Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Bag-of-Concepts representation for document classification based on automatic knowledge acquisition from probabilistic knowledge base[☆]

Pengfei Li [a], Kezhi Mao [a,*], Yuecong Xu [a], Qi Li [b], Jiaheng Zhang [a]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore
[b] Interdisciplinary Graduate School, Nanyang Technological University, 21 Nanyang Link, 637371, Singapore

ARTICLE INFO

Article history:
Received 8 January 2019
Received in revised form 30 October 2019
Accepted 24 December 2019
Available online xxxx

Keywords:
Natural language processing
Text representation
Document classification
Knowledge base
Interpretability

ABSTRACT

Text representation, a crucial step for text mining and natural language processing, concerns about transforming unstructured textual data into structured numerical vectors to support various machine learning and data mining algorithms. For document classification, one classical and commonly adopted text representation method is Bag-of-Words (BoW) model. BoW represents document as a fixed-length vector of terms, where each term dimension is a numerical value such as term frequency or tf-idf weight. However, BoW simply looks at surface form of words. It ignores the semantic, conceptual and contextual information of texts, and also suffers from high dimensionality and sparsity issues. To address the aforementioned issues, we propose a novel document representation scheme called Bag-of-Concepts (BoC), which automatically acquires useful conceptual knowledge from external knowledge base, then conceptualizes words and phrases in the document into higher level semantics (i.e. concepts) in a probabilistic manner, and eventually represents a document as a distributed vector in the learned concept space. By utilizing background knowledge from knowledge base, BoC representation is able to provide more semantic and conceptual information of texts, as well as better interpretability for human understanding. We also propose Bag-of-Concept-Clusters (BoCCl) model which clusters semantically similar concepts together and performs entity sense disambiguation to further improve BoC representation. In addition, we combine BoCCl and BoW representations using an attention mechanism to effectively utilize both concept-level and word-level information and achieve optimal performance for document classification.

1. Introduction

In this era of information explosion, text data is everywhere, from news articles, social media to customer feedback, machine logs and medical reports. Irrespective of industry, company or individual, the demands of automatic text analyses using Natural Language Processing (NLP) techniques are continuously increasing for better informed decisions or uncovering valuable insights from texts. Among various text analysis tasks, document classification is the most classical one, whose goal is to allocate documents into different categories. It has been widely applied in various applications such as spam detection [1], ontology mapping [2], document recommendation [3], topic labeling [4], and sentiment classification [5,6]. In this paper, we focus on text representation learning for document classification.

A typical document classification system consists of three main processes, namely text preprocessing, document representation and classification. The system performance largely relies on the quality of document representation, whose goal is to convert raw documents into structured numerical vectors (feature vectors) to support various machine learning and data mining techniques. The quality of a document representation can be evaluated from two perspectives: semantic quality and statistical quality [7]. Semantic quality refers to the semantic meaning and interpretability of the feature vector, i.e. to what extent can each feature describes the content of the document. For example, Bag-of-Words (BoW) model represents document as a fixed-length vector of all terms (words or n-grams) occurred in the corpus. Each term is weighted with a numerical value such as term frequency or tf-idf weighting. Statistical quality refers to the discriminative power of the feature vector in distinguishing documents from different classes. Recently, deep neural networks such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [8–10], especially the recent proposed pre-trained language models [11–13] have demonstrated excellent

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.105436.

[*] Corresponding author.
E-mail addresses: pli006@e.ntu.edu.sg (P. Li), ekzmao@ntu.edu.sg (K. Mao), xuyu0014@e.ntu.edu.sg (Y. Xu), liqi0024@e.ntu.edu.sg (Q. Li), e160003@e.ntu.edu.sg (J. Zhang).

https://doi.org/10.1016/j.knosys.2019.105436
0950-7051/© 2019 Published by Elsevier B.V.

statistical quality for many NLP tasks. However, majority models focus only on the statistical quality and neglect the semantic quality of representations. The resulted representations are very hard to interpret since the value of each feature are computed through complex compositions of the neural network weights. However, many real text mining applications demand for interpretable representations in order to provide human deeper understandings of texts and clearer operation logic for reasoning and decision making. The focus of this paper is to propose a good document representation that is able to capture the underlying characteristics and distinct features of documents in different classes while preserving its interpretability.

BoW model is commonly adopted for document classification due to its simplicity, acceptable accuracy and good interpretability. However, BoW model suffers from several shortcomings which hinder the performances of BoW-based models. Firstly, BoW simply look at the surface form of words and ignore the semantic relatedness and conceptual information of words and phrases. For example, the two sentences “Ferrari delivers the best driving experience” and “Lamborghini has powerful engine” share no common words, and BoW will represent them as two totally different vectors. However, the two sentences are closely related because “Ferrari” and “Lamborghini” are both brands of sports car. Secondly, BoW does not consider word order, hence the context information of texts is lost. However, context information offers a lot of semantics and is very useful to distinguish polysemous words and recognize synonyms. For example, the word “apple” can be a kind of fruit or a company, by looking into its context we can distinguish the two senses: if the word comes along with “fruit”, “orange”, “lemon” etc, we can recognize it as a kind of fruit; if it comes along with “company”, “Google”, “Microsoft” etc, we can recognize it as a company. Thirdly, BoW suffers from high dimensionality and sparsity since it uses hard/crisp mapping of the terms in the whole vocabulary. Besides, the model considers single words only, however, sometimes phrases deliver more semantic meanings than single words. For example, “galaxy note” is actually a Samsung phablet, however the word “galaxy” and “note” themselves deliver no such meaning. Even though Bag-of-n-Grams considers n-grams as basic terms, it will further increase dimensionality and sparsity severely. Besides, most of n-grams are meaningless.

To address the limitations of BoW model, we propose a novel Bag-of-Concepts (BoC) document representation scheme, which provides more semantic and conceptual information for document classification. The idea comes from human way of text understanding: we understand “Ferrari”, “Lamborghini” as sports car brands and “Google”, “Microsoft” as companies, because our mental world contains many concepts about such worldly facts. Besides acquiring information from the text we are reading, we simultaneously use these conceptual knowledge stored in our brain to help us understand text better. Inspired by this, the proposed BoC model conceptualizes words and phrases in the document into higher level semantics (i.e. concepts) based on the conceptual knowledge from external knowledge base, and finally represents document as a distributed vector in the learned concept space. The useful conceptual knowledge for certain classification task is automatically acquired from a large scale probabilistic knowledge base called Probase, these knowledge serves as the background knowledge in human brain. Compared with BoW model, the proposed model is actually transforming a high dimensional and sparse term vector space into a low dimensional and more dense concept vector space in a non-linear way based on the conceptual information of texts.

The main contributions of our work are summarized as follows:

∗ We propose an efficient way of automatically acquiring useful knowledge from large probabilistic knowledge base (Probase) that will benefit document classification task. The proposed knowledge acquisition scheme not only reduces the search space and querying time significantly, but also preserves the useful information for the certain classification task and reduces noises that are irrelevant at the meanwhile.

∗ Based on the acquired knowledge, we propose a document conceptualization scheme which reveals the higher level semantics of documents, and constructs a Bag-of-Concepts (BoC) model for document representation using a novel concept score-inverse soft document frequency ($cs - idf^s$) weighting scheme. Compared with BoW representation, BoC has lower dimensionality and sparsity due to the probabilistic mapping of document in concept space. More importantly, BoC representation is able to capture the semantic relatedness and conceptual information of texts, which will benefit document classification tasks.

∗ We further propose concept clustering and entity sense disambiguation techniques, and construct a Bag-of-Concept-Clusters (BoCCl) model to improve the BoC representation. The concept clustering process clusters similar concepts together to further reduce concept space dimensionality and sparsity. The entity sense disambiguation process disambiguates vague entities according to the context information, making the conceptualization of entities more accurate. Compared with BoW representation which use sparse words or n-grams as indexing terms, BoCCl representation is more interpretable and provides much deeper understanding of the document.

∗ Comprehensive experiments are conducted to evaluate both statistical and semantic qualities of the proposed BoC and BoCCl representations. Experiment results demonstrate that our proposed Bag-of-Concepts representation is able to improve document classification performance and also provide great interpretability for human understanding.

This paper is organized as follows. In Section 2, some related works of document representation are reviewed. Our proposed Bag-of-Concepts (BoC) representation for document classification is presented in Section 3, and some refined approaches to improve BoC representation are described in Section 4. Experimental results and detailed analyses are presented in Section 5. Finally, our work is concluded in Section 6.

2. Related work

In literature, document representation approaches fall into two categories: one is representation based on data only, where the representation is solely learned from training data; the other is incorporating knowledge from external knowledge bases into the representation. In this section, we give a comprehensive review of these approaches.

2.1. Representation based on data

As mentioned in the Introduction, the most established BoW model has limitations including high dimensionality and sparsity, and incapability of capturing contextual and conceptual information of texts. Many works are proposed to improve BoW representation. The most well-known models are latent semantic analysis (LSA) [14] and topic models including probabilistic latent semantic analysis (pLSA) [15] and latent dirichlet allocation (LDA) [16]. Based on the statistical analyses of term–document matrix, these models discover the latent semantic structures of the document and transform BoW representation into low dimension and dense vectors. In LSA, singular value decomposition (SVD) is applied on term–document matrix to represent documents in a new latent semantic space, where each latent semantic dimension is a linear combination of terms in the original BoW representation. In topic models, a document is represented as mixtures of topics that spit out words with certain probabilities. It is assumed that each topic is a distribution of words and each document is a mixture of topics. The topic models then try to backtrack from the documents to find a set of topics that are likely to have generated the documents. The term "Bag-of-Concepts" is originally proposed in [17]. However, the concepts are some synonym sets or latent dimensions of "BoW" instead of semantic hyponymies. All these models rely on word occurrence statistics to perform dimensionality reduction. However, the semantic relevance among words is ignored, and phrases cannot be captured since only single terms are considered. Besides, the derived latent dimensions are highly corpus-dependent and lack of semantic interpretation.

