Joint Recognition of Names and Publications in Academic Homepages

Yimeng Dai
The University of Melbourne
yimengd@student.unimelb.edu.au

Jianzhong Qi
The University of Melbourne
jianzhong.qi@unimelb.edu.au

Rui Zhang*
The University of Melbourne
rui.zhang@unimelb.edu.au

ABSTRACT

Academic homepages are an important source for learning researchers' profiles. Recognising person names and publications in academic homepages are two fundamental tasks for understanding the identities of the homepages and collaboration networks of the researchers. Existing studies have tackled person name recognition and publication recognition separately. We observe that these two tasks are correlated since person names and publications often co-occur. Further, there are strong position patterns for the occurrence of person names and publications. With these observations, we propose a novel deep learning model consisting of two main modules, an *alternatingly updated memory* module which exploits the knowledge and correlation from both tasks, and a *position-aware memory* module which captures the patterns of where in a homepage names and publications appear. Empirical results show that our proposed model outperforms the state-of-the-art publication recognition model by 3.64% in F1 score and outperforms the state-of-the-art person name recognition model by 2.06% in F1 score. Ablation studies and visualisation confirm the effectiveness of the proposed modules.

ACM Reference Format:
Yimeng Dai, Jianzhong Qi, and Rui Zhang. 2020. Joint Recognition of Names and Publications in Academic Homepages. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371771

1 INTRODUCTION

Recognition of person names and publications from academic homepages are two essential tasks for analysing researchers' profiles. There have been extensive research interests in the extraction and mining of such information from academic homepages [3, 8, 16, 20, 28, 31]. The recognition process has become a necessary part of many online systems, such as AMiner [24] and CiteSeerX [16], and the extracted person names and publications can bring interesting applications. For example, person names can provide valuable insights for analysing researchers' collaboration networks. Publications can be used to mine the evolution of a researcher's research interests and predict the development directions of the researcher.

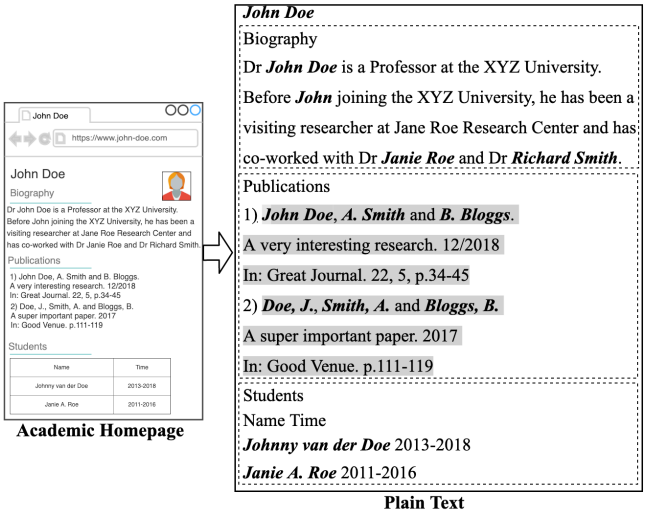

Figure 1: An example of recognising person names and publications in academic homepages. Names are marked in bold italic and publications are marked with grey background.

Figure 1 shows an example of the two tasks. Given the plain text of an academic homepage, the aim is to recognise every person name and every publication as a text string shown in the example.

Recently, deep learning based methods have been developed to address these problems. The state-of-the-art for publication recognition [31] uses a Bi-LSTM-CRF based model to learn the page-level and line-level structure. The state-of-the-art for person name in academic homepages [1] uses a co-guided neural network to learn from fine-grained annotation of names. Despite their success, these studies have tackled the two tasks separately. We observe that there is a strong correlation between person names and publications. For example, a string is more likely to refer to a publication if it contains multiple names consecutively. Also, strings that appear in multiple publications are likely to be a person's name, e.g., the page owner or a frequent coauthor. Such an observation motivates us to design joint learning models for publication and person names simultaneously. A straightforward method to learn a model for the two tasks jointly is to train them together by minimising the total loss of the two tasks or simply concatenating the representation of publication and person name when training [6, 13]. However, our experimental study shows that such a straightforward approach performs poorly. The issue is that they cannot capture the correlation between the two tasks well since the learned signals from the

*Corresponding author.

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371771

two tasks do not have enough intermediate interaction with each other at each iteration of training.

Further, we observe that there are strong patterns of where in a homepage names and publications appear. Academic homepages usually present information in separate blocks, e.g., one for biography and another for publications (cf. Figure 1). These blocks may use different formatting styles, such as paraghraphs, lists, and tables. The grouping of similar contents into separate blocks and the similar formatting styles within the same block lead to strong position patterns in the plain text of academic homepages. Specifically, the contents of the same block may run across multiple consecutive lines, while the contents of different blocks may be separated. Further, each line or several consecutive lines in a block may describe one piece of information. For example, the block for publications in Figure 1 consists of six consecutive lines and each publication consists of three lines. The position of lines and blocks provides valuable signals for the recognition tasks.

To address the issues in straightforward joint models, and to better utilise the correlation and position patterns of person names and publications, we propose a novel *Position-aware Alternating Memory* (PAM) network. PAM consists of two main modules, an *alternatingly updated memory* (AM) module which exploits the knowledge and correlation from both tasks, and a *position-aware memory* (PM) module which captures the patterns of where in a homepage names and publications appear. In the AM module, an attention-based memory updating controller is used to activate hidden representation from a name encoder and a publication encoder alternatingly, and update the memory representation alternatingly to enhance the intermediate interaction in each iteration. The correlation representation between person names and publications is captured in the alternating updates of memories. In the PM module, position representations are integrated into the correlation representation between person names and publications. The position representations consist of local and global positions. The local position representations capture the difference in line numbers between tokens. The global position representations capture the attention distribution of all the lines in a homepage with respect to the publication block.

