Automated Social Text Annotation with Joint Multi-Label Attention Networks

Hang Dong, Wei Wang, Kaizhu Huang, 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 multi-label classification problem. We propose a novel deep learning based method for this problem, and design an attention-based neural network with semantic-based regularisation, which can mimic users' reading and annotation behaviour 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 regularisers, i.e. similarity and subsumption, that enforce the output of the network to conform to label semantics. The model with the semantic-based loss regularisers is referred to as the Joint Multi-label 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 datasets. In terms of F_1, JMAN significantly outperformed Bi-GRU (Bidirectional Gated Recurrent Unit) relatively by around 12.8% to 78.6%, and the Hierarchical Attention Network (HAN) by around 3.9% to 23.8%. The JMAN model demonstrates advantages in convergence and training speed. Further improvement of performance was observed against LDA (Latent Dirichlet Allocation) and SVM (Support Vector Machine). When applying the semantic-based loss regularisers, 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*—Automated Social Annotation, Multi-Label Classification, Deep Learning, Attention Mechanisms, Recurrent Neural Networks**

I. Introduction

Tagging is a popular approach to organise 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 organise 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 to questions for better search and recommendation; in microblogging services like 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 organising 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 to other textual features, such as titles, descriptions and comments, for document classification [5]. Figure 1 displays an example of a published paper 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 of maintaining 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 organisation. 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 encode the input texts as continuous vector representations and approximate the matching from the input to the label space, where labels are often assumed to be orthogonal or independent to each other.

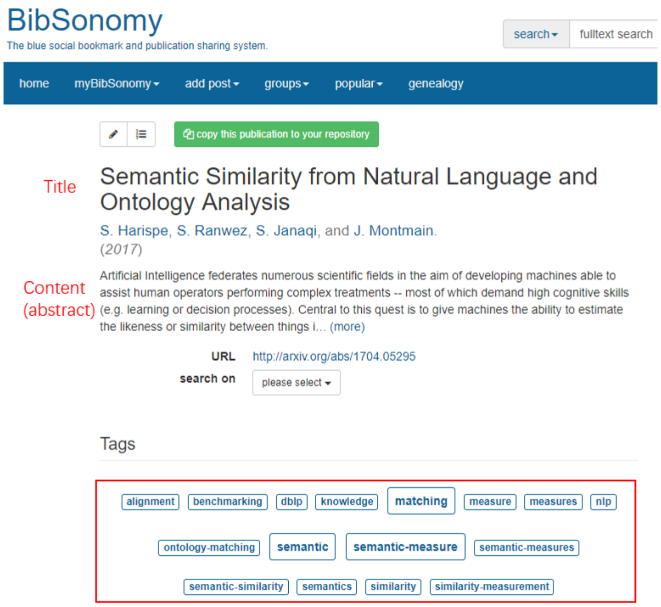
Our study shows that the existing deep learning based methods at least suffer two issues: (1) *The modelling of*

Manuscript received xxx xx, 2019; revised xxx xx, 2020; accepted xxx xx, 2020. This research is funded by the Research Development Fund at Xi'an Jiaotong-Liverpool University (XJTLU), contract number RDF-14-01-10, and partially supported by the following: The National Natural Science Foundation of China under no. 61876155; Natural Science Foundation of Jiangsu Province BK20181189; Key Program Special Fund in XJTLU under no. KSF-A-01, KSF-T-06, KSF-E-26, KSF-P-02, and KSF-A-10. H. Dong is also supported by the Human Phenotype Project in Health Data Research UK Scotland. 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/). *(Corresponding author: Wei Wang.)*

H. Dong and F. Coenen are with the Department of Computer Science, University of Liverpool (e-mail: HangDong@liverpool.ac.uk, Coenen@liverpool.ac.uk). H. Dong is also with the Department of Computer Science and Software Engineering, Xi'an Jiaotong Liverpool University, and Centre for Medical Informatics, Usher Institute, University of Edinburgh (email: hang.dong@ed.ac.uk).

W. Wang is with the Department of Computer Science and Software Engineering, Xi'an Jiaotong Liverpool University (email: Wei.Wang03@xjtlu.edu.cn).

K. Huang is with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong Liverpool University (email: Kaizhu.Huang@xjtlu.edu.cn), and Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou, China.



Semantic Similarity from Natural Language and Ontology Analysis

S. Harispe, S. Ranwez, S. Janaqi, and J. Montmain.
(2017)

Artificial Intelligence federates numerous scientific fields in the aim of developing machines able to assist human operators performing complex treatments -- most of which demand high cognitive skills (e.g. learning or decision processes). Central to this quest is to give machines the ability to estimate the likeness or similarity between things i... (more)

URL http://arxiv.org/abs/1704.05295
search on please select

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

reading and annotation behaviour (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 categorisation and tag recommendation [5]; and (2) *The semantics in the labels (label correlation)* - In prediction, the most common *multi-hot* (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 multi-label 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 behaviour 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 behaviour. To annotate a document, a user attempts to digest the meaning of the title first; 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 sentences 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 realised through a dynamic alignment of title and sentences, which also enables better explainability in the modelling and visualisation.