With the development of word embeddings, a distributed representation of words in vector space, semantic and syntactic properties of words can be captured in low-dimensional and dense space [18,19]. The word embeddings can be obtained through unsupervised learning from large-scale text corpus [20–22]. Many document classification methods are proposed by utilizing pre-trained word embeddings. One simple approach is averaging the word embeddings of all words in a document [23], assuming that the semantic space is shared by words and documents. Le and Mikolov [24] proposed doc2vec model by extending word2vec algorithm [20,21], which utilizes contextual information of words and documents to learn document representations in a continuous vector space. Zhao and Mao [25] proposed fuzzy Bag-of-Words (FBoW) model which adopts fuzzy mapping of words based on word embedding similarities to capture the semantic correlation among words. Kim et al. [26] also proposed a bag-of-concepts model for document representation. Different from our model, they treat each word cluster as a concept by perform clustering of word embeddings generated from Word2Vec. However, the performance of the model strongly depends on the quality of word embeddings. In many cases, Word2Vec model may not be able to correctly capture the semantic relationship among all the words since the training of word embeddings is based on the short context of words. For example, the cosine similarity between "good" and "bad" is 0.72, whereas the cosine similarity between "good" and "amazing" is only 0.48. Besides, word embeddings cannot deal with polysemous words such as "bank" and "apple". Consequently, the clusters based on word embedding similarities may not be reliable, and the errors or outliers in the clusters will affect the performance of the model. Besides, since word embeddings are single-word based, the semantics of phrases cannot be captured.

Recently, with the prevalence of deep learning [27], many deep compositional models have been proposed for document classification. A deep compositional model contains multilayer neural networks with various forms, including convolution [8,28], recurrent [9,10], recursive [29,30] neural networks as well as attention-based transformers [11,31] to learn text representations by performing compositions over word embeddings. The advantages of deep compositional models are that word order and contextual information are preserved, and more complex latent semantics can be captured by training the models using large amount of labeled data. Even though high classification accuracy can be obtained from deep compositional models, the models require sufficient high-quality annotated training corpus, heavy computations and tricky hyper-parameter settings. Besides, the resulted representations lack of interpretability since the features are computed through complex composition of neural networks.

2.2. *Representation leveraging knowledge bases*

Solely data-based text representation learning requires sufficient human-annotated and high-quality training corpus, especially for deep compositional models. This is sometimes impractical due to the lexical, morphological and syntactic variations of natural language. As a result, the learned representations may not be reliable and are normally domain specific. Many works are proposed to incorporate existing knowledge from external knowledge bases as prior knowledge for machine learning systems, in order to reduce the reliance on training data and provide additional useful information for text representation. In NLP, external knowledge bases can be categorized into lexical knowledge base and encyclopedic knowledge base.

The most widely used lexical knowledge bases include WordNet [32], SentiWordNet [33], VerbNet [34] and FrameNet [35]. Many works utilize the synonym sets and lexical categories of WordNet as concepts to improve text representation [36–38]. In [39], knowledge from WordNet is utilized to perform word sense disambiguation (WSD) by modeling the semantic space and semantic path using LSA and PageRank respectively. Some works use the lexicons in SentiWordNet to perform sentiment classification of reviews [40–42]. In [43], knowledge from WordNet and FrameNet is incorporated into CNN for causal relation extraction from texts. Despite the usefulness of lexical knowledge bases for text representation, these knowledge bases are normally manually constructed dictionaries which have limited coverage of worldly facts and conceptual knowledge. To capture more semantic and conceptual information of texts, many studies incorporate the knowledge from encyclopedic knowledge bases such as Wikipedia,[1] DBpedia [44], Open Directory Project (ODP),[2] Yago [45], Probase [46] etc. In [47], both Wikipedia and WordNet are used to exploit useful features for short text representation. Some works use the units of knowledge from Wikipedia and DBpedia as concepts for text representation [48–50]. In [51] and [52], categorical information in ODP is utilized for text representation. In [53], word sense in BabelNet[3] is embedded into document embeddings. In [54] and [55], the probabilistic knowledge of Probase is incorporated into topic models. Some works utilize Probase to learn concept embeddings which can be used for document representation [56,57]. A few studies on short text understanding and classification use the conceptual knowledge in Probase to discover the semantics from short texts [58–60]. We also utilize Probase in our work because it is a state-of-art encyclopedic knowledge base which has a broad coverage of concepts and entities, and the associations between entities and concepts are measured using probabilities instead of Boolean variable, which makes the subsequent inferences more flexible.

3. Bag-of-Concepts (BoC) representation

In this section, the proposed BoC representation is presented. The overall architecture of the model is shown in Fig. 1. The model consists of offline knowledge acquisition phase including automatic knowledge acquisition and entity conceptualization, as well as online representation learning phase including entity and concept recognition, document conceptualization and bag-of-concepts construction. The details of the two phases are described in Sections 3.1 and 3.2 respectively.

[1] https://en.wikipedia.org.

[2] http://curlie.org.

[3] https://babelnet.org.

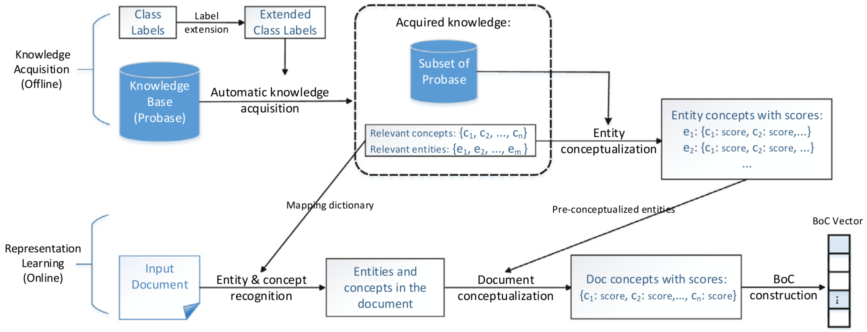
Knowledge
Acquisition
(Offline)
Class Labels
Label extension
Extended Class Labels
Knowledge Base (Probase)
Automatic knowledge acquisition
Acquired knowledge:
Subset of Probase
Relevant concepts: {c1, c2, ..., cn}
Relevant entities: {e1, e2, ..., em }
Entity conceptualization
Entity concepts with scores:
e1: {c1: score, c2: score,...}
e2: {c1: score, c2: score, ...}
...
Mapping dictionary
Pre-conceptualized entities
BoC Vector
Representation
Learning
(Online)
Input Document
Entity & concept recognition
Entities and concepts in the document
Document conceptualization
Doc concepts with scores:
{c1: score, c2: score,..., cn: score}
BoC construction
⋮

Fig. 1. Overall architecture of BoC model.

3.1. Offline knowledge acquisition

3.1.1. Probase and typicality scores

Probase[4] is a large scale encyclopedic knowledge base as well as a probabilistic semantic network. It contains over 5 million unique concepts, 12 million unique entities, and 87 million Is-A relations. The Is-A relationship is harvested from billions of web pages using Hearst patterns [61]. For example, from the sentence "Singapore's University such as Nanyang Technological University...", it extracts the concept–entity pair in which the entity "Nanyang Technological University" is an instance of the concept "Singapore's University".

One important characteristic of Probase is that the concepts and entities in the knowledge base are associated with each other in a probabilistic manner instead of Boolean association. This probabilistic manner is similar as our human mind. For example, given an entity "apple", people will conceptualize it as "fruit" or "company" more likely than "movie" or "song"; given a concept "illness", people are more likely to think of "flu" than "lupus". In Probase, the probabilistic associations between concepts and entities are represented by typicality scores, including the probabilities of concepts for a certain entity ($P(c|e)$), as well as the probabilities of entities belonging to a certain concept ($P(e|c)$). For above examples, $P(fruit|apple) > P(song|apple)$ and $P(flu|illness) > P(lupus|illness)$. Formally, the typicality scores are derived from the co-occurrences of concepts and entities as follows:

$$P(c|e) = \frac{n(e,c)}{\sum_{e \in c_i} n(e,c_i)} \tag{1}$$

$$P(e|c) = \frac{n(e,c)}{\sum_{e_i \in c} n(e_i,c)} \tag{2}$$

where $n(e, c)$ is the co-occurrence of entity e and concept c in the Hearst patterns from web documents.

The typicality scores allow the knowledge representation to be more accurate and reliable, and also provide more flexibilities on querying and operating on the knowledge base. However, when conceptualizing an entity, the two typicality scores tends to give higher scores to "extreme" concepts [62]. $P(c|e)$ tends to assign higher scores to more general concepts as the score is proportional to the co-occurrence of c and e when e is given; and $P(e|c)$ tends to assign higher scores to more specific concepts as the score will be higher for the specific concepts that only contains e or majority of its entities are e. General concepts provide less discriminative powers, and specific concepts cover less entities, making the concept space even more sparse. To balance between general and specific, and find the most suitable concepts for an entity, we use a modified $Rep(e, c)$ score proposed by Wang et al. [62] to conceptualize an entity, as shown in Eq. (3).

$$Rep(e,c) = P(c|e) \cdot P(e|c)_{k-smooth} \tag{3}$$

where $P(e|c)_{k-smooth}$ is the smoothed typicality score to avoid the extreme values caused by the specific concepts that contains few entities:

$$P(e|c)_{k-smooth} = \frac{n(e,c)+k}{\sum_{e_i \in c} n(e_i,c) + kN_e} \tag{4}$$

where N_e is the total number of entities, and k is a small constant assuming that every concept–entity pair has a small co-occurrence whether we observe it or not.

3.1.2. Automatic knowledge acquisition from Probase

Even though Probase has a broad coverage of concepts and entities, fully utilization of Probase knowledge for document classification is not a good idea. The reasons are as follows:

- As mentioned in Section 3.1.1, Probase contains large amount of concepts, entities, and Is-A relation pairs. Querying on the whole knowledge base is quite time-consuming due to the enormous search space. This will cause both offline and online learning processes to be very slow, hence hinder the computational efficiency of document classification.
- Using all the concepts in Probase will cause the dimensionality of concept space even large than the vocabulary space, making the model suffer from curse of dimensionality as well as sparsity issue. Besides, many concepts and entities in Probase are not relevant to the specific classification task. These irrelevant concepts and entities will add noise to the model and affect the accuracy of document classification.

