Automated Social Text Annotation With Joint Multilabel Attention Networks

Automated Social Text Annotation With Joint Multilabel Attention Networks

Hang Dong, Wei Wang, Kaizhu Huang, *Member, IEEE*, and Frans Coenen

***Abstract*— Automated social text annotation is the task of suggesting a set of tags for shared documents on social media platforms. The automated annotation process can reduce users' cognitive overhead in tagging and improve tag management for better search, browsing, and recommendation of documents. It can be formulated as a multilabel classification problem. We propose a novel deep learning-based method for this problem and design an attention-based neural network with semantic-based regularization, which can mimic users' reading and annotation behavior to formulate better document representation, leveraging the semantic relations among labels. The network separately models the title and the content of each document and injects an explicit, title-guided attention mechanism into each sentence. To exploit the correlation among labels, we propose two semantic-based loss regularizers, i.e., similarity and subsumption, which enforce the output of the network to conform to label semantics. The model with the semantic-based loss regularizers is referred to as the joint multilabel attention network (JMAN). We conducted a comprehensive evaluation study and compared JMAN to the state-of-the-art baseline models, using four large, real-world social media data sets. In terms of F_1, JMAN significantly outperformed bidirectional gated recurrent unit (Bi-GRU) relatively by around 12.8%–78.6% and the hierarchical attention network (HAN) by around 3.9%–23.8%. The JMAN model demonstrates advantages in convergence and training speed. Further improvement of performance was observed against latent Dirichlet allocation (LDA) and support vector machine (SVM). When applying the semantic-based loss regularizers, the performance of HAN and Bi-GRU in terms of F_1 was also boosted. It is also found that dynamic update of the label semantic matrices (JMAN$_d$) has the potential to further improve the performance of JMAN but at the cost of substantial memory and warrants further study.**

***Index Terms*— Attention mechanisms, automated social annotation, deep learning, multilabel classification, recurrent neural networks (RNNs).**

I. Introduction

TAGGING is a popular approach to organize various resources on many social media platforms, which allows users to share and annotate resources with their own vocabularies. In academic social bookmarking systems, such as Bibsonomy (http://bibsonomy.org) and CiteULike (http://citeulike.org), tags are used to organize academic publications; on social question & answering (Q&A) sites, such as Quora (http://quora.com), StackOverFlow (https://stackoverflow.com), and Zhihu (https://zhihu.com/), tags are associated with questions for better search and recommendation; in microblogging services, such as Twitter (https://twitter.com), tags are in the form of hashtags to produce alternative access points to tweets. These accumulated tags are commonly referred to as Folksonomies, which have been used for organizing online resources [1], browsing [2], semantic-based search and recommendation [3], and learning knowledge structures [4]. It is also reported that tags have higher descriptive and discriminative power compared with other textual features, such as titles, descriptions, and comments, for document classification [5]. Fig. 1 shows an example of a published article and its associated tags on Bibsonomy.

Many shared online documents are, however, not annotated, for example, on Zhihu, more than 18% of questions are not associated with any tags, as reported in [6]. Moreover, many user-generated tags are noisy and of low quality. These problems can be alleviated to a great extent by automated annotation, which learns to assign a set of meaningful tags for (unannotated) documents. The perceived benefits include efficient annotation, tag reuse, and easy maintenance of the quality of folksonomies [7].

Automatic social annotation is highly relevant to "tag recommendation" in the literature [8], which suggests tags from the list of candidates for different objects to support overall resource organization. Previous studies applied term frequency-based lexical features [9], adaptive hypergraph learning [6], and probabilistic graphical models [10], [11] to model the automated tagging process. Recent studies explored the use of deep learning [12]–[16], which encodes the input texts as continuous vector representations and approximates the matching from the input to the label space, where labels are often assumed to be orthogonal or independent to each other.

Manuscript received August 30, 2019; revised March 5, 2020; accepted June 11, 2020. This work was supported in part by the Research Development Fund at Xi'an Jiaotong-Liverpool University (XJTLU) under Contract RDF-14-01-10, in part by the National Natural Science Foundation of China under Grant 61876155, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20181189, and in part by the Key Program Special Fund in XJTLU under Grant KSF-A-01, Grant KSF-T-06, Grant KSF-E-26, Grant KSF-P-02, and Grant KSF-A-10. The work of Hang Dong was supported by the Human Phenotype Project in Health Data Research UK Scotland. *(Corresponding author: Wei Wang.)*

Hang Dong is with the Department of Computer Science, University of Liverpool, Liverpool L69 7ZX, U.K., also with the Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China, and also with the Centre for Medical Informatics, Usher Institute, The University of Edinburgh, Edinburgh EH16 4UX, U.K. (e-mail: hangdong@liverpool.ac.uk).

Wei Wang is with the Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (e-mail: wei.wang03@xjtlu.edu.cn).

Kaizhu Huang is with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China, and also with the Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310000, China (e-mail: kaizhu.huang@xjtlu.edu.cn).

Frans Coenen is with the Department of Computer Science, University of Liverpool, Liverpool L69 7ZX, U.K. (e-mail: coenen@liverpool.ac.uk).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3002798

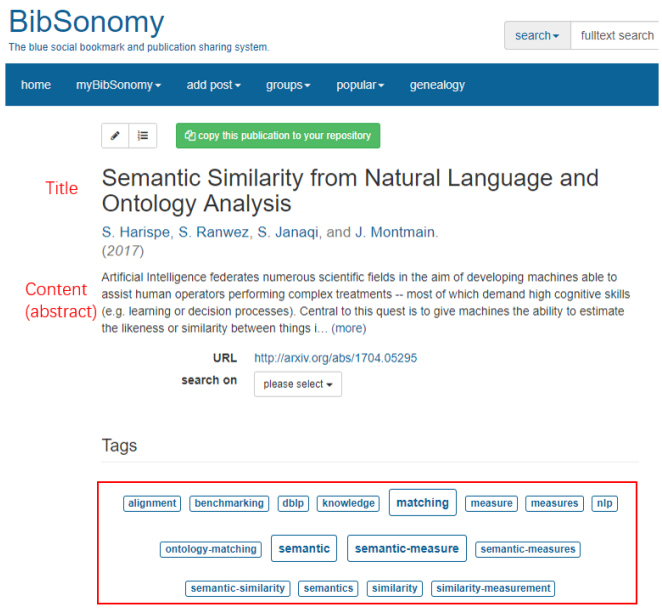


Fig. 1. Example of a document and its associated metadata and tags on Bibsonomy. The metadata consists of title and the content (i.e., abstract of this article). Tags are surrounded with a red box.

Our study shows that the existing deep learning-based methods at least suffer two issues.

1) *Modeling of Reading and Annotation Behavior (Encoding):* In encoding, mainstream methods simply scan the texts in the document and do not fully model the way how users read and annotate it. Recurrent neural networks (RNNs) typically encode a sequence of text one word by another into a fixed-length vector while not considering the internal structure of documents. The hierarchical attention network (HAN) [17] models the hierarchical (word-sentence) structure of a document; however, it does consider how a document is annotated by a human user with the presence of different metadata, e.g., a user may digest the title before reading the document. Studies have explored the impact and importance of title on users' annotation choice [18], document categorization, and tag recommendation [5].
2) *Semantics in the Labels (Label Correlation):* In prediction, the most common multihot (as opposed to one-hot) representation for each label set [19] assumes orthogonality among labels and does not consider their correlation, which represents the semantic relations among tags. However, it is a key issue in multilabel classification, especially when the label size is large [20], [21]. Studies show that co-occurring tags in documents often exhibit similarity or subsumption relations [22], [23].

We present a novel deep learning framework to seamlessly integrate users' reading and annotation behavior in the encoding and prediction for automated annotation, leveraging the guided attention mechanisms and label correlation encoded in external knowledge sources. We propose a new attention mechanism to simulate users' reading behavior. To annotate a document, a user attempts to digest the meaning of the title first and then, based on her or his understanding, proceeds to the content (e.g., abstract of the document). The key is the use of a title-guided attention mechanism that allows the meaning of the title to govern the "reading" of each sentence to form a final representation of the document. The idea is different from the attention mechanism used in the HAN model, which is implemented through an implicit vector. In our approach, it is realized through a dynamic alignment of title and sentences, which also enables better explainability in modeling and visualization.

Current studies mostly consider the symmetric, similarity relation among labels [24]–[26]. The asymmetric relation, i.e., subsumption, among labels needs further exploration, as suggested in [25]. To incorporate both types of label semantics in one deep network, we propose two semantic-based loss regularizers to constrain the network output to satisfy the similarity and subsumption relations among labels. The regularizers allow the model to leverage semantic relations that can be either matched to existing knowledge bases or inferred from data sets. We further explore the dynamic update of the semantic relations when optimizing the loss regularizers.

The main contributions of the work are highlighted as follows.

1) We propose a joint multilabel attention network (JMAN) that models users' reading and annotation behavior through title-guided attention mechanisms to encode the document.
2) We propose two semantic-based loss regularizers to enforce the output of the neural network to conform to label similarity and subsumption relations. The semantic-based loss is independent of the deep network and also can be applied to other deep learning models that need to exploit external knowledge.
3) We carry out extensive experiments on four large, social media data sets. The results produced by our model show significant improvement over the state-of-the-art and other baseline models, in terms of Hamming loss, accuracy, precision, recall, and F_1-score with a substantial reduction of training time.

The rest of the article is organized as follows. In Section II, we review the related work on the task of automated social text annotation. In Section III, we formally define the problem and elaborate on the joint multilabel learning method, including the title-guided attention mechanism and the semantic-based loss regularizers. In Section IV, the experiment and evaluation results are presented and discussed, with analysis on model convergence, multisource components, and attention visualization. In Section V, we conclude the article and discuss future research directions.

II. Related Work

In this section, we review the related research on automated social text annotation. Specifically, as our work is related to deep learning and multilabel learning, we focus on discussing the attention mechanisms in deep learning for text classification and the label correlation issue.

A. Automated Social Text Annotation

Automated annotation can support users' tagging process, reduce their cognitive overhead, and help produce more stable,

quality folksonomies on social media platforms [6]–[8]. It is natural to automatically annotate new documents with an existing collection of cleaned tags originally contributed by users. The task is closely related to tag recommendation, which aims at suggesting tags for existing or previously unseen resources to facilitate users' tagging [8]. The study in [8] classified tag recommendation as either object-centered or personalized. Object-centered recommendation predicts a set of tags that are descriptive to an object regardless of the target user. This type of recommendation aims at enhancing the quality of tagging and, thus, can benefit information retrieval in general. In contrast, personalized recommendation takes the users' interests or preferences into consideration. Automated social text annotation can be considered as an object-centered tag recommendation task.

Various methods and techniques have been proposed for tag recommendation, as reviewed in [8], including tag co-occurrence-, content-, matrix factorization-, clustering-, graph-, and learning to rank- approaches. On social Q&A sites, existing research explores the annotation for a question by using the descriptive tags of its similar questions through probabilistic hypergraph construction, adaptive probabilistic hypergraph learning, and heuristic-based tag selection [6]. In microblogging services, such as Twitter, various models have been proposed for content-based hashtag recommendation [9], [11], [13]–[16], that is, to suggest tags according to textual features. The research in [9] extracted term frequency-based lexical features and applied probabilistic graphical models [11] to suggest hashtags.

Recent studies formulated the automated annotation task as a multilabel classification problem and started using deep learning-based methods for automated hashtag annotation [13]–[16] and publication annotation [12]. These deep models usually encoded the input with multiple layers of nodes and nonlinear activations to a vector representation and tried to approximate the matching from the input to the labels. The advantage of multilabel deep learning models lies in their relatively straightforward problem formulation with strong approximation power on large data sets, resulting in better performance over traditional approaches [27]. Some of the notable deep models adapted for multilabel classification included variations of RNNs [12], [15], [16] and convolutional neural networks (CNNs) [13], [14] with attention or memory mechanisms.