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 regularisers to constrain the network output to satisfy the similarity and subsumption relations among labels. The regularisers allow the model to leverage semantic relations that can be either matched to existing knowledge bases or inferred from datasets. We further explore the dynamic update of the semantic relations when optimising the loss regularisers.

The main contributions of the work are highlighted as follows: (1) we propose a Joint Multi-label Attention Network (JMAN) that models users' reading and annotation behaviour through title-guided attention mechanisms to encode the document; (2) we propose two semantic-based loss regularisers 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; and (3) we carry out extensive experiments on four large, social media datasets. 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 paper is organised 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 the joint multi-label learning method, including the title-guided attention mechanism and the semantic-based loss regularisers. In Section IV, the experiment and evaluation results are presented and discussed, with analysis on model convergence, multi-source components, and attention visualisation. In Section V, we conclude the paper 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 multi-label 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 *personalised*. 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, personalised 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-based*, *content-based*, *matrix factorisation-based*, *clustering-based*, *graph-based*, *learning to rank-based* 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 *multi-label 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 non-linear activations to a vector representation and tried to approximate the matching from the input to the labels. The advantage of multi-label deep learning models lies in their relatively straightforward problem formulation with strong approximation power on large datasets, resulting in better performance over traditional approaches [27]. Some of the notable deep models adapted for multi-label classification included variations of Recurrent Neural Networks (RNN) [12], [15], [16] and Convolutional Neural Networks (CNN) [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 feed-forward layer with softmax activation to model this alignment. This soft alignment can be visualised, 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] utilised three different alignments, dot product alignment for self-attention, element-wise alignment for cross-attention, concatenation-based alignment for co-attention 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 Hierarchical Attention Network [17] proposed *word-level* 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 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 lights 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 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] modelled the mutual and external alignment between texts and images in a microblog with a co-attention network for hashtag annotation. Our study, however, focuses on the relations between the title and content of a document, which naturally simulates users' reading behaviour during document annotation.

C. Label Correlation in Multi-Label Learning

In multi-label classification, each instance (document) is associated with a set of labels and the labels are usually correlated to each other [21], [34]. This is different from multi-class classification in which classes (labels) are assumed to be disjoint. Social annotation can be seen as a multi-label 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-word 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 *optimisation*, which are either the related terms (*text_mining* being a related application domain), or narrower terms (the specific algorithm *svm* and the sub-domain *optimisation*). The relations among these labels represent additional knowledge that can be exploited to potentially improve the performance of multi-label 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 multi-label classification is to construct many binary classifiers, one for each label. This approach, often referred to as *binary relevance* or *one-vs-rest*, however, completely ignores the correlations among labels [19], [20]. One main strategy to address this issue was to re-generate a feature space incorporating information on label correlation. An example was adapting discriminative classifier like Support Vector Machine (SVM) [26]. The Classifier Chain method extends this idea through incorporating the binary classification results in a chain as features to predict the next label [36]. The classifier chain can be randomised and embedded into an ensemble learning architecture [37] or mined using clustering and graph-based methods [38]. Instead of organising classifiers as a chain, the Hierarchy Of Multilabel classifiER (HOMER) [39] created a tree of classifiers, based on the hierarchical structure of labels pre-learned in an unsupervised manner. *Probabilistic graphical models* were also used to encode the correlation among labels, including *Gibbs Random Fields* [40] and *Bayesian Networks* [41].

Existing studies using *deep learning* for multi-label 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 *multi-hot* representation, e.g. [0 1 0 1 1] in a 5-dimensional 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 initialisation* [24]: initialising 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 initialisation. 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 organise neural networks as a chain for hierarchical label prediction, i.e., assigning a chained feed-forward neural network for each layer in a label hierarchy. Similarly to the idea of assigning dedicated neurons, this cannot be easily scaled to a massive number of label similarity and subsumption relations.

III. The Proposed Approach

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

A. Problem Statement

The automated annotation task can be formulated as a multi-label classification problem [19], [20]. Suppose X denoting the collection of textual sequences or instances (e.g. documents), and $Y = \{y_1, y_2, ..., y_n\}$ 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 *multi-hot* vector, $\overrightarrow{Y_i} = [y_{i1}, y_{i2}, ..., y_{in}]$ 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 illustrated in Figure 2, is an extension to our previous work [44]. Instead of feeding the whole text sequence X into the neural network as in Hierarchical Attention Network (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 *multi-hot* representation, $\overrightarrow{Y_i} \in \{0, 1\}^{|Y|}$.