In fact, instead of acquiring all the knowledge from external resources, our human selectively acquire useful knowledge based on their needs for certain tasks. For document classification tasks, the classes or categories are normally pre-defined, and only the concepts and entities related to these classes are useful for document classification. Motivated by this, we propose an automatic knowledge acquisition scheme to select a subset of Probase for subsequent processes. Based on the class labels and class-related information, we automatically select as much as possible the concepts and entities from Probase that are relevant to the classes. The details are as follows:

We first find the most relevant words that can best describe each classes, such as class labels. Then we extend the class labels using an external lexical resource called WordNet.[5] WordNet groups semantically similar English words into sets of synonyms

[4] Available online at https://concept.research.microsoft.com.

[5] Available online at https://wordnet.princeton.edu.

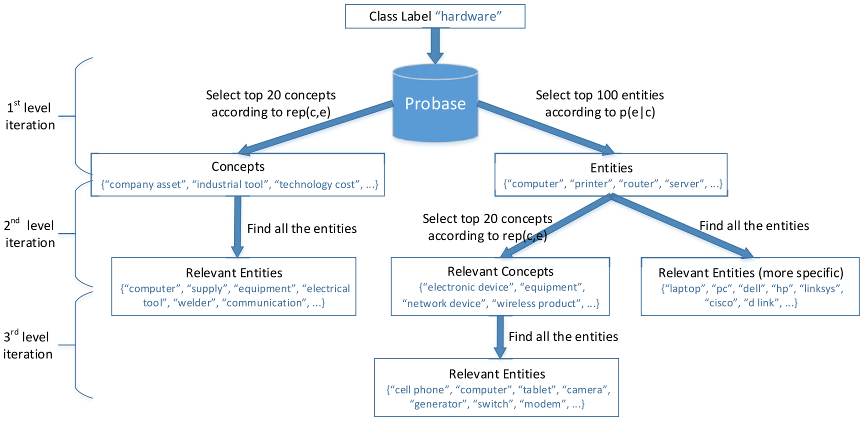
Class Label "hardware"
Probase
Select top 20 concepts according to rep(c,e)
Select top 100 entities according to p(e|c)
1st level iteration
2nd level iteration
3rd level iteration
Concepts
{"company asset", "industrial tool", "technology cost", ...}
Entities
{"computer", "printer", "router", "server", ...}
Find all the entities
Select top 20 concepts according to rep(c,e)
Find all the entities
Relevant Entities
{"computer", "supply", "equipment", "electrical tool", "welder", "communication", ...}
Relevant Concepts
{"electronic device", "equipment", "network device", "wireless product", ...}
Relevant Entities (more specific)
{"laptop", "pc", "dell", "hp", "linksys", "cisco", "d link", ...}
Find all the entities
Relevant Entities
{"cell phone", "computer", "tablet", "camera", "generator", "switch", "modem", ...}

Fig. 2. Example of concept and entity extraction from Probase using 3-level iterations based on the class label "hardware".

Algorithm 1 Automatic knowledge acquisition

```
Step 1: Find the class labels that best describe each document class.
    $labels = \{l_1, l_2, \ldots, l_c\}$
Step 2: Extend the class labels using the WordNet synsets of the class labels.
    for term in labels do
        for synset in WordNet synsets of term do
            for lemma in WordNet lemmas of synet do
                if lemma not in labels then
                    labels = labels + lemma
                end if
            end for
        end for
    end for
Step 3: Concept and entity extraction from Probase using 3-level iterations.
    for each term $t$ in the extended class labels do
        1st-level iteration:
            Find top 20 concepts $c$ according to $Rep(t, c)$.
            Find top 100 entities $e$ according to $P(e|t)$.
        2nd-level iteration:
            For each concept $c$ extracted from the 1st level, find all its entities.
            For each entity $e$ extracted from the 1st level, find its top 20 concepts $c$ according to $Rep(e, c)$; treat the entity $e$ as concept and find all the entities belong to it.
        3rd-level iteration:
            For each concept extracted from the 2nd level, find all its entities.
    end for
    Aggregate all the concepts found during the 3-level iteration and form a concept set: $c\_set = \{c_1, c_2, \ldots, c_n\}$;
    Aggregate all the entities found during the 3-level iteration and form an entity set: $e\_set = \{e_1, e_2, \ldots, e_m\}$.
```

(called synsets) to represent different meanings [32]. By searching the synsets of class labels, we can find all the synonyms of the class labels and hence extend the class labels to cover more relevant terms. Finally, we extract all the relevant concepts and entities from Probase based on the extended class labels. The detailed algorithm for automatic knowledge acquisition is shown in Algorithm 1. Fig. 2 shows an example of concept and entity extraction from Probase using the proposed 3-level iterations based on the class label "hardware". The concepts extracted based on the first level entities are all the relevant concepts of the class label, as they contain common entities. The entities extracted based on the concepts of first level and second level concepts are the relevant entities of the class label, as they either belong to the same concept as class label, or belong to the relevant concepts of the class label. Besides, the entities found by treating the second level entities as concepts are the more specific entities related to the class label.

Through the proposed automatic knowledge acquisition algorithm, all the useful concepts and entities that are relevant to the document classification task can be extracted from Probase. This greatly reduces the search space and accelerates the subsequent processes. More importantly, by keeping only the relevant knowledge, the noises are removed and the dimensionality as well as sparsity of concept space are reduced significantly.

3.1.3. Entity conceptualization

After selecting relevant entities and concepts from Probase, we conceptualize the selected entities by mapping each entity to the selected concepts with probability scores. For each entity, we rank its concepts according to the $Rep(e, c)$ score described in Section 3.1.1. Then we keep only the top k_c concepts and transform the $Rep(e, c)$ scores into probability scores according to Eq. (5):

$$Rep(e, c_i)_p = \frac{Rep(e, c_i)}{\sum_{j=1}^{k_c} Rep(e, c_j)} \tag{5}$$

Finally, a mapping dictionary of the conceptualized entities is constructed, which map entities (words/phrases) into higher level semantics (i.e. concepts) with probability scores. The conceptualized entities can be directly used for online representation learning processes.

3.2. Online representation learning

3.2.1. Entity and concept recognition

We use the entities and concepts acquired from the automatic knowledge acquisition process as mapping dictionary to recognize the entities and concepts in the document. The matching algorithm we used is called Backward Maximum Matching (BMM) [63]. BMM is originally proposed for Chinese word segmentation as most Chinese words contain many characters. Similarity, we found that BMM is also efficient to recognize English phrases which are composed of many words. The algorithm selects a string with *maxLen* words from right to left of the text, and use it to map the entities and concepts in Probase.

Before applying the matching algorithm, we first pre-process the document including sentence segmentation, word tokenization and part-of-speech (POS) tagging. Then we perform entity

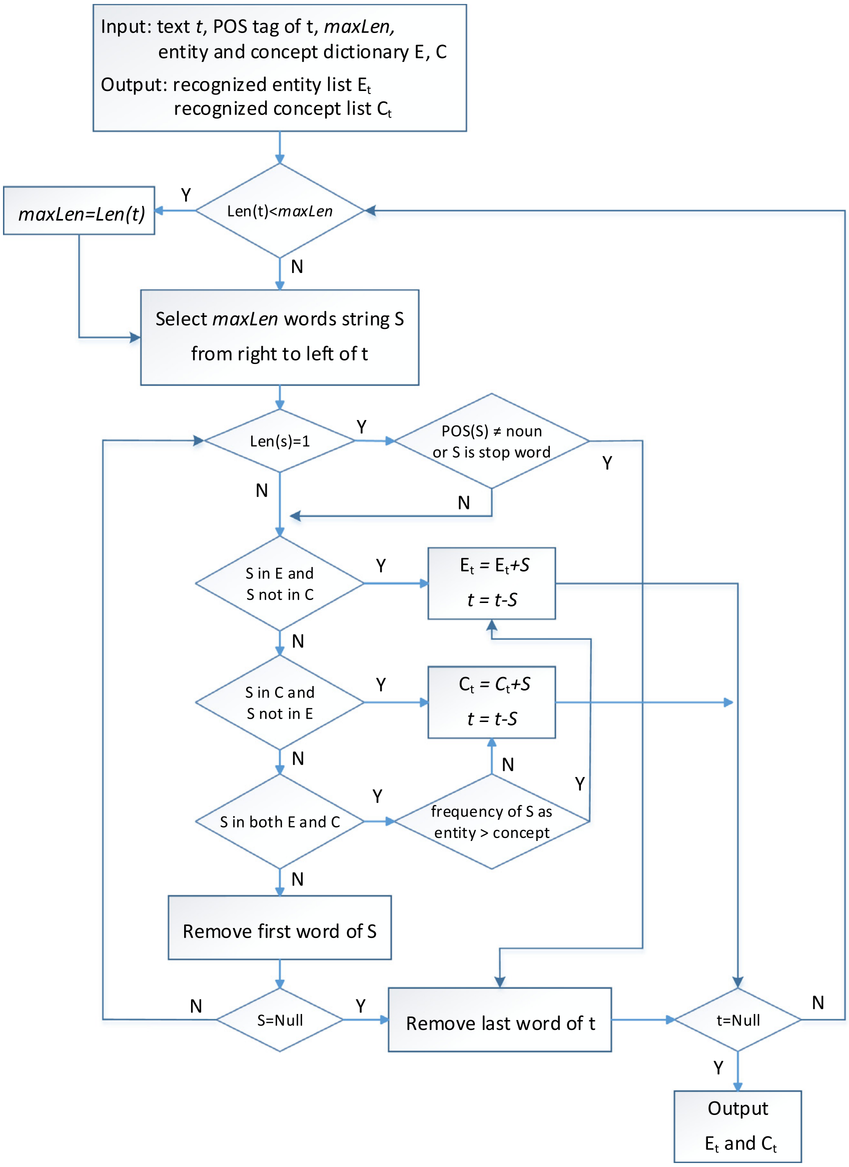
Input: text t, POS tag of t, maxLen, entity and concept dictionary E, C
Output: recognized entity list E_t recognized concept list C_t
Len(t)<maxLen
Y
maxLen=Len(t)
N
Select maxLen words string S from right to left of t
Len(s)=1
Y
POS(S) ≠ noun or S is stop word
Y
N
N
S in E and S not in C
Y
$E_t = E_t+S$ $t = t-S$
N
S in C and S not in E
Y
$C_t = C_t+S$ $t = t-S$
N
S in both E and C
Y
frequency of S as entity > concept
Y
N
N
Remove first word of S
S=Null
N
Y
Remove last word of t
t=Null
N
Y
Output E_t and C_t

Fig. 3. The flow chart of Backward Maximum Matching (BMM) for entity and concept recognition.

and concept recognition sentence by sentence, as shown in the flow chart in Fig. 3. We use two mapping dictionaries: one for entities and the other for concepts, so that the algorithm can output the recognized entities and concepts separately. For a single word, we do not consider it as concept or entity candidate if the word is a stop word (similar like stop words removal), or the POS tag of the word is not noun. For example, in the sentence "watch star war movie", the word "watch" is a verb and is not an entity or concept. However, if we do not consider the POS tag, the system may recognize it as an entity which belongs to concepts "timepiece", "valuable" etc. The wrong recognition of entities or concepts will introduce noise and mislead the document classification system.

