

Received April 2, 2019, accepted May 21, 2019, date of publication June 21, 2019, date of current version July 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924250

Resource Mention Extraction for MOOC Discussion Forums

YA-HUI AN^{1,2}, LIANGMING PAN^{2,3}, MIN-YEN KAN^{2,4}, QIANG DONG¹, AND YAN FU¹

¹Complex Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

²Web IR/NLP Group, National University of Singapore, Singapore

³NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore

⁴NUS Institute for Application of Learning Science and Educational Technology, National University of Singapore, Singapore

Corresponding author: Liangming Pan (peterpan10211020@gmail.com)

This work was supported in part by the China Scholarship Council (CSC), in part by a Learning Innovation–Technology Grant from NUS (LIF-T, NUS), and in part by the UESTC Fundamental Research Funds for the Central Universities under Grant ZYGX2016J196.

ABSTRACT In discussions hosted on discussion forums for massive online open courses (MOOCs), references to online learning resources are often of central importance. They contextualize the discussion, anchoring the discussion participants' presentation of the issues and their understanding. However, they are usually mentioned in free text, without appropriate hyperlinking to their associated resource. Automated learning resource mention hyperlinking and categorization will facilitate discussion and searching within the MOOC forums and also benefit the contextualization of such resources across disparate views. We propose the novel problem of learning resource mention identification in MOOC forums, i.e., to identify resource mentions in discussions and classify them into pre-defined resource types. As this is a novel task with no publicly available data, we first contribute a large-scale labeled dataset—dubbed the forum resource mention (FoRM) dataset—to facilitate our current research and future research on this task. The FoRM contains over 10 000 real-world forum threads in collaboration with Coursera, with more than 23 000 manually labeled resource mentions. We then formulate this task as a sequence tagging problem and investigate solution architectures to address the problem. Importantly, we identify two major challenges that hinder the applications of sequence tagging models to the task: 1) the diversity of resource mention expression and (2) long-range contextual dependencies. We address these challenges by incorporating character-level and thread context information into an LSTM–CRF model. First, we incorporate a *character encoder* to address the out-of-vocabulary problem caused by the diversity of mention expressions. Second, to address the context dependency challenge, we encode thread contexts using an RNN-based context encoder and apply the attention mechanism to selectively leverage useful context information during sequence tagging. The experiments on FoRM show that the proposed method improves the baseline deep sequence tagging models notably, significantly bettering performance on instances that exemplify two challenges.

INDEX TERMS Artificial intelligence, deep learning, hyperlinking, learning resources, MOOC discussion forums, name entity recognition.

I. INTRODUCTION

With the efforts towards building an interactive online learning environment, discussion forum has become an indispensable part in the current generation of MOOCs. In discussion forums, students or instructors could post problems or instructions directly by starting a thread or posting in an existing thread. During discussions, it is natural for students or instructors to refer to a learning resource, such as a certain quiz, this week's lecture video, or a particular page of slides. These references to resources are called

resource mentions, which compose the most informative parts among a long thread of posts and replies. The right side of Figure 1 shows a real-world forum thread from Coursera,¹ in which resource mentions are highlighted in bold, with same color reference to the same resource on the left. From this example, we find that if we identify and highlight resource mentions in forum threads, it will greatly facilitate learners to efficiently seek for useful information in discussion forums, and also establish a strong linkage between a course and its forum.

The associate editor coordinating the review of this manuscript and approving it for publication was Mervat Adib Bamiah.

¹Coursera (<https://www.coursera.org/>) is one of the largest MOOC platforms in the world.

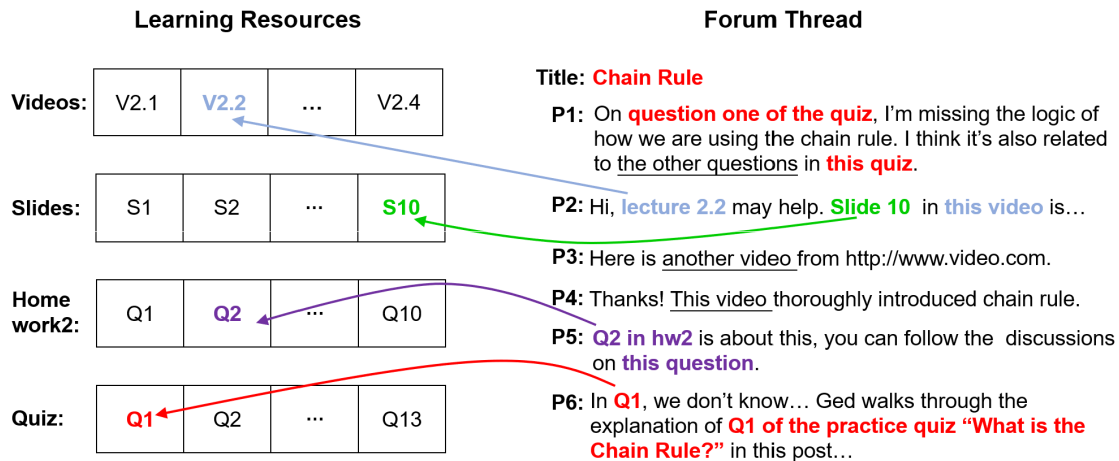


FIGURE 1. An example of resource mention identification. The left shows the learning resources of a course, the right is a forum thread. P1–P6 are six posts in the thread. Resource mentions are marked in bold, with same color refer to the same learning resource. The underlined text are not valid mentions.

We propose and study the problem of resource mention identification in MOOC forums. Specifically, given a thread from MOOC discussion forum, our goal is to automatically identify all resource mentions present in this thread, and categorize each of them to its corresponding resource type. For resource types, we adopt the categorization proposed in [1], where learning resources are categorized into videos, slides, assessments, exams, transcripts, readings, and additional resources.

Our task can be formulated as a sequence tagging problem. Given a forum thread as a word sequence $\mathcal{T} = \{w_1, \dots, w_n\}$, we apply a sequence tagging model to assign a tag t_i to each word w_i , where t_i represents either the Beginning, Inside or Outside (BIO) of a certain type of resource mention (e.g., the tag “Videos_B” for w_i indicates that w_i is the first word of a resource mention with type “Videos”). To train a sequence tagger, we need a large amount of labeled resource mentions in MOOC forums. However, to the best of our knowledge, no public labeled dataset is available since we are the first to investigate this task. To closely investigate this problem and also facilitate the following research on this task, we manually construct a large-scale dataset, namely *Forum Resource Mention (FoRM)* dataset, in which each example is a forum post with labeled resource mentions. We first crawl real-world forum posts from Coursera, and then perform human annotations to identify resource mentions and their resource types. During the annotation, we find that resource mentions are hard to be identified even for human annotators. Compared with some well-studied sequence tagging problems such as POS tagging [2], [3], and Named Entity Recognition (NER) [4]–[6], resource mention identification in MOOC forums poses several unique challenges.

The most challenging issue is the *context dependency*. Compared with other sequence tagging tasks such as POS tagging and NER, in which lexical patterns or local contexts

serves as strong clues for identification, resource mention identification usually requires more, an understanding of the whole context in the thread. For example, in Figure 1, both the post P2 and P4 contain the mention “this video”. The mention in P2 is a valid resource mention, as it refers to a specific resource (Video 2.2) within the course. However, in P4, “this video” actually refers to an external resource, thus is not a valid resource mention. As another example, the mention “the other questions” in P1 is also an invalid resource mention, because it makes a general reference to the quiz questions. These examples reflect some of the typical scenarios in MOOC forums, in which the identification deals with long-range context dependencies, and require an in-depth understanding of the thread context. Another challenge comes from the *variety of expressions*. Since the discussion forum is a colloquial communication environment, it is often filled with typos, abbreviations, compound words, new words, and other words that are not included in the dictionary, i.e., Out-of-Vocabulary (OOV) words. As shown in the post P6 of Figure 1, the word “Q1” is a valid resource mention but also an OOV word. Identifying “Q1” requires not only the context, but also an understanding of character-level semantics (e.g., “Q” stands for “Question”), which further increases the difficulty of this task.