B. Attention Mechanisms for Text Classification

Attention mechanisms have been widely used in many natural language processing tasks. Originally, the idea was proposed in machine translation to cope with the bottleneck issue arising from compressing a long sentence to a single fixed-length vector. Instead of generating only one vector representation for each sentence, the attention mechanism allows generating a distinct vector representation with respect to each target word to be decoded, selectively focusing on parts of the input sentence [28], [29].

Technically, attention mechanisms compute a weighted average of hidden states or the representations of input words, based on alignments or similarities [28], [29], i.e., computing the similarity between the current target word representation and each of the input word representations (hidden states in the encoder) to determine how much weight (attention) can be assigned to the input. The work in [28] applied an additional feedforward layer with softmax activation to model this alignment. This soft alignment can be visualized, showing agreement with human intuition [28]. The study in [29] further investigated other alignment models with different functions and explored a local attention that focuses on a subset of words in a sentence, achieving improved results in neural machine translation. The study [30] utilized three different alignments, dot product alignment for self-attention, elementwise alignment for cross-attention, and concatenation-based alignment for coattention to model questions and answers for duplicated question annotation.

The idea that attention mechanisms can learn to select the important parts from a sentence has been applied to text classification. The HAN [17] proposed word- and sentence-level attention mechanisms to capture the hierarchical pattern of a document and to focus on each word or sentence distinctively for classification. Unlike the attention mechanism in machine translation, there is no target representation that can be aligned to. As such, an "informative" learnable vector was added and attended to each word or sentence. The idea of aligning each word or sentence to the learnable vectors, although it has been used in later studies for sentiment classification [31] and document annotation [12], does not properly model the users' reading and understanding. In fact, the importance of each word or sentence can be reflected by aligning it to the main themes of a document. A more explainable approach would be to transform the title of a document into an explicit representation of the themes so that words and sentences in the document can be aligned. Besides, while sentences are key elements in document understanding for human beings, recent studies only model social documents with word-level attention mechanisms, e.g., answers in [30] and conversations in [32]. In this study, we shed light on an explicitly guided sentence-level attention mechanism for social text annotation.

Attention mechanisms have also been widely used in computer vision, including image captioning [33] and multimodal image and text annotation [13]. To model the attention in the human visual system, the work in [33] proposed both hard and soft attention mechanisms for image captioning, aligning each part of an image to the sequence of previous words to generate the next word, as inspired by the alignment in machine translation. The work in [13] modeled the mutual and external alignment between texts and images in a microblog with a coattention network for hashtag annotation. Our study, however, focuses on the relations between the title and content of a document, which naturally simulates users' reading behavior during document annotation.

C. Label Correlation in Multilabel Learning

In multilabel classification, each instance (document) is associated with a set of labels and the labels are usually correlated with each other [21], [34]. This is different from multiclass classification in which classes (labels) are assumed to be disjoint. Social annotation can be seen as a multilabel

classification problem, in which a document might be an abstract or a publication, a question or an image, and the tags contributed by online users correspond to labels.

In real-world data with a large number of labels, the correlation among labels is common and cannot be ignored. In collaborative tagging, different users use tags in various semantic forms and granularities [23], [35]. For example, in the Bibsonomy data, many documents tagged with machine_learning are also tagged with text_mining, svm, or optimization, which are either the related terms (text_mining being a related application domain), or narrower terms (the specific algorithm svm and the subdomain optimization). The relations among these labels represent additional knowledge that can be exploited to potentially improve the performance of multilabel classification [21]. Many of such relations have already been captured and stored as human knowledge in existing knowledge bases. Relations among the labels can be extracted by grounding the labels to terms and concepts in those knowledge bases.

A traditional approach for multilabel classification is to construct many binary classifiers, one for each label. This approach, often referred to as binary relevance or one-versus-rest and, however, completely ignores the correlations among labels [19], [20]. One main strategy to address this issue was to regenerate a feature space incorporating information on label correlation. An example was adapting discriminative classifiers, such as support vector machine (SVM) [26]. The classifier chain method extends this idea by incorporating the binary classification results in a chain as features to predict the next label [36]. The classifier chain can be randomized and embedded into an ensemble learning architecture [37] or mined using clustering and graph-based methods [38]. Instead of organizing classifiers as a chain, the Hierarchy Of Multilabel classifiER (HOMER) [39] created a tree of classifiers, based on the hierarchical structure of labels prelearned in an unsupervised manner. Probabilistic graphical models were also used to encode the correlation among labels, including the Gibbs random fields [40] and the Bayesian networks [41].

Existing studies using deep learning for multilabel classification have reported superior performance over the traditional methods [20], [27]; however, they have not adequately solved the issue of label correlation. Neural network models usually represent the label space with an orthogonal vector: one label with one-hot representation, and each label set with a multihot representation, e.g., [0 1 0 1 1] in a 5-D label space, as in [12], [13], [15], [16], [19]. This, however, assumes independence among labels.

One recent approach to leverage label correlation in neural networks was through weight initialization [24]: initializing higher weights for some dedicated neurons (each represents a co-occurring pattern among labels) between the last hidden layer and the output layer. This idea was extended in [42] to include subsumption relations among labels. It is, however, difficult to interpret how the randomly chosen "dedicated" neurons really work in such settings. Computationally, it is also extremely expensive (if not infeasible) to place many neurons, equal to the number of co-occurring patterns, in the last hidden layer for weight initialization. Therefore, a desired deep learning model should not only incorporate the label relations (e.g., similarity and subsumption) from external knowledge bases to improve the classification performance but also ensure that the computation is practically feasible. The study in [43] explored tree-like architectures to organize neural networks as a chain for hierarchical label prediction, i.e., assigning a chained feedforward neural network for each layer in a label hierarchy. Similar to the idea of assigning dedicated neurons, this cannot be easily scaled to a massive number of label similarity and subsumption relations.

III. Proposed Approach

We first define the problem in a formal way and then propose a parallel, two-layered attention network, called the JMAN, to model the users' reading and annotation process.

A. Problem Statement

The automated annotation task can be formulated as a multilabel classification problem [19], [20]. Suppose X denoting the collection of textual sequences or instances (e.g., documents) and $Y = \{y_1, y_2, \ldots, y_n\}$ denotes the label space with n possible labels (i.e., user-generated tags). Each instance in X, $x \in \mathbb{R}^d$, is a word sequence, in which each word is represented as a d-dimensional vector. Each x is associated with a label set $Y_i \subseteq Y$. Each $\overrightarrow{Y_i}$ is an n-dimensional multihot vector, $\overrightarrow{Y_i} = [y_{i1}, y_{i2}, \ldots, y_{in}]$, and $y_{ij} \in \{0, 1\}$, where a value of 1 indicates that the jth label y_j has been used to annotate (is relevant to) the ith instance, and 0 indicates irrelevance of the label to the instance. The task is to learn a complex function $h : X \rightarrow Y$ based on a training set $D = \{x_i, \overrightarrow{Y_i} | i \in [1, m]\}$, where m is the number of instances in the training set.

B. Overall Design

The JMAN model, as shown in Fig. 2, is an extension to our previous work [44]. Instead of feeding the whole text sequence X into the neural network as in HAN [12], [17], JMAN takes as inputs the title, x_t, and the content (in this work, the abstract of a document is treated as the content), x_a, and processes them separately, where $x = \{x_t, x_a\}$. Each target is a multihot representation, $\overrightarrow{Y_i} \in \{0, 1\}^{|Y|}$.

There are four attention modules, shown as dotted edges in Fig. 2: two word-level attention modules for the words in the title and in each sentence in the content, respectively; two sentence-level attention mechanisms, one guided by the title representation ("title-guided") and the other guided by an "informative" vector ("original"). JMAN's key distinctions from the previous models include: 1) the multisource hierarchical architecture allows different metadata in a document to be processed in different ways in parallel (see Section III-C); 2) the title-guided sentence-level attention mechanism aims to explicitly model the reading behavior of users during annotation (see Section III-D); and 3) the semantic-based loss regularizers aim to enhance the learning process by enforcing the output of the network to conform to the label correlation as specified in external knowledge bases (see Section III-E).

C. Multisource Hierarchical Architecture

The title of a document is a key feature that can greatly influence the decision of tagging [18] and the performance of

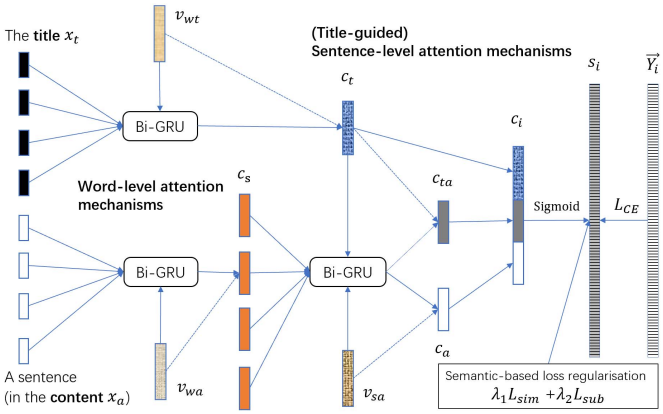


Fig. 2. JMAN.

classification [5]. We process the title and the content separately, and this multisource hierarchical architecture constitutes the backbone of the JMAN model.

1) Embedding Layer: Each input title or content (usually multiple sentences) is an ordered set of words, represented as $x_t = (v_t^{(1)}, v_t^{(2)}, \ldots, v_t^{(n_t)})$ and $x_a = (v_a^{(1)}, v_a^{(2)}, \ldots, v_a^{(n_a)})$, where n_t or n_a denotes the number of words in the title or content, respectively. The embedding layer transforms the input v into low-dimensional vectors, which are formally defined as $e_t = W_e v_t$ and $e_a = W_e v_a$, where $W_e \in \mathbb{R}^{d_e \times |V|}$ is the embedding weights that are usually pretrained via neural word embedding algorithms, e.g., Word2Vec [45] or Glove [46]. The embedding dimensionality d_e is far less than the vocabulary size $|V|$, i.e., $d_e \ll |V|$.

2) Bi-GRU Layer: A problem in the vanilla RNN is the vanishing gradient, e.g., when reading a lengthy sequence, the RNN "reader" may forget the previous words before it completes processing the whole sequence. Long short-term memory (LSTM) [47] and gated recurrent units (GRUs) [48] have been proposed to address this problem. GRUs have been applied to the original HAN model [17] and neural machine translation [28] due to their efficiency in training. We follow this setting and use GRUs as the basic recurrent unit.

GRUs introduce two gates, a reset gate $r^{(t)}$ and an update gate $z^{(t)}$, to control and generate a new hidden state $h^{(t)}$ from the previous hidden state $h^{(t-1)}$. RNN with GRUs can be formally defined in (1), where σ refers to a nonlinear activation function (here, we use the logistic sigmoid function), and $W_{er}, W_{ez}, W_{e\tilde{h}} \in \mathbb{R}^{d_h \times d_e}$ and $W_{hr}, W_{hz}, W_{h\tilde{h}} \in \mathbb{R}^{d_h \times d_h}$ are weights, where d_h is the number of hidden units. We use the model with bias terms $b_r, b_z \in \mathbb{R}^{d_h}$ as in [17]

$$\begin{aligned} r^{(t)} &= \sigma(W_{er} e^{(t)} + W_{hr} h^{(t-1)} + b_r) \\ z^{(t)} &= \sigma(W_{ez} e^{(t)} + W_{hz} h^{(t-1)} + b_z) \\ \tilde{h}^{(t)} &= \tanh(W_{e\tilde{h}} e^{(t)} + W_{h\tilde{h}}(r^{(t)} \circ h^{(t-1)})) \\ h^{(t)} &= (1 - z^{(t)}) \circ h^{(t-1)} + z^{(t)} \circ \tilde{h}^{(t)}. \end{aligned} \tag{1}$$

The idea of bidirectional-RNN [49] with GRUs, denoted as bidirectional GRUs (Bi-GRUs), are proposed to capture the fact that a word in a sequence is not only related to its previous words but also to its following words. Bi-GRUs consist of forward GRUs and backward GRUs. The forward GRUs read the embedding of each word in the input sequentially from left to right, e.g., from $e^{(1)}$ to $e^{(n)}$, to produce forward hidden states $(\overrightarrow{h^{(1)}}, \ldots, \overrightarrow{h^{(n)}})$, whereas the backward GRUs read the sequence reversely from $e^{(n)}$ to $e^{(1)}$ to calculate backward hidden states $(\overleftarrow{h^{(n)}}, \ldots, \overleftarrow{h^{(1)}})$. Both hidden states are concatenated to construct a new fixed-length vector as the output hidden state, $h^{(i)} = [\overrightarrow{h^{(i)}}; \overleftarrow{h^{(i)}}]$.