3.2.2. Document conceptualization

For each recognized entity in the document, we conceptualize it into higher level concepts by looking for the offline mapping dictionary of the conceptualized entities:

$$e_i \rightarrow \{c_1 : Rep(e_i, c_1)_p, c_2 : Rep(e_i, c_2)_p, \ldots\}$$

where $Rep(e_i, c_j)_p$ is the concept probability score of concept c_j for entity e_i. For each recognized concept in the document, we set its concept probability score as 1:

$$c_i \rightarrow \{c_i : 1\}$$

After that, we conceptualize the sentences in the document by aggregating the concept probability scores of the recognized entities and concepts in the sentence.

$$S_i \rightarrow \{c_1 : Score(S_i, c_1), c_2 : Score(S_i, c_2), \ldots\}$$

where $Score(S_i, c_j)$ is the concept score of c_j for sentence S_i:

$$Score(S_i, c_j) = \sum_{e_k \in S_i} Rep(e_k, c_j)_p + \sum_{c_j \in S_i} 1 \tag{6}$$

Similarly, the document is conceptualized by aggregating the concepts and their scores of all the sentences in the document.:

$$D_i \rightarrow \{c_1 : Score(D_i, c_1), c_2 : Score(D_i, c_2), \ldots\}$$

where $Score(D_i, c_j)$ is the concept score of c_j for document D_i:

$$Score(D_i, c_j) = \sum_{S_k \in D_i} Score(S_k, c_j) \tag{7}$$

Eventually, each document is conceptualized into a set of concepts which are associated with concept scores reflecting the relevance of the concepts to the document.

3.2.3. Bag-of-Concepts construction

After conceptualizing all the documents in the collection, we find the concepts vocabulary and their document frequencies. In BoW model, document frequency of a term t is the total number of documents containing t, as the term t either appears or not appears in a document (Boolean association). However, this is not applicable to our proposed BoC model, because each concept term is associated with the documents using concept scores (fuzzy association), and the concepts with very small concept scores in a document should not be counted as one occurrence. Therefore, we propose a soft document frequency df_i^s for concept c_i, as calculated in Eq. (8).

$$df_i^s = \sum_{D_j \in \mathfrak{D}} Occ(D_j, c_i) \tag{8}$$

where $\mathfrak{D}$ is the set of all the documents, and $Occ(D_j, c_i)$ is the soft occurrence of concept c_i in document D_j:

$$Occ(D_j, c_i) = \begin{cases} Score(D_j, c_i) & \text{if } Score(D_j, c_i) < 1 \\ 1 & \text{if } Score(D_j, c_i) \geq 1 \end{cases} \tag{9}$$

We discard the concepts whose soft document frequency is less 1, which means the concepts rarely appear in the documents and provide little information for document classification. Finally, we represent a document D_j as a distributed vector in the learned concept space: $d_j = (w_{1j}, w_{2j}, \dots, w_{lj})$, where l is the dimensionality of the concept space, and w_{ij} is the weight of concept c_i in document d_j. Similar to term frequency–inverse document frequency ($tf - idf$) weighting scheme, we propose a concept score-inverse soft document frequency ($cs - idf^s$) weighting scheme for calculating w_{ij}, as shown in Eq. (10).

$$w_{ij} = Score(D_j, c_i) \times log(\frac{N}{df_i^s}) \tag{10}$$

where N is total number of documents in the collection. The first term $Score(D_j, c_i)$ is the concept score which indicates the relevance of concept c_i in document D_j; the second term $log(\frac{N}{df_i^s})$ is the inverse soft document frequency to reduce the weights of frequent concepts since the concepts appear in majority documents are less informative than those only appear in certain classes.

Compared with BoW representation, BoC has lower dimensionality due to the fact that concept space has much lower dimension than word or n-gram space. Besides, BoC representation is less sparse than BoW representation, because the conceptualization process in BoC model maps words and phrases into diverse concepts in a probabilistic way instead the hard mapping of words as in BoW model. More importantly, BoC model is able to capture the semantic relatedness and conceptual information of words and phrases as well as higher-level semantics of documents, which will benefit the document classification tasks.

4. Refined approaches

To further improve BoC representation, we propose some refined approaches including Bag-of-Concept-Clusters (BoCCl) model (Section 4.1), entity sense disambiguation (Section 4.2) and combination of BoC and BoW representations using attention mechanism (Section 4.3).

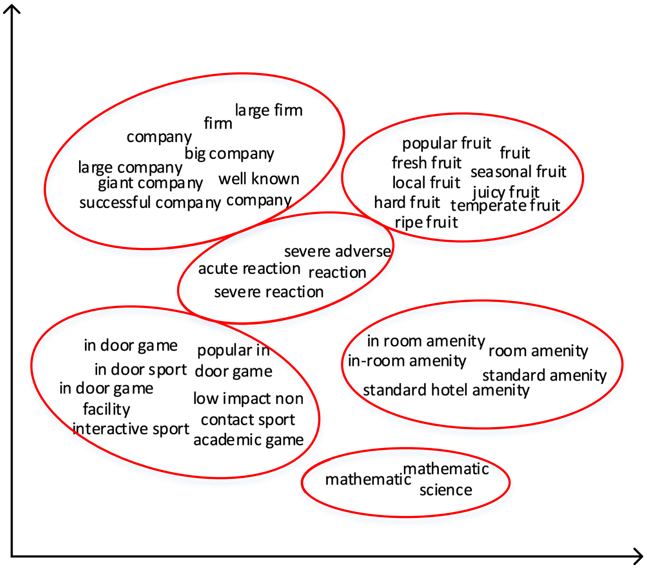

Fig. 4. Illustration of clustering of similar concepts.

4.1. Bag-of-Concept-Clusters (BoCCl)

Among the concepts acquired from Probase, many of them are similar with each other, which contain a lot of common entities. For example, "company", "firm", "large company", "big company", "giant company", "large firm" are similar concepts representing "company"; "in room amenity", "standard hotel amenity", "room amenity", "in-room amenity", "standard amenity" are similar concepts representing "in-room amenity". These similar concepts are redundant features which will increase the dimensionality as well as sparsity of the concept space.

To further reduce dimensionality and sparsity of concept space, we perform concept clustering to group similar concepts together. Firstly, we represent each concept c_i as a vector in the entity space $\mathbb{R}^m$, where m is the total number of entities.

$$c_i \rightarrow \{P(e_1|c_i), P(e_2|c_i), \dots, P(e_m|c_i)\}$$

where $P(e_i|c_i)$ is the typicality score of entity e_i belonging to concept c_i as described in Section 3.1.1.

Secondly, we compute pair-wise distance between two concepts and obtain a distance matrix of all the concepts. The pair-wise distance is calculated as shown in Eq. (11).

$$d(c_i, c_j) = 1 - cosine(c_i, c_j) \tag{11}$$

where $cosine(c_i, c_j)$ is the cosine similarity of concepts c_i and c_j in the entity space.

Thirdly, agglomerative clustering [64] is used to group similar concepts together based on the pair-wise distance matrix. Agglomerative clustering is a bottom-up hierarchical clustering method, which treats each concept as one cluster at the beginning, and successively merges close clusters together until we get the desired number of clusters or meet certain distance threshold between clusters. Finally, the Probase concepts are grouped into clusters of similar concepts, as illustrated in Fig. 4.

Based on the concept clusters, we propose Bag-of-Concept-Clusters (BoCCl) model, where an entity is conceptualized into concept clusters instead of concepts. The entity conceptualization process is also done offline: for each entity, we rank its concept clusters according to the $Rep(e, ccl)$ score as defined in Eq. (12).

$$Rep(e, ccl_i) = \sum_{c_j \in ccl_i} Rep(e, c_j) \tag{12}$$

Please cite this article as: P. Li, K. Mao, Y. Xu et al., Bag-of-Concepts representation for document classification based on automatic knowledge acquisition from probabilistic knowledge base, Knowledge-Based Systems (2020) 105436, https://doi.org/10.1016/j.knosys.2019.105436.

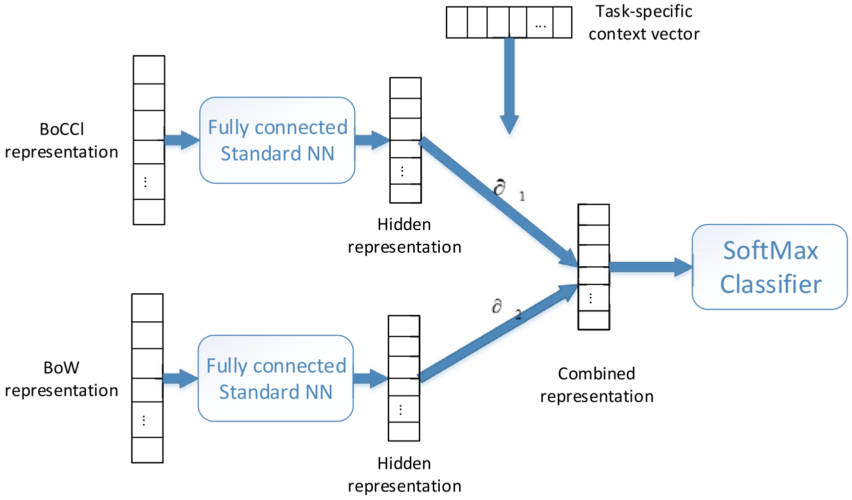

Fig. 5. Combination of BoCCl and BoW representations using attentional neural network.

Similar as in Section 3.1.3, we keep only the top k_c concept clusters and transform the $Rep(e, ccl_i)$ scores into probabilities $Rep(e, ccl_i)_p$ in the same way as Eq. (5). The online document conceptualization and BoCCl construction processes are basically same as BoC model, except that the basic term is concept clusters instead of concepts.