We propose to add a *character encoder* and a *context encoder* to LSTM-CRF [7], a state-of-the-art model for sequence tagging, to address the above challenges. First, to better capture the semantics of OOV words caused by the variety of expressions, we incorporate *Character Encoder* to the original LSTM-CRF model, which encodes character-level information via LSTMs. This helps us better capture the correlation between abbreviations (e.g., “Q1” and “Q2”) and the prefix or postfix information (e.g., “dishdetail.html”). As for the context dependency problem, we need an effective way to

leverage thread contexts, since LSTM-CRF usually has a hard time dealing with long-range context dependencies. To resolve this problem, we propose to add an attentive-based *Context Encoder*, which encodes each context sentence with LSTMs, and selectively attends to useful contexts using the attention mechanism [8] during the decoding process of sequence tagging.

Based on the constructed FoRM dataset, we subsequently evaluate the performance of different sequence tagging models, and conduct further analysis on how the proposed method solves the major challenges in resource mention identification. We evaluate the models on two versions of FoRM datasets: a medium-scale version (FoRM-M), which contains around 9,000 annotated resource mentions, and has high agreement between human annotators; a large-scale version (FoRM-L), which contains more than 25,000 annotated resource mentions, but with relatively lower annotation agreement. The resource mentions in FoRM-M are easier to identify from surface forms (e.g., “Week 2 Quiz 1”); while mentions in FoRM-L are more ambiguous and dependent on the context. The experimental results show that our incremental LSTM-CRF model outperforms the baselines on both FoRM-M and FoRM-L, with noticeable effects on alleviating the above two challenges via incorporating character encoder and context encoder.

The main contributions of this paper can be summarized as follows:

- The first attempt, to the best of our knowledge, to systematically investigate the problem of resource mention extraction in MOOC forums.
- We propose an incremental model of LSTM-CRF that incorporates character encoder and context encoder, to solve the expression variety and context dependency problems. The model achieves an average improvement F_1 score of 3.16% (*c.f.* Section V-C) over LSTM-CRF.
- We construct a novel large-scale dataset, FoRM, from forums in Coursera, to evaluate our proposed method.

The rest of the paper is organized as follows: In Section II, we will first discuss some related works. In Section III, we will introduce our dataset, FoRM. In Section IV, we formalize the problem, and illustrate our proposed model. We will provide the experimental results and analysis of the proposed method in Section V. Finally, Section VI will summarize the paper and discuss future research directions.

II. RELATED WORKS

The task of resource mention identification can be regarded as a twin problem of *named entity recognition* and *anaphora resolution*, and we will elaborate both in the following.

A. NAMED ENTITY RECOGNITION

Despite some works have investigated extracting key concepts in MOOCs [9]–[11], our work is different because the objective of our task is to jointly identify the position and type of resource mentions from plain texts. Therefore, it is more similar to *Named Entity Recognition (NER)*, which

seeks to locate named entities in texts and classify them into pre-defined categories. Neural sequence tagging models have become the dominate methodology for NER since the emerge and flourish of deep learning. Hammerton [12] attempted a single-direction LSTM network to perform sequence tagging, and Collobert *et al.* [13] employed a deep feed-forward neural network for NER, and achieved near state-of-the-art results. However, these NER models only utilize the input sequence when predicting the tag for a certain time-step, but ignoring the interaction between adjacent predictions. To address this problem, Huang *et al.* [7] proposed to add a CRF layer on top of a vanilla LSTM sequence tagger. This LSTM-CRF model has achieved the state-of-the-art results for NER when using the bidirectional LSTM (BLSTM).

One problem of LSTM-CRF is that it only captures the word-level semantics. This causes a problem when intra-word morphological and character-level information are also very important for recognizing named entities. Recently, dos Santos *et al.* [14] augmented the work of Collobert *et al.* [13] with character-level CNNs. Chiu and Nichols [6] incorporated the character-level CNN to BLSTM and achieved a better performance in NER. In our task, resource mention identification, the widely existing OOV words, such as “Q1”, “Q2”, “hw2” in Figure 1, greatly increase the difficulty of capturing word-semantics. Therefore, we also incorporate the character-level semantics by proposing a character encoder via LSTM.

However, incorporating character embeddings is insufficient for resource mention identification, as this task is different from NER with respect to the reliance on long-range contexts. Compared to NER, which typically requires limited context information, resource mention identification is a more context-dependent task. A common scenario is to judge whether a pronoun phrase, such as “this video”, refers to a resource mention or not. For example, to understand that “this video” in *P4* of Figure 1 actually does not refer to any resource within the course requires the contexts from at least *P2*, *P3* and *P4*. In this case, this problem is more related to *Anaphora Resolution*, which is another challenging problem in natural language processing (NLP).

B. ANAPHORA RESOLUTION

In computational linguistics, anaphora is typically defined as references to items mentioned earlier in the discourse or “pointing back” reference as described by [15]. *Anaphora Resolution (AR)* is then defined as resolving anaphora to its corresponding entities in a discourse. Resolving repeated references to an entity is similar to differentiating whether a mention is a valid resource mention within the course.

Most of the early AR algorithms were dependent on a set of hand-crafted rules. These early methods were a combination of salience, syntactic, semantic and discourse constraints to do the antecedent selection. In 1978, Hobbs *et al.* [16] firstly combined the rule-based, left to right breadth-first traversal of the syntactic parse tree of a sentence with

selectional constraints to search for a single antecedent. Lappin and Leass [17] discussed a discourse model to solve the pronominal AR. Then the centering theory [18], [19] was proposed as a novel algorithm used to explain phenomenon like anaphora using discourse structure. During the late nineties, the research in AR started to shift towards statistical and machine learning algorithms [20]–[23], which combines the rules or constraints of early works as features. Recently, the relevant research shifted to deep learning models for *Coreference Resolution (CR)*, which includes AR as a sub-task. Wiseman et al. [24] designed mention ranking model by learning different feature representations for anaphoricity detection and antecedent ranking by pre-training on these two individual subtasks [25]. Later, they proved that coreference task can benefit from modeling global features about entity clusters [26]. Meanwhile, Clark and Manning [27] proposed another cluster ranking model to derive global information. Up to now, the state-of-the-art model was proposed by [28], an end-to-end CR system that jointly modeled mention detection and CR.

Most of the AR works take as input the candidate key phrases extracted from the discourse, and then resolve these phrases to entities by casting the problem as either a classification or ranking task. However, our task is defined as a sequence tagging problem, which requires anaphora resolution implicitly when predicting the type of an ambiguous resource mention. In our model, we incorporate a context encoder to implement a mechanism of sequence-to-sequence tagging with attention to help the model to learn anaphora resolution within the contexts implicitly during training.

III. THE FORM DATASET

In this section, we introduce the construction of our experimental dataset, i.e., Forum Resource Mention (FoRM) dataset. To the best of our knowledge, there is no publicly available dataset that contains labeled resource mentions in MOOC forums. We construct our dataset via a three-stage process: (1) data collection, (2) data annotation, and (3) dataset construction.

A. DATA COLLECTION

Our data comes from Coursera, one of the largest MOOC platforms in the world. Coursera was founded in 2012 and up to August 2018, it has offered more than 2,700 courses and attracted about 33 million registered learners. Each course has a discussion forum for students to post/reply questions and to communicate with each other. Each forum contains all the threads started by students or instructors, which consists of one *thread title* (main idea of a problem), one or more *thread posts* (details about the problem) and replies (see Figure 1 as an example).

As the distribution of resource mentions may vary for courses in different domains, we consider a wide variety of course domains when collecting the data. Specifically, we collect the forum threads from 142 completed courses

in 10 different domains² Note that in Coursera, each course may have multiple sessions; each session is an independent learning iteration of the course, with a fixed start date and end date (e.g., “Machine Learning” (from 2018-08-20 to 2018-12-20)). Different sessions of a course may have different organization and notation systems for the same set of learning resources, which involves ambiguity if we consider them all. Therefore, we only select the latest completed session for each course, resulting a total number of 102,661 posts.³ Finally, we exclude the posts that belong to the “General Discussion” and “Meet & Greet” forums, which are unlikely to contain resource mentions, and only select the posts in “Week Forums”, as they are designed for “Discuss and ask questions about Week X”. This gives us a data collection of 84,945 posts from 11,679 different forum threads.