There are four attention modules, shown as dotted edges in Figure 2: two word-level attention modules for the words in the title and in each sentence in the content, respectively; and 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 Multi-Source Hierarchical architecture allows different metadata in a document to be processed in different ways in parallel (Section III-C); (2) the title-guided sentence-level attention mechanism aims to explicitly model the reading behaviour of users during annotation (Section III-D); and (3) the semantic-based loss regularisers 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 (Section III-E).

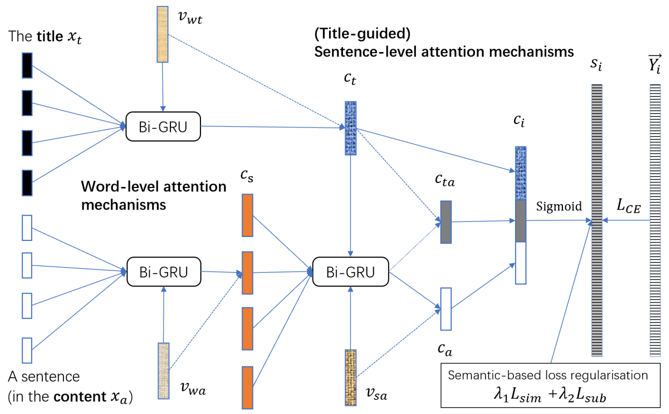


Fig. 2. The Joint Multi-label Attention Network (JMAN)

C. Multi-Source Hierarchical Architecture

The title of a document is a key feature which can greatly influence the decision of tagging [18] and the performance of classification [5]. We process the title and the content separately, and this Multi-Source 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 $x_t = (v_t^{(1)}, v_t^{(2)}, ..., v_t^{(\mathrm{n_t})})$ and $x_a = (v_a^{(1)}, v_a^{(2)}, ..., v_a^{(\mathrm{n_a})})$, where $\mathrm{n_t}$ or $\mathrm{n_a}$ 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 $e_t = W_e v_t$, $e_a = W_e v_a$, where $W_e \in \mathbb{R}^{d_e \times |V|}$ is the embedding weights that are usually pre-trained 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 Equation (1), where σ refers to a non-linear activation function (here we use the logistic sigmoid function), and $W_{er}, W_{ez}, W_{e\tilde{h}} \in \mathbb{R}^{d_h \times d_e}$, $W_{hr}, W_{hz}, W_{h\tilde{h}} \in \mathbb{R}^{d_h \times d_h}$ 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 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 $e^{(\mathrm{n})}$, to produce forward hidden states $(\overrightarrow{h^{(1)}},...,\overrightarrow{h^{(\mathrm{n})}})$; whereas the backward GRUs read the sequence reversely from $e^{(\mathrm{n})}$ to $e^{(1)}$ to calculate backward hidden states $(\overleftarrow{h^{(\mathrm{n})}},...,\overleftarrow{h^{(1)}})$. 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 Figure 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 emphasise the words towards the end of the sequence. Therefore, the attention mechanisms need to be applied to re-calculate 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 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 Figure 2, each accompanied by an attention layer(s): two word-level attention layers, for title and sentences in the abstract, respectively; and 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], [28]. The attention scores are based on an alignment of each hidden representation in a sequence to a non-static 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 Equation (2).

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

In Equation (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 sentences 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 sentences in the document, she would try to align the meaning of each sentences to the title. If a sentence conveys a piece of meaningful information based on her knowledge, especially the one aligns well to 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 the previous section are not enough to make a clear distinction among sentences. Firstly, the impact of the title on the document annotation is not considered, which is, however, particularly important during the tagging process [5], [18]. Secondly, in the attention mechanisms described in Equation (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 summarisation 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 Figure 2, which can be modelled using Equation (3):

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

where $h_s^{(r)}$ is the hidden state of the rth sentence; c_t is the title representation obtained from Equation (2); $\mathrm{n_s}$ denotes the number of sentences in the abstract; $\alpha_s^{(r)}$ is the sentence-level attention score; W_s, 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 behaviour, compared to 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 illustrated in Figure 2. The idea of the guided attention can be naturally generalised 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 against a number of state-of-the-art and baseline models.

E. Semantic-based Loss Regularisers

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 multi-label 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 initialisation approach [24], [42] that assigns a neuron in the penultimate layer of the neural network to "memorise" just one of the numerous label relations.