4.2. Conceptualization with entity sense disambiguation

In natural language, there often exists polysemous words or phrases which have several different meanings. If such polysemous word or phrase is an entity, then it will be conceptualized into different concepts reflecting different meanings. For example, the entity “apple” can be conceptualized into “fruit” or “company”, which are totally different concepts. Hence, entity sense disambiguation is important to achieve better representation of texts as well as better performance of NLP systems.

Inspired by Firth [65] who stated that “A word is characterized by the company it keeps”, we believe that context entities provide useful information for entity sense disambiguation. For example, if the context of “apple” contains “fruit”, “orange”, “banana” etc, we can conceptualize it as “fruit” with larger probability; if the context contains “company”, “Google”, “Microsoft” etc, we can conceptualize it as “company” with larger probability.

We first evaluate the ambiguity of an entity e_i using the entropy of e_i’s concept clusters distribution, as shown in Eq. (13).

$$H(e_i) = -\sum_{ccl_j \in CCl_{e_i}} P(ccl_j|e_i) \times log_2 P(ccl_j|e_i) \tag{13}$$

$$P(ccl_j|e_i) = \sum_{c_k \in ccl_j, c_k \in C_{e_i}} P(c_k|e_i) \tag{14}$$

where CCl_{e_i} is the set of all the concept clusters of e_i, $P(ccl_j|e_i)$ is the probability of concept cluster ccl_j, and C_{e_i} is the set of all the concepts of e_i.

A higher entropy indicates higher uncertainty or ambiguity of the sense of an entity. We can set an entropy threshold h_t to determine whether an entity is ambiguous or not based on empirical studies. If an entity is ambiguous, then we can disambiguate the entity utilizing unambiguous context entities. To find the unambiguous context entities, we choose a context window of size 3 as an entity’s context: the sentence which contains the entity, and the sentences before and after this sentence. Then we find all the unambiguous entities E_u in the context whose entropy is smaller than h_t. Finally, we re-weight the $Rep(e, ccl)$ scores of the concept clusters of the ambiguous entity according to Eq. (15).

$$Rep(e, ccl_i)' = Rep(e, ccl_i) + \frac{\sum_{ccl_i \in CCl_{e_j}, e_j \in E_u} Rep(e_j, ccl_i)}{|E_u|} \tag{15}$$

The intuition behind Eq. (15) is to increase the ambiguous entity’s concept cluster score if the unambiguous context entities also belong to this concept cluster. Consequently, the probability scores of concept clusters ($Rep(e, ccl_i)_p$) for an ambiguous entity will be dynamically adjusted based on the context information, making the entity conceptualization process more accurate.

4.3. Combination of BoCCl and BoW representations

As majority of entities in Probase are nouns or noun phrases, BoCCl (BoC) model focuses on the concepts of these words/phrases in the document. However, for some document classification tasks, other types of words such as verbs, adjectives and adverbs also play an important role in identifying the categories of the documents. Even though BoCCl provides higher level semantics of the document, only considering the concept-level information may not yield the best classification result. On the contrary, BoW model provides only word-level information but neglect concept-level information. To achieve the best performance for document classification, both concept-level and word-level information are needed. Therefore, we consider combining BoCCl and BoW representations to preserve both concept-level and word-level information for document representation.

To solve the high dimensionality and sparsity of BoW representation and allow the model to select concept-level and word-level information attentionally for different classification tasks, we propose an attention mechanism to effectively combine BoCCl and BoW representations. Fig. 5 shows the architecture of the proposed attention mechanism. We first feed BoCCl and BoW representations into a fully connected layer of standard neural network to get their hidden representations with same dimensionality. As in Eq. (16), **W** is the weight matrix and b is a bias term of the fully connected layer. Then we measure the importance of two representations by comparing them with a task-specific context vector. As shown in Eq. (17), dot product is used to evaluate the relatedness between the representation h_i and the context vector c, and softmax function is used to get a normalized attention weight α_i for each representation. Based

Table 1
Statistics of six document classification datasets.

Statistics	Document classification tasks					
	20NG	R8	R52	BBC	BBCSport	Yahoo
Vocab. size	117 160	19 956	22 248	21 914	9637	883 894
Class No.	20	8	52	5	5	10
Sample No.	18 846	7674	9100	2225	737	1460k
Avg. words	241	98	104	370	323	90
Train/Test	11 314/7532	5485/2189	6532/2568	1112/1113	368/369	1400k/60k

on the attention weight, the two representations are combined as shown in Eq. (18). The task specific context vector c is a high level representation reflecting which information from concept-level and word-level representations is important for certain classification task. It is randomly initialized and adjusted by learning from training data.

$$h_i = tanh(\mathbf{W}v_i + b) \tag{16}$$

$$\alpha_i = \frac{exp(h_i^T c)}{\sum_i exp(h_i^T c)} \tag{17}$$

$$r = \sum_i \alpha_i h_i \tag{18}$$

By using attention mechanism, the model is able to select word-level and concept-level information attentionally and make good use of both of them to meet the classification needs.

5. Experiments

5.1. Datasets

Six real-life datasets for document classification are used to evaluate different document representation methods, especially our proposed BoC representation. To prove the robustness of our model, we select the datasets that are diverse in the aspect of size, type and number of categories. The statistics of the six datasets are shown in Table 1.

20Newsgroups (20NG) dataset is a collection of newsgroup posts which contains 20 different categories describing different topics. There are around 20,000 documents in the dataset and partitioned (nearly) evenly into the 20 categories. We adopted the version of 20NG sorted by date, and the duplicates as well as some headers are removed.[6] Furthermore, we remove the “headers” and “footers” in each document to make the document classification task more realistic. Because our goal is to train a document classification system based on the documents themselves, not other information. Even though the information in the “headers” and “footers” of 20NG can improve the classification accuracy, the classifier can easily overfit to these information, resulting bad generalization on other documents without “headers” and “footers”.

Reuters_8 (R8) and *Reuters_52* (R52) are both generated from Reuters-21578 dataset which contains 21 578 articles on the Reuters newswire in 1987 and were manually classified into 90 categories by Reuters Ltd.[7] As we only consider single-labeled documents, we eliminated documents with no topics or more than one topics. Following the original train-test splitting, we kept the classes with at least one train and one test sample. Then we created two datasets R8 and R52, each containing 8 and 52 most frequent classes respectively.

BBC and *BBCSport* are originated from BBC News and BBC Sport websites respectively from year 2004 to 2005 [66].[8] *BBC* contains 2225 documents corresponding to stories in five topical areas: business, entertainment, politics, sport, and tech. *BBCSport* contains 737 documents corresponding to sports news articles in five topical areas: athletics, cricket, football, rugby, and tennis. Since train-test splitting were not specified for the two datasets, we splitted the datasets into training and test subsets equally.

Yahoo! Answers Topic (Yahoo) is a very large dataset which contains 1,460,000 samples obtained from ten largest categories of Yahoo! Answers Comprehensive Questions and Answers version 1.0, each category contains 140,000 training samples and 6000 testing samples [67]. We combine the question title, question content and the best answer as a document to classify its category.

5.2. Experiment settings

Our proposed BoC and BoCCl models are compared with the following benchmark document representation methods:

1. BoW: Bag-of-Words model which serves as baseline model in our experiment. Our implementation is based on *sklearn*.[9] Stop-words and punctuations are eliminated, and words with document frequency less than 2 are removed.
2. LSA: Latent Semantic Analysis [14]. Our implementation is based on *sklearn*.
3. LDA: Latent Dirichlet Allocation [16]. Our implementation is based on *sklearn*. We use the default document topic distribution prior α and topic word distribution prior η because we found they are the optimum hyper-parameter settings.
4. AE: Document representation by taking the average of word embeddings of all words in the document [23]. Pre-trained word embeddings *Word2Vec*[10] from Google is used. The word embeddings have a dimensionality of 300 and were trained on part of Google News dataset containing about 100 billion words.
5. Doc2Vec: Also known as paragraph vector model which learns paragraph and document embeddings via distributed memory (DM) and distributed bag of words (DBoW) models [24]. Our implementation is based on *Gensim*.[11] The dimensionality of embeddings is also set to 300, and DM model with negative sampling is used to learn document embeddings because neither DBoW nor the concatenation of DBoW and DM resulted in better performance for our document classification tasks.

[6] The dataset can be downloaded from http://qwone.com/~jason/20Newsgroups.

[7] The dataset can be downloaded from http://www.daviddlewis.com/resources/testcollections/reuters21578/.

[8] The datasets can be downloaded from http://mlg.ucd.ie/datasets/bbc.html.

[9] https://scikit-learn.org/stable/.

[10] https://code.google.com/archive/p/word2vec/.

[11] https://radimrehurek.com/gensim/index.html.

Table 2
Classification accuracies (%) of BoW, LSA and LDA using different weighting schemes.

Datasets	BoW				LSA				LDA			
	tf	tf_n	$tf-idf$	$tf-idf_n$	tf	tf_n	$tf-idf$	$tf-idf_n$	tf	tf_n	$tf-idf$	$tf-idf_n$
20NG	69.97	78.24	66.99	**80.32**	69.62	78.16	67.98	**80.24**	74.58	58.63	**75.25**	56.81
R8	95.93	97.26	96.21	**97.53**	96.48	96.99	96.62	**97.40**	**95.20**	91.14	94.20	90.45
R52	92.72	94.74	92.48	**95.33**	92.68	94.74	92.87	**95.29**	**90.30**	81.43	88.16	82.32
BBC	96.32	96.68	96.05	**96.86**	96.32	96.77	96.14	**97.21**	**95.15**	91.28	95.06	92.09
BBCSport	97.56	97.56	97.83	**98.37**	**98.64**	97.83	98.37	98.37	**92.41**	72.09	90.79	81.84

Table 3
Number of concepts and entities automatically acquired from Probase for six different document classification tasks.

Datasets	No. of classes	No. of concepts	No. of entities
BBCSport	5	1294	92 359
BBC	5	10 901	461 850
R8	8	14 848	457 931
Yahoo	10	25 672	417 683
20NG	20	21 642	586 078
R52	52	27 515	600 804

Table 4
Classification accuracies (%) of BoC using different weighting schemes.