B. DATA ANNOTATION

Based on the above collected data, we then manually annotate resource mentions for each thread. We employ 16 graduate students from technical backgrounds to annotate the data. As mentioned before, our data collection consists of 11,679 forum threads from 142 courses; each thread is a time-ordered list of posts, including thread title and a series of thread/reply posts. We split the 11,679 threads into 8 portions, and assign each portion to 2 annotators. For simplicity of annotation, for each thread, we concatenate all contents of its posts, to get a single document of sentences for annotation. For each thread document, the task of the annotator is to identify all the resource mentions in the document, and tag each of them with one of the pre-defined 7 resource types defined in Section I (refer to Table 9 for details). We define a resource mention as any one or more consecutive words in a sentence that represents an unambiguous learning resource in the course. We use the *brat rapid annotation tool*,⁴ an online environment for collaborative text annotation, which is widely used in entity, relation and event annotations [29]–[31], as our annotation platform.

To help annotators better understand the above process and relevant concepts, we conduct an one-hour training for annotators; the complete training process is documented in VII. Then, we start the real annotation; the whole annotation process takes around one month. In the end, each thread is doubly annotated, and we denote the two copies of the annotated data as Group 1 and Group 2, respectively. Table 1 summarizes the number of annotated resource mentions for each resource type. Note that we integrate the 4 resource types representing teaching materials, i.e., ‘*Readings*’, ‘*Slides*’, ‘*Transcripts*’, and ‘*Additional Resources*’, into one single resource type

²The selected domains are: ‘*Arts and Humanities*’, ‘*Business*’, ‘*Computer Science*’, ‘*Data Science*’, ‘*Language Learning*’, ‘*Life Sciences*’, ‘*Math and Logic*’, ‘*Personal Development*’, ‘*Physical Science and Engineering*’ and ‘*Social Sciences*’.

³Our data was collected at January 31, 2017, and we are in partnership with Coursera at the time of the dataset collection.

⁴<http://brat.nlplab.org/>

TABLE 1. Annotation result from group 1 and group 2 on assessments, exams, videos, and coursewares. Coursewares = readings, slides, transcripts, and additional resources. P_{pos} is the positive specific agreement.

Resource Type	Group 1	Group 2	Intersection	Union	P_{pos}
Assessments	8,047	8,520	5,451	11,116	0.658
Exams	1,891	3,624	1,146	4,369	0.416
Videos	1,852	3,037	1,236	3,653	0.506
Coursewares	3,281	4,286	1,557	6,010	0.412
Total	15,071	19,467	9,390	25,148	0.544

‘Coursewares’, to form a dataset with more balanced training examples for each class.

To evaluate the inter-annotator agreement between two groups, we use the *Positive Specific Agreement* [32], a widely-used measure of agreement when the positive cases are rare compared with the negative cases. In summary, there are 4 possible cases when comparing the result of the annotated mentions between Group 1 and Group 2, summarized in Table 2. For example, *AG* denotes the number of cases that both groups agree are resource mentions and also have an agreement about its type. Based on all the conditions listed in Table 2, the calculation of the positive specific agreement (denoted as P_{pos}) between two groups’ annotations is given in Equation 1. The agreement scores for different resource types are shown in the column P_{pos} of Table 1.

$$P_{pos} = \frac{2 \times AG}{2 \times AG + (TD + G1) + (TD + G2)} \quad (1)$$

To give an explanation for P_{pos} values to better understand whether our annotation achieves an acceptable agreement, we analyze the value of P_{pos} by referring to Kappa coefficient, because [32] proves that κ approaches the positive specific agreement when the number of negative cases grows large, which is exactly our case. We find that the P_{pos} value for *Exams*, *Videos* and *Coursewares* are in the range of *moderate agreement*,⁵ and for *Assessments*, the value shows a *substantial agreement* [33]. The possible reasons that the agreement for *Assessments* is higher than the other types are: 1) samples for four types of resource are unbalanced; the ratio of *Assessments* is higher than others, thus has a lower annotation bias; 2) *Assessments* is easier for annotators to distinguish compared to other types of resource. In summary, the overall annotation result achieves a moderate agreement between two group of annotators.

⁵The values for κ : [-1, 0): less than chance agreement; 0: random; [0.01, 0.20]: slight agreement; [0.21, 0.40]: fair agreement; [0.41, 0.60]: moderate agreement; [0.61, 0.80]: substantial agreement; and [0.81, 0.99] almost perfect agreement; 1: perfect agreement.

TABLE 2. Four possible cases when comparing the annotation results of group 1 and 2.

Type	Description	Notation
Agree	text span overlaps, and annotated type same	<i>AG</i>
Type-Disagrees	text span overlaps, but annotated type different	<i>TD</i>
G1-Only	the annotation exists only in Group 1	<i>G1</i>
G2-Only	the annotation exists only in Group 2	<i>G2</i>

C. DATASET CONSTRUCTION

Based on the annotation results, we construct two versions of datasets with different characteristics. First, to provide a dataset with high-quality resource mentions, we only use the ‘‘Agree’’ cases in Table 2 as the ground-truth resource mentions to construct the *FoRM-M* dataset. For the ‘‘Agree’’ case, we joint the text spans of annotated mentions from Group 1 and Group 2 as the ground truth. For example, if the annotated mentions are ‘‘the video 1’’ (Group 1) and ‘‘video 1 of week 2’’ (Group 2), we create a ground-truth of ‘‘the video 1 of week 2’’ by unioning the texts. In this way, we tend to obtain more specific mentions (e.g., ‘‘the video 1 of week 2’’) rather than general ones (e.g., ‘‘video 1’’). The number of ‘‘Agree’’ resource mentions is 9,390 as shown in the column ‘‘Intersection’’ in Table 1. We also construct a larger but relatively more noisy dataset, namely *FoRM-L*, by using all the cases as ground-truths, which represents a ‘‘union’’ of the annotations from the two groups. The statistics are shown in the ‘‘Union’’ column of Table 1.

As mentioned in Section I, we formulate the task of resource mention identification in MOOC forums as a sequence tagging problem. Therefore, we associate each word in the dataset with a corresponding tag, based on the ground-truth we obtained in the previous step. A word is associated with the *Beginning (B)*/*Inside (I)* tag if it is the beginning/inside of a resource mention with type *T*, denoted as *T_B/I*. Otherwise, the *Outside (O)* tag is assigned to the word.

The statistics of the constructed datasets are shown in Table 3, where *# Examples* is the total number of sentences containing at least one resource mention, *# Tokens* is the total number of words in the dataset. *# Average Length* denotes the average number of words in a sentence. The total number of B-tags (e.g., *Coursewares_B*) and I-tags (e.g., *Exams_I*) for different resource types, as well as the number of O-tags, are also listed in the table.

IV. METHODS

We present our neural model for identifying and typing resource mentions in MOOC forums. We first formulate the problem and then present the general architecture of the proposed model. Followed by that, we introduce the major components of our model in detail in the remaining sections.

A. PROBLEM FORMULATION

We first introduce some basic concepts, and formally define the task of resource mention identification in MOOC forums.

TABLE 3. Statistics of the forum resource mention dataset (FoRM).

Dataset	FoRM-M	FoRM-L
# Examples	8,390	19,952
# Tokens	150,597	395,958
# Average Length	17.95	19.84
Coursewares_B	1,398	5,183
Coursewares_I	1,273	1,273
Exams_B	1,094	3,989
Exams_I	1,901	4,166
# Tags		
Assessments_B	5,202	10,432
Assessments_I	7,359	12,885
Videos_B	1,223	3,403
Videos_I	2,121	4,670
O	129,026	346,693

Definition 1 (Post): A post P is the smallest unit of communication in MOOC forums that contains user-posted contents. Each post is composed of the text contents written by the user, and some associated meta-data such as user ID, posting time etc. In our task, we focus on extracting resource mentions from text contents; thus we simply formulate a post as a sequence of sentences, i.e., $P = \{s_1, \dots, s_{|P|}\}$, where each sentence is a word sequence $s = \{w_1, \dots, w_{|s|}\}$.