We take a different strategy by using the semantic-based loss regularisation, in which two loss regularisers are proposed to deal with the similarity and subsumption relations, respectively, jointly optimised 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 dataset itself or extracted through grounding the labels to concepts or terms in external knowledge bases. The whole joint loss is defined in Equation (4) below:

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

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 multi-label text classification with a feed-forward neural network. In Equation (5) below, $y_{ij} \in \{0, 1\}$ 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.

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

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 Equation (6) define how the output values conform to the label relations as defined in external knowledge bases or learned from a dataset.

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

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 Figure 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, $Sim \in (0,1)^{|Y|*|Y|}$, 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, $Sub \in \{0,1\}^{|Y|*|Y|}$, indicates whether the label y_j is a child label of y_k. Both the Sim and Sub matrices can be pre-computed 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 multi-label learning, labels with high semantic similarity tend to be predicted together with similar values. The L_{sim} is a multiplication between two terms, Sim_{jk} and $|s_{ij}-s_{ik}|^2$. To minimise 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 a low similarity value. L_{sim} has a distinct form to the label manifold regulariser proposed in [25]. The latter considers minimising the differences of vector representations for low-rank approximation, while L_{sim} minimises 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 a paper using “LSTM”, but researchers from other areas may annotate the same paper 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 penalised. Such a case will result in a positive penalty, while the penalty will be 0 in all other cases.

As the pre-defined label relations may not be compatible with the semantics of the labels in the dataset. 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 regularisers 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 favoured. 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 optimise the joint loss function in Equation (4) with the L_2 regularisation using the Adam optimiser [52].

IV. Experiments

We carried out experiments on four large, social media datasets for academic research (Bibsonomy and CiteULike, three datasets) and question&answering (Zhihu, one dataset). 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 regularisation parameters and analysed the attention through visualisation. The code, implementation details and prediction results are available at https://github.com/acadTags/Automated-Social-Annotation.

A. Datasets

On Bibsonomy and CiteUlike, users can share and annotate publications. Metadata of the documents such as title and abstract are also available. The Bibsonomy dataset [53] version “2015-07-01”[1] was used, which contains 3,794,882 annotations, 868,015 resources, 283,858 distinct tags from 11,103 users, accumulated from 2003 to 2015. We used the cleaned dataset 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 datasets CiteULike-a and CiteULike-t released in [10]. We applied the same preprocessing steps as in [54] and removed the tags occurring less than 10 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 3 million questions and 1,999 labels. The dataset was preprocessed before its release: all the Chinese words were segmented and replaced with an unknown codebook due to privacy issues. We

[1] https://www.kde.cs.uni-kassel.de/bibsonomy/dumps

[2] https://www.acm.org/publications/class-2012

[3] https://biendata.com/competition/zhihu/

randomly sampled around 100,000 questions having both the title and content.

To extract the subsumption relations for all tags in each of the datasets (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 which can be directly used to find subsumption relations.

Statistics of the cleaned datasets are shown in Table I, including 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 for each dataset 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 datasets is large, from around 2K to 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.

TABLE I
STATISTICS OF THE FOUR DATASETS

Dataset	$\lvert X\rvert$	$\lvert Y\rvert$	$\lvert V\rvert$	*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

B. Experiment Settings

To calculate the similarity matrix Sim in Equations (6), we used the cosine similarity of the pre-trained skip-gram embeddings [45] on all labels in each dataset. 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 10-fold cross-validation[5]. We implemented the proposed JMAN model and its variants on Tensorflow [55]. Seven models were implemented for comparison:

1) SVM-ovr: an one-versus-rest multi-label Support Vector Machine 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 modelling 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 multi-label 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 1,000 iterations and tuned the number of topics T as 200 and k as 1 for all datasets based on the validation sets. The baseline was also used in [59].
3) Bi-GRU: the Bidirectional-RNN [49] with GRUs for multi-label 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 Hierarchical Attention Network 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 regularisers.
6) JMAN-s-tg: the proposed model without semantic-based loss regularisers 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 regularisers 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 are based on brightmart's TextRNN and Hierarchical Attention Network under the MIT license[7]. We trained all the models using 10-fold cross-validation and then tested on a separate, fixed 10% randomly held-out dataset. 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 dataset was set to 128, and the batch size for the Zhihu dataset was set to 1,024. 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; 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 initialised as a 100-dimension pre-trained 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 (11G GPU RAM), except for the dynamic update of Sim and Sub on Intel® Xeon® Processor E5-2630 v3 or v4 with 30G RAM; experiments on SVM-ovr and LDA were run on an Intel® Xeon® CPU E5-1620 v2 with 16G RAM.