In the proposed network (see Fig. 2), after the reading in both directions is completed, the title and content are represented as context vectors $\mathbf{c}_t$ or $\mathbf{c}_a$, respectively. These vectors are normally set as the last concatenated hidden states $h^{(n)}$; however, doing so tends to emphasize the words toward the end of the sequence. Therefore, the attention mechanisms need to be applied to recalculate the vectors $\mathbf{c}_t$ or $\mathbf{c}_a$.

3) Hierarchical Attention Layers: The idea of hierarchical attention is closely related to how users read and comprehend documents. The HAN model assumes that, to understand a document, users read the document word by word in each sentence and then sentence-by-sentence. During the reading, users would pay special attention to the most informative words or sentences, which might be considered to annotate that document later. There are three Bi-GRU layers in JMAN, as shown in Fig. 2, each accompanied by an attention layer(s): two word-level attention layers, for title and sentences in the abstract, respectively; two sentence-level attention layers, one is the original sentence-level attention proposed in [17] and the other is the title-guided sentence-level attention (see Section III-D).

To model the different amount of attention paid on each word or sentence, a weighted average of hidden representations is applied, as suggested in [17] and [28]. The attention scores are based on an alignment of each hidden representation in a sequence to a nonstatic and learnable, "informative" vector representation, which is supposed to encode "what is the informative word (or sentence)" in the sequence [17] and commonly used in document classification tasks [12], [31]. The dot product is naturally used as the alignment measure to calculate vector similarity. The word-level attention models the importance of each word in the title or sentence, while the sentence-level attention mechanism makes a distinction for each of the sentences. The word-level attention mechanism in the title (or sentences) is described in the following equation:

$$\begin{aligned} v^{(i)} &= \tanh(W_t h^{(i)} + b_t) \\ \alpha^{(i)} &= \frac{\exp(v_{wt} \bullet v^{(i)})}{\sum_{i \in [1, n_t]} \exp(v_{wt} \bullet v^{(i)})} \\ c_t &= \sum_{i \in [1, n_t]} \alpha^{(i)} h^{(i)}. \end{aligned} \tag{2}$$

In (2), a fully connected layer is added to transform the hidden state $h^{(i)}$ to a vector representation $v^{(i)}$, followed by alignment to the attention vector v_{wt} with the dot product operation (denoted as $\bullet$). A softmax function is applied to obtain the attention weights $\alpha^{(i)}$. The context vector c_a, which is the representation of the sequence, is computed as the weighted average of all hidden state vectors $h^{(i)}$. In a similar way, we can compute the word-level attention for each sentence and the original sentence-level attention.

D. Guided Attention at Sentence Level

Given a document, we naturally assume that a user would try to read and understand first the title, which often represents the main themes of that document and keep her understanding in the mind. When reading each sentence in the document, she would try to align the meaning of each sentence to the title. If a sentence conveys a piece of meaningful information based on her knowledge, especially the one that aligns well with the main themes of the document, she would keep it for annotation either immediately or later; otherwise, that sentence would be skipped.

The attention mechanisms presented in Section III-C are not enough to make a clear distinction among sentences. First, the impact of the title on the document annotation is not considered, which is, however, particularly important during the tagging process [5], [18]. Second, in the attention mechanisms described in (2), the "informative" vector v_{wt}, commonly treated as weights to be learned in the model [12], [31], does not reflect any explicit object in humans' reading and understanding.

Selection of the important sentences in the content should ideally conform to the main themes of the document. Title is a short, abstractive summarization of the main themes, and a good starting point to understand the document. We propose the title-guided sentence-level attention mechanism, as shown in Fig. 2, which can be modeled using the following equation:

$$\begin{aligned} v_s^{(r)} &= \tanh\left(W_s h_s^{(r)} + b_s\right) \\ \alpha_s^{(r)} &= \frac{\exp\left(c_t \bullet v_s^{(r)}\right)}{\sum_{k\in[1,n_s]} \exp\left(c_t \bullet v_s^{(k)}\right)} \\ c_{ta} &= \sum_{r\in[1,n_s]} \alpha_s^{(r)} h_s^{(r)} \end{aligned} \tag{3}$$

where $h_s^{(r)}$ is the hidden state of the rth sentence, c_t is the title representation obtained from (2), n_s denotes the number of sentences in the abstract, $\alpha_s^{(r)}$ is the sentence-level attention score, W_s and b_s are learnable weights in the network. This title-guided attention mechanism is distinct from a recent study in [50], which used the title at the word level to enhance the annotation for keyphrase generation. The "title-guided encoding" in [50] calculates a different title representation for each word in the document. However, it did not model the human reading behavior compared with the proposed title-guided attention mechanism.

Guiding the sentences solely with the title may cause the final document representation to be overly dependent on the title. The actual content of a document usually contains (far) more information not described in the title, which can help suggest more tags during annotation [5]. For example, some sentences may highlight an innovative and important evaluation study, which is not present in the title. To avoid such an overemphasis on the effect of the title and form a more comprehensive document representation, the original sentence-level attention is also considered. The final representation of a documents is the concatenation of the title representation c_t, the title-guided sentence representation c_{ta}, and the original sentence representation c_a, i.e., $c_i = [c_t, c_{ta}, c_a]$, as shown in Fig. 2. The idea of guided attention can be naturally generalized to other sources of metadata that can affect the annotation process, such as the users' preferences, bookmarks, or reading history. We will show the effectiveness of this design by comparing it against a number of state-of-the-art and baseline models.

E. Semantic-Based Loss Regularizers

Studies show that tags have hidden semantic structures (e.g., similarity and subsumption) and users collectively annotate documents with semantically related tags of various forms and granularities [7], [22], [23], [35]. If we treat each tag as a label, then we have to take the label correlation into account for multilabel classification. Leveraging the label correlation is particularly challenging as the number of relation pairs might be enormously large when there are many labels [20]. In this case, it is infeasible or computationally inefficient to apply the weight initialization approach [24], [42] that assigns a neuron in the penultimate layer of the neural network to "memorize" just one of the numerous label relations.

We take a different strategy by using the semantic-based loss regularization, in which two loss regularizers are proposed to deal with the similarity and subsumption relations, respectively, jointly optimized with the binary cross-entropy loss. The idea is to enforce the output of the neural network to satisfy the semantic constraints from the label relations. Such relations can be either inferred from the data set itself or extracted through grounding the labels to concepts or terms in external knowledge bases. The whole joint loss is defined in the following equation:

$$L = L_{\text{CE}} + \lambda_1 L_{\text{sim}} + \lambda_2 L_{\text{sub}} \tag{4}$$

where L_{CE} is the binary cross entropy loss [19], which obtained superior results with faster convergence over the pairwise ranking loss proposed in [27] for multilabel text classification with a feedforward neural network. In (5), $y_{ij} \in \{0, 1\}$ indicates the true value whether a label $y_j \in Y$ has been used to annotate the document i, and s_{ij} is the actual value after the sigmoid layer

$$L_{\text{CE}} = -\sum_i \sum_j (y_{ij} \log(s_{ij}) + (1 - y_{ij}) \log(1 - s_{ij})). \tag{5}$$

While the binary cross-entropy loss defines the matching between the output values and the true label set, the proposed L_{sim} and L_{sub} shown in (6) define how the output values conform to the label relations as defined in external knowledge bases or learned from a data set

$$\begin{aligned} L_{\text{sim}} &= \frac{1}{2} \sum_i \sum_{j,k|y_j,y_k\in Y_i} \text{Sim}_{jk} |s_{ij} - s_{ik}|^2 \\ L_{\text{sub}} &= \frac{1}{2} \sum_i \sum_{j,k|y_j,y_k\in Y_i} \text{Sub}_{jk} R(s_{ij})(1 - R(s_{ik})) \end{aligned} \tag{6}$$

where Y_i is the set of labels for the ith document; j and k are the indices of a co-occurring pair of labels y_j and y_k in the label set Y_i, corresponding to the indices of nodes s_{ij} and s_{ik} in the output layer s_i in Fig. 2. $R()$ represents the rounding function for binary prediction $R(s_{ij}) = 0$ if $s_{ij} < 0.5$, otherwise $R(s_{ij}) = 1$.

The label similarity matrix $\text{Sim} \in (0,1)^{|Y|*|Y|}$ stores pairwise label similarity; the larger the value of Sim_{jk}, the more similar the labels y_j and y_k are to each other. Each element Sub_{jk} in the label subsumption matrix, $\text{Sub} \in \{0,1\}^{|Y|*|Y|}$, indicates whether the label y_j is a child label of y_k. Both the Sim and Sub matrices can be precomputed from the training data or obtained from external knowledge bases before the training. In the implementation, Sim (if a threshold is used for all entries) and Sub can be treated as sparse matrices to reduce computational complexity.

The idea for L_{sim} is that, in collective tagging, besides the same labels, users tend to annotate documents with different labels that have very similar meanings. In multilabel learning, labels with high semantic similarity tend to be predicted together with similar values. L_{sim} is a multiplication between two terms: Sim_{jk} and $|s_{ij}-s_{ik}|^2$. To minimize L_{sim}, intuitively, for very similar co-occurring labels y_j and y_k, i.e., with high Sim_{jk} close to 1, their corresponding nodes in the output layer should have minimal difference so that $|s_{ij}-s_{ik}|^2$ is low; for labels having low similarity with Sim_{jk} close to 0, there is almost no strict requirement on their corresponding output, as the squared difference $|s_j-s_k|^2$ will be scaled down by low similarity value. L_{sim} has a distinct form to the label manifold regularizer proposed in [25]. The latter considers minimizing the differences of vector representations for low-rank approximation, while L_{sim} minimizes node differences in the output layer in a neural network.

The idea for L_{sub} is that, in collective tagging, besides the same labels, users often annotate documents using different labels with different levels of specificity based on their knowledge and understanding. An analogy for this is "A birder sees a 'robin' when a normal person only sees a 'bird"' [35], [51]. For example, a researcher from the machine learning area would annotate an article using "LSTM," but researchers from other areas may annotate the same article using more general labels, such as "Neural Networks" or "Deep Learning." Distinct from similarity relations, the subsumption relations between labels are asymmetric. For two tags having a subsumption relation, if the child tag is associated with the document, there is a higher likelihood that the parent tag is related to the same document than others. In L_{sub}, if two labels having a subsumption relation $< y_j \rightarrow y_k >$ are both present in the label set Y_i, the case that the parent label y_k is predicted as false (i.e., $R(s_{ik})=0$) when its child label y_j is predicted as true (i.e., $R(s_{ij})=1$), will be penalized. Such a case will result in a positive penalty, while the penalty will be 0 in all other cases.

As the predefined label relations may not be compatible with the semantics of the labels in the data set, it would be interesting to allow label correlation (represented by Sim and Sub) to be updated dynamically with training data. In doing this, both Sim and Sub become continuous representations and can have negative entries, which has an impact on the two regularizers L_{sim} and L_{sub}. Taking L_{sim} as the example, the more negative the value of Sim_{jk}, the less similar the labels y_j and y_k. Then, the case of $|s_{ij}-s_{ik}|^2$ being large (e.g., label y_j predicted as true and label y_k predicted as false) will be favored. Dynamic update of Sim and Sub with a large number of labels, however, requires substantial memory. We first focus on the fixed Sim and Sub and compare the results between dynamic and fixed Sim and Sub in the experiments.