Definition 2 (Thread): Typically, a thread T in MOOC forums is composed of a thread title t , an initiating post I , and a set of reply posts R [34]. Initiating post is the first post in the thread and initiates discussions. All other posts in a thread are the reply posts that participate in the discussion started by the initiating post. For simplicity, we do not differentiate between the initiating post and the reply posts, and we also treat the thread title as a special post P_0 . In this case, a thread T can be represented as an ordered list of posts, i.e., $T = \{P_0, P_1, \dots, P_{|T|}\}$. A thread T with n posts can be unfolded as a long document of N sentences $T = \{s_1, \dots, s_N : s_i \in P_{I(i)}\}$, where $I(i)$ is the index of the post that sentence s_i belongs to.

Definition 3 (Resource Mention): A course C in MOOCs is defined as a set of resources, where each *resource* represents a specific learning resource/material in C (e.g., “Video 2.1”), and is associated with a *resource type* (e.g., “Video”). In a thread that belongs to course C , we define any semantically complete single/multi-word phrase that represents a resource of C as a *resource mention* (e.g., “the first video of chapter 2”).

Definition 4 (Resource Mention Identification): The task of resource mention identification in MOOC threads is defined as follows: Given a thread T in the discussion forums of course C , the objective is to identify all resource mentions appearing in T , and for each identified resource mention, to categorize it into one of the pre-defined resource types.

This task involves identifying both the location and the type of a resource mention, so it can be formulated as a sequence tagging problem. Specifically, given a thread T , our task is to assign a tag t to each word $w \in T$. The tag t can be either \mathcal{T}_B (the beginning of a resource mention of type \mathcal{T}), \mathcal{T}_I (inside a resource mention of type \mathcal{T}), or \mathcal{O} (outside any resource mention). Under this problem formulation, state-of-the-art sequence tagging models, such as LSTM-CRF, can be

applied to our task. However, they suffer from the two major challenges discussed in Section I. Therefore, we propose an incremental neural model based on LSTM-CRF to address the challenges. In the following sections, we will introduce our model in detail, and more specifically, discuss how we address the above two challenges by incorporating the context encoder and the character encoder.

B. GENERAL ARCHITECTURE

A thread T with n posts is unfolded as a sequence of N sentences $T = \{s_1, \dots, s_N\}$, where s_i is the i -th sentence in the entire thread T . Given T as input, our model performs sentence-level sequence tagging for each sentence in the thread T . Specifically, to decode the sentence $s_i \in T$, we consider all or part of the previous sentences of s_i as its contexts, denoted as C_i . Then, our goal is to learn a model that assigns each word in s_i with a tag; we denote the output tag sequence as t_i . Therefore, our model essentially approximates the following conditional probability.

$$p(Y | T ; \Theta) = \prod_{i=1}^N p(t_i | s_i, C_i ; \Theta) \quad (2)$$

where Θ is the model parameters, and $p(t_i | s_i, C_i ; \Theta)$ denotes the conditional probability of the output tag sequence t_i given the sentence s_i and its context C_i .

To model the conditional probability $p(t_i | s_i, C_i ; \Theta)$, our model includes three components: (1) the context encoder, (2) the character encoder, and (3) the attentive LSTM-CRF tagger. Figure 2 shows the framework of our proposed neural model. First, to encode the context information C_i , we incorporate the context encoder: a set of recurrent neural network (RNN) to encode each context sentence (Section IV-C). Our context encoder is generic to any textual contexts that can be additionally provided (e.g., from external resources), while in our model, we use the previous sentences of the thread as the context, to address the context dependency problem proposed in Section I. To alleviate the OOV challenge in our task, we employ the character encoder to build word embeddings using BLSTMs [35] over the characters (Section IV-D). The character-level word embeddings are then combined with the word-level embeddings as inputs to our model. Finally, we use the BLSTM-CRF [7] to generate the output tag sequence. Different from the original model in [7], we add an attention module [8] that acts over the encoded textual contexts (attentive LSTM-CRF tagger), to make use of important context information during sequence tagging (Section IV-E).

C. CONTEXT ENCODER

As discussed in Section I, context information is crucial for identifying resource mentions. For the i -th sentence s_i in the input thread T , a straightforward way is to use the **thread context**, which is to encode all the previous sentences of s_i in T as its context, i.e., $C_i = \{s_1, \dots, s_{i-1}\}$. The thread context contains complete information for inferring resource mentions in s_i , but also makes it harder for the model to

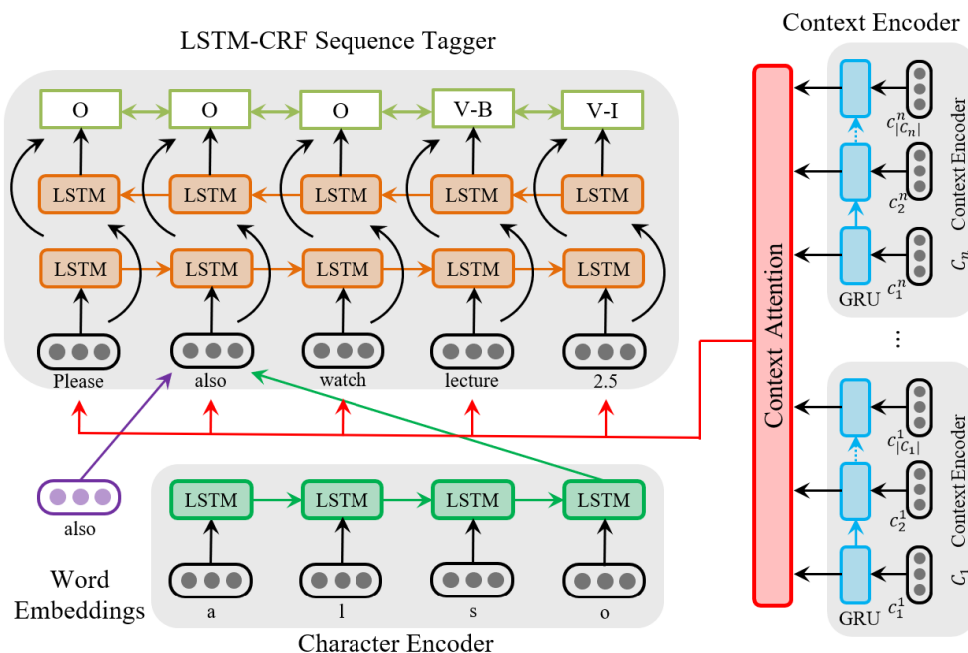


FIGURE 2. The general architecture of the proposed model. Our model consists of three parts: Context encoder, character encoder, and LSTM-CRF, which are shaded in gray.

learn the inherent patterns from these long and noisy contexts. We address this problem by introducing the attention mechanism into the decoding process, which will be further illustrated in Section IV-E.

We denote the thread context C as a sequence of m sentences $C = \{c_1, \dots, c_m \mid c_i = (c_1^i, \dots, c_{|c_i|}^i)\}$, where c_j^i represents the one-hot encoding of the j -th token in the i -th context sentence c_i , and $|c_i|$ is the length of the sentence c_i (cf Figure 2, each gray block represents the encoding of a sentence c_i in context C). We employ the method in [36] to use a set of m Gated Recurrent Neural Networks (GRU) [37] to encode each of the context sentence separately:

$$h_j^{c_i} = \text{GRU}_i(\mathbf{E}_c c_j^i, h_{j-1}^{c_i}) \quad (3)$$

where GRU_i denotes the GRU used to encode the i -th context sentence c_i , \mathbf{E}_c is the input word embedding matrix, and $h_j^{c_i} \in \mathbb{R}^{H_c}$ is the GRU hidden state in the j -th time step, which is determined by the input token c_j^i and the previous hidden state $h_{j-1}^{c_i}$. We concatenate the last hidden state $h_{|c_i|}^{c_i}$ for each encoded context sentence c_i to obtain our **context vector** h_c as follows:

$$h_c = [h_{|c_1|}^{c_1}; \dots; h_{|c_i|}^{c_i}; \dots; h_{|c_m|}^{c_m}] \quad (4)$$

The context vector will further be used by the attention mechanism in Section IV-E to provide contextual information in the sequence tagging process.

D. CHARACTER ENCODER

As discussed in Section I, our task suffers from the OOV problem, i.e., a large portion of words in forums (e.g., “Q4”)

are not in the vocabulary. This problem can be alleviated by incorporating the character-level semantics (e.g., the postfix “.pdf” in the word “intro.pdf”). In fact, introducing the character-level inputs to build word embeddings has already been proved to be effective in various NLP tasks, such as part-of-speech tagging [38] and language modeling [39]. In our model, we build up a character encoder to encode character-level embeddings to fight against the OOV problem. For each word, we use bidirectional LSTMs to process the sequence of its characters from both sides and their final state vectors are concatenated. The resulting representation is then concatenated with the word-level embeddings to feed to the sequence tagger in Section IV-E.

We denote \mathcal{V}_C as the alphabet of characters, including uppercase and lowercase letters as well as numbers and punctuation, with dimensionality in the low hundreds. The input word w is decomposed into a sequence of characters $x_1, \dots, x_{|w|}$, with each x_i represented as an one-hot vector over \mathcal{V}_C . We denote $\mathbf{E}_c \in \mathbb{R}^{d_c \times \mathcal{V}_C}$ as the input character embedding matrix, where d_c is the dimension of character embeddings. Given $x_1, \dots, x_{|w|}$, a bidirectional LSTM computes the forward state h_i^f by applying $h_i^f = \text{LSTM}(\mathbf{E}_c c_i, h_{i-1}^f)$, and computes the backward state h_i^b by applying $h_i^b = \text{LSTM}(\mathbf{E}_c c_{|w|-i+1}, h_{i-1}^b)$. Finally, the input vector v_w to the sequence tagger is the concatenation of word and character embeddings, i.e., $v_w = [\mathbf{E}_w w; h_{|w|}^f; h_{|w|}^b]$.

E. LSTM-CRF TAGGER

After defining the input vector v_w and the context vector h_c , we build up the attentive LSTM-CRF tagger to assign a tag to each word. Given a sentence with n words $s = \{w_1, \dots, w_T\}$

in the input thread T with context C , to obtain its tag sequence $l = \{l_1, \dots, l_T\}$, we are actually approximating the conditional probability $p(l_1, \dots, l_T | w_1, \dots, w_T, C)$. This can be effectively modeled by the LSTM-CRF tagger [7] in the following way.

$$p(l_1, \dots, l_T | w_1, \dots, w_T, C) = \frac{\exp(r(s, l|C))}{\sum_{l'} \exp(r(s, l'|C))} \quad (5)$$

where $r(s, l|C)$ is a scoring function indicating how well the tag sequence l fits the given input sentence s , given the context C . In LSTM-CRF, $r(s, l|C)$ is parameterized by a transition matrix A and a non-linear neural network f , as follows:

$$r(s, l|C) = \sum_{t=1}^T (A_{l_{t-1}, l_t} + f(w_t, l_t|C)) \quad (6)$$

where $f(w_t, l_t|C)$ is the score output by the LSTM network for the t -th word w_t and the t -th tag l_t , conditioned on the context C . The matrix A is the transition score matrix, $[A]_{ij}$ is the transition score from i -th tag to j -th for a consecutive time steps.

To model the score $f(w_t, l_t|C)$, we build a bidirectional-LSTM network with attention over the contexts C . In time step t , the current hidden state h_t is updated as follows:

$$h_t = LSTM([v_{w_t}; a_c^t], h_{t-1}) \quad (7)$$

where v_{w_t} is input vector for word w_t , a_c^t is the attended context vector of h_c at time step t , which will be discussed in detail later. Then, the score $f(w_t, l_t|C)$ is computed through a linear output layer with softmax, as follows:

$$o_t = W_o h_t \quad (8)$$

$$f(w_t, l_t|C) = \frac{\exp(o_{l_t, t})}{\sum_j \exp(o_{j, t})} \quad (9)$$

where W_o is the matrix that maps hidden states h_t to output states o_t .

F. CONTEXT ATTENTION ON THE TAGGER

To effectively select useful information from the contexts, we introduce an attention mechanism over all the hidden states of the context sentences $h_{|c_1|}^{c_1}, \dots, h_{|c_i|}^{c_i}, \dots, h_{|c_m|}^{c_m}$. We denote α_i^t as the scalar value determining the attention weight of the context vector $h_{|c_i|}^{c_i}$ at time step t . Then, the input context vector to the LSTM-CRF tagger a_c^t is calculated as follows:

$$a_c^t = \sum_{i=1}^m \alpha_i^t h_{|c_i|}^{c_i} \quad (10)$$

Given the previous state of the LSTM h_{t-1} , the attention mechanism calculates the context attention weights $\alpha^t = \alpha_1^t, \dots, \alpha_m^t$ as a vector of scalar weights, where α_i^t is calculated as follows:

$$e_i^t = v_a^\top \tanh(W_a h_{t-1} + U_a h_{|c_i|}^{c_i}) \quad (11)$$

$$\alpha_i^t = \frac{\exp(e_i^t)}{\sum_j \exp(e_j^t)} \quad (12)$$

where v_a, W_a, U_a are trainable weight matrices of the attention modules. Note that we actually calculate an attention over all context sentences, but not on the word level, which greatly reduce the scale of parameters. Another reason to use sentence-level attention is based on the observation that the useful information tends to appear coherently in one context sentence, rather than separated in different sentences.

V. EXPERIMENTS

A. BASELINES

Since we formulate our task as a sequence tagging problem, to evaluate the performance of the proposed method, we conduct experiments on several widely-used sequence tagging models as follows:

- **BLSTM**: the bidirectional LSTM network (BLSTM) [40] has been widely used for sequence tagging task. In predicting the tag of a specific time frame, it can efficiently make use of past features (via forward states) and future features (via backward states). We train the BLSTM using back-propagation through time (BPTT) [41] with each sentence-tag pair (s, l) as a training example.
- **CRF**: Conditional Random Fields (CRF) [42] is a sequence tagging model that utilizes neighboring tags and sentence-level features in predicting current tags. In our implementation of CRF, we use the following features: (1) current word, (2) the first/last two/three characters of the current word, (3) whether the word is digit/title/in upper case, (4) the POS tag, (5) the first two symbols of the POS tag, and (6) the features (1)-(5) for the previous and next two words.
- **BLSTM-CRF**: As we illustrated in Section IV-E, BLSTM-CRF [7] is a state-of-the-art sequence tagging model that combines a BLSTM network with a CRF layer. It can efficiently use past input features via a LSTM layer and sentence level tag information via a CRF layer.
- **BLSTM-CRF-CE**: This model adds a character encoder (CE), as described in Section IV-D, into the BLSTM-CRF model. It can be regarded as a simplified version of the proposed model, i.e., without the context encoder.
- **BLSTM-CRF-CE-CA**: The full version of the proposed method, i.e., an incremental model of BLSTM-CRF that takes into account the character-level inputs and the thread context information.

B. EXPERIMENTAL SETTINGS

1) DATASETS

We test LSTM, CRF, LSTM-CRF, LSTM-CRF-CE and our model on both the FoRM-M and the FoRM-L datasets. For each dataset, we randomly split the data into 2 parts: 90% for training and 10% for testing. This results in 6,796 training and 839 testing examples for FoRM-M, and 16,160 training and 1,996 testing examples for FoRM-L.

TABLE 4. Overall performance of different methods on the FoRM dataset (%). The best performances for each metric are in bold.

Models	FoRM-M			FoRM-L		
	Precision	Recall	F_1 Score	Precision	Recall	F_1 Score
BLSTM	71.53	47.32	56.96	64.46	48.04	55.05
CRF	78.08	69.09	73.31	73.82	62.03	67.41
BLSTM-CRF	75.40	72.38	73.90	74.39	66.01	69.94
BLSTM-CRF-CE	73.76	77.62	75.64	76.20	70.21	73.08
BLSTM-CRF-CE-CA	72.91	79.20	75.92	74.32	74.17	74.24

TABLE 5. The F_1 scores of different methods for each resource mention type on the FoRM-M dataset. The best results are in bold.

Resource Type	Tag	BLSTM	CRF	BLSTM-CRF	BLSTM-CRF-CE	BLSTM-CRF-CE-CA
Assessments	B	47.67	74.48	78.27	79.83	80.15
	I	61.32	76.01	79.66	81.06	81.44
Exams	B	45.89	70.27	72.52	73.97	77.46
	I	62.55	65.48	72.55	77.75	76.20
Videos	B	51.73	75.48	69.23	67.31	66.41
	I	62.33	74.94	70.70	74.28	70.90
Coursewares	B	45.75	67.23	64.45	72.92	74.30
	I	46.98	50.42	50.48	54.48	56.93

2) SETUP

For deep learning models, we set the size of the word representation to 200, and initialize the word embedding matrix with pre-trained GloVe [43] vectors. In the LSTM-CRF-CE and our model, we set the dimensionality of characters to 64. Each hidden state used in the LSTM and GRU is set to 256. We train all models by stochastic gradient descent, with a minibatch size of 16, using the ADAM optimizer⁶. For the CRF model, we implement it using the keras-contrib⁶ package. To evaluate the overall performance, we use the micro-precision/recall/ f_1 score on all the resource mention tags, i.e., all tags excluding the O tag, calculated as follows:

$$\text{micro-}P = \frac{\sum_{t \in L/O} TP_t}{\sum_{t \in L/O} (TP_t + FP_t)} \quad (13)$$

$$\text{micro-}R = \frac{\sum_{t \in L/O} TP_t}{\sum_{t \in L/O} (TP_t + FN_t)} \quad (14)$$

$$\text{micro-}F_1 = \frac{2 \times \text{micro-}R \times \text{micro-}P}{\text{micro-}R + \text{micro-}P} \quad (15)$$

where L is the tag set, TP_t , FP_t and FN_t represents the number of true positive, false positive, and false negative examples for the tag $t \in L$, respectively.

C. EXPERIMENTAL RESULTS

We train models using training data and monitor performance on validation data. During training, 10% of training data are held out for validation (10-fold cross validation). The model is re-trained on the entire training data with the best parameter settings, and finally evaluated on the test data. For deep learning models, we use a learning rate of 0.01, and the

training process requires less than 20 epochs to converge and it in general takes less than a few hours.

We report models' performance on test datasets in Table 4, in which the best results are in bold cases. On both FoRM-M and FoRM-L dataset, BLSTM-CRF-CE-CA achieves the best F_1 score, which indicates the robustness and effectiveness of the proposed method. Specifically, we also have the following observations.

- (1) BLSTM is the weakest baseline for both two data sets. It obtains relatively high precision but poor recall. When predicting current tags, BLSTM only considers the previous and post words, without making use of the neighboring tags to predict the current one. This problem greatly limits its performance, especially in identifying the Begin tags, which will be further demonstrated in Table 5.
- (2) The CRF forms strong baselines in our experiments, especially in precision. In the FoRM-M dataset, it achieves the best precision of 78.08% among all the models. This is as expected, because hand-crafted local linguistic features are used in the CRF, making it easy for the model to capture the phrases with strong "indicating words", such as "quiz 1.1" and "video of lecture 4". However, the recall for CRF is relatively low (11.3% lower than the proposed method in average), because in many cases, local linguistic features are not enough in identifying resource mentions, and long-range context dependencies need to be considered (e.g., the phrase "Chain Rule" in Figure 1).
- (3) The BLSTM-CRF performs close to CRF on precision, but is better than CRF on recall (+3.64%

⁶<https://github.com/keras-team/keras-contrib>

TABLE 6. The performance comparison between BLSTM-CRF and BLSTM-CRF-CE on the test set of FoRM-M. ‘Correct/Total’ refers to the correct/total number of predictions, ‘Ratio’ is the ratio of correct prediction.

Performance	Correct / Total / Ratio		
	All Mentions	None-OOV Mentions	OOV Mentions
LSTM-CRF	564 / 876 / 64.38%	471 / 713 / 66.06%	93 / 163 / 57.06%
LSTM-CRF-CE	600 / 876 / 68.49%	493 / 713 / 69.14%	107 / 163 / 65.64%
Improvements	4.11%	3.08%	8.58%

in average). During prediction, the model can make use of the full context information encoded in LSTM cell rather than only local context features.

- (4) After considering character embeddings, the change of precision is not obvious, but the recall improves 4.72% in average compared with BLSTM-CRF. This demonstrates the effectiveness of incorporating character-level semantics. We will further analyze how character embeddings alleviates the OOV problem in Section V-D. Encoding the thread contexts further improves the recall (+2.77% in average), at the cost of a slightly drop on precision (−1.37% in average). The thread contexts bring in enough information for inferring long-term dependencies, but also burdens the model to filter out irrelevant information introduced.
- (5) As expected, the F_1 score of all models drops when moving from the FoRM-M to the FoRM-L dataset that contains more noisy annotations. This decrease in performance is more obvious on recall, with an average of 5.03% drop. The most significant performance drop comes from CRF (−5.9% in F_1 score), which further exposes its limitation in handling the variability of resource mentions. The proposed method, with a 1.68% decrease in F_1 , proves to be the most robust model, owing to its high model complexity.

To further investigate how different models perform on identifying each type of resource mention, we report models’ micro- F_1 scores for each type of tag on the FoRM-M dataset. The results are summarized in Table 5, and we get several interesting observations. For BLSTM, the F_1 score of Begin tags (47.76% in average) is much lower than that of Inside Tags (58.29% in average). A reasonable explanation is that there are less training data for B-tags compared with I-tags, and BLSTM does not utilize the neighboring tags to predict the current one. After adding the CRF layer, the BLSTM-CRF model makes a significant improvement in identifying B-tags (+23.35% in average). Among the four mention types, the models achieve best results in identifying the Assessments. There are two reasons: (1) there are about 3 times labeled data for the Assessments, compared with the other 3 types, and (2) identifying the mention of assessments does not rely much on long-range contexts (e.g., “Assignment 1.3”). The Coursewares is the most difficult resource type to identify; all models achieve the lowest F_1 scores in identifying the Coursewares. This is due to the high variety

of this type, since it is a mixture of transcripts, readings, slides, and other additional resources. Furthermore, long-range context dependency is more common in this type (e.g., “sgd.py”), which further increases its variety.

D. EFFECT OF THE CHARACTER ENCODER

This section examines how our introduction of the *Character Encoder* addresses the problem of Out-of-Vocabulary. To this end, we first evaluate the severity of the OOV problem on our data. We define OOV words as the words that cannot be found in the pre-trained GloVe embeddings, which has a vocabulary size of 400K.⁷ As OOV words do not have pre-trained word embeddings, we need their character-level information to be taken into account. The FoRM-M dataset contains a vocabulary size of 9,761, with 3,045 (31.19%) of them are OOV words. This reveal the severe of the OOV problem in our task.

To understand how character encoder addresses the OOV problem, we analyze the prediction results of BLSTM-CRF and BLSTM-CRF-CE on the test set of FoRM-M, which contains 876 ground-truth resource mentions within its 839 testing examples. Among these 876 resource mentions, 163 of them contain at least one OOV word. We call these resource mentions as *OOV Mentions*; identifying OOV Mentions require both word-level and character-level semantics. Other resource mentions are then denoted as *None-OOV Mentions*.

Table 6 shows the performance comparison between BLSTM-CRF and BLSTM-CRF-CE on both the OOV mentions and none-OOV mentions. Among the 876 testing resource mentions, the rate of correct predictions⁸ increases from 64.38% to 68.49%, with a 4.11% improvement. But the performance improvement for the none-OOV mentions only increase 3.08%. For the OOV mentions, however, the performance boost is 8.58%, much higher than the overall improvement of performance. This indicates that incorporating character-level information significantly benefits the identification of OOV resource mentions, which makes a major contribution to the overall performance improvement.

E. ERROR ANALYSIS

The micro- F_1 score is a proper evaluation metric for models’ performance on individual tags; however, does not tell us why errors are made. To provide an in-depth analysis of

⁷<https://nlp.stanford.edu/projects/glove/>

⁸A correct prediction means that the prediction of scope and type for a resource mention are both correct.

TABLE 7. Prediction error types and examples. (Pred.) is the model prediction, and (G.T.) is the ground truth. The bold texts with [] are identified/true resource mentions associated with type label.

Types	Examples
Exactly Correct	(Pred.) Problem with fminunc when I run [<i>ex2regm</i>] _{Coursewares} . (G.T.) Problem with fminunc when I run [<i>ex2regm</i>] _{Coursewares} .
Missing	(Pred.) I was wondering about the file location of house_data_g1. (G.T.) I was wondering about the file location of [<i>house_data_g1</i>] _{Coursewares} .
Wrongly Extracted	(Pred.) I was wondering about [<i>the file</i>] _{Coursewares} location of house_data_g1. (G.T.) I was wondering about the file location of house_data_g1.
Scope Wrong Type Right	(Pred.) Hi, I completed [<i>thequiz</i>] _{Exams} for week 2. (G.T.) Hi, I completed [<i>thequizforweek2</i>] _{Exams} .
Scope Right Type Wrong	(Pred.) How about to understand it better from [<i>thesimulationlecture</i>] _{Assessments} . (G.T.) How about to understand it better from [<i>thesimulationlecture</i>] _{Videos} .
Scope Wrong Type Wrong	(Pred.) Anybody have the zip file of [<i>assignment</i>] _{Assessments} for week 3? (G.T.) Anybody have [<i>thezipfileofassignmentforweek3</i>] _{Coursewares} ?

TABLE 8. Error analysis of BLSTM-CRF-CE-CA on the FoRM-M dataset. The table shows the number/percentage of different prediction cases for different resource types.

Types	Assessments	Exams	Videos	Coursewares	Total
Exactly Correct	385 (69.4%)	61 (68.6%)	66 (54.1%)	88 (59.5%)	600 (65.6%)
Missing	15 (2.7%)	2 (2.2%)	4 (3.3%)	19 (12.8%)	40 (4.4%)
Wrongly Extracted	19 (3.4%)	3 (3.4%)	5 (4.1%)	11 (7.4%)	38 (4.2%)
Scope Wrong Type Right	129 (23.2%)	22 (24.7%)	44 (36.1%)	20 (13.5%)	215 (23.5%)
Scope Right Type Wrong	4 (0.7%)	1 (1.1%)	1 (0.8%)	4 (2.7%)	10 (1.1%)
Scope Wrong Type Wrong	3 (0.6%)	0 (0.0%)	2 (1.6%)	6 (4.1%)	11 (1.2%)
Total	555	89	122	148	914

the proposed model's performance, we list the six possible conditions that happen during the prediction, summarized in Table 7, together with examples. The model makes an Exactly Correct prediction if the scope of the prediction exactly matches the ground truth, and the predicted type is also correct. There are cases when scopes are matched but the predicted type is incorrect or conversely, these are summarized as three cases: Scope Right/Wrong Type Right/Wrong. The remaining conditions happen when the prediction has no overlap with the ground truth in the sentence, which are divided into Missing and Wrongly Extracted errors.

Table 8 summarizes the performance of BLSTM-CRF-CE-Context on the FoRM-M test set. Among all the 914 cases obtained from the 839 testing examples, 600(65.6%) of them are predicted completely correctly by the model. We observe that most of the errors come from the Scope Wrong Type Right, holding a high percentage of 23.5%. Compared to this, other errors are less obvious. However, we further discover that a large portion (178 out of 215 cases) of this error happens because the model selects a more 'general' mention from a longer ground truth. For example, as given by the example in Table 7, the model selects the phrase "the quiz" from the ground truth mention "the quiz

for week 2". This behavior can be explained by the feature of sequence tagging; the decoder tends to select shorter and general patterns, as they are more frequently present as training signals. To some extent, both general and specific mentions are acceptable in practice, but teaching model to identify more specific mentions is a future direction for improvement. A potential solution is to take into account the grammatical structure of the sentence in decoding. Another observation is that besides the scope error, the Missing error holds a high percentage of 12.8% in identifying Coursewares. This is consistent with the relative low recall presented in Table 4, which poses the challenges of dealing with noisy expressions and long-range context dependency. Encoding thread context partially addresses the challenge, but there is still much room for improvement.

VI. CONCLUSION AND FUTURE WORKS

We propose and investigate the problem of automatically identifying resource mentions in MOOC discussion forums. We precisely define the problem and introduce the major challenges: the variety of expressions and the context dependency. Based on the vanilla LSTM-CRF model, we propose a character encoder to address the OOV problem caused by

TABLE 9. The description and examples for the pre-defined 7 types of resources mentions.

Types	Description	Examples
Assessments	Assessments are any form of assessments, exercises, homeworks or assignments.	the/that/this/my assignment; assignment 3; question 1 in assignment 2; assignment [name].
Exams	Exams are any form of evaluations to the knowledge and/or skills of learners, including quizzes, tests, mid-exams and final exams.	the/that/this quiz; quiz 3; question 1 in quiz 2; quiz [name].
Videos	Videos present the lecture content.	this/that/the video; this/that/the lecture; video [name]; slide 4 in video.
Readings	Reading materials optionally provide a list of other learning resources provided by courses.	the exercise instructions; this week's video tutorials; the lecture notes.
Slides	Slides provide the lecture content for download and separate review, which are often aligned to those in the video.	slide pdfs; slide 2; the lecture slides.
Transcripts	Transcripts or subtitles of the videos.	transcript of video 1; the subtitle files.
Additional Resources	Additional Resources help learners to catch other materials made available for specialized discipline-specific courses.	ex2regm.m; ex2reg scripts.

the variety of expressions, and a context encoder to capture the information of thread contexts. To evaluate the proposed model, we manually construct a large scale dataset FoRM based on real online forum data collected from Coursera. The FoRM dataset will be published as the first benchmark dataset for this task. Experimental results on the FoRM dataset validate the effectiveness of the proposed method.

To build up a more efficient and interactive environment for learning and discussing in MOOCs, it requires the inter-linkings between resource mentions and real resources. Our work takes us closer towards this goal. A promising future direction is to investigate how to properly resolve the identified resource mentions to real learning resources. However, it is also worthy to notice that the current identification performance still has much room for improvement; there are still challenges that are not fully addressed, such as identifying more specific resource mentions, as discussed in Section V-E. Addressing these challenges by utilizing more features from both static materials and dynamic interactions in MOOCs are also promising future directions to be explored.

VII. APPENDIX ANNOTATION DETAILS

We train the annotators in advance, before starting the annotation at June, 2018. First, we email every annotator with an annotation instruction document, which contains the detailed description and examples for different types of resource resources, cf Table 9. We then provide them with a link to our brat platform with an example annotation file containing

TABLE 10. Original annotation result from group 1 and group 2 on assessments, exams, videos, readings, slides, transcripts, and additional resources. The values in the parentheses are the ratios of intersections (column 4) for each type of resource mentions of group 1 (column 2) and group 2 (column 1).

Resource Type	Group 1	Group 2	Intersection
Assessments	8,047 (67.7%)	8,520 (64.0%)	5,451
Exams	1,891 (60.6%)	3,624 (31.6%)	1,146
Videos	1,852 (66.7%)	3,037 (40.7%)	1,236
Readings	821 (42.3%)	1,136 (30.5%)	347
Slides	190 (51.1%)	287 (33.8%)	97
Transcripts	163 (19.6%)	147 (21.8%)	32
Additional Resources	2,107 (51.3%)	2,716 (39.8%)	1,081
Total	15,071 (62.3%)	19,467 (48.2%)	9390

formatted annotation data and typical examples. They are required to complete an one hour training to learn the usage of the annotation tool and try out some practical annotations to better understand the annotation instruction. Finally, we add every annotator to a Wechat group to coordinate questions and answers about unclear examples. We observe that a few questions are raised at the beginning of the annotation, and later the annotators become more confident and fluent in their annotation. Table 10 summarized the detailed information of the annotated data for 7 types of learning resource mentions as described in Table 9. We find that the percentages of intersections for two groups of each type of resource mentions are

from 19.6% (Transcripts for Group 1) to 67.7% (Assessments for Group 1).

REFERENCES

- [1] Y.-H. An, M. K. Chandrasekaran, M.-Y. Kan, and Y. Fu, "The MUIR framework: Cross-linking MOOC resources to enhance discussion forums," in *Proc. Int. Conf. Theory Pract. Digit. Libraries*. Germany, Berlin: Springer, 2018, pp. 208–219.
- [2] E. Brill, "A simple rule-based part of speech tagger," in *Proc. 3rd Conf. Appl. Natural Lang. Process.*, 1992, pp. 152–155.
- [3] T. Brants, "TnT: A statistical part-of-speech tagger," in *Proc. 6th Conf. Appl. Natural Lang. Process.*, 2000, pp. 224–231.
- [4] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Linguistic Invest.*, vol. 30, no. 1, pp. 3–26, Jan. 2007.
- [5] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. NAACL-HLT*, 2016, pp. 260–270.
- [6] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 357–370, Dec. 2016.
- [7] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*. [Online]. Available: <https://arxiv.org/abs/1508.01991>
- [8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: <https://arxiv.org/abs/1409.0473>
- [9] F. Wang, X. Li, W. Lei, C. Huang, M. Yin, and T.-C. Pong, "Constructing learning maps for lecture videos by exploring wikipedia knowledge," in *Proc. Pacific Rim Conf. Multimedia*. Berlin, Germany: Springer, 2015, pp. 559–569.
- [10] L. Pan, X. Wang, C. Li, J. Li, and J. Tang, "Course concept extraction in MOOCs via embedding-based graph propagation," in *Proc. 8th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2017, pp. 875–884.
- [11] L. Pan, C. Li, J. Li, and J. Tang, "Prerequisite relation learning for concepts in moocs," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1447–1456.
- [12] J. Hammerton, "Named entity recognition with long short-term memory," in *Proc. 7th Conf. Natural Lang. Learn. HLT-NAACL*, 2003, pp. 172–175.
- [13] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.
- [14] C. D. Santos, V. Guimaraes, R. J. Niterói, and R. de Janeiro, "Boosting named entity recognition with neural character embeddings," in *Proc. 5th Named Entity Workshop*, 2015, pp. 25–33.
- [15] R. Mitkov, "Multilingual anaphora resolution," *Mach. Transl.*, vol. 14, nos. 3–4, pp. 281–299, 1999.
- [16] J. R. Hobbs, "Resolving pronoun references," *Lingua*, vol. 44, no. 4, pp. 311–338, 1978.
- [17] S. Lappin and H. J. Leass, "An algorithm for pronominal anaphora resolution," *Comput. Linguistics*, vol. 20, no. 4, pp. 535–561, 1994.
- [18] B. J. Grosz, S. Weinstein, and A. K. Joshi, "Centering: A framework for modeling the local coherence of discourse," *Comput. Linguistics*, vol. 21, no. 2, pp. 203–225, 1995.
- [19] J. P. Walker, *Centering Theory in Discourse*. New York, NY, USA: Oxford Univ. Press, 1998.
- [20] C. Aone and S. W. Bennett, "Evaluating automated and manual acquisition of anaphora resolution strategies," in *Proc. 33rd Annu. Meeting Assoc. Comput. Linguistics*, 1995, pp. 122–129.
- [21] H. Lee, M. Surdeanu, and D. Jurafsky, "A scaffolding approach to coreference resolution integrating statistical and rule-based models," *Natural Lang. Eng.*, vol. 23, no. 5, pp. 733–762, 2017.
- [22] R. Mitkov, R. Evans, and C. Orasan, "A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*. Berlin, Germany: Springer, 2002, pp. 168–186.
- [23] N. Ge, J. Hale, and E. Charniak, "A statistical approach to anaphora resolution," in *Proc. 6th Workshop Very Large Corpora*, 1998, pp. 161–170.
- [24] S. J. Wiseman, A. M. Rush, S. M. Shieber, and J. Weston, "Learning anaphoricity and antecedent ranking features for coreference resolution," in *Proc. Assoc. Comput. Linguistics*, 2015, pp. 1416–1426.
- [25] R. Sukthankar, S. Poria, E. Cambria, and R. Thirunavukarasu, "Anaphora and coreference resolution: A review," 2018, *arXiv:1805.11824*. [Online]. Available: <https://arxiv.org/abs/1805.11824>
- [26] S. Wiseman, A. M. Rush, and S. M. Shieber, "Learning global features for coreference resolution," 2016, *arXiv:1604.03035*. [Online]. Available: <https://arxiv.org/abs/1604.03035>
- [27] K. Clark and C. D. Manning, "Deep reinforcement learning for mention-ranking coreference models," 2016, *arXiv:1609.08667*. [Online]. Available: <https://arxiv.org/abs/1609.08667>
- [28] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," 2017, *arXiv:1707.07045*. [Online]. Available: <https://arxiv.org/abs/1707.07045>
- [29] T. Ohta, S. Pyysalo, J. Tsujii, and S. Ananiadou, "Open-domain anatomical entity mention detection," in *Proc. Workshop Detecting Struct. Scholarly Discourse*, 2012, pp. 27–36.
- [30] J.-D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, "Overview of BioNLP'09 shared task on event extraction," in *Proc. Workshop Current Trends Biomed. Natural Lang. Process. Shared Task*, 2009, pp. 1–9.
- [31] J.-D. Kim, S. Pyysalo, T. Ohta, R. Bossy, N. Nguyen, and J. Tsujii, "Overview of BioNLP shared task 2013," in *Proc. BioNLP Shared Task Workshop*, 2013, pp. 1–7.
- [32] G. Hripcsak and A. S. Rothschild, "Agreement, the F-measure, and reliability in information retrieval," *J. Amer. Med. Informat. Assoc.*, vol. 12, no. 3, pp. 296–298, 2005.
- [33] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: The kappa statistic," *Family Med.*, vol. 37, no. 5, pp. 360–363, 2005.
- [34] S. Bhatia and P. Mitra, "Adopting inference networks for online thread retrieval," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 1300–1305.
- [35] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [36] H. Elshahar, C. Gravier, and F. Laforest, "Zero-shot question generation from knowledge graphs for unseen predicates and entity types," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2018, pp. 218–228.
- [37] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.
- [38] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2741–2749.
- [39] W. Ling, T. Luís, L. Marujo, R. F. Astudillo, S. Amir, C. Dyer, A. W. Black, and I. Trancoso, "Finding function in form: Compositional character models for open vocabulary word representation," in *Proc. Empirical Methods Natural Lang. Process.*, 2015, pp. 1520–1530.
- [40] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2013, pp. 6645–6649.
- [41] M. Boden, "A guide to recurrent neural networks and backpropagation," School Inf. Sci., Halmstad Univ., Halmstad, Sweden, SICS Tech. Rep. T2002:03, 2002.
- [42] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, vol. 951, no. 2001, 2001, pp. 282–289.
- [43] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.



YA-HUI AN received the B.S. degree in computer science and technology from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2013, where she is currently pursuing the Ph.D. degree in computer science. From 2016 to 2018, she was with the School of Computing (SoC), National University of Singapore (NUS), as a Research Intern. Her main research topics include the applications of recommender systems and the

user behaviors in e-commerce platforms, social media platforms, and news platforms. She is also interested in natural language processing and digital library.



LIANGMING PAN received the B.S. degree in computer science from Beihang University, Beijing, in 2014, and the M.S. degree in computer science from Tsinghua University, Beijing, in 2017. He is currently pursuing the Ph.D. degree in computer science with the National University of Singapore (NUS), Singapore. His research interests include knowledge base, natural language processing, and data mining. He is currently involved in discovering research dynamics in scientific literature.



QIANG DONG received the B.S. and Ph.D. degrees in computer science and technology from Chongqing University, in 2006 and 2010, respectively. From 2008 to 2009, he visited the Department of Computer Science, Montclair State University, NJ, USA. He is currently an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. He has published over 20 academic papers in peer-reviewed journals.

His research interests include combinatorial networks, complex networks, and recommender systems.



MIN-YEN KAN received the B.S., M.S., and Ph.D. degrees in computer science from Columbia University, NY, USA. He is currently an Associate Professor with the School of Computing, National University of Singapore. His research interests include the areas of digital libraries, natural language processing, information retrieval, and human-computer interaction. His main research interests include applying natural language (NL) techniques to help researchers find the information

they need to act on. A key focus of his work is improving information retrieval user interfaces with the use of NL technology to make search results easier to understand and disseminate to collaborators.



YAN FU received the B.S. degree from the Department of Mathematics, Sichuan University, Chengdu, Sichuan, China, in 1982, and the M.S. degree from the Department of Computer Software, University of Electronic Science and Technology of China, in 1988, where she is currently a Professor with the School of Computer Science and Engineering. Her research interests include the areas of database, data mining, and neural networks.

...