusters techniques. *Knowledge Engineering Review, 22*, 379–403.

K., Sarkar (2013). A hybrid approach to extract keyphrases from medical documents. *International Journal of Computer Applications, 63*, 14–19. Published by Foundation of Computer Science, New York, USA..

Shih, C.-W., Chen, M.-Y., Chu, H.-C., & Chen, Y.-M. (2011). Enhancement of domain ontology construction using a crystallizing approach. *Expert Systems with Applications, 38*, 7544–7557.

Tonelli, S., Rospocher, M., Pianta, E., & Serafini, L. (2011). Boosting collaborative ontology building with key-concept extraction. In 2011 Fifth IEEE international conference on semantic computing (ICSC) (pp. 316–319).

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In Proceedings of HLT-NAACL (pp. 173–180).

Turpin, A., & Scholer, F. (2006). User performance versus precision measures for simple search tasks. In Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval SIGIR '06 (pp. 11–18).

Villaverde, J., Persson, A., Godoy, D., & Amandi, A. (2009). Supporting the discovery and labeling of non-taxonomic relationships in ontology learning. *Expert Systems with Applications, 36*, 10288–10294.

Wang, W., Mamaani Barnaghi, P., & Bargiela, A. (2010). Probabilistic topic models for learning terminological ontologies. *IEEE Transactions on Knowledge and Data Engineering, 22*, 1028–1040.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*, 80–83.

Wong, W., Liu, W., & Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM Computing Surveys, 44*, 20:1–20:36.

Xu, F., Kurz, D., Piskorski, J., & Schmeier, S. (2002). An domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping. In Proceedings of the third international conference on language resources and evaluation (LREC). Canary island, Spain.