We also re-implemented three representative algorithms for comparison, which transform either the feature space or label space of a base classifier for multi-label classification: (1) Classification Chain (CC) [36], [37], (2) Hierarchy Of

[4] https://concept.research.microsoft.com/Home

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

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

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

Multilabel classifiER (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 RBF kernel for the methods. Due to large numbers of documents and labels, the program took much longer than the SVM-ovr implementation (see Table IV) 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. We thus 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, F-measure, to assess the performance of the algorithms [20], [26], [66], [67]. For the metrics below, 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.

- Hamming loss (H) measures the number of misclassified labels, $\text{H}(f) = \frac{1}{|D_t|}\sum_{i\in D_t}\frac{1}{Q}|f(x_i)\Delta y_i|$, where Δ is the symmetric difference between two sets and Q is a normalisation 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.
- 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 $\text{A}(f) = \frac{1}{|D_t|}\sum_{i\in D_t}\frac{|f(x_i)\cap y_i|}{|f(x_i)\cup y_i|}$.
- Precision (P), defined as the fraction of the correctly predicted labels to all the predicted labels, $\text{P}(f) = \frac{1}{|D_t|}\sum_{i\in D_t}\frac{|f(x_i)\cap y_i|}{|f(x_i)|}$.
- Recall (R), defined as the fraction of the correctly predicted labels to all the actual labels, $\text{R}(f) = \frac{1}{|D_t|}\sum_{i\in D_t}\frac{|f(x_i)\cap y_i|}{|y_i|}$.
- F-measure (F_1), defined as the harmonic mean between precision and recall, $F_1(f) = \frac{2P(f)R(f)}{P(f)+R(f)}$.

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 regularisers.

1) Main Results: Table II shows the evaluation and comparison results using JMAN and others based on the four datasets[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 datasets. Most results of JMAN$_d$ were better than JMAN on the CiteULike-a/t datasets, 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 dataset.

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 dataset; and 5.9% (by 31.2%) and 4.5% (by 22.2%) over Bi-GRU and HAN for the CiteULike-t dataset. A similar performance gain was achieved using the Bibsonomy dataset, 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 datasets of 2.4% (by 13.4%) over Bi-GRU, and 0.8% over (by 3.4%) HAN. This overall improvement showed that the separate modelling 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 modelling the hierarchical pattern of a document with attention mechanisms, and validated the results in [17].

Effectiveness of the semantic-based loss regularisers was observed by comparing the results produced by JMAN and JMAN-s (without semantic-based loss regularisers). The regularisers 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 dataset, largely due to its distinct characteristics: compared to other datasets, Zhihu has much shorter texts (around 1/3 of the texts in other datasets), larger vocabularies (about 3-4 folds), 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 Hamming Loss was not always consistent with the other four metrics. 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 Hamming loss among HAN, JMAN and its downgraded variants, JMAN-s, JMAN-s-tg and JMAN-s-att, were all marginal. Compared to SVM and LDA, JMAN and its variants performed significantly better in terms of all metrics on all datasets, except a few cases where the LDA produced higher recall but much lower precision and F_1.

2) Results on Semantic-based Loss Regularisers: To test the effectiveness of the semantic-based loss regularisers L_{sim} and L_{sub}, we applied them (either separately or collectively) on Bi-GRU, HAN and JMAN-s, and reported the results with 10-fold cross-validation on the testing data.

From Table III, it can be seen that models with the semantic-based loss regularisers (either one or both) consistently performed better than the original models. 0.9% to 1.6% absolute gain of F_1 was observed for Bi-GRU, and 0.6% to 1.6% for HAN. For the JMAN-s model, the improvement with the

[8] We were not able to obtain the results of SVM-ovr on the Zhihu dataset as the training time for each fold in 10-fold 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 datasets.

TABLE II

COMPARISON RESULTS OF JMAN AND OTHERS ON THE FOUR SOCIAL ANNOTATION DATASETS 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.

TABLE III

COMPARISON RESULTS OF USING THE SEMANTIC-BASED LOSS REGULARISERS 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.

semantic-based loss regularisers is less obvious; there was only 0.1% to 0.5% absolute increase of F_1. It is hard to draw a clear conclusion on which of the L_{sim} and L_{sub} was 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 regularisers 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 multi-label classification. The work using a weight initialisation approach in [42] reported performance gain of less than 1% in F_1 in most experimental settings. The proposed approach is more feasible than the weight initialisation 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 regularisers 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 multi-label 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 models in 10-fold cross-validation. With the efficient and highly scalable implementation of 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 of 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 datasets. The training time increased when the semantic-based loss regularisers 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 dataset. 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 datasets 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 Figure 3. The total number of epochs for each model was determined by early stopping based on the validation set. On all four datasets, 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 better representation of the input documents with fewer epochs than HAN and Bi-GRU.

F. Analysis of Multi-Source 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 analysing how 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 datasets, without the use of semantic regularisers. 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 the 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 multi-source information for document representation.