We finally optimize the joint loss function in (4) with the L_2 regularization using the Adam optimiser [52].

IV. Experiments

We carried out experiments on four large, social media data sets for academic research (Bibsonomy and CiteULike, three data sets) and question&answering (Zhihu, one data set). The evaluation showed significant performance gain of JMAN over the state-of-the-art models in terms of a number of metrics, with a substantial improvement of convergence speed. We also discussed the impact of the regularization parameters and analyzed the attention through visualization. The code, implementation details, and prediction results are available at https://github.com/acadTags/Automated-Social-Annotation.

A. Data Sets

On Bibsonomy and CiteUlike, users can share and annotate publications. Metadata of the documents, such as title and abstract, are also available. The Bibsonomy data set [53] version "2015-07-01"[1] was used, which contains 3 794 882 annotations, 868 015 resources, and 283 858 distinct tags from 11 103 users, accumulated from 2003 to 2015. We used the cleaned data set from our previous work [54] and selected only the documents containing both the title and the abstract. For better qualitative analysis, we further selected the documents having at least one tag matched to the concepts in the ACM Computing Classification System.[2] For CiteUlike, we used the benchmark data sets CiteULike-a and CiteULike-t released in [10]. We applied the same preprocessing steps as in [54] and removed the tags occurring less than ten times.

Zhihu is a leading Chinese social Q&A site in all domains. Each question has a title and a detailed description. We used the official benchmark open data from the Zhihu Machine Learning Challenge 2017,[3] containing more than three million questions and 1999 labels. The data set was preprocessed before its release: all the Chinese words were segmented and replaced with an unknown codebook due to privacy issues. We randomly sampled around 100 000 questions having both the title and content.

To extract the subsumption relations for all tags in each of the data sets (except Zhihu), we grounded the tags to concepts in the external knowledge base, the Microsoft Concept Graph (MCG).[4] MCG has around 1.8M concepts and instances and 8.5M subsumption relations. Zhihu released its crowdsourced tag hierarchies that can be directly used to find subsumption relations.

Statistics of the cleaned data sets are shown in Table I, including the number of documents $|X|$, number of labels $|Y|$, vocabulary size in documents $|V|$, average number of labels per document Ave, and the number of label subsumption pairs

[1] https://www.kde.cs.uni-kassel.de/bibsonomy/dumps
[2] https://www.acm.org/publications/class-2012
[3] https://biendata.com/competition/zhihu/
[4] https://concept.research.microsoft.com/Home

TABLE I
STATISTICS OF THE FOUR DATA SETS

Dataset	$\|X\|$	$\|Y\|$	$\|V\|$	*Ave*	Σ_{Sub}
Bibsonomy (clean)	12,101	5,196	17,619	11.59	101,084
CiteULike-a (clean)	13,319	3,201	17,489	11.60	107,273
CiteULike-t (clean)	24,042	3,528	23,408	7.68	141,093
Zhihu (sample)	108,168	1,999	62,519	2.45	2,655

for each data set Σ_{Sub}. The average number of labels per document in Zhihu is much less than the ones in Bibsonomy and CiteULike, but the former has a larger number of documents and vocabulary size. The number of labels in all data sets is large, from around 2k–5.2k. The number of subsumption relations grounded to MCG is also large, all above 100k except Zhihu. There are more than 2.5k subsumption relations in Zhihu.

B. Experiment Settings

To calculate the similarity matrix Sim in (6) , we used the cosine similarity of the pretrained skip-gram embeddings [45] on all labels in each data set. To construct the label subsumption matrix Sub, we used the subsumption pairs from MCG and Zhihu. The values of λ_1 and λ_2 in L were tuned using tenfold cross-validation.[5] We implemented the proposed JMAN model and its variants on Tensorflow [55]. Seven models were implemented for comparison:

1) *SVM-Ovr:* A one-versus-rest multilabel SVM with word embedding features, implemented using the scikit-learn Python package.[6] We used the RBF kernel and tuned the C and γ to achieve the best F_1. This baseline was also used in [15].
2) *LDA:* The probabilistic topic modeling approach, latent Dirichlet allocation (LDA) [56], was applied to represent each document as a probability distribution over hidden topics, implemented with the wrapper in the Python Gensim package [57] for the JAVA-based MALLET toolkit [58]. The algorithm was adapted to multilabel classification by assigning each new document the tags of its k most similar documents based on the document-topic distributions $p(\text{topic}|\text{document})$. We trained the LDA model for 1000 iterations and tuned the number of topics T as 200 and k as 1 for all data sets based on the validation sets. The baseline was also used in [59].
3) *Bi-GRU:* The bidirectional-RNN [49] with GRUs for multilabel classification. The algorithm treated the title and content together as the input sequence. The document representation $\mathbf{c_i}$ is set as the last concatenated hidden state.
4) *HAN:* The HAN in [17], which was used in [12] for tag recommendation. We combined the title and abstract and fed into the HAN model, as implemented in [12]. This is the state-of-the-art deep learning model for document classification.
5) *JMAN-s:* The proposed model without semantic-based loss regularizers.
6) *JMAN-s-tg:* The proposed model without semantic-based loss regularizers and the title-guided sentence-level attention, i.e., $c_i = [c_t, c_a]$.
7) *JMAN-s-Att:* The proposed model without semantic-based loss regularizers and the original sentence-level attention, i.e., $c_i = [c_t, c_{ta}]$.
8) *JMAN$_d$:* The proposed model with dynamic update of Sim and Sub during training.

The implementation of neural network models is based on brightmart's TextRNN and HAN under the MIT license.[7] We trained all the models using tenfold cross-validation and then tested on a separate, fixed 10% randomly held-out data set. The number of hidden units, learning rate, and dropout rate [60] were set as 100, 0.01, and 0.5, respectively, for all models. The batch size for the Bibsonomy and CiteULike-a/t data set was set to 128, and the batch size for the Zhihu data set was set to 1024. The sequence lengths of the title (also the length of each sentence) and the content were padded to 30 and 300 for Bibsonomy, CiteULike-a, and CiteULike-t and 25 and 100 for Zhihu. We parsed the sentences of Bibsonomy and CiteULike based on punctuations and padded the sentences to a fixed length. For Zhihu, as the data had been masked, we simply set a fixed length to split the content into "sentences." Input embeddings for the title and the sentences were initialized as a 100-D pretrained skip-gram embedding [45] from the documents. We decayed the learning rate by half when the loss on the validation set increased and set an early stopping point when the learning rate was below a threshold (2e-5 for Bibsonomy and Zhihu; 1e-3 for CiteULike-a/t). Experiments on the neural network models were run on a GPU server, NVIDIA GeForce GTX 1080 Ti (11-GB GPU RAM), except for the dynamic update of Sim and Sub on Intel Xeon Processor E5-2630 v3 or v4 with 30-GB RAM; experiments on SVM-ovr and LDA were run on an Intel Xeon CPU E5-1620 v2 with 16-GB RAM.

We also reimplemented three representative algorithms for comparison, which transforms either the feature space or label space of a base classifier for multilabel classification: 1) classification chain (CC) [36], [37]; 2) HOMER [39]; and 3) principal label space transformation (PLST) [61], adapting the Python scikit-multilearn [62] wrapper of MEKA [63] (based on WEKA [64] and MULAN [65]). The base classifier was SVM with an RBF kernel for the methods. Due to large numbers of documents and labels, the program took much longer than the SVM-ovr implementation and required substantial memory. With the default parameters in MEKA, the results of the three methods were not better than the ones of the SVM-ovr classifier. Thus, we do not report their results here but provide an open implementation for reproducibility.

C. Evaluation Metrics

Five widely used example-based metrics were applied for evaluation, including hamming loss, accuracy, precision, recall, and F-measure, to assess the performance of the

[5]We tuned λ_1 and λ_2 using a two-step parameter tuning process: first, finding the best $\lambda_1 \in \{1E{-}1, 1E{-}2, \ldots, 1E{-}6\}$ by setting λ_2 as 0, and second, finding the best $\lambda_2 \in \{1E{+}1, 1E{+}0, \ldots, 1E{-}4\}$ while fixing the tuned λ_1.

[6]https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestzClassifier.html

[7]https://github.com/brightmart/text_classification

TABLE II

COMPARISON RESULTS OF JMAN AND OTHERS ON THE FOUR SOCIAL ANNOTATION DATA SETS IN TERMS OF HAMMING LOSS (H), ACCURACY (A), PRECISION (P), RECALL (R), AND F_1-SCORE (F_1)

		SVM-ovr	LDA	Bi-GRU	HAN	JMAN-s-tg	JMAN-s-att	JMAN-s	JMAN	JMAN$_d$
Bib	H	*107.7±0.2*(7)	*142.3±2.0*(8)	*90.1±0.7*(6)	*86.1±0.4*(5)	***84.5±0.5***(1)	*84.6±0.3*(2)	85.2±0.5(4)	85.1±0.6(3)	-
	A	19.2±0.2(8)	*21.0±0.5*(6)	*19.2±1.3*(7)	*22.0±1.0*(5)	*24.1±0.6*(4)	*24.2±0.6*(3)	24.8±0.4(2)	**25.1±0.4**(1)	-
	P	39.2±0.3(7)	*31.1±0.8*(8)	*52.2±2.0*(6)	*57.2±0.8*(5)	59.1±1.0(2)	**59.2±1.0**(1)	58.6±0.4(4)	58.8±0.8(3)	-
	R	*25.2±0.2*(6)	***31.1±0.7***(1)	*21.7±1.6*(8)	*24.6±1.2*(7)	*26.9±0.6*(5)	*27.2±0.7*(4)	*28.2±0.5*(3)	28.6±0.3(2)	-
	F_1	*30.7±0.2*(7)	*31.1±0.7*(6)	*30.6±1.9*(8)	*34.4±1.3*(5)	*37.0±0.7*(4)	*37.3±0.8*(3)	*38.0±0.5*(2)	**38.5±0.4**(1)	-
C-a	H	*118.1±0.3*(8)	*168.2±1.5*(9)	*100.0±0.7*(7)	96.0±0.5(5)	*94.6±0.5*(2)	***94.5±0.3***(1)	95.5±0.5(3)	95.7±0.6(4)	*97.2±1.3(6)*
	A	*8.6±0.1*(8)	*9.5±0.3*(7)	*7.5±1.6*(9)	*11.0±0.8*(6)	13.5±0.6(4)	13.4±0.4(5)	13.6±0.8(3)	13.9±0.8(2)	**14.4±0.6(1)**
	P	*26.1±0.2*(8)	*18.5±0.5*(9)	*32.6±4.5*(7)	*42.9±1.4*(6)	47.9±1.2(2)	***48.4±0.8***(1)	47.2±1.6(4)	47.3±1.5(3)	47.1±1.1(5)
	R	*12.3±0.1*(8)	***18.6±0.6***(1)	*8.9±2.0*(9)	*13.2±1.1*(7)	*16.3±0.8*(5)	*16.0±0.6*(6)	16.6±1.2(4)	17.0±1.1(3)	*17.8±0.7(2)*
	F_1	*16.7±0.1*(8)	*18.6±0.5*(7)	*14.0±2.9*(9)	*20.2±1.4*(6)	24.3±1.0(4)	*24.1±0.7*(5)	24.6±1.5(3)	25.0±1.3(2)	**25.8±0.8(1)**
C-t	H	*113.5±0.3*(8)	*171.8±2.2*(9)	*97.1±0.7*(7)	***93.6±0.3***(1)	*94.2±0.3*(3)	*94.0±0.4*(2)	95.2±0.5(4)	95.2±0.6(5)	96.3±0.8(6)
	A	*8.7±0.2*(9)	*9.2±0.2*(8)	*10.9±2.3*(7)	*11.9±1.0*(6)	*13.6±0.6*(4)	*13.5±0.3*(5)	14.4±0.6(3)	14.5±0.4(2)	***15.2±0.8(1)***
	P	*24.5±0.3*(8)	*17.2±0.2*(9)	*34.9±5.1*(7)	*38.2±1.8*(6)	*39.8±1.2*(5)	*40.0±0.8*(4)	40.9±0.9(2)	40.9±0.6(3)	***42.3±0.9(1)***
	R	*12.2±0.2*(9)	*17.7±0.5*(3)	*13.0±2.9*(8)	*13.8±1.3*(7)	*16.2±0.8*(5)	*16.2±0.4*(6)	17.6±0.9(4)	17.8±0.7(2)	***18.7±1.0(1)***
	F_1	*16.3±0.2*(9)	*17.4±0.3*(8)	*18.9±3.9*(7)	*20.3±1.7*(6)	*23.0±1.0*(4)	*23.0±0.5*(5)	24.6±1.0(3)	24.8±0.7(2)	***26.0±1.1(1)***
Zhi	H	-	*187.9±0.7*(7)	95.3±0.3(5)	***93.4±0.2***(1)	*94.3±0.3*(2)	*94.6±0.3*(3)	95.3±0.5(6)	95.2±0.6(4)	-
	A	-	*3.9±0.2*(7)	*13.9±0.8*(6)	15.3±0.8(4)	15.5±0.3(3)	15.3±0.4(5)	**15.6±0.5**(1)	**15.6±0.5**(1)	-
	P	-	*5.6±0.2*(7)	*23.8±1.1*(6)	25.7±1.2(2)	25.7±0.5(4)	25.4±0.7(5)	25.7±0.8(3)	**25.8±0.9**(1)	-
	R	-	*5.6±0.2*(7)	*15.4±0.9*(6)	*16.7±1.0*(5)	17.5±0.3(3)	*17.4±0.5*(4)	17.7±0.5(2)	**17.8±0.6**(1)	-
	F_1	-	*5.6±0.2*(7)	*18.7±1.0*(6)	20.3±1.1(5)	20.8±0.3(3)	*20.7±0.5*(4)	21.0±0.7(2)	**21.1±0.7**(1)	-