Datasets	cs	cs_n	$cs-idf^s$	$cs-idf_n^s$
20NG	66.75	72.29	65.49	**74.35**
R8	94.20	94.75	93.79	**95.43**
R52	88.98	90.81	89.21	**91.94**
BBC	94.98	95.27	96.23	**96.51**
BBCSport	88.86	88.10	91.24	**93.86**

For our proposed BoC and BoCCl models, concepts or concept clusters with soft document frequency less than 1 are removed. We use $k = 0.0001$ in the smoothed typicality score since it will affect the typicality score significantly if the value of k is too large. k_c is set to 20 to select top 20 concepts or concept clusters during entity conceptualization, and $maxLen$ is set to 3 for entity and concept recognition. For BoCCl model, the inter-cluster distance of the agglomerative clustering algorithm is evaluated using average linkage, which is the average of the distances between all observations of pairs of clusters. We choose a cluster size of 1000 for BBCSport dataset and 5000 for other datasets. For entity sense disambiguation, the entropy threshold h_t is set to be 2.5 based on empirical studies. We also investigate the combination of BoCCl and BoW representations. Two combination methods are studied: direct concatenation or combine using attention mechanism.

Linear support vector machine (SVM) [68] is used as document classifier based on the document representations learned from the aforementioned methods. When training the SVM classifier, one-vs-rest training strategy is applied for multi-class case, and square of hinge loss as well as L2 penalty are used. For BoCCl+BoW model using attention mechanism, neural network and softmax classifier are used for learning the attentional combination of BoCCl and BoW representations. The dimensionality of context vector as well as hidden representations are set to 200. The model is trained by minimizing the categorical cross entropy loss function using mini-batch stochastic gradient descent (SGD) with the Adam update rule [69]. However, for fair comparison, we used the same SVM classifier for the final classification of documents based on the combined representations learned (before softmax layer).

5.3. Results and analyses

5.3.1. BoC knowledge acquisition results

We first evaluated the number of related concepts and entities automatically acquired from the original knowledge base which consists of over 5 million concepts and 12 million entities. Table 3 shows the number of concepts and entities acquired from Probase for different document classification tasks.

It is observed that with increasing number of document classes, the amount of concepts and entities automatically acquired from Probase also increases. This is because more conceptual knowledge is needed for document classification tasks with many categories. Besides, more concepts and entities are acquired for general categories than specific categories. This explains the reason why BBC acquired more concepts and entities than BBCSport, and Yahoo acquired more concepts than 20NG, since the categories in BBC and Yahoo are more general. Compared with the original size of the knowledge base, the amount of acquired concepts and entities are significantly reduced, resulting a much more smaller search space. Besides, the noises in the model are reduced since irrelevant concepts and entities are eliminated.

5.3.2. Weighting schemes evaluation

Besides determining the indexing terms or features for document representation, the term weighting scheme which assigns weight for each term also plays an important role in document representation. In this experiment, we evaluated various weighting schemes for BoW, LSA, LDA, as well as the proposed BoC model.

For LSA and LDA, the weighting schemes of the input matrix were studied, which is the term–document matrix for dimensionality reduction. The optimal number of latent dimensions/topics for each dataset was found by grid search, and the best classification accuracy was reported. Table 2 shows the performance of BoW, LSA and LDA using different weighting schemes including term frequency (tf), term frequency–inverse document frequency ($tf-idf$), as well as L2 normalized tf (tf_n) and tf-idf ($tf-idf_n$).[12] The results of L1 normalization are not reported because the performance of L1 normalization is much more worse than L2 normalization. From Table 2, it can be concluded that for BoW and LSA, L2 normalization can improve the performance significantly, and tf-idf weighting with L2 normalization achieved the best performance for document classification; whereas for LDA, tf weighting achieved the best performance, and normalization of term–document matrix will degrade the model performance significantly since LDA depends on count-based term–document statistics to infer probabilities. Although other weighting schemes may achieve the best performance for some particular datasets due to the special characteristics of the dataset such as size and word distribution, the performance difference with the best weighting scheme is marginal.

Table 4 shows the performance of BoC using different weighting schemes including concept score (cs), concept score-inverse soft document frequency ($cs-idf^s$), as well as L2 normalized cs (cs_n) and $cs-idf^s$ ($cs-idf_n^s$). Similar as BoW and LSA, L2 normalization improves the performance of BoC significantly. Besides, incorporating inverse soft document frequency also improves the

[12] Yahoo dataset is not used in this experiment because it requires significant amount of time and memory to obtain a single result.

Table 5
Classification accuracies (%) of various models on six document classification datasets.

Datasets	BoW	LSA	LDA	AE	Doc2Vec	BoC	BoCCl	BoCCl+BoW (concat)	BoCCl+BoW (atten)
20NG	80.32	80.24	75.25	68.70	68.36	74.35	75.52	80.68	**81.78**
R8	97.53	97.40	95.20	96.98	94.95	95.43	96.69	97.62	**97.85**
R52	95.33	95.29	90.30	92.72	91.07	91.94	92.25	95.27	**95.87**
BBC	96.86	97.21	95.15	96.41	96.13	96.51	97.09	97.53	**98.04**
BBCSport	98.37	98.64	92.41	98.10	97.37	93.86	94.33	98.30	**99.08**
Yahoo	72.10	72.13	69.91	68.00	66.46	70.02	71.15	73.88	**75.26**

Table 6
Pair-wise comparison of algorithms using Wilcoxon Signed-Rank Test with $\alpha = 0.05$. Z statistics are shown above the diagonal, bold indicates significantly different between algorithms with $z < -1.96$. Comparisons of the algorithms in the row with the column are shown below the diagonal, + and − indicate significantly better or worse respectively; = indicates no statistical significant difference.

	BoW	LSA	LDA	AE	Doc2Vec	BoC	BoCCl	Concat	Atten
BoW		−0.31	−**2.20**	−**2.20**	−**2.20**	−**2.20**	−**1.99**	−1.57	−**2.20**
LSA	=		−**2.20**	−**2.20**	−**2.20**	−**2.20**	−**2.20**	−1.15	−**2.20**
LDA	−	−		−0.31	−0.10	−1.57	−**2.20**	−**2.20**	−**2.20**
AE	−	−	=		−**2.20**	−0.10	−0.52	−**2.20**	−**2.20**
Doc2Vec	−	−	=	−		−1.36	−1.36	−**2.20**	−**2.20**
BoC	−	−	=	=	=		−**2.20**	−**2.20**	−**2.20**
BoCCl	−	−	+	=	=	+		−**2.20**	−**2.20**
Concat	=	=	+	+	+	+	+		−**2.20**
Atten	+	+	+	+	+	+	+	+	

performance since it is able to reduce the weights of frequent but less informative concepts for document classification. Therefore, the L2 normalized $cs - idf^s$ is the best weighting scheme for BoC model.

5.3.3. Statistical quality evaluation

We compared our proposed BoC models with several other benchmark document representation methods as mentioned in Section 5.2. For BoW, LSA, LDA, BoC and BoCCl, the best weighting scheme of each model was used. Particularly, for LSA and LDA, the optimal number of latent dimensions were used and the best results are reported. For Doc2Vec, we tuned the negative sampling rate, number of training epochs, as well as initial and minimum learning rates, and the best results are reported. Table 5 shows the classification accuracies of all the above mentioned models on six real-life document classification datasets. We also perform pair-wise comparison of algorithms using Wilcoxon Signed-Rank Test [70] to test the statistical significance of our results, the test results are shown in Table 6.

We found that the classification accuracy of BoW (with L2 normalized tf-idf weighting) outperforms many other document representations, indicating that BoW is a very strong baseline for document classification tasks. The reason is that BoW can effectively capture word-level statistics of a document, and majority documents can be classified into correct categories based on these word-level statistics. Even though high dimensionality and sparsity are associated with BoW representations, SVM classifier is very robust to the dimensionality and sparsity of feature vectors and is able tolerate irrelevant information effectively [71]. However, unlike our proposed BoC model, BoW cannot capture the semantic relatedness and conceptual information of texts, and also losses word order and phrasal information. The conceptual information are very important for making document classification decisions and it is the key to further improve the document classification performance.

LSA and LDA are both dimensionality reduction techniques based on BoW representations. The performance of LSA is comparable with BoW, with the cost of searching the optimal number of latent dimensions. The performance of LDA is worse than BoW and LSA, because it is a generative model that focuses on the generative process of document collections instead of discrimination of document categories. The latent dimensions or topics of LSA and LDA are derived from word frequencies or co-occurrences, hence the semantic relatedness and conceptual information are also ignored. Besides, LSA and LDA are highly specific to the training corpus, resulting in bad generalization capability. On the contrary, our proposed BoC model utilizes external probabilistic encyclopedic knowledge base to capture the semantic relatedness and conceptual information of texts. The higher level semantics (i.e. concepts) captured in BoC are more reliable and robust since the conceptualization is based on human knowledge.

AE and Doc2Vec are both document representation methods based on distributed representation of words or documents. Results show that AE cannot achieve comparable results as BoW model, because the averaging operation of the word embeddings is too simple to capture the complex semantics of documents. For Doc2Vec, complicated hyper-parameter tuning is needed to achieve better performance. However, the results of Doc2Vec are even worse than AE for our document classification tasks. The reason is that AE utilizes pre-trained word embeddings that are learned from large corpus, whereas Doc2Vec needs to learn embeddings from scratch based on our small and domain specific corpus, hence it is very hard for Doc2Vec to effectively capture the semantic information on document embeddings.

Compare our proposed BoC and BoCCl models, results show that BoCCl substantially outperforms BoC. The reason of the improvement is two-folds: one is that the concept clustering algorithm of BoCCl groups similar concepts into clusters. Hence, redundant concepts are reduced, and the dimensionality as well as sparsity of the feature space are reduced significantly, making the training on small datasets more efficient. The other is that BoCCl utilizes context information to perform entity sense disambiguation, making the conceptualization process more accurate. Even though the performances of BoCCl is better than LDA, and also comparable with AE, they are not as good as BoW representation. This is because that BoCCl focuses on the conceptual information of entities within the document, but ignores part of the word-level information which is also important for determining the document categories. On the contrary, BoW focuses only on the word-level information but ignore the concept-level information of texts. Hence, the two representations complement with each other, and once combined properly, the best document classification performance can be achieved. As shown in Table 5, the combination of BoCCl and BoW representations achieved the best classification results. The direct concatenation of BoCCl and BoW results in little improvement, the reason may be that the two representations use different weighting scheme and the dimensionality of BoW is significantly larger than BoCCl. However, when we combine the two representations using attention mechanism, the final representation is able to select word-level and concept-level information attentionally for different classification tasks and the performance improved significantly.