G. Attention Visualisation

We can further understand how the hierarchical attention mechanisms work, especially the guided attention mechanism, by visualising the attention weights in Figure 4. Four attention weights in JMAN were illustrated for sample documents from Bibsonomy, CiteULike-a and CiteULike-t: (1) word-level attention for title, (2) word-level attention for each sentences 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 dataset were not interpretable as all words had been officially masked with an unknown codebook.

In Figure 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 colour, 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 diagrams.

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", "personalised" and "visualisation" in the Bibsonomy example; "implicit", "feedback", "ir", "models", and "searcher" in the CiteULike-a example; and "machine", "virtualising", "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 colour in the blocks).

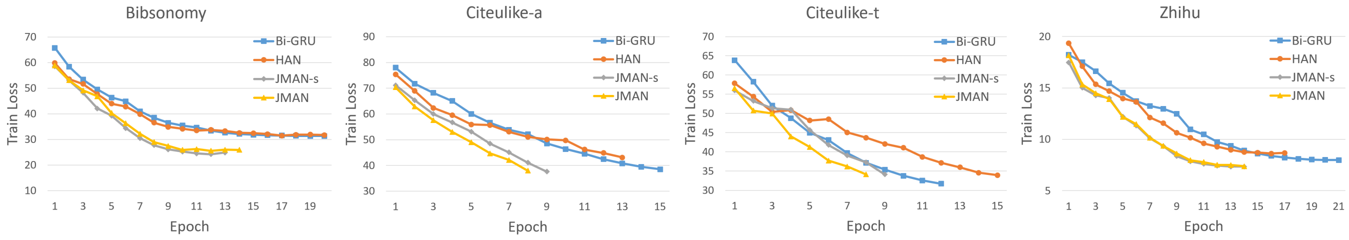
The title-guided sentence-level attention ("tg") assigned different weights and provided a distinct "view" from the original sentence-level attention ("ori"). In the Bibsonomy example, the "ori" weights highlighted mostly the second sentence (a general statement that identifies the gap in the literature), while the "tg" weights highlighted more the fourth (a statement of a tool that allows integrating personal knowledge into 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 that 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 which aligned

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.



(a) Bibsonomy (b) CiteUlike-a (c) CiteUlike-t (d) Zhihu

Fig. 3. Convergence plot: training loss with respect to the number of training epochs for the Bi-GRU, HAN, JMAN-s and JMAN models

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 DATASETS 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**

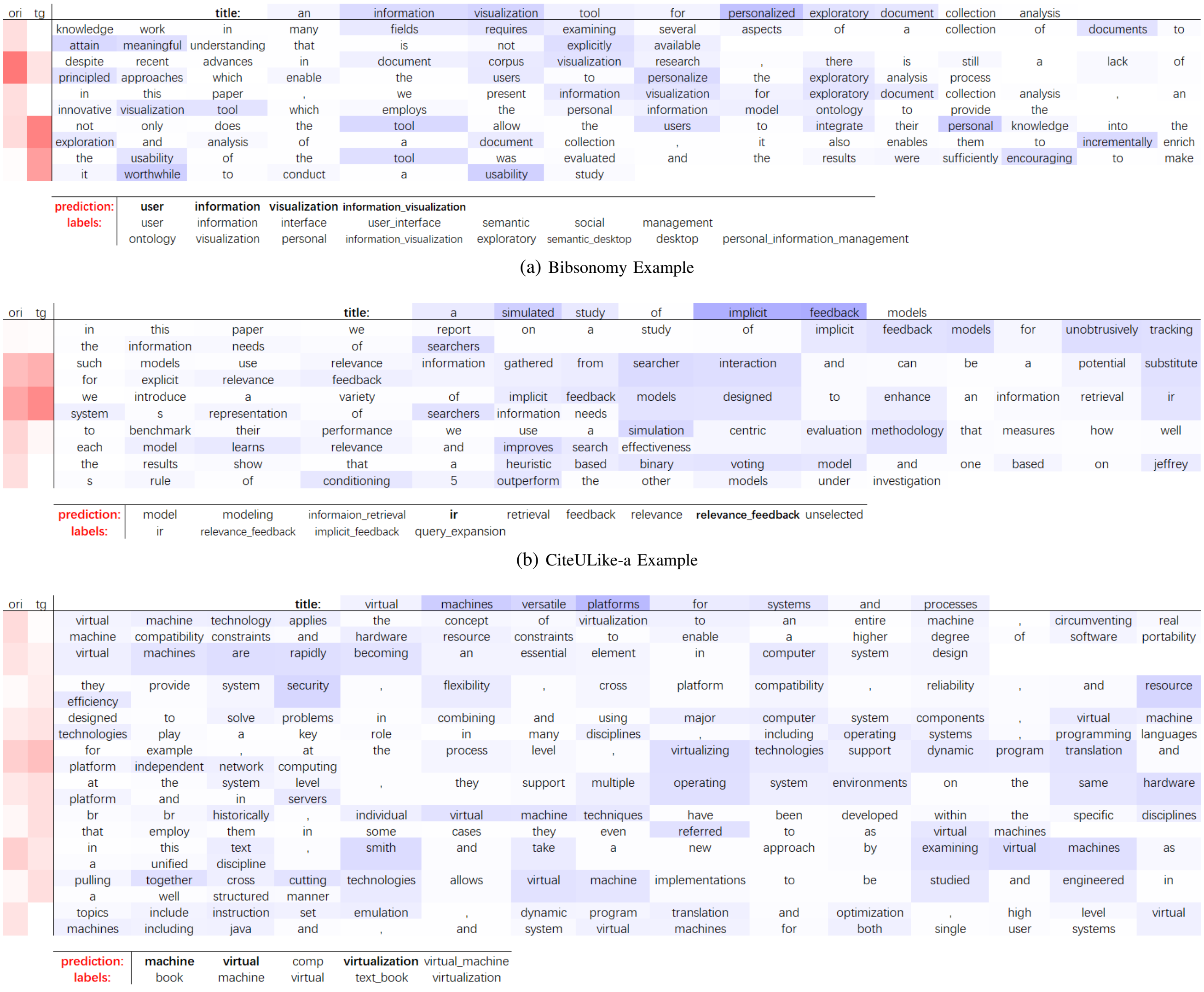
For H, the smaller the better; for A, P, R, and F_1, the larger, the better. The best results are in **bold**.