In summary, this paper makes the following contributions:

- We address the tasks of person name recognition and publication recognition in academic homepages simultaneously by modeling their correlation and the position patterns.
- We propose a deep learning model named PAM, which consists of two main modules, an *alternatingly updated memory* module and a *position-aware memory* module.
- We conduct a thorough experimental study using real datasets. The empirical results show that our model PAM outperforms the state-of-the-art publication recognition model by 3.64% in F1 score and outperforms the state-of-the-art person name recognition model by 2.06% in F1 score. Ablation studies and visualisation confirm the effectiveness of the proposed modules in our model.

2 RELATED WORK

Previous studies on academic homepages usually use rule-based [8, 30] or a hybrid of machine learning and rule-based methods [3] on the HTML DOM trees of webpages. Yang and Ho [30] use heuristic rules to locate the publications in a DOM tree. They assume that publications are listed as nodes at the same level in the DOM tree. Chung et al. [3] uses a linear chain CRF model to analyse the content in a DOM tree and then refines the publication boundaries by rules.

Recent studies on academic homepages usually treat the plain text of a homepage as a document and recognise information from the plain text using deep learning based natural language processing methods. For example, state-of-the-art techniques for publication recognition [31] and for person names recognition [1] use Bi-LSTM-CRF based models to recognise information from the plain text of the homepages. However, they solve the two tasks separately. To the best of our knowledge, no existing work has taken a joint learning approach to recognising person names and publications simultaneously from the plain text of academic homepages.

A few other studies recognise person names and publications from research papers and digital libraries [15, 18, 24, 25]. Such a recognition problem is simpler since the text in research papers and digital libraries is usually well-formatted with few format variations. After recognition, these studies may need to solve the name disambiguation problem (i.e., different people with identical names) [23] before mining the collaboration networks or research interests of a researcher. Such a problem can be alleviated by recognising information from academic homepages.

Models based on memory networks [22, 27] are proposed for question answering in recent years. Dynamic Memory Network (DMN) [12] uses a gated recurrent unit [2] based controller to update the memory, while Working Memory Network (W-MemNN) [17] uses a multi-head attention [26] based controller. All these networks use a memory module for a single task and update the memory repeatly, while our model updates the memory alternatingly using the knowledge from two correlated tasks.

Moreover, we use different methods to capture the position patterns. The state-of-the-art for publication recognition [31] trains webpage-level and line-level models together to capture the position information of academic homepage, whereas our model captures position information by integrating them into the memory updating process. Studies have exploited relative token position and importance in a sentence [21, 29], whereas our algorithm focuses on relative line position and importance in a page.

3 JOINT LEARNING FOR BOTH TASKS

Usually, the plain text of an academic homepage is saved first, then the recognition tasks are conducted on text [1, 31]. Given the plain text of an academic homepage, we aim to recognise all the person names and publications from the plain text simultaneously. To accomplish this, a straightforward method is to train a model for the two tasks together by solving a joint optimization problem, i.e., minimising the total loss of the two tasks, or using simple concatenation procedures when training [6, 13, 32]. However, our experimental study (Section 4.3.2) shows that such a naive way of joint learning does not yield good performance since the correlation between the two tasks cannot be captured well.

To address the problem, we propose a *Position-aware Alternating Memory* (PAM) network. Figure 2 illustrates our proposed PAM network. Our model consists of four modules including *input processor*, *alternatingly updated memory* (AM), *position-aware memory* (PM) and *joint recognition*. AM an PM are the two main modules for joint recognition of names and publications. In input processor

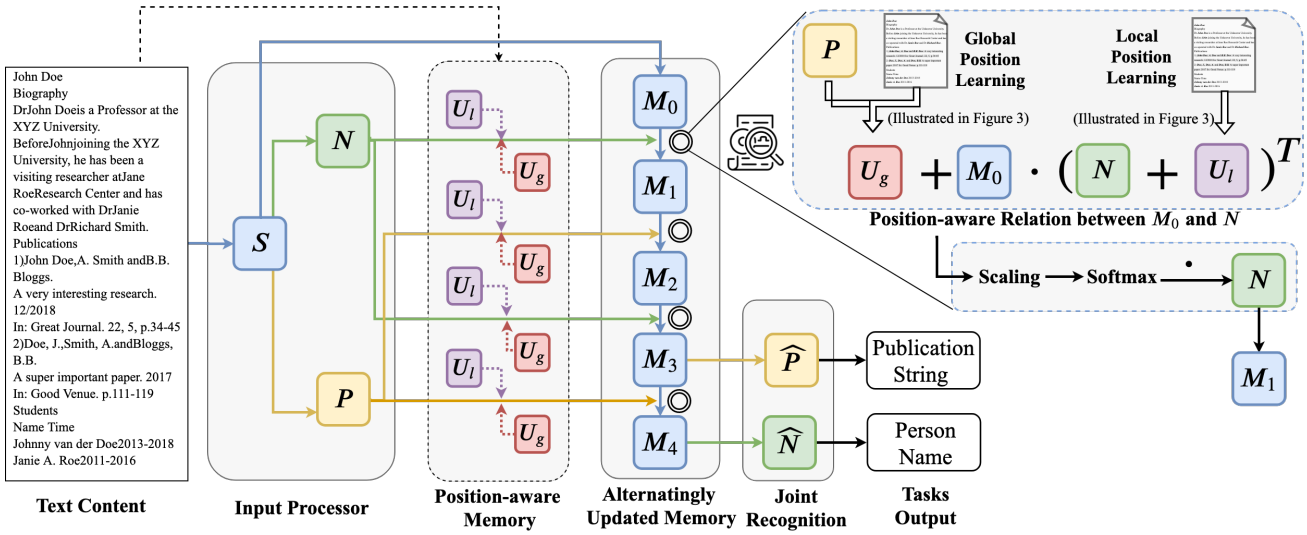

Figure 2: Overview of the PAM network. The *Alternatingly updated memory* (AM) module exploits the knowledge from both tasks. The *Position-aware memory* (PM) module integrates local and global position into the memory updating process.

(Section 3.1), we tokenise the plain text, use word embeddings to represent the tokenised text, and encode them through two encoders to get two sequential hidden representations, one for person names and the other for publications. Then, the hidden representations are passed to AM module (Section 3.2) to capture the correlation between person names and publications. Specifically, we use a memory updating controller to activate hidden representation and update the memory alternatingly. In position-aware memory (Section 3.3), to take advantage of the position patterns in academic homepages, we expand the memory updating controller and integrate local and global position representations into the memory updating process. In the joint recognition module (Section 3.4), we produce recognition reuslts based on the updated memory and jointly learn the two tasks. Table 1 summarises the symbols frequently used in the following discussions.