Table 7
Dimensionality of different document representations learned from six datasets.

Datasets	BoW	LSA	LDA	BoC	BoCCl
20NG	49 241	8000	6000	12 035	4168
R8	10 553	700	600	5776	2997
R52	12 029	5000	4000	8624	3172
BBC	12 150	500	1000	4662	2891
BBCSport	5198	100	80	476	419
Yahoo	32 1210	6000	5000	23 501	4939

Table 8
Top 3 features of different representations for two documents from “sport” and “entertainment” categories respectively.

Models	Document 1 (sport)
BoW	Villa, cole, leary
LSA	l_{110}: ibm, patents, villa, fiat, students l_{109}: children, lee, england, kennedy, spider l_{111}: children, villa, israel, radio, award
LDA	t_{447}: ankle, villa, cole, injury, six t_{586}: villa, leary, us, southampton, three t_{768}: foot, robben, weeks, injury, six
BoC	Club: manchester united, chelsea, arsenal, real madrid, liverpool Operation: welding, sawing, copying, drilling, burning Player: lebron james, kobe bryant, cristiano ronaldo, messi
BoCCl	Club: manchester united, chelsea, arsenal, real madrid, liverpool Operation: welding, sawing, copying, drilling, burning Player: lebron james, kobe bryant, cristiano ronaldo, messi

Models	Document 2 (entertainment)
BoW	Wars, lucas, rating
LSA	l_0: said, mr, would, people, year l_{347}: green, galloway, hybrid, glazer, song l_4: film, best, awards, award, economy
LDA	t_{482}: film, awards, said, star, nominations t_{618}: film, michael, said, life, different t_{327}: morrison, album, label, record, mark
BoC	Film: star wars, matrix, panic room, sin city, harry potter Movie: star wars, matrix, jurassic park, avatar, terminator Star war character: darth vader, luke skywalker, han solo, r2d2
BoCCl	Movie: star wars, matrix, jurassic park, harry potter, avatar Star war character: darth vader, luke skywalker, han solo, r2d2 Rating: default, star, high, low, medium

5.3.4. *Semantic quality evaluation*

We first evaluated the dimensionalities of the document representations learned using the above mentioned approaches. High dimensional feature vector means using a large amount of features to represent a document, which will affect both semantic interpretability and statistical quality. Although SVM is robust to the dimensionality of feature vectors, this is not the case for other classifiers where feature selection is needed to reduce the feature space to manageable dimensions. Besides, it is computationally cheaper to operate the classifier in low dimensional spaces.

Among the above mentioned models, AE and Doc2Vec have the lowest dimensionality, which is equal to the dimensionality of word embeddings (300). However, the features are derived from distributed representation of words, and they are not interpretable. Hence, is it very hard for human to discover the semantic meanings from the representations. Table 7 shows the representation dimensionality of BoW, LSA, LDA, BoC and BoCCl learned from different document classification tasks. For LSA and LDA, the dimensionality is equal to the number of latent dimensions/topics, which is a pre-defined hyperparameter. We report the dimensionalites of LSA and LDA which achieved the best performance on each dataset. Results show that LSA and LDA have the lowest dimensionalities, however, the optimal dimensionality differs a lot for different datasets, hence careful hyper-parameter tuning is needed. The dimensionality of BoC is much lower than BoW, since BoC is operated in the concept space which has much lower dimensionality than word space. BoCCl further reduces the dimensionality by clustering similar concepts together, resulting more appropriate dimensionalities.

In order to show the representation interpretability of different models, we discovered the top-weighted features from different representations. We took two documents from different categories as examples. The first document is from “sport” category and the second document is from “entertainment” category of BBC dataset, as shown below:

1. Aston Villa’s Carlton Cole could be out for six weeks with a knee injury. The striker, who is on a season-long loan from Chelsea, picked up the knock in an England Under-21 match against Holland earlier this month. “Carlton will be out of action for four to six weeks after a bad challenge”, said Villa boss David O’Leary. “I won’t be able to tell you whether he will need an operation until maybe next week. Whether he has an operation has got to be left to Chelsea”. Cole, who also struggled with an ankle problem earlier in the season, was unable to rest because O’Leary had a shortage of strikers. The return to fitness of Darius Vassell after four months out with a broken ankle and the emergence of Luke Moore has alleviated some of the Villa’s manager’s problems in that department.
2. The sixth and final Star Wars movie may not be suitable for young children, film-maker George Lucas has said. He told US TV show 60 Minutes that Revenge of the Sith would be the darkest and most violent of the series. “I don’t think I would take a five or six-year-old to this”, he told the CBS programme, to be aired on Sunday. Lucas predicted the film would get a US rating advising parents some scenes may be unsuitable for under-13s. It opens in the UK and US on 19 May. He said he expected the film would be classified PG-13 — roughly equivalent to a British 12A rating. The five previous Star Wars films have all carried less restrictive PG – parental guidance – ratings in the US. In the UK, they have all been passed U – suitable for all – with the exception of Attack of The Clones, which got a PG rating in 2002. Revenge of the Sith – the third prequel to the original 1977 Star Wars film – chronicles the transformation of the heroic Anakin Skywalker into the evil Darth Vader as he travels to a Hell-like planet composed of erupting volcanoes and molten lava. “We’re going to watch him make a pact with the devil”, Lucas said. “The film is much more dark, more emotional. It’s much more of a tragedy”.

Table 8 shows the top 3 weighted features of different representations for the two documents. For BoW, the top-weighted words are shown directly. For LSA and LDA, the top-weighted latent dimensions/topics are recorded, and the meaning of each latent dimension/topic is shown using the top words it covers. For BoC and BoCCl, the name of the top-weighted concepts/concept clusters are recorded, and the meaning of each concept/concept cluster is shown using the top entities it contains. From the two examples, it is obvious that BoC and BoCCl provide much greater interpretability than other representations. BoW only provides sparse words from the document, which is very hard to discover the meaning of the document. LSA performs linear combination of words which will benefit statistical discrimination, however, semantic interpretation of each latent dimension is loss. LDA is a generative model that discovers document-topic and topic-word distributions based on word co-occurrences, and it provides better interpretability than LSA. However, the topics of LDA still lack

Please cite this article as: P. Li, K. Mao, Y. Xu et al., Bag-of-Concepts representation for document classification based on automatic knowledge acquisition from probabilistic knowledge base, Knowledge-Based Systems (2020) 105436, https://doi.org/10.1016/j.knosys.2019.105436.

Table 9
Classification accuracy (%) of 50 samples from BBC dataset based on human judgment of the top 5 weighted features of each algorithm. "avg" and "std" indicate average and standard deviation values of the three evaluators respectively.

Evaluator	BoW	LSA	LDA	BoC	BoCCl
No. 1	66	48	74	74	76
No. 2	46	42	58	70	70
No. 3	62	62	68	70	72
avg ± std	58.0 ± 10.6	50.7 ± 10.3	66.7 ± 8.1	71.3 ± 2.3	72.7 ± 3.1

of intuitive interpretations because of the error-prone corpus-based inferences. Besides, these models only consider the surface form of words, hence the semantic and conceptual information of words and phrases cannot be captured.

Our proposed BoC and BoCCl conceptualize documents into concepts based on external knowledge base, the concepts reveal the higher-level semantic and conceptual meanings of texts and provide intuitive interpretations. For example, in Document 1, our model is able to recognize "Aston Villa" as "club" and "Carlton Cole" as "player", which is impossible for other models. Contrary to LSA and LDA, the meaning of each concept is explicitly given, and the entities involved are reliable and understandable as they are inferred based on human knowledge. The difference between BoC and BoCCl is that semantic similar concepts are grouped together in BoCCl to further reduce the concept space dimensionality. For example, in Document 2, the concepts "film" and "movie" are grouped into a single concept cluster "movie".

We further conduct quantitative evaluation to prove the better semantic interpretability of our model. We randomly sample 50 documents from BBC dataset for human evaluation. The experiment is conducted on three evaluators and only the top 5 weighted features of each sample are shown to the evaluators for them to determine the category of the document. For BoW, the top-weighted words are shown directly; for LSA and LDA, the meaning of each top-weighted latent dimensions/topics is shown using the top words it covers; for BoC and BoCCl, the top-weighted concepts/concept clusters are shown. Table 9 shows the classification accuracy of different algorithms based on human judgment of the top 5 weighted features. Evaluation results show that the proposed BoC and BoCCl are the most interpretable representations which achieve higher evaluation accuracy than other representations. Besides, the lower standard deviation of BoC and BoCCl also demonstrates that the understanding of the semantic meaning of features is more consistent across different evaluators. With the interpretable BoC and BoCCl representations, we are able to have a deeper understanding of documents and a clearer operation logic for further reasoning and decision making.

6. Conclusion

In this paper, we propose a novel Bag-of-Concepts (BoC) representation for document classification. The useful conceptual knowledge for certain document classification task is automatically acquired from a large probabilistic knowledge base. Based on these conceptual knowledge, the entities (words and phrases) in the document are conceptualized into concepts in a probabilistic way, which capture their semantic relatedness and higher-level conceptual meanings. By aggregating the conceptualized entities within the document, a document is conceptualized into relevant concepts and eventually being represented as a distributed vector in the learned concept space using the proposed concept score-inverse soft document frequency ($cs-idf^s$) weighting scheme. We also propose Bag-of-Concept-Clusters (BoCCl) model to further improve BoC representation by grouping semantically similar concepts into clusters and performing entity sense disambiguation. Furthermore, we combine the BoCCl and BoW representations using an attention mechanism to effectively utilize both concept-level and word-level information to achieve optimal performance for document classification.