well to the title; while the "ori" weights also emphasised the fourth and fifth sentences which 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 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 J-MAN model suggested meaningful labels (more prediction results are available at https://github.com/acadTags/Automated-Social-Annotation). The predicted labels had a 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", "modelling" 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

Our work focused on two main issues in using a deep learning based method for automated social annotation as a



(a) Bibsonomy Example

(b) CiteULike-a Example

(c) CiteULike-t Example

Fig. 4. Attention visualisation 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 colour, the greater amount of attention was paid to the word or sentence in JMAN. The predicted labels and the "ground truth" labels are displayed below each diagrams.

multi-label classification problem: (i) how to design a deep network according to users' reading and annotation behaviour to achieve better classification performance; and (ii) how to leverage label correlation to further improve the performance of the classification. The proposed model, 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 statistical analysis of users' annotation behaviour 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 regularisers which 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 datasets 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 regularisers. Analysis of the multi-source 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 multi-label 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 regularisers 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 datasets, but with the cost of increased computation. More efficient method for dynamic update of Sim and Sub in the loss regularisers 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 multi-label 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 have been commonly used for text processing, it is also interesting to integrate the semantic-based loss regularisers and ensemble our model with other neural networks for social text annotation, including sequence-to-sequence networks [32], [70], Convolutional Neural Networks [71], attention-based network Transformer [72] and transfer-learning-based approaches, Bidirectional Encoder Representations from Transformers (BERT) [73].

References

[1] A. Zubiaga, V. Fresno, R. Martnez, and A. P. Garca-Plaza, "Harnessing folksonomies to produce a social classification of resources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 8, pp. 1801–1813, Aug 2013.

[2] D. R. Millen and J. Feinberg, "Using social tagging to improve social navigation," in *Workshop on the Social Navigation and Community based Adaptation Technologies*. 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: Springer New York, 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, H. Pinto, F. Belm, J. Almeida, M. Gonalves, D. Fernandes, and E. Moura, "Assessing the quality of textual features in social media," *Information Processing & Management*, vol. 49, no. 1, pp. 222 – 247, 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. 5:1–5:23, Jan. 2014.

[7] F. Jabeen and S. Khusro, "Quality-protected folksonomy maintenance approaches: a brief survey," *The Knowledge Engineering Review*, vol. 30, no. 5, p. 521544, 2015.

[8] F. M. Belém, J. M. Almeida, and M. A. Gonçalves, "A survey on tag recommendation methods," *Journal of the Association for Information Science and Technology*, vol. 68, no. 4, pp. 830–844, 2017.

[9] E. Zangerle, W. Gassler, and G. Specht, "Recommending#-tags in twitter," in *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, 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 *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '13. 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 *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 *Proceedings of the 12th ACM Conference on Recommender Systems*, ser. RecSys '18. New York, NY, USA: ACM, 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 *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, ser. IJCAI'17. AAAI Press, 2017, pp. 3420–3426.