3.1 Input Processor

Given the plain text of an academic homepage, we first tokenise it and get a sequence $\boldsymbol{S}$ of n tokens and each token is represented as a d_e-dimension word embedding, i.e., $\boldsymbol{S} \in \mathcal{R}^{n \times d_e}$. Following state-of-the-art methods [1, 31], we use GloVe [19] to learn word embeddings on an academic homepage dataset (detailed in Section 4.1.1), although other pre-training methods may be used here without loss of generality. Then, we encode the input sequence $\boldsymbol{S}$ via two recurrent neural networks (RNNs), one for person name recognition and the other for publication recognition. Specifically, we use LSTM [7] as the RNN unit:

$$\boldsymbol{N} = LSTM(\boldsymbol{S}) \quad \text{and} \quad \boldsymbol{P} = LSTM(\boldsymbol{S}) \tag{1}$$

Here, $\boldsymbol{N} \in \mathcal{R}^{n \times d_h}$ is the hidden representation from the person name encoder (i.e., an LSTM), $\boldsymbol{P} \in \mathcal{R}^{n \times d_h}$ is the hidden representation from the publication encoder (another LSTM), and d_h is the dimensionality of each token in hidden representations.

Next, $\boldsymbol{S}$, $\boldsymbol{N}$, and $\boldsymbol{P}$ are passed to the AM module.

Table 1: Frequently used symbols

Symbols	Description
n	the number of tokens
d_e	the dimension of word embeddings
d_h	the dimension of hidden representations
d_m	the dimension of memeory representations
d_z	the dimension of each head
i	the hop number
$\boldsymbol{A}$	the alternating hidden representation
$\boldsymbol{M}$	the memory representation
$\boldsymbol{W}$	the projection matrix
$\boldsymbol{R}$	the relationship representation
$\boldsymbol{U_g}$	the global position representation
$\boldsymbol{U_l}$	the local position representation
$\boldsymbol{Q}$	the predicted publication block
l_μ	the median line of $\boldsymbol{Q}$
σ	half of the total number of lines in $\boldsymbol{Q}$
k	the number of predicted publication tokens in a line
l_t	the line number of token t
ε	the total number of lines in the homepage

3.2 Alternatingly Updated Memory

Different from the traditional Memory Networks [22, 27], which updates a memory representation repeatedly for a *single* task, we propose to update the memory representation *alternatingly* using the knowledge from *two* correlated tasks. Our intuition is to improve the learned representations for one task by taking into account the knowledge from the other task.

Specifically, AM initialises the memory representation $\boldsymbol{M}$ with $\boldsymbol{S}$, i.e., $\boldsymbol{M}_0 = \boldsymbol{S}$, and updates it using an alternating hidden representation $\boldsymbol{A}$, which is obtained from the person name encoder and publication encoder by:

$$\boldsymbol{A} = f(i+1)\boldsymbol{N} + f(i)\boldsymbol{P} \tag{2}$$

where $\boldsymbol{A} \in \mathcal{R}^{n \times d_h}$, i is the hop number and function f activate N and P alternatingly in two consecutive hops by providing alternating boolean values for even and odd values of i:

$$f(i) = \frac{1}{2}[(-1)^i + 1] \tag{3}$$

When updating the memory representation $\boldsymbol{M}$, we use a memory updating controller based on multi-head attention [26], which is similar to that used in Working Memory Network [17]. Multi-head attention allows the model to jointly attend to different representation subspaces using projection matrices.

Let $\boldsymbol{Z}_j$ denote the memory representation in head j. The memory representation in hop i, denoted by $\boldsymbol{M}_i$, is the concatenated memory representations of all the heads:

$$\boldsymbol{M}_i = (\boldsymbol{Z}_1 \oplus \boldsymbol{Z}_2 \oplus ... \oplus \boldsymbol{Z}_h)\boldsymbol{W}^Z \tag{4}$$

where $\boldsymbol{M}_i \in \mathcal{R}^{n \times d_m}$, $\boldsymbol{W}^Z \in \mathcal{R}^{d_m \times d_m}$ is a projection matrix, d_m is the dimensionality of each token in $\boldsymbol{M}_i$, $\oplus$ is the concatenation procedure, and h is the number of heads.

Let $\boldsymbol{R}_j$ denote the encoding of the specific relationship between the most recent memory representation $\boldsymbol{M}_{i-1}$ and $\boldsymbol{A}$ in head j. Then $\boldsymbol{Z}_j$ is:

$$\boldsymbol{Z}_j = softmax(\frac{\boldsymbol{R}_j}{\sqrt{d_z}})\boldsymbol{A}\boldsymbol{W}_j^{\boldsymbol{A}} \tag{5}$$

where $j \in [1, h]$, $\boldsymbol{Z}_j \in \mathcal{R}^{n \times d_z}$, $\boldsymbol{W}_j^{\boldsymbol{A}} \in \mathcal{R}^{d_h \times d_z}$ is a projection matrix, $\frac{1}{\sqrt{d_z}}$ is the scaling factor (cf. Figure 2), and $d_z = \frac{d_m}{h}$.

$\boldsymbol{R}_j$ is the dot-product between the projection of $\boldsymbol{M}_{i-1}$ in head j and the transpose of the projection of $\boldsymbol{A}$ in head j, which is a key step for capturing the correlation in the alternating updates, given by the following equation:

$$\boldsymbol{R}_j = \boldsymbol{M}_{i-1}\boldsymbol{W}_j^{\boldsymbol{M}}(\boldsymbol{A}\boldsymbol{W}_j^{\boldsymbol{A}'})^\top \tag{6}$$

where $\boldsymbol{R}_j \in \mathcal{R}^{n \times n}$, $\boldsymbol{W}_j^{\boldsymbol{M}} \in \mathcal{R}^{d_m \times d_z}$ and $\boldsymbol{W}_j^{\boldsymbol{A}'} \in \mathcal{R}^{d_h \times d_z}$ are projection matrices.