Both statistical and semantic qualities of the proposed BoC and BoCCl representations are evaluated and compared with other benchmark document representations. Experiment results show that BoC and BoCCl are able to effectively capture the concept-level information of documents and improve the performance of document classification. More importantly, BoC and BoCCl provide great interpretabilities which allow human to have a deeper understanding of the document and a clearer operation logic for further reasoning and decision making. As a future work, effectively utilizing BoC/BoCCl for sentence-level representation are to be studied. Particularly, we plan to incorporate conceptual knowledge into deep neural networks for better semantic understandings of natural language while preserving model interpretability.

References

[1] T.A. Almeida, T.P. Silva, I. Santos, J.M.G. Hidalgo, Text normalization and semantic indexing to enhance instant messaging and sms spam filtering, Knowl.-Based Syst. 108 (2016) 25–32.

[2] J. Dang, M. Kalender, C. Toklu, K. Hampel, Semantic search tool for document tagging, indexing and search, Google Patents, 2017, US Patent 9, 684, 683.

[3] Y. Xiao, B. Liu, J. Yin, Z. Hao, A multiple-instance stream learning framework for adaptive document categorization, Knowl.-Based Syst. 120 (2017) 198–210.

[4] S. Joty, G. Carenini, R.T. Ng, Topic segmentation and labeling in asynchronous conversations, J. Artificial Intelligence Res. 47 (2013) 521–573.

[5] G. Lee, J. Jeong, S. Seo, C. Kim, P. Kang, Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network, Knowl.-Based Syst. 152 (2018) 70–82.

[6] Y. Li, H. Guo, Q. Zhang, M. Gu, J. Yang, Imbalanced text sentiment classification using universal and domain-specific knowledge, Knowl.-Based Syst. 160 (2018) 1–15.

[7] W. Zhang, T. Yoshida, X. Tang, A comparative study of tf* idf, lsi and multi-words for text classification, Expert Syst. Appl. 38 (3) (2011) 2758–2765.

[8] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 655–665.

[9] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1422–1432.

[10] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489.

[11] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[12] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, Technical report, OpenAI, 2018.

[13] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Vol. 1, 2018, pp. 2227–2237.

[14] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Amer. Soc. Inf. Sci. 41 (6) (1990) 391–407.

[15] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Mach. Learn. 42 (1–2) (2001) 177–196.

[16] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.

[17] M. Sahlgren, R. Cöster, Using bag-of-concepts to improve the performance of support vector machines in text categorization, in: Proceedings of the 20th International Conference on Computational Linguistics, Association for Computational Linguistics, 2004, p. 487.

Please cite this article as: P. Li, K. Mao, Y. Xu et al., Bag-of-Concepts representation for document classification based on automatic knowledge acquisition from probabilistic knowledge base, Knowledge-Based Systems (2020) 105436, https://doi.org/10.1016/j.knosys.2019.105436.

[18] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (Feb) (2003) 1137–1155.

[19] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (Aug) (2011) 2493–2537.

[20] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.

[21] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[22] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.

[23] C. Xing, D. Wang, X. Zhang, C. Liu, Document classification with distributions of word vectors, in: Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, IEEE, 2014, pp. 1–5.

[24] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, 2014, pp. 1188–1196.

[25] R. Zhao, K. Mao, Fuzzy bag-of-words model for document representation, IEEE Trans. Fuzzy Syst. 26 (2) (2018) 794–804.

[26] H.K. Kim, H. Kim, S. Cho, Bag-of-concepts: Comprehending document representation through clustering words in distributed representation, Neurocomputing 266 (2017) 336–352.

[27] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, Vol. 1, MIT press Cambridge, 2016.

[28] Y. Kim, Convolutional Neural Networks for Sentence Classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.

[29] R. Socher, B. Huval, C.D. Manning, A.Y. Ng, Semantic compositionality through recursive matrix-vector spaces, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, 2012, pp. 1201–1211.

[30] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[32] C. Fellbaum, Wordnet, in: Theory and Applications of Ontology: Computer Applications, Springer, 2010, pp. 231–243.

[33] A. Esuli, F. Sebastiani, Sentiwordnet: a high-coverage lexical resource for opinion mining, Evaluation 17 (2007) 1–26.

[34] K.K. Schuler, Verbnet: A broad-coverage, comprehensive verb lexicon, 2005.

[35] J. Ruppenhofer, M. Ellsworth, M.R. Petruck, C.R. Johnson, J. Scheffczyk, FrameNet II: Extended theory and practice, Institut für Deutsche Sprache, Bibliothek, 2016.

[36] A. Hotho, S. Staab, G. Stumme, Ontologies improve text document clustering, in: Third IEEE International Conference on Data Mining, IEEE, 2003, pp. 541–544.

[37] Q. Luo, E. Chen, H. Xiong, A semantic term weighting scheme for text categorization, Expert Syst. Appl. 38 (10) (2011) 12708–12716.

[38] M. Liu, G. Haffari, W. Buntine, M. Ananda-Rajah, Leveraging linguistic resources for improving neural text classification, in: Proceedings of the Australasian Language Technology Association Workshop 2017, 2017, pp. 34–42.

[39] Y. Wang, M. Wang, H. Fujita, Word sense disambiguation: A comprehensive knowledge exploitation framework, Knowl.-Based Syst. (2019) http://dx.doi.org/10.1016/j.knosys.2019.105030.

[40] T.T. Thet, J.-C. Na, C.S. Khoo, Aspect-based sentiment analysis of movie reviews on discussion boards, J. Inf. Sci. 36 (6) (2010) 823–848.

[41] C. Hung, S.-J. Chen, Word sense disambiguation based sentiment lexicons for sentiment classification, Knowl.-Based Syst. 110 (2016) 224–232.

[42] F.H. Khan, U. Qamar, S. Bashir, A semi-supervised approach to sentiment analysis using revised sentiment strength based on sentiwordnet, Knowl. Inf. Syst. 51 (3) (2017) 851–872.

[43] P. Li, K. Mao, Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts, Expert Syst. Appl. 115 (2019) 512–523.

[44] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: The Semantic Web, Springer, 2007, pp. 722–735.

[45] F.M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, in: Proceedings of the 16th International Conference on World Wide Web, ACM, 2007, pp. 697–706.

[46] W. Wu, H. Li, H. Wang, K.Q. Zhu, Probase: A probabilistic taxonomy for text understanding, in: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, ACM, 2012, pp. 481–492.

[47] X. Hu, N. Sun, C. Zhang, T.-S. Chua, Exploiting internal and external semantics for the clustering of short texts using world knowledge, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM, 2009, pp. 919–928.

[48] E. Gabrilovich, S. Markovitch, Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge, in: AAAI, Vol. 6, 2006, pp. 1301–1306.

[49] P. Wang, J. Hu, H.-J. Zeng, Z. Chen, Using wikipedia knowledge to improve text classification, Knowl. Inf. Syst. 19 (3) (2009) 265–281.

[50] A. Alahmadi, A. Joorabchi, A.E. Mahdi, A new text representation scheme combining bag-of-words and bag-of-concepts approaches for automatic text classification, in: GCC Conference and Exhibition (GCC), 2013 7th IEEE, IEEE, 2013, pp. 108–113.

[51] J.-H. Lee, J. Ha, J.-Y. Jung, S. Lee, Semantic contextual advertising based on the open directory project, ACM Trans. Web (TWEB) 7 (4) (2013) 24.

[52] H. Shin, G. Lee, W.-J. Ryu, S. Lee, Utilizing wikipedia knowledge in open directory project-based text classification, in: Proceedings of the Symposium on Applied Computing, ACM, 2017, pp. 309–314.

[53] R.A. Sinoara, J. Camacho-Collados, R.G. Rossi, R. Navigli, S.O. Rezende, Knowledge-enhanced document embeddings for text classification, Knowl.-Based Syst. 163 (2019) 955–971.

[54] L. Yao, Y. Zhang, B. Wei, H. Qian, Y. Wang, Incorporating probabilistic knowledge into topic models, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2015, pp. 586–597.

[55] Y.-K. Tang, X.-L. Mao, H. Huang, X. Shi, G. Wen, Conceptualization topic modeling, Multimedia Tools Appl. 77 (3) (2018) 3455–3471.

[56] Y. Li, B. Wei, Y. Liu, L. Yao, H. Chen, J. Yu, W. Zhu, Incorporating knowledge into neural network for text representation, Expert Syst. Appl. 96 (2018) 103–114.

[57] W. Shalaby, W. Zadrozny, H. Jin, Beyond word embeddings: learning entity and concept representations from large scale knowledge bases, Inf. Retr. J. (2018) 1–18.

[58] Y. Song, H. Wang, Z. Wang, H. Li, W. Chen, Short text conceptualization using a probabilistic knowledgebase, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Three, AAAI Press, 2011, pp. 2330–2336.

[59] F. Wang, Z. Wang, Z. Li, J.-R. Wen, Concept-based short text classification and ranking, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM, 2014, pp. 1069–1078.

[60] W. Hua, Z. Wang, H. Wang, K. Zheng, X. Zhou, Short text understanding through lexical-semantic analysis, in: Data Engineering (ICDE), 2015 IEEE 31st International Conference on, IEEE, 2015, pp. 495–506.

[61] M.A. Hearst, Automatic acquisition of hyponyms from large text corpora, in: Proceedings of the 14th Conference on Computational Linguistics-Volume 2, Association for Computational Linguistics, 1992, pp. 539–545.

[62] Z. Wang, H. Wang, J.-R. Wen, Y. Xiao, An inference approach to basic level of categorization, in: Proceedings of the 24th Acm International on Conference on Information and Knowledge Management, ACM, 2015, pp. 653–662.

[63] M.-h. Chen, J.-h. Yin, C. Shu, M.-j. Wang, A chinese word segmentation system design based on forward-backward maximum matching algorithm, Inf. Technol. 6 (2009) 042.

[64] L. Rokach, O. Maimon, Clustering methods, in: Data Mining and Knowledge Discovery Handbook, Springer, 2005, pp. 321–352.

[65] J.R. Firth, A synopsis of linguistic theory, 1930-1955, Stud. Linguist. Anal. (1957).

[66] D. Greene, P. Cunningham, Practical solutions to the problem of diagonal dominance in kernel document clustering, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 377–384.

[67] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems, 2015, pp. 649–657.

[68] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: A library for large linear classification, J. Mach. Learn. Res. 9 (Aug) (2008) 1871–1874.

[69] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[70] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.

[71] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in: European Conference on Machine Learning, Springer, 1998, pp. 137–142.