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

[15] Y. Li, T. Liu, J. Jiang, and L. Zhang, "Hashtag recommendation with topical attention-based lstm," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 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 *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 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 *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[18] M. Lipczak and E. Milios, "The impact of resource title on tags in collaborative tagging systems," in *Proceedings of the 21st ACM conference on Hypertext and hypermedia*. New York, NY, USA: ACM, 2010, pp. 179–188.

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

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

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

[22] W. G. Stock, "Concepts and semantic relations in information science," *Journal of the American Society for Information Science and Technology*, 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*, ser. Knowledge and Information. 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 *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 521–526.

[25] Y. Zhu, J. T. Kwok, and Z. Zhou, "Multi-label learning with global and local label correlation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1081–1094, June 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, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 22–30.

[27] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, 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," *arXiv preprint arXiv:1409.0473*, 2014.

[29] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[30] D. Liang, F. Zhang, W. Zhang, Q. Zhang, J. Fu, M. Peng, T. Gui, and X. Huang, "Adaptive multi-attention network incorporating answer information for duplicate question detection," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR19. New York, NY, USA: Association for Computing Machinery, 2019, p. 95104. [Online]. Available: https://doi.org/10.1145/3331184.3331228

[31] A. Kumar, D. Kawahara, and S. Kurohashi, "Knowledge-enriched two-layered attention network for sentiment analysis," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 1624–1633.

[33] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[34] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” *International Journal of Data Warehousing and Mining (IJDWM)*, 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, Technical Report 2006-10, April 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, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 254–269.

[37] ——, “Classifier chains for multi-label classification,” *Machine learning*, vol. 85, no. 3, pp. 333–359, 2011.

[38] B. Chen, W. Li, Y. Zhang, and J. Hu, “Enhancing multi-label classification based on local label constraints and classifier chains,” in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 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 2008 Workshop on Mining Multidimensional Data (MMD08)*, vol. 21. sn, 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 *Proceedings of the 15th ACM International Conference on Multimedia*, ser. MM ’07. New York, NY, USA: ACM, 2007, pp. 17–26.

[41] M.-L. Zhang and K. Zhang, “Multi-label learning by exploiting label dependency,” in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’10. New York, NY, USA: ACM, 2010, pp. 999–1008.

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

[43] J. Wehrmann, R. C. Barros, S. N. d. Dôres, and R. Cerri, “Hierarchical multi-label classification with chained neural networks,” in *Proceedings of the Symposium on Applied Computing*, ser. SAC 17. New York, NY, USA: Association for Computing Machinery, 2017, p. 790795. [Online]. Available: https://doi.org/10.1145/3019612.3019664

[44] H. Dong, W. Wang, K. Huang, and F. Coenen, “Joint multi-label attention networks for social text annotation,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 1348–1354.

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

[46] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

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

[48] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[49] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, 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,” *arXiv preprint arXiv:1808.08575*, 2019, aAAI 19.

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

[52] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[53] D. Benz, A. Hotho, R. Jschke, B. Krause, F. Mitzlaff, C. Schmitz, and G. Stumme, “The social bookmark and publication management system BibSonomy,” *The VLDB Journal*, 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 *iConference 2017 Proceedings*. iSchools, 2017, pp. 661–666.

[55] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: A system for large-scale machine learning,” in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI’16. Berkeley, CA, USA: USENIX Association, 2016, pp. 265–283.

[56] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[57] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[58] A. K. McCallum, “Mallet: A machine learning for language toolkit,” 2002, 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. 4:1–4:31, Feb. 2011.

[60] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[61] F. Tai and H.-T. Lin, “Multilabel classification with principal label space transformation,” *Neural Computation*, vol. 24, no. 9, pp. 2508–2542, 2012.

[62] P. Szymański and T. Kajdanowicz, “A scikit-based Python environment for performing multi-label classification,” *ArXiv e-prints*, Feb. 2017.

[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, p. 667671, Jan. 2016.

[64] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, p. 1018, Nov. 2009. [Online]. Available: https://doi.org/10.1145/1656274.1656278

[65] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, “Mulan: A java library for multi-label learning,” *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

[66] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Mining multi-label data,” in *Data Mining and Knowl. Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2010, pp. 667–685.

[67] M. S. Sorower, “A literature survey on algorithms for multi-label learning,” Department of Computer Science, Oregon State University, Tech. Rep., 2010.

[68] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, “Deep learning for extreme multi-label text classification,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR 17. New York, NY, USA: Association for Computing Machinery, 2017, p. 115124. [Online]. Available: https://doi.org/10.1145/3077136.3080834

[69] Y. Zhu, K. M. Ting, and Z. Zhou, “Multi-label learning with emerging new labels,” *IEEE Transactions on Knowledge and Data Engineering*, 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 *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS’14. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112.

[71] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 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,” *arXiv preprint arXiv:1810.04805*, 2018.