For H, the smaller the better; for A, P, R, and F_1, the larger, the better. The best results are in **bold**. The results in *italics* indicate that the difference between JMAN and others is statistically significant with paired t-tests at a 95% significance level. The number in round brackets "()" shows ranking of the algorithm.

algorithms [20], [26], [66], [67]. For the metrics given in the following, D_t denotes the instances in the testing data and $|D_t|$ the number of the instances; $f(x_i)$ and y_i denote the predicted and actual label sets for the ith instance, respectively.

1) Hamming loss (H) measures the number of misclassified labels, $H(f) = (1)/(|D_t|)\sum_{i\in D_t}(1)/(Q)|f(x_i)\Delta y_i|$, where Δ is the symmetric difference between two sets and Q is a normalization constant. We set Q as the average number of labels per document, Ave, in the data (see Table I). The lower the value, the better the performance.
2) Accuracy (A), defined as the fraction of the correctly predicted labels to the total number of labels presented (union of predicted and actual ones), computed as $A(f) = (1)/(|D_t|)\sum_{i\in D_t}(|f(x_i)\cap y_i|)/(|f(x_i)\cup y_i|)$.
3) Precision (P), defined as the fraction of the correctly predicted labels to all the predicted labels, $P(f) = (1)/(|D_t|)\sum_{i\in D_t}(|f(x_i)\cap y_i|)/(|f(x_i)|)$.
4) Recall (R), defined as the fraction of the correctly predicted labels to all the actual labels, $R(f) = (1)/(|D_t|)\sum_{i\in D_t}(|f(x_i)\cap y_i|)/(|y_i|)$.
5) F-measure (F_1), defined as the harmonic mean between precision and recall, $F_1(f) = (2P(f)R(f))/(P(f)+R(f))$.

D. Evaluation and Comparison

We presented the evaluation results using the metrics and compared the performance of JMAN to the state-of-the-art and popular classification models. In particular, we highlighted the performance of using the semantic-based loss regularizers.

1) Main Results: Table II shows the evaluation and comparison results using JMAN and others based on the four data sets.[8] The proposed model JMAN and JMAN$_d$ performed the best in terms of accuracy and F_1-score and among the top or comparably well in terms of precision, recall, and Hamming Loss, on all data sets. Most results of JMAN$_d$ were better than JMAN on the CiteULike-a/t data sets, which indicated the usefulness of the dynamic update of the label semantic matrices: Sim and Sub. The results of JMAN were significantly better (denoted in *italics*) than HAN and Bi-GRU in terms of accuracy, precision, recall, and F_1-score, with few exceptions for HAN on the Zhihu data set.

In terms of F_1, JMAN provided an absolute increase up to 11.0% (by 78.6%) and 4.8% (by 23.7%) over Bi-GRU and HAN for the CiteULike-a data set and 5.9% (by 31.2%) and 4.5% (by 22.2%) over Bi-GRU and HAN for the CiteULike-t data set. A similar performance gain was achieved using the Bibsonomy data set, with an absolute increase of 7.9% (by 25.8%) over Bi-GRU and 4.1% over HAN (by 11.9%), and a relatively smaller increase using the Zhihu data sets of 2.4% (by 13.4%) over Bi-GRU and 0.8% over (by 3.4%) HAN. This overall improvement showed that the separate modeling of the metadata and the title-guided attention on the sentences clearly boosted the performance on automated annotation. The results of HAN were better than Bi-GRU in most settings, which showed the effectiveness of modeling the hierarchical pattern of a document with attention mechanisms and validated the results in [17].

Effectiveness of the semantic-based loss regularizers was observed by comparing the results produced by JMAN and JMAN-s (without semantic-based loss regularizers). The regularizers helped improve the recall and F_1 although with a relatively low margin. In terms of accuracy, precision, and F_1 in most evaluation settings, the results of JMAN were significantly better than JMAN-s-tg and JMAN-s-att, where either the title-guided or the original sentence-level attention was removed.

Only little improvement was observed with the Zhihu data set, largely due to its distinct characteristics: compared with other data sets, Zhihu has much shorter texts (around 1/3 of the texts in other data sets), larger vocabularies

[8]We were not able to obtain the results of SVM-ovr on the Zhihu data set as the training time for each fold in tenfold cross-validation was more than one day, which prevented efficient parameter tuning. JMAN$_d$ also requires substantial memory, and we failed to obtain results with the specified settings on the Bibsonomy and the Zhihu data sets.

TABLE III

COMPARISON RESULTS OF USING THE SEMANTIC-BASED LOSS REGULARIZERS ON DIFFERENT MODELS IN TERMS OF HAMMING LOSS (H), ACCURACY (A), PRECISION (P), RECALL (R), AND F_1-SCORE (F_1)

		Bi-GRU	+L_{sim}	+L_{sub}	+both	HAN	+L_{sim}	+L_{sub}	+both	JMAN-s	+L_{sim}	+L_{sub}	+both (JMAN)
Bib	H	90.1±0.7	90.2±0.4	**89.7±0.6**	90.0±0.9	86.1±0.4	86.1±0.5	86.0±0.6	**85.9±0.5**	85.2±0.5	85.1±0.6	**84.6±0.7**	85.1±0.6
	A	19.2±1.3	19.5±0.7	19.5±0.7	**20.1±0.5**	22.0±1.0	22.2±0.7	22.5±0.5	**22.5±0.8**	24.8±0.4	24.9±0.5	**25.2±0.6**	25.1±0.4
	P	52.2±2.0	52.4±1.7	52.7±1.5	**53.3±1.7**	57.2±0.8	**57.3±1.2**	57.1±1.0	57.3±1.1	58.6±0.4	58.4±0.8	**59.2±0.9**	58.8±0.8
	R	21.7±1.6	22.1±0.9	21.9±0.9	**22.8±0.6**	24.6±1.2	24.7±0.8	25.2±0.7	**25.2±0.9**	28.2±0.5	28.4±0.5	28.5±0.7	**28.6±0.3**
	F_1	30.6±1.9	31.0±1.1	31.0±1.1	**31.9±0.8**	34.4±1.3	34.6±0.9	35.0±0.8	**35.0±1.1**	38.0±0.5	38.2±0.6	**38.5±0.8**	38.5±0.4
C-a	H	100.0±0.7	**99.2±0.8**	100.3±0.5	99.6±0.4	96.0±0.5	**95.5±0.4**	95.9±0.5	95.7±0.4	**95.5±0.5**	95.9±0.8	95.9±0.6	95.7±0.6
	A	7.5±1.6	**8.5±1.1**	7.7±1.2	8.2±1.3	11.0±0.8	11.4±0.8	11.0±0.6	**11.5±0.5**	13.6±0.8	13.8±0.7	13.8±0.6	**13.9±0.8**
	P	32.6±4.5	**35.8±3.3**	32.8±3.3	35.2±3.7	42.9±1.4	**43.8±1.2**	42.7±1.1	43.4±0.1	47.2±1.6	47.1±1.3	46.9±1.1	**47.3±1.5**
	R	8.9±2.0	**10.0±1.3**	9.2±1.5	9.7±1.6	13.2±1.1	13.6±1.0	13.2±0.8	**13.7±0.7**	16.6±1.2	**17.1±1.0**	17.0±0.9	17.0±1.1
	F_1	14.0±2.9	**15.6±1.9**	14.3±2.1	15.2±2.4	20.2±1.4	20.7±1.3	20.2±1.0	**20.9±0.9**	24.6±1.5	**25.1±1.2**	24.9±1.1	25.0±1.3
C-t	H	97.1±0.7	96.6±0.5	96.9±0.6	**96.4±0.3**	93.6±0.3	**93.5±0.2**	93.6±0.3	93.6±0.3	95.2±0.5	95.3±0.7	**95.1±0.5**	95.2±0.6
	A	10.9±2.3	**11.8±0.8**	11.0±1.2	11.8±0.4	11.9±1.0	12.4±0.6	**12.8±0.6**	12.4±1.0	14.4±0.6	**14.5±0.4**	14.4±0.5	**14.5±0.4**
	P	34.9±5.1	36.8±1.5	35.4±2.5	**37.4±1.2**	38.2±1.8	38.7±0.8	**39.4±0.9**	38.6±1.8	40.9±0.9	41.1±0.6	**41.1±0.8**	40.9±0.6
	R	13.0±2.9	**13.9±1.1**	13.0±1.5	13.9±0.7	13.8±1.3	14.5±0.8	**15.1±0.9**	14.5±1.4	17.6±0.9	17.7±0.8	17.7±0.8	**17.8±0.7**
	F_1	18.9±3.9	20.2±1.3	19.0±2.0	**20.3±0.9**	20.3±1.7	21.1±0.9	**21.9±1.1**	21.1±1.7	24.6±1.0	24.7±0.8	24.7±0.9	**24.8±0.7**
Zhi	H	**95.3±0.3**	95.4±0.4	95.5±0.4	95.4±0.3	93.4±0.2	**93.3±0.2**	**93.3±0.2**	93.4±0.3	95.3±0.5	**95.1±0.4**	95.3±0.3	95.2±0.6
	A	13.9±0.8	**14.6±0.3**	14.4±0.7	14.3±0.5	15.3±0.8	**15.7±0.5**	15.6±0.7	**15.7±0.5**	**15.6±0.5**	15.6±0.3	15.6±0.2	**15.6±0.5**
	P	23.8±1.1	**24.9±0.5**	24.7±1.0	24.5±0.8	25.7±1.2	**26.5±0.7**	26.3±1.1	26.4±0.9	25.7±0.8	**25.9±0.5**	25.8±0.5	25.8±0.9
	R	15.4±0.9	**16.2±0.4**	16.1±0.9	15.9±0.6	16.7±1.0	**17.3±0.6**	17.0±0.8	17.2±0.6	17.7±0.5	17.8±0.4	17.8±0.2	**17.8±0.6**
	F_1	18.7±1.0	**19.6±0.5**	19.5±1.0	19.3±0.7	20.3±1.1	**20.9±0.7**	20.7±0.9	20.8±0.7	21.0±0.7	21.1±0.4	21.1±0.3	**21.1±0.7**

For H, the smaller the better; for A, P, R, and F_1, the larger, the better. The best results are in **bold** font for each category of models.

(about threefold to fourfold), a fewer number of labels (around 40%–60%), and fewer average number of labels per document (around 20%–30%), as shown in Table I. We also noticed that the result of the hamming loss was not always consistent with the other four metrics. The Hamming loss measures the symmetric difference between two sets, which treats every label equally; while the example-based metrics, accuracy, precision, recall, and F_1-score, are scaled by the length of the actual label set and/or the predicted label set. From the results, we observed that the relative difference of the hamming loss among HAN and JMAN, and its downgraded variants, JMAN-s, JMAN-s-tg, and JMAN-s-att, were all marginal. Compared with SVM and LDA, JMAN and its variants performed significantly better in terms of all metrics on all data sets, except a few cases where the LDA produced higher recall but much lower precision and F_1.

2) Results on Semantic-Based Loss Regularizers: To test the effectiveness of the semantic-based loss regularizers L_{sim} and L_{sub}, we applied them (either separately or collectively) on Bi-GRU, HAN, and JMAN-s and reported the results with tenfold cross-validation on the testing data.

From Table III, it can be seen that models with the semantic-based loss regularizers (either one or both) consistently performed better than the original models; 0.9%–1.6% absolute gain of F_1 was observed for Bi-GRU and 0.6%–1.6% for HAN. For the JMAN-s model, the improvement with the semantic-based loss regularizers is less obvious; there was only 0.1%–0.5% absolute increase of F_1. It is hard to draw a clear conclusion in which the L_{sim} and L_{sub} were more effective in further improving the model performance. This may depend on which of the semantic relations, similarity or subsumption were more prominent in the label sets. The results showed that L_{sim} and L_{sub} complement to each other and achieved the best results in around half of the experimental settings. For other cases, using either L_{sim} or L_{sub} performed better than using them together.

The results produced by adding the semantic-based loss regularizers indeed coincided with our initial perception and expectation that model performance could be further improved by exploiting the label correlations with help of external knowledge bases. However, most of the differences in the evaluation settings were not statistically significant. The evaluation result was generally in line with the one produced in the existing research that also leveraged label correlation in multilabel classification. The work using a weight initialization approach in [42] reported a performance gain of less than 1% in F_1 in most experimental settings. The proposed approach is more feasible than the weight initialization approach [42] for data with large label sizes, typically in the context of automated annotation, as explained in Section II-C.

The marginal improvement from experiments was probably due to the fact that the shared weights in the layers prior to the output layer in the neural networks might already indirectly model some of the correlations among the output nodes. This might also explain why JMAN-s is less boosted by the regularizers than Bi-GRU and HAN. We also noticed that the work in [19] reported somehow different results, i.e., that the binary cross-entropy loss L_{CE} achieved better performance than the pairwise ranking loss [27], which also considers label correlation. We believe that exploiting label correlation from external knowledge bases for a wide array of multilabel classification problems is necessary and useful, but, obviously, this is a challenging problem and needs further studies.

E. Training Time and Model Convergence

In Table IV, we reported the mean and standard deviation of training time spent per fold for each model in tenfold cross-validation. With the efficient and highly scalable implementation of the Gibbs sampling in MALLET [58], the LDA model took the least time for training. Among the other models, JMAN-s was the most efficient in training despite its relatively more complex architecture, by around 21.2%–54.7% faster than Bi-GRU and around 13.3%–23.2% faster than HAN on all data sets. The training time increased when the semantic-based loss regularizers were used. The increased time was related to the document size $|X|$, label size $|Y|$, and the average length of the label sets Ave of the data set.

TABLE IV

COMPARISON OF TRAINING TIME FOR ALL MODELS IN SECONDS

	SVM	LDA	Bi-GRU	Bi-GRU+s	HAN	HAN+s	JMAN-s-tg	JMAN-s-att	JMAN-s	JMAN
Bib	1107 ± 12	**110 ± 2**(1)	1480 ± 92	1683 ± 78	1164 ± 52	1434 ± 74	1075 ± 87	**1024 ± 100**(3)	**894 ± 55**(2)	1138 ± 86
C-a	1660 ± 31	**113 ± 3**(1)	869 ± 288	877 ± 57	462 ± 63	554 ± 45	434 ± 49	**429 ± 41**(3)	**394 ± 33**(2)	468 ± 38
C-t	4796 ± 50	**210 ± 7**(1)	1635 ± 1034	1469 ± 276	858 ± 100	947 ± 115	**752 ± 52**(3)	780 ± 69	**744 ± 62**(2)	839 ± 49
Zhi	over 1 day	**903 ± 31**(1)	1455 ± 69	2459 ± 151	1387 ± 78	2388 ± 275	**1220 ± 81**(3)	1275 ± 99	**1147 ± 44**(2)	1712 ± 105

Training time of the three most efficient models are in **bold** and marked with a ranking index in brackets. BiGRU+s and HAN+s denote the models with semantic-based loss regularisers.

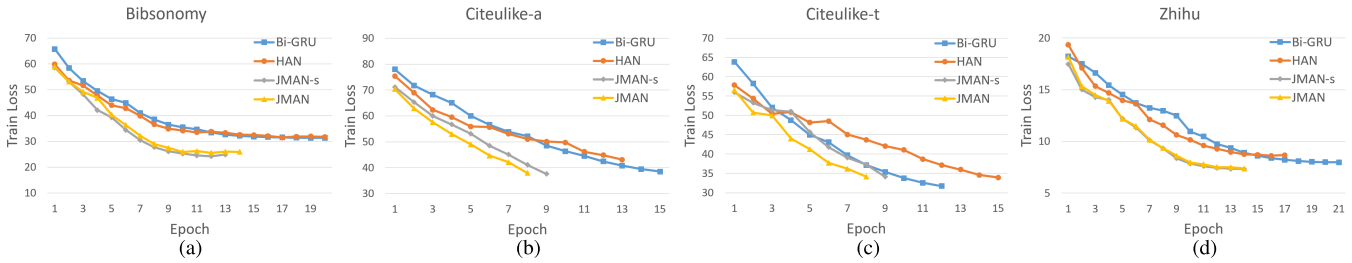


Fig. 3. Convergence plot: training loss with respect to the number of training epochs for the Bi-GRU, HAN, JMAN-s and JMAN models. (a) Bibsonomy. (b) CiteUlike-a. (c) CiteUlike-t. (d) Zhihu.

TABLE V

COMPARISON RESULTS OF USING DIFFERENT SOURCE INFORMATION (TITLE, CONTENT, AND TITLE-GUIDED CONTENT REPRESENTATIONS) IN THE JMAN MODEL ON THE FOUR SOCIAL ANNOTATION DATA SETS IN TERMS OF HAMMING LOSS (H), ACCURACY (A), PRECISION (P), RECALL (R), AND F_1-SCORE (F_1)

		Title (c_t)	Content (c_a)	Content, title-guided (c_{ta})	JMAN-s-tg ($[c_t, c_a]$)	JMAN-s-att ($[c_t, c_{ta}]$)	JMAN-s ($[c_t, c_{ta}, c_a]$)
Bib	H	88.7 ± 0.8	87.7 ± 0.7	86.8 ± 0.5	**84.5 ± 0.5**	84.6 ± 0.3	85.2 ± 0.5
	A	17.0 ± 1.1	20.4 ± 1.1	21.2 ± 0.5	24.1 ± 0.6	24.2 ± 0.6	**24.8 ± 0.4**
	P	50.4 ± 1.6	54.7 ± 1.7	55.4 ± 0.6	59.1 ± 1.0	**59.2 ± 1.0**	58.6 ± 0.4
	R	18.4 ± 1.2	22.8 ± 1.3	23.7 ± 0.6	26.9 ± 0.6	27.2 ± 0.7	**28.2 ± 0.5**
	F_1	26.9 ± 1.5	32.2 ± 1.6	33.2 ± 0.7	37.0 ± 0.7	37.3 ± 0.8	**38.0 ± 0.5**
C-a	H	96.4 ± 0.2	97.1 ± 0.3	97.0 ± 0.3	94.6 ± 0.5	**94.5 ± 0.3**	95.5 ± 0.5
	A	7.3 ± 0.4	9.5 ± 0.5	9.6 ± 0.9	13.5 ± 0.6	13.4 ± 0.4	**13.6 ± 0.8**
	P	34.0 ± 1.5	39.2 ± 1.4	39.5 ± 1.4	47.9 ± 1.2	**48.4 ± 0.8**	47.2 ± 1.6
	R	8.3 ± 0.6	11.4 ± 0.7	11.5 ± 1.3	16.3 ± 0.8	16.0 ± 0.6	**16.6 ± 1.2**
	F_1	13.3 ± 0.8	17.6 ± 1.0	17.8 ± 1.7	24.3 ± 1.0	24.1 ± 0.7	**24.6 ± 1.5**
C-t	H	96.1 ± 0.3	95.4 ± 0.5	95.2 ± 0.3	94.2 ± 0.3	**94.0 ± 0.4**	95.2 ± 0.5
	A	5.7 ± 0.9	10.3 ± 0.4	10.5 ± 1.1	13.6 ± 0.6	13.5 ± 0.3	**14.4 ± 0.6**
	P	21.2 ± 2.5	33.3 ± 1.0	34.0 ± 1.7	39.8 ± 1.2	40.0 ± 0.8	**40.9 ± 0.9**
	R	6.5 ± 1.1	12.1 ± 0.6	12.3 ± 1.5	16.2 ± 0.8	16.2 ± 0.4	**17.6 ± 0.9**
	F_1	9.9 ± 1.5	17.8 ± 0.8	18.0 ± 1.9	23.0 ± 1.0	23.0 ± 0.5	**24.6 ± 1.0**
Zhi	H	97.0 ± 0.2	97.2 ± 0.2	94.9 ± 0.2	**94.3 ± 0.3**	94.6 ± 0.3	95.3 ± 0.5
	A	7.1 ± 0.8	7.4 ± 0.4	9.7 ± 0.8	15.5 ± 0.3	15.3 ± 0.4	**15.6 ± 0.5**
	P	12.2 ± 1.1	12.6 ± 0.7	17.2 ± 1.2	25.7 ± 0.5	25.4 ± 0.7	**25.7 ± 0.8**
	R	7.8 ± 0.9	8.1 ± 0.5	10.4 ± 0.9	17.5 ± 0.3	17.4 ± 0.5	**17.7 ± 0.5**
	F_1	9.5 ± 1.0	9.9 ± 0.6	13.0 ± 1.0	20.8 ± 0.3	20.7 ± 0.5	**21.0 ± 0.7**

For H, the smaller the better; for A, P, R, and F_1, the larger, the better. The best results are in **bold**.

The SVM-ovr model was the least efficient as it trained one SVM RBF classifier for every single label, and the number of unique labels in the data sets was large.

The difference in training time among the neural network-based models, Bi-GRU, HAN, JMAN-s, and JMAN, can also be explained by the convergence plots in Fig. 3. The total number of epochs for each model was determined by early stopping based on the validation set. On all four data sets, JMAN and JMAN-s converged much faster than Bi-GRU and HAN, with fewer training epochs and steeper convergence plots. This showed that JMAN and JMAN-s can learn a better representation of the input documents with fewer epochs than HAN and Bi-GRU.

F. Analysis of Multisource Components

The architecture described in Section III-C combines the title representation c_t, content c_a, and title-guided content c_{ta}. It is worth analyzing how the different source of the representations contributes to the performance of annotation. Table V presents the results with c_t, c_a, c_{ta}, and different combinations of them on the four data sets, without the use of semantic regularizers. The JMAN-s model concatenates all three representations, while JMAN-s-tg and JMAN-s-att are combinations of title representation and one of the content representations. It is clear that the JMAN-s model, with the representation of [c_t, c_a, c_{ta}], performed the best among all models. A similar level of performance was observed in using JMAN-s-tg and JMAN-s-att, where either the title-guided content representation ("-tg") or the original content representation ("-att") was excluded. When only one type of representation was used, the title-guided content representation performed the best. While a single user may tend to provide annotations based on the title or the abstract only and browse the content selectively, their collective annotations tend to reflect the whole document. The results confirmed the advantage of using multisource information for document representation.

G. Attention Visualization

We can further understand how the hierarchical attention mechanisms work, especially the guided attention mechanism,

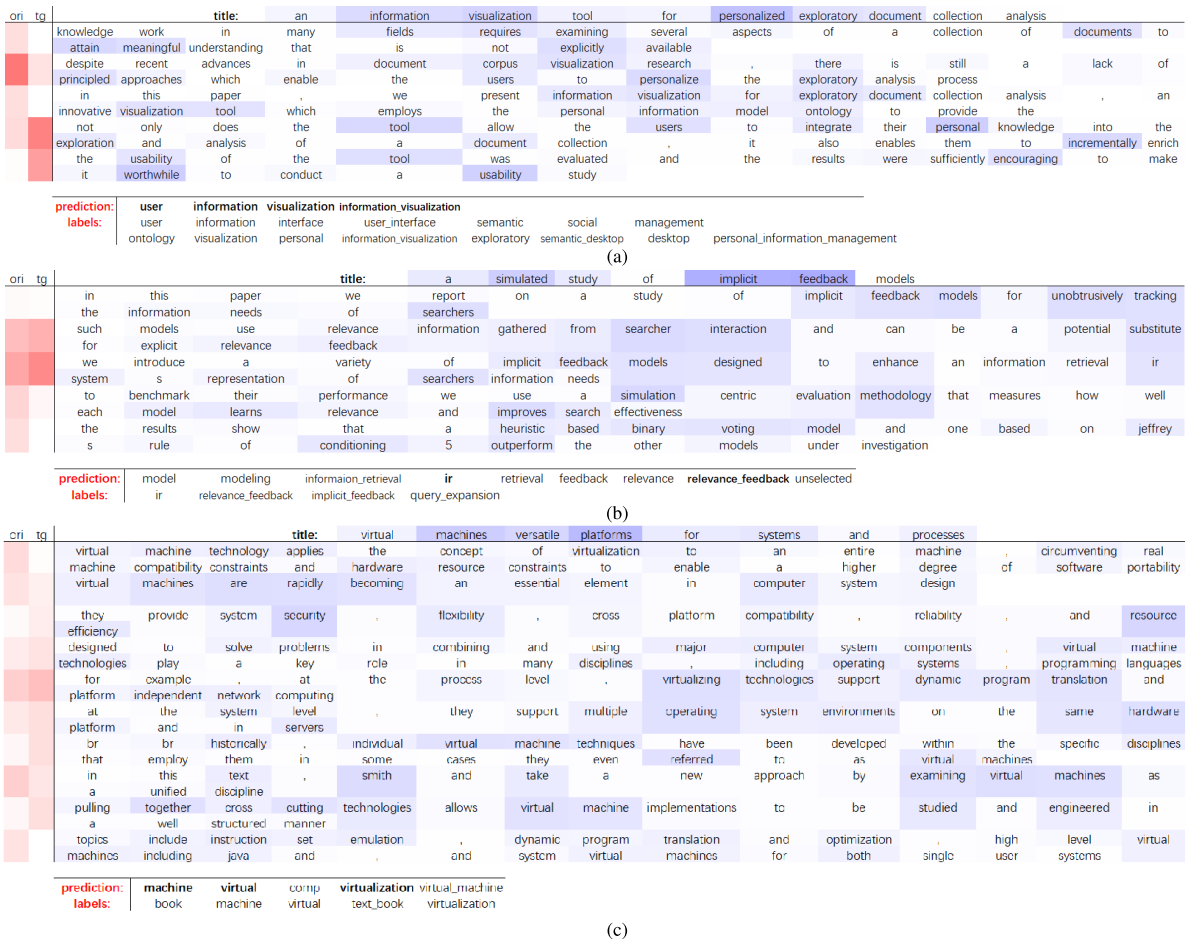


Fig. 4. Attention visualization of the proposed JMAN model for documents in Bibsonomy, CiteULike-a, and CiteULike-t. Red blocks in the leftmost two columns show the original ("ori") and the title-guided ("tg") sentence-level attention weights, respectively. Purple blocks mark the word-level attention weights for the title (the first row) and each sentence (every two rows) in the abstract. The darker the color, the greater the amount of attention was paid to the word or sentence in JMAN. The predicted labels and the "ground truth" labels are displayed below each diagram. (a) Bibsonomy Example. (b) CiteULike-a Example. (c) CiteULike-t Example.

by visualizing the attention weights in Fig. 4. Four attention weights in JMAN were illustrated for sample documents from Bibsonomy, CiteULike-a, and CiteULike-t: 1) word-level attention for the title; 2) word-level attention for each sentence in the abstract; 3) original sentence-level attention for the abstract; and 4) title-guided attention for the abstract. Documents and labels in the Zhihu data set were not interpretable as all words had been officially masked with an unknown codebook.

In Fig. 4, the purple blocks denote the attention weights of each word in the title (the first row) or a sentence (below the first row every two rows represent a sentence). The red blocks in the leftmost columns denote the sentence-level attention weights, where the left one ("ori") displays the original sentence-level attention weights and the right one ("tg") displays the title-guided sentence-level attention weights. The darker the color, the greater the amount of attention was paid to a word or sentence. The predicted labels by the JMAN model and the ground truth labels are shown below each diagram.

It can be seen that the word-level attention indeed highlighted many of the most informative words (from either the title or sentences). These informative words were either the same as or highly related to the true labels or the topics of the document, for example, "information," "user," "personalized," and "visualization" in the Bibsonomy example; "implicit," "feedback," "ir," "models," and "searcher" in the CiteULike-a example; and "machine," "virtualizing," "platform," "virtual," and "operating" in the CiteULike-t example. Words that conveyed no meanings regarding the topics of the document, such as the stop words and many uninformative ones, were assigned nearly zero weight (e.g., white color in the blocks).

The title-guided sentence-level attention ("tg") assigned different weights and provided a distinct "view" from the original sentence-level attention ("ori"). In the Bibsonomy example, the "ori" weights highlighted mostly the second sentence (a general statement that identifies the gap in the literature), while the "tg" weights highlighted more the fourth (a statement of a tool that allows integrating personal knowledge into the exploration of a document collection) and fifth sentences (continuation of the previous statement on the tool's usability). These two sentences are well aligned to the title and intuitive for users to determine the main themes of the document for annotation.

This difference was also present in the other two examples. As discussed in Section III-D, concatenating the output from both attention mechanisms would help gain a more

comprehensive understanding of the documents and provide more accurate annotation (as indicated by the comparison results with JMAN-s-tg, JMAN-s-att, and JMAN-s in Table II). This is because the abstract of a document may contain more useful and important information that is not present in the title. For example, in the CiteULike-a example, the "tg" weights highlighted only the second and third sentences that aligned well to the title, while the "ori" weights also emphasized the fourth and fifth sentences that talked about the "simulation," "evaluation" and two specific models. Although they were not well aligned to the title, they represented important information for document understanding. There was also a certain degree of agreement between the two attention weights, for instance, in the CiteULike-a example, both attention weights were low for the first sentence (a general introduction) and high for the second (more detail about the topic) and the third sentences (more on the authors' work). The degree of agreement was even higher in the CiteULike-t example.

From the predicted results, we can see that the JMAN model suggested meaningful labels (more prediction results are available at https://github.com/acadTags/Automated-Social-Annotation). The predicted labels had substantial overlap with the "ground truth" labels but still have the potential for improvement, especially in terms of recall. We also noticed that the true labels also contained some that were useless or not related to the topics of the document, for example, "book" and "text_book" in the CiteULike-t example. It was very interesting to see that the predicted labels not included in the "ground truth" were indeed highly relevant to the themes of the documents, which should have been used for annotation, e.g., "information_retrieval," "retrieval," "modeling," and "relevance" in the CiteULike-a example and "virtual_machine" in the CiteULike-t example. Besides automated annotation, the proposed approach also has the potential to enhance the quality of existing annotations.

V. Conclusion

This work focused on two main issues in using a deep learning-based method for automated social annotation as a multilabel classification problem: 1) how to design a deep network according to users' reading and annotation behavior to achieve better classification performance and 2) how to leverage label correlation to further improve the performance of the classification. The proposed model, i.e., JMAN, introduces a title-guided attention mechanism that can extract informative sentences from a document to aid annotation. The design is in line with the previous studies on a statistical analysis of users' annotation behavior and the impact of the titles of documents [5], [18]. To tackle the challenging issue of label correlation in the high-dimensional label space [20], [21], we proposed two semantic-based loss regularizers that can enforce the output of the neural network to conform to the semantic relations among labels, i.e., similarity and subsumption. Extensive experiments on four large, real-world social media data sets demonstrated the superior performance of JMAN, in terms of accuracy and F_1-score, over the state-of-the-art baseline models and their variants. Furthermore, there was a substantial reduction of training time for the JMAN without using the semantic-based loss regularizers. Analysis of the multisource components showed the advantage of using the title-guided content representation and the proposed multiple sources in the document representation.

While it is a consensus that making use of the label correlation from quality external knowledge bases for multilabel classification is necessary and useful, we did find that the performance gain tended to be marginal. In addition, the parameter tuning for the semantic-based loss regularizers was a time-consuming process even though, without them, the proposed JMAN still greatly outperformed the state-of-the-art deep learning-based models. As a potential remedy, we showed that through a dynamic update of Sim and Sub, the results were improved in two of the data sets but with the cost of increased computation. A more efficient method for dynamic update of Sim and Sub in the loss regularizers merits further study. It is also worth exploring other types of guided attention mechanisms, for example, in microblog annotation, a message may be guided by the profile or historical microblogs from the same user and comments of the microblog, or even guided by external information of different modalities, such as sensor data in annotating events. The proposed model could also shed light on the open problem of extreme multilabel text classification problem [68], where there are hundreds of thousands or even millions of possible labels. Another important direction is to extend the current approach to deal with emerging new labels, as discussed in [69]. Although we mainly focused on RNN-based classification models in this work, which has been commonly used for text processing, it is also interesting to integrate the semantic-based loss regularizers and ensemble our model with other neural networks for social text annotation, including sequence-to-sequence networks [32], [70], CNNs [71], attention-based network transformer [72] and transfer-learning-based approaches, and bidirectional encoder representations from transformers (BERT) [73].

Acknowledgment

Part of this work has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF) (http://www.ecdf.ed.ac.uk/).

References

[1] A. Zubiaga, V. Fresno, R. Martinez, and A. P. Garcia-Plaza, "Harnessing folksonomies to produce a social classification of resources," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1801–1813, Aug. 2013.

[2] D. R. Millen and J. Feinberg, "Using social tagging to improve social navigation," in *Proc. Workshop Social Navigat. Community Based Adaptation Technol.* Princeton, NJ, USA: Citeseer, 2006.

[3] F. Gedikli and D. Jannach, "Recommender systems, semantic-based," in *Encyclopedia of Social Network Analysis and Mining*, R. Alhajj and J. Rokne, Eds. New York, NY, USA: Springer, 2014, pp. 1501–1510.

[4] M. Strohmaier, D. Helic, D. Benz, C. Körner, and R. Kern, "Evaluation of folksonomy induction algorithms," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 4, pp. 74:1–74:22, Sep. 2012.

[5] F. Figueiredo *et al.*, "Assessing the quality of textual features in social media," *Inf. Process. Manage.*, vol. 49, no. 1, pp. 222–247, Jan. 2013.

[6] L. Nie, Y.-L. Zhao, X. Wang, J. Shen, and T.-S. Chua, "Learning to recommend descriptive tags for questions in social forums," *ACM Trans. Inf. Syst.*, vol. 32, no. 1, pp. 1–23, Jan. 2014.

[7] F. Jabeen and S. Khusro, "Quality-protected folksonomy maintenance approaches: A brief survey," *Knowl. Eng. Rev.*, vol. 30, no. 5, pp. 521–544, Nov. 2015.

[8] F. M. Belém, J. M. Almeida, and M. A. Gonçalves, "A survey on tag recommendation methods," *J. Assoc. Inf. Sci. Technol.*, vol. 68, no. 4, pp. 830–844, Apr. 2017.

[9] E. Zangerle, W. Gassler, and G. Specht, "Recommending#-tags in twitter," in *Proc. Workshop Semantic Adapt. Social Web (SASWeb). CEUR Workshop*, vol. 730, 2011, pp. 67–78.

[10] H. Wang, B. Chen, and W.-J. Li, "Collaborative topic regression with social regularization for tag recommendation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.* Palo Alto, CA, USA: AAAI Press, 2013, pp. 2719–2725.

[11] Z. Ding, X. Qiu, Q. Zhang, and X. Huang, "Learning topical translation model for microblog hashtag suggestion," in *Proc. 23rd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2013, pp. 2078–2084.

[12] H. A. M. Hassan, G. Sansonetti, F. Gasparetti, and A. Micarelli, "Semantic-based tag recommendation in scientific bookmarking systems," in *Proc. 12th ACM Conf. Recommender Syst.* New York, NY, USA: ACM, Sep. 2018, pp. 465–469.

[13] Q. Zhang, J. Wang, H. Huang, X. Huang, and Y. Gong, "Hashtag recommendation for multimodal microblog using co-attention network," in *Proc. 26th Int. Joint Conf. Artif. Intell.* Palo Alto, CA, USA: AAAI Press, Aug. 2017, pp. 3420–3426.

[14] Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network," in *Proc. IJCAI*, 2016, pp. 2782–2788.

[15] Y. Li, T. Liu, J. Jiang, and L. Zhang, "Hashtag recommendation with topical attention-based LSTM," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 3019–3029.

[16] H. Huang, Q. Zhang, Y. Gong, and X. Huang, "Hashtag recommendation using end-to-end memory networks with hierarchical attention," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 943–952.

[17] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[18] M. Lipczak and E. Milios, "The impact of resource title on tags in collaborative tagging systems," in *Proc. 21st ACM Conf. Hypertext Hypermedia (HT)*. New York, NY, USA: ACM, 2010, pp. 179–188.

[19] J. Nam, J. Kim, E. L. Mencia, I. Gurevych, and J. Fürnkranz, *Large-Scale Multi-label Text Classification—Revisiting Neural Networks*. Berlin, Germany: Springer, 2014, pp. 437–452.

[20] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[21] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 52:1–52:38, 2015.

[22] W. G. Stock, "Concepts and semantic relations in information science," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 61, no. 10, pp. 1951–1969, 2010.

[23] I. Peters, "Knowledge representation in Web 2.0: Folksonomies," in *Folksonomies. Indexing and Retrieval in Web 2.0* (Knowledge and Information). Berlin, Germany: De Gruyter, 2009, pp. 153–282.

[24] G. Kurata, B. Xiang, and B. Zhou, "Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 521–526.

[25] Y. Zhu, J. T. Kwok, and Z.-H. Zhou, "Multi-label learning with global and local label correlation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1081–1094, Jun. 2018.

[26] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Advances in Knowledge Discovery and Data Mining*, H. Dai, R. Srikant, and C. Zhang, Eds. Berlin, Germany: Springer, 2004, pp. 22–30.

[27] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[29] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.

[30] D. Liang *et al.*, "Adaptive multi-attention network incorporating answer information for duplicate question detection," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.* New York, NY, USA: Association Computing Machinery, Jul. 2019, pp. 95–104, doi: 10.1145/3331184.3331228.

[31] A. Kumar, D. Kawahara, and S. Kurohashi, "Knowledge-enriched two-layered attention network for sentiment analysis," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, 2018, pp. 253–258.

[32] Y. Wang, J. Li, I. King, M. R. Lyu, and S. Shi, "Microblog hashtag generation via encoding conversation contexts," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 1624–1633.

[33] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[34] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[35] P. Heymann and H. Garcia-Molina, "Collaborative creation of communal hierarchical taxonomies in social tagging systems," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 2006-10, Apr. 2006.

[36] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Machine Learning and Knowledge Discovery in Databases*, W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, Eds. Berlin, Germany: Springer, 2009, pp. 254–269.

[37] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.

[38] B. Chen, W. Li, Y. Zhang, and J. Hu, "Enhancing multi-label classification based on local label constraints and classifier chains," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 1458–1463.

[39] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *Proc. ECML/PKDD Workshop Mining Multidimensional Data*, vol. 21, Sep. 2008, pp. 53–59.

[40] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *Proc. 15th Int. Conf. Multimedia*. New York, NY, USA: ACM, 2007, pp. 17–26.

[41] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)* New York, NY, USA: ACM, 2010, pp. 999–1008.

[42] S. Baker and A. Korhonen, "Initializing neural networks for hierarchical multi-label text classification," in *Proc. BioNLP*, 2017, pp. 307–315.

[43] J. Wehrmann, R. C. Barros, S. N. D. Dôres, and R. Cerri, "Hierarchical multi-label classification with chained neural networks," in *Proc. Symp. Appl. Comput. (SAC)* New York, NY, USA: Association Computing Machinery, 2017, pp. 790–795, doi: 10.1145/3019612.3019664.

[44] H. Dong, W. Wang, K. Huang, and F. Coenen, "Joint multi-label attention networks for social text annotation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 1348–1354.

[45] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[46] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[48] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[49] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[50] W. Chen, Y. Gao, J. Zhang, I. King, and M. R. Lyu, "Title-guided encoding for keyphrase generation," 2018, *arXiv:1808.08575*. [Online]. Available: http://arxiv.org/abs/1808.08575

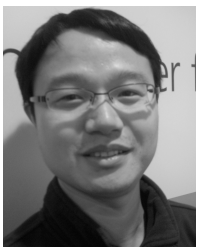
[51] J. W. Tanaka and M. Taylor, "Object categories and expertise: Is the basic level in the eye of the beholder?" *Cognit. Psychol.*, vol. 23, no. 3, pp. 457–482, Jul. 1991.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[53] D. Benz *et al.*, "The social bookmark and publication management system bibsonomy," *VLDB J.*, vol. 19, no. 6, pp. 849–875, Dec. 2010.

[54] H. Dong, W. Wang, and C. Frans, "Deriving dynamic knowledge from academic social tagging data: A novel research direction," in *Proc. iConference*. Wuhan, China: iSchools, 2017, pp. 661–666.

[55] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Conf. Operating Syst. Design Implement.* Berkeley, CA, USA: USENIX Association, 2016, pp. 265–283.
[56] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[57] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.
[58] A. K. McCallum, "Mallet: A machine learning for language toolkit," Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep., 2002. [Online]. Available: http://mallet.cs.umass.edu
[59] Y. Song, L. Zhang, and C. L. Giles, "Automatic tag recommendation algorithms for social recommender systems," *ACM Trans. Web*, vol. 5, no. 1, pp. 1–31, Feb. 2011.
[60] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[61] F. Tai and H.-T. Lin, "Multilabel classification with principal label space transformation," *Neural Comput.*, vol. 24, no. 9, pp. 2508–2542, Sep. 2012.
[62] P. Szymański and T. Kajdanowicz, "A scikit-based Python environment for performing multi-label classification," 2017, *arXiv:1702.01460*. [Online]. Available: http://arxiv.org/abs/1702.01460
[63] J. Read, P. Reutemann, B. Pfahringer, and G. Holmes, "Meka: A multi-label/multi-target extension to Weka," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 667–671, Jan. 2016.
[64] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, Nov. 2009, doi: 10.1145/1656274.1656278.
[65] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A Java library for multi-label learning," *J. Mach. Learn. Res.*, vol. 12, pp. 2411–2414, Jun. 2011.
[66] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA, USA: Springer, 2010, pp. 667–685.
[67] M. S. Sorower, "A literature survey on algorithms for multi-label learning," Oregon State Univ., Corvallis, OR, USA, 2010, vol. 18, pp. 1–25.
[68] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.* New York, NY, USA: Association Computing Machinery, Aug. 2017, pp. 115–124, doi: 10.1145/3077136.3080834.
[69] Y. Zhu, K. M. Ting, and Z.-H. Zhou, "Multi-label learning with emerging new labels," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 10, pp. 1901–1914, Oct. 2018.
[70] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112.
[71] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.
[72] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[73] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805



Hang Dong received the bachelor's degree in library science from Wuhan University, Wuhan, China, in 2013, the master's degree in information systems from The University of Sheffield, Sheffield, U.K., in 2015, and the Ph.D. degree in computer science from the University of Liverpool, Liverpool, U.K., and Xi'an Jiaotong-Liverpool University, Suzhou, China, in 2020.

He is currently a Research Fellow with the Usher Institute of Population Health Science and Informatics, The University of Edinburgh, Edinburgh, U.K. His research interests include data mining, natural language processing, and machine learning.

Dr. Dong is a member of the Association of Computational Linguistics (ACL).

Wei Wang received the Ph.D. degree in computer science from the University of Nottingham, Nottingham, U.K., in 2009.

He was a Lecturer with the University of Nottingham Malaysia Campus, Semenyih, Malaysia, and later a Post-Doctoral Research Fellow with the Centre for Communication Systems Research (now known as the Institute for Communication Systems), University of Surrey, Guildford, U.K. He is currently an Associate Professor with the Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China. He has published more than 50 articles in reputed journals and conferences related to the Internet of Things and knowledge discovery. His research interests lie in the broad area of data and knowledge engineering, in particular, knowledge discovery from textual data, social media data and smart city data, semantic search, and deep learning for data processing.

Dr. Wang also serves as a reviewer for a number of prestigious international journals.

Kaizhu Huang (Member, IEEE) is the Founding Director of the Suzhou Municipal Key Laboratory of Cognitive Computation and Applied Technology and was the Head of the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China. He is currently a Professor with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, and a Visiting Scholar at the Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou, China. He has been working on machine learning, neural information processing, and pattern recognition. Until October 2018, he has published eight books in Springer and over 160 international research articles at prestigious journals and conferences.

Dr. Huang was a recipient of the 2011 Asia Pacific Neural Network Society (APNNS) Younger Researcher Award. He also received the Best Book Award in the National Three 100 Competition 2009. He has served as the chair for many international conferences and workshops. He also serves as an associate editor for three international journals and a board member in three international book series. He has been sitting in the grant evaluation panels in Hong Kong RGC, Singapore AI Programs, and NSFC, China.

Frans Coenen is currently a Professor with the Department of Computer Science, University of Liverpool, Liverpool, U.K., where he is also the Director for the University of Liverpool Doctoral Network in AI for Future Digital Health. He has a general background in AI and has been working in the field of data mining and knowledge discovery in data (KDD) for the last 15 years. He is interested in the application of the techniques of data mining and knowledge discovery in data to unusual data sets, such as: 1) graphs and social networks; 2) time series; 3) free text of all kinds; 4) 2-D and 3-D images, particularly medical images; and 5) video data. He is also interested in data mining over encrypted data. He currently leads a small research group working on many aspects of data mining and KDD. He has some 390 refereed publications on KDD and AI-related research and has been on the program committees for many KDD conferences and related events.