3.3 Position-aware Memory

To exploit the position patterns in academic homepages, we further learn a position-aware memory. Specifically, we integrate the global position representation $\boldsymbol{U}_g$ and the local position representation $\boldsymbol{U}_l$ into the memory updating process by extending Equation (6) :

$$\boldsymbol{R}_j = \boldsymbol{M}_{i-1}\boldsymbol{W}_j^{\boldsymbol{M}}(\boldsymbol{A}\boldsymbol{W}_j^{\boldsymbol{A}} + \boldsymbol{U}_l)^\top + f(i)\boldsymbol{U}_g \tag{7}$$

3.3.1 Global Position. We observe that the contents of a block run across consecutive lines. We utilise such a position pattern and model the attention distribution of all the lines in a homepage in $\boldsymbol{U}_g$ when recognising publications, i.e., the lines around the median line of a publication block should have more attention, while the lines far from the median line should have less. The attention distribution is assumed to follow a normal distribution.

Let $\boldsymbol{G}_t$ denote the attention distribution for token t. Then $\boldsymbol{U}_g$ is the concatenation of all such distributions:

$$\boldsymbol{U}_g = \boldsymbol{G}_0 \oplus \boldsymbol{G}_1 \oplus ... \oplus \boldsymbol{G}_n \tag{8}$$

where $\boldsymbol{U}_g \in \mathcal{R}^{n \times n}$.

Let l_μ denote the median line of the predicted publication block $\boldsymbol{Q}$ and σ denote half of the total number of lines in $\boldsymbol{Q}$ (cf. Figure 3).

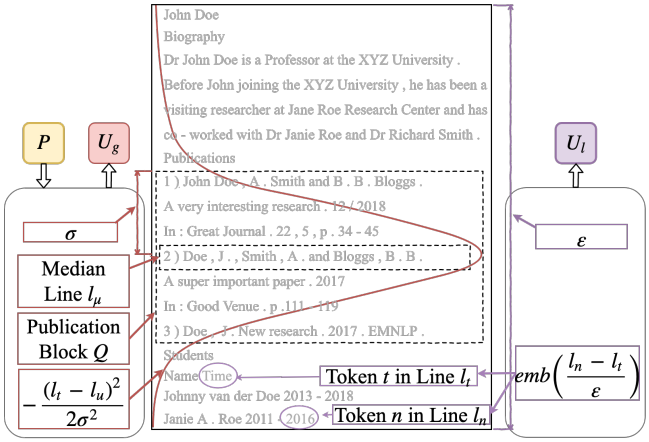

Figure 3: Example of the computation of U_g and U_l.

Then we have:

$$\boldsymbol{G}_t = [-\frac{(l_1 - l_\mu)^2}{2\sigma^2}, ..., -\frac{(l_n - l_\mu)^2}{2\sigma^2}] \tag{9}$$

where $\boldsymbol{G}_t \in \mathcal{R}^n$ and details for optimizing l_μ are in Section 3.4.

Let k denote the number of predicted publication tokens in a line, $\bar{k}$ denote the mean of all the k values in a homepage, then the predicted publication block $\boldsymbol{Q}$ is the set with the most consecutive lines that contains more predicted publication tokens than $\bar{k}$.

Let V denote the number of tokens in line l and $\boldsymbol{T}_v^l$ denote the prediction for token v in l, then the number of predicted publication tokens in l, denoted by k_l, is:

$$k_l = \sum_{v=1}^{V} \boldsymbol{T}_v^l \tag{10}$$

where $\boldsymbol{T}_v^l$ is computed by:

$$\boldsymbol{T}_v^l = \begin{cases} 0, & top(\boldsymbol{B}_v^l) \notin pub \\ 1, & top(\boldsymbol{B}_v^l) \in pub \end{cases} \tag{11}$$

$$\boldsymbol{B} = softmax(\boldsymbol{P}\boldsymbol{W}^B + \boldsymbol{b}^B) \tag{12}$$

Here, $\boldsymbol{W}^P \in \mathcal{R}^{n \times d_P}$, $\boldsymbol{b}^P \in \mathcal{R}^{d_P}$, d_P is the number of token labels (e.g., BIO) in the publication recognition task, and $\boldsymbol{B}$ is the probability distribution of the possible token labels for all the tokens based on $\boldsymbol{P}$. Function $top()$ finds the label with the largest posibility for each token: $top(\boldsymbol{B}_v^l) \in pub$ means that the found label is a publication label while $top(\boldsymbol{B}_v^l) \notin pub$ means otherwise.

3.3.2 Local Position. $\boldsymbol{U}_l$ captures the difference in line numbers between tokens. Let $\boldsymbol{L}_t$ denote the embedding of the relative line distances between token t and every other token in the homepage, then $\boldsymbol{U}_l$ is the concatenation of all such embeddings:

$$\boldsymbol{U}_l = \boldsymbol{L}_0 \oplus \boldsymbol{L}_1 \oplus ... \oplus \boldsymbol{L}_n \tag{13}$$

where $\boldsymbol{U}_l \in \mathcal{R}^{n \times n \times d_z}$.

Let l_t denote the line number of token t, ε denote the total number of lines in the homepage (cf. Figure 3), emb denote a function that yields a d_z-dimension embedding, then $\boldsymbol{L}_t$ is:

$$\boldsymbol{L}_t = emb([\frac{l_1 - l_t}{\varepsilon}, ..., \frac{l_n - l_t}{\varepsilon}]) \tag{14}$$

where $t \in [1, n]$, $\boldsymbol{L}_t \in \mathcal{R}^{n \times d_z}$ and emb is applied to reduce the space complexity when computing $\boldsymbol{U}_l$ in multi-head attention [21].

3.4 Joint Recognition

The improved memory representations from PAM is used to produce our final output by:

$$\hat{\boldsymbol{N}} = softmax(\boldsymbol{M}_c \boldsymbol{W}^N + \boldsymbol{b}^N) \tag{15}$$

$$\hat{\boldsymbol{P}} = softmax(\boldsymbol{M}_{c-1} \boldsymbol{W}^P + \boldsymbol{b}^P) \tag{16}$$

where $\boldsymbol{W}^N \in \mathcal{R}^{d_m \times d_N}, \boldsymbol{b}^N \in \mathcal{R}^{d_N}, \boldsymbol{W}^P \in \mathcal{R}^{d_m \times d_P}, \boldsymbol{b}^P \in \mathcal{R}^{d_P}$, d_N is the number of token labels (e.g., BIO) in person name recognition task and c is the final hop. $\hat{\boldsymbol{N}}$ and $\hat{\boldsymbol{P}}$ contain the learned probability distributions of the labels for all the tokens. For each token in each task, the label with the highest posibillity is our final output.

Our model is trained by minimising the following loss function:

$$\mathcal{L} = \mathcal{L}_{\boldsymbol{N}} + \mathcal{L}_{\boldsymbol{P}} + \lambda \mathcal{L}_{\boldsymbol{D}} \tag{17}$$

where $\mathcal{L}_{\boldsymbol{N}}$ and $\mathcal{L}_{\boldsymbol{P}}$ are the loss for the person name recognition task and the publication recognition task, respectively, and $\mathcal{L}_{\boldsymbol{D}}$ is added to optimise l_μ with λ as the weight. Specifically:

$$\mathcal{L}_{\boldsymbol{N}} = -\frac{1}{n} \sum_{i=1}^{n} \tilde{\boldsymbol{N}}_i \log(\hat{\boldsymbol{N}}_i) \tag{18}$$

$$\mathcal{L}_{\boldsymbol{P}} = -\frac{1}{n} \sum_{i=1}^{n} \tilde{\boldsymbol{P}}_i \log(\hat{\boldsymbol{P}}_i) \tag{19}$$

$$\mathcal{L}_{\boldsymbol{D}} = \frac{\|\tilde{l} - l_\mu\|}{\varepsilon} \tag{20}$$

where $\tilde{\boldsymbol{N}}$ and $\tilde{\boldsymbol{P}}$ are the ground turth for the two tasks and $\mathcal{L}_{\boldsymbol{N}}$ and $\mathcal{L}_{\boldsymbol{P}}$ are the average of cross-entropies for the two tasks, respectively. $\|\tilde{l} - l_\mu\|$ is the distance between the ground truth median line $\tilde{l}$ in the publication block and the predicted median line l_μ. $\tilde{l}$ is computed in the same way as l_μ, except that k_l is based on the ground truth $\tilde{\boldsymbol{P}}$.

4 EXPERIMENTS

We report experimental results in this section. We start with the experimental setup (Section 4.1). To evaluate the effectiveness of our proposed model, we compare it with state-of-the-art models that solve the two task separately (Section 4.2). We also compare our model with other naive joint learning models (Section 4.3.1). To evaluate the effectiveness of the architectural choices of our model, we perform an ablation study and compare our model with variants of our models (Section 4.3.2). To better understand how the model works, we also show and analyse the visualisation result (Section 4.4) and conduct an error analysis (Section 4.5).

4.1 Experimental Setup

4.1.1 Dataset and Preprocessing. We use the same datasets used by the state-of-the-art for publication recognition [31] and person name recognition [1]. Table 2 summarises the dataset statistics.

- **HomePub** dataset [31] contains the plain text of 2,087 homepages from different universities and research institutes with 12,796 publications annotated.
- **HomeName** dataset [1] is constructed from the HomePub dataset by further labeling the person names. All the 70,864 person names are annotated with fine-grained forms such as whether a name is a first or last name. We keep only the annotation for names.

Table 2: Statistics of the dataset used in experiments.

Summary of Dataset	
# homepages	2,087
# homepages for training	1342
# homepages for development	335
# homepages for testing	410
# publications	12,796
# homepages containing publications	702
Avg. # publications per page	18.23
Std. # publications per page	36.15
# person names	70,864
# homepages containing names	2087
Avg. # names per page	34
Std. # names per page	133
# pages with names only	1385
# pages with names and pubs	702
# names in pages with names only	9,490
# names in pages with names and pubs	61,372

We focus on English webpages and convert any text in Unicode to ASCII using Unidecode[1]. The text is tokenised on whitespace, newline characters, and punctuations. Every punctuation and newline character is considered as a single token. Standard BIO tagging scheme is adopted. Word embeddings are initialised with 100 dimensional GloVe [19] vectors trained on the tokenised dataset.

4.1.2 Evaluation Metric. We measure precision (P), recall (R), and F1-score ($F1$). For person name recognition, we report the *Token Level* performance, which reflects the model capability to recognise each person name token. We also report the *Name Level* performance, in which the model needs to recognise a whole person name without missing any token. For example, for the name 'John Doe', recognising either 'John' or 'Doe' is a true positive at token level, while only recognising 'John Doe' in full is a true positive at name level. For publication recognition, we report the *String Level* performance, in which the model needs to recognise a whole publication without missing any token.

4.1.3 Model Implementation . We implement our PAM model in TensorFlow[2] and train it on an NVIDIA GTX1080 GPU. The model is trained with the Adam optimiser [11] with the learning rates tuned among {**0.01**, 0.005, 0.001}. The batch size is tuned among {**32**, 64, 128} and the maximum sequence length is tuned among {100, **200**, 500}. The dimensions of the encoders are tuned among {**100**, 200} and the dropout rate is set to 0.5. The number of hops is tuned among {2, **4**, 6, 8}, the number of attention heads is tuned among {2, **5**, 8} and λ is tuned among {1, 10, **15**}. We use a development set to select the best hyperparameters through grid search and the optimal hyperparameters are highlighted above in bold. The model is trained for a maximum of 20 epochs with early stopping if the performance on the development set does not improve after 3 epochs.

[1] https://pypi.org/project/Unidecode/
[2] https://www.tensorflow.org/

4.1.4 Baselines. We compare our proposed model with the following single-task models for publication recognition:

- **ParsCit** [4] is an open-source package[3] for parsing publications based on feature engineering and CRF.
- **CNN-Sentence** [10] is used to classify whether each line in a webpage is a publication. It has filter windows with sizes of 3, 4, 5 and 100 feature maps for each, a dropout rate of 0.5, a batch size of 40 and a learning rate of 0.01.
- **Bi-LSTM-CNN-CRF** [14] has a filter window size of 3 with 30 feature maps, a hidden dimension of 100, a dropout rate of 0.5, a batch size of 40 and a learning rate of 0.01.
- **PubSE** [31] is the state-of-the-art for publication recognition based on Bi-LSTM-CNN-CRF. It has a filter window size of 3 with 30 feature maps, a hidden dimension of 100, a learning rate of 0.01 and a dropout rate of 0.5 for both the line-level model and the webpage-level model. The batch size is 40 for the line-level model and is 1 for the webpage-level model. The coefficients in the loss function are 1, 0.05, 1, 0.3.

We compare with the following single-task models for person name recognition:

- **Stanford-NER** [5] is a named entity recognisor based on CRF provided by the Stanford NLP Group[4].
- **Bi-LSTM-CRF** [9] has a hidden dimension of 100, dropout rate of 0.5, batch size of 32 and an initial learning rate of 0.01 with a decay rate of 0.05.
- **CogNN** [1] uses fine-grained name form annotations through co-attention. This is the state-of-the-art model for person name recognition. It has a hidden dimension of 100, dropout rate of 0.5, batch size of 32 and an initial learning rate of 0.01 with a decay rate of 0.05.

All the above models are trained on the same training dataset. The parameters are selected using the same development set with the optimisers and early stopping mechanisms reported in the corresponding papers. If these are not reported, we use the Adam optimiser [11] to train the model for a maximum of 20 epochs with early stopping if the performance on the development set does not improve after 3 epochs.

4.1.5 Variants. Since there are no existing models that jointly recognise person names and publications, we compare with the following variants of our proposed model:

- **Joint-Naive** is resulted from removing the AM and PM modules, in which the outputs of the encoders are fed into the network's final output layers directly, and training the two tasks directly by minimising the total loss.
- **Joint-Concat** is resulted from replacing the AM and PM modules with a concatenation procedure similar to Ma et al. [13] and Hashimoto et al. [6]. This model has a pipeline architecture for two jointly trained tasks. In the N → P direction, the output of the name encoder and the publication encoder is concatenated to be the input of the the publication predictor. In the P → N direction, the output of the publication encoder and the name encoder are concatenated to be the input of the name predictor.
- **Joint-Gate** is resulted from replacing the AM and PM modules with the gating function in DMN [12]. This model has a pipeline architecture for two jointly trained tasks. In the N → P direction, the output of the name encoder and the publication encoder are summed up by the gates for several hops to be the input of the publication predictor. And the gates are learned based on the output of the name encoder and the publication encoder. In the P → N direction, the output of the publication encoder and the name encoder are summed up by the gates for several hops to be the input of the name predictor. And the gates are learned based on the output of the publication encoder and the name encoder.
- **Joint-Att** is resulted from replacing the AM and PM modules with multi-layer multi-head attention [26]. This model has a pipeline architecture for two jointly trained tasks. In the N → P direction, the attention is computed using the output of the name encoder and the publication encoder, then the output of the publication encoder is weighted by the attention for several hops before feeding into the publication predictor. In the P → N direction, the attention is computed using the output of the publication encoder and the name encoder, then the output of the name encoder is weighted by the attention for several hops before feeding into the name predictor.
- **Joint-Stack** is resulted from replacing the AM and PM modules with stacked two groups of multi-layer multi-head attention, each for one task. The attention is computed using the output of the name encoder and the initial word representation, then the output of the name encoder is weighted by the attention for several hops before feeding into the name predictor. After that, the attention is computed using the output of the publication encoder and the updated output from name encoder, then the output of the publication encoder is weighted by the attention for several hops before feeding into the publication predictor.
- **AM** is resulted from removing the PM module from our proposed model.

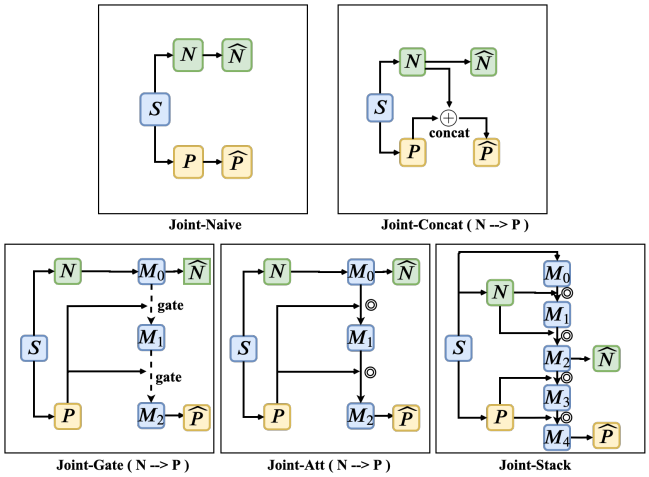

Figure 4: Illustration of Different Joint Models.

[3] http://parscit.comp.nus.edu.sg/
[4] https://nlp.stanford.edu/software/CRF-NER.html

Table 3: Experimental results on single-task models for publication and person name recognition.

Model	Pub (String Level)			Model	Name (Token Level)			Name (Name Level)		
	R	P	F1		R	P	F1	R	P	F1
ParsCit [4]	70.34	18.22	28.94	–	–	–	–	–	–	–
CNN-sentence [10]	73.39	76.69	75.00	Stanford-NER [5]	64.94	94.68	77.04	41.31	40.98	41.15
Bi-LSTM-CNN-CRF [14]	74.15	77.22	75.65	Bi-LSTM-CRF [9]	87.97	89.64	88.79	79.48	82.34	80.89
PubSE [31]	84.12	91.12	87.48	CogNN [1]	93.06	92.85	92.95	86.40	85.32	85.85
PAM (proposed)	**89.02**	**93.34**	**91.12**	PAM (proposed)	**95.51**	**93.21**	**94.35**	**88.40**	**87.42**	**87.91**

- **Local-AM** is resulted from removing the global position representation from our proposed model.
- **Global-AM** is resulted from removing the local position representation from our proposed model.

Figure 4 illustrates the architectures of different joint models. All the above models are trained jointly by minimising the total loss of the two tasks on the NVIDIA GTX1080 GPU with a batch size of 32, a dropout rate of 0.5, a maximum sequence length of 200, encoder dimensions of 100 and a learning rate of 0.01. Joint-Gate and Joint-Att have hops of 2 and Joint-Stack has hops of 2 in each group. Joint-Att and Joint-Stack have attention heads of 5. All the models are trained with the Adam optimiser [11] and the hyperparameters are selected using the same development set reported in Section 4.1.1. The models are trained for a maximum of 20 epochs with early stopping if the performance on the development set does not improve after 3 epochs.

4.2 Comparison with the State-of-the-Art

Table 3 reports the performance comparison result with the single-task models. Overall, our PAM model outperforms the state-of-the-art models on publication recognition and person name recognition by considerable margins and the improvements mainly lie in the recall. The improvements are statistically significant, with $p < 0.05$ based on McNemar's test.

4.2.1 Publication Recognition. The advantage of PAM over neural baselines such as CNN-sentence [10] and Bi-LSTM-CNN-CRF [14] is over 15.47% in terms of F1 score since CNN-sentence and Bi-LSTM-CNN-CRF can hardly handle complex homepages without the extra information about the position patterns and person names. PAM also outperforms the hierarchical PubSE [31] model, which can capture the positional diversity, by 3.64% in F1 score. The advantage of our model is more significant in recall than in precision. This may be explained by our use of the global position representation, which helps yield higher attention to the publications on a publication block and helps capture more publications. PubSE may miss some publications since the block information are not captured well in their models.

4.2.2 Person Name Recognition. Our proposed PAM model outperforms the baselines that use standard NER models, such as Stanford NER [5] and Bi-LSTM-CRF [9], by at least 5.56% on token level and 7.09% on name level in F1 score. Our improvements mainly lie in the recall, which is consistent with the observation on the publication recognition task. This indicates that our model has better capability to cover more person names with the knowledge from the publication recognition task. PAM also outperforms CogNN [1] by 1.40% on token level and 2.06% on name level in F1 score. Note that CogNN relies on extra labelling information such as whether the tokens are first names or family names, while our model does not have this requirement.

4.3 Ablation Study

Table 4 reports the results where we compare our PAM model with other joint models and variants of PAM. Overall, PAM outperforms other joint models. AM makes effective improvements to the overall model performance, and both global and local position representations in PM contribute substantially to the model performance, especially for the publication recognition task. The improvements achieved by both PM and AM modules are statistically significant, with $p < 0.05$ based on McNemar's test.

4.3.1 Model Performance without AM. The models with a name prefix of 'Joint-' (i.e., Joint-Naive, Joint-Concat, Joint-Gate, Joint-Att, Joint-Stack) do not contain AM module. We can see from Table 4 that they perform worse than AM, with an up to 46.3% drop in F1 score on the string level for publication recognition and an up to 6.7% drop in F1 score on the name level for person name recognition. Models having more architecture similarity with AM achieve better result than others, i.e., Joint-Att and Joint-Stack perform better than others. Joint-Concat tends to introduce noise from one task into the other task, which leads to worser results than Joint-Naive. Joint-Gate tends to use the original information in the corresponding task and discard the new information from the other task, which leads to similar results as Joint-Naive. We also observe that with higher frequency we alternatingly updated the representations, we achieve better results, i.e., AM performs better than Joint-Stack. The reason is that the correlation between person names and publications can be better captured in alternating updates.

4.3.2 Model Performance without PM. The AM, Local-AM and Global-AM models do not contain complete PM module. We can see from Table 4 that they perform much worse than the full PAM model, with an up to 13.4% drop in F1 score on the string level for publication recognition and an up to 5.4% drop in F1 score on the name level for person name recognition. This indicates that both global and local position representations are critical to the performance of PAM, i.e., removing either or both would result in a drop in performance. Global position is more important than local position, i.e., performance drops more when the global position representation is removed. We also note that PM is more important for the publication recognition task than the person name recognition task, i.e., performance for publication recognition drops more when PM is removed. This is expected as publications have stronger position patterns than person names.

Table 4: Experimental results of joint models and variants of PAM for publication and person name recognition. F1-score is reported. Models with * have pipeline architectures for two jointly trained tasks and the reported results for each task are from the corresponding pipeline directions, i.e., the result for publication is from N → P and vice versa.

Model	Pub		Name	
	Token	String	Token	Name
Joint-Naive	88.5	33.0	88.9	77.5
Joint-Concat*	89.5	31.3	89.4	75.8
Joint-Gate*	88.8	33.8	89.1	77.7
Joint-Att*	93.6	65.4	90.9	81.5
Joint-Stack	94.8	73.1	90.6	82.2
AM	95.2	77.4	91.4	82.5
Local-AM	96.3	84.8	92.1	84.1
Global-AM	96.7	88.7	92.6	85.7
PAM	**97.2**	**91.1**	**94.3**	**87.9**

4.4 Visualisation

We visualise the attention weights on an academic homepage to better examine and understand how the memory module works. Figure 5 shows the attention heatmaps with corresponding tokens in different hops of the memory; tokens with higher attention are in darker colour.

The attention is generally more focused in fewer tokens as more updating hops have been run. For example, Hop 3 and Hop 4 have higher attention weights on the intended recognition targets (names and publications) than Hop 1 and Hop 2; meanwhile, Hop 3 and Hop 4 have much lower weights on other tokens than Hop 1 and Hop 2 have. We also note that the alternatingly updating mechanism can shift and correct attention weights to the intended level. For example, in Hop 1 (the first name round), we observe that the attention focuses on *Mag . Wu Shengqian*, while *Mag* is an academic degree and should not be recognised as a name. In Hop 3 (the second name round), *Wu Shengqian* gains more attention while Mag gets less attention, so they can be recognised correctly. Similarly, in Hop 2 (the first publication round), *Evaluation of the chondrocyte phenotype in health, disease and therapy* has high attention,but actually it relates to the researcher's research interest and should not be recognised as a publication. In Hop 4 (the second publication round), this string gains lower attention and can be discarded correctly.

4.5 Error Analysis

We perform a manual inspection of recognition results of state-of-the-art models for the two tasks (i.e., PubSE [31] and CogNN [1]) and our proposed PAM model on 50 randomly selected homepages. We focus on string level performance for the publication recognition task and on name level performance for the person name recognition task. In total, we inspect 1,137 publications and 5,542 person names.

For publication string recognition, we observe that PubSE [31] misrecognises strings about patents, grants, and research projects as publications. PAM avoids these errors since it can capture publication block information and these strings are usually listed on other blocks. Both PubSE and PAM make mistakes when publications are listed together with invited talks or presentations. These strings have high similarity to the publications and are difficult to distinguish when they are listed together. For example, the string *Kelly Schrum. "Teaching Hidden History: Creating An Effective Hybrid Graduate Course" Conference on Higher Education Pedagogy, Virginia Tech (Feb 2016)* is a talk given by the page owenr but not a publication.

For person name recognition, we observe that CogNN [1] tends to produce false negative predictions in groups, i.e., a series of person names in a publication string cannot be recognised. PAM does not make such mistakes since it captures the correlation between names and publications. However, PAM may misrecognise person names with complex name format while CogNN is better on those cases since CogNN uses many fine-grained name form annotations. For example, the string *Tyler, L.M.K. Mellor, D. Hauser, KD.* contain three person names, which are marked underlined, while PAM cannot distinguish them. Both CogNN and PAM may not recognise some person names hidden in a long paraghraphs, such as a long biography section. We conjecture that this may be caused by the LSTM-based encoder. We aim to solve this problem in future work.

5 CONCLUSION

Based on the observations that there is correlation between person names and publications and that there are position patterns in academic homepages, we propose to jointly recognise person names and publications while taking into account the important feature of position patterns. We proposed a Position-aware Alternating Memory network. The network has an alternatingly updated memory module to exploit the correlation of two tasks, and a position-aware memory module to exploit global and local position information. Empirical results show that our model outperforms the state-of-the-art publication recognition model by 3.64% in F1 score and outperforms the state-of-the-art person name recognition model by 2.06% in F1 score. Our model also outperforms naive joint models by up to 59.80% and 12.10% in F1 score for publication and person name recognition, respectively. Ablation studies and visualisation confirm the effectiveness of the proposed modules in our model.

Our proposed way of modelling interdependency may be applied to other tasks which are inherently correlated, such as entity recognition and relation extraction. Our framework may also be applied to IE tasks on other datasets which have strong position patterns, such as resumes and other webpages.

ACKNOWLEDGMENTS

This work is supported by Australian Research Council Discovery Project DP180102050 and by the China Scholarship Council.

REFERENCES

[1] Anonymous. 2018. A Co-guided Neural Network for Person Name Recognition in Academic Homepages. *OpenReview Preprint* (2018). https://openreview.net/pdf?id=H1eBBGbdnN

[2] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *EMNLP*. 1724–1734.

[3] Jen-Ming Chung, Ya-Huei Lin, Hahn-Ming Lee, and Jan-Ming Ho. 2012. Mining publication records on personal publication web pages based on conditional random fields. In *WI-IAT*. 319–326.

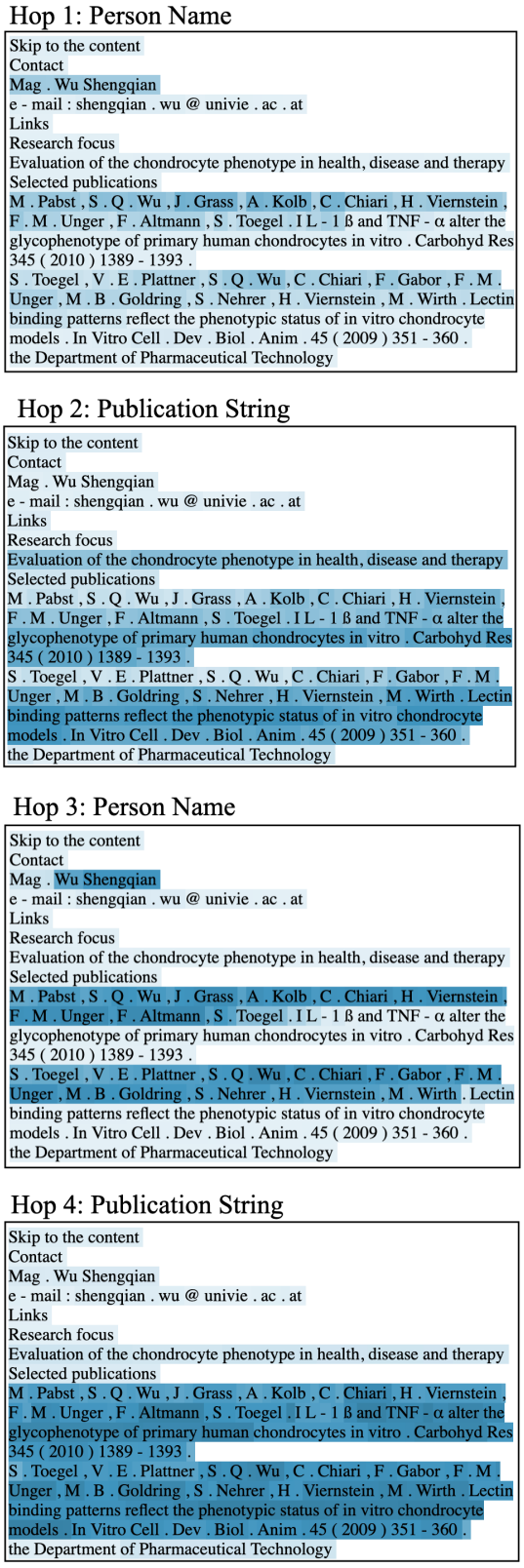

Figure 5: Visualisation of attention weights along with corresponding tokens in Alternatingly Updated Memory. The colour scale represents the strength of the attention weights. Each box represents the attention weights in a memory updating hop.

[4] Isaac G Councill, C Lee Giles, and Min-Yen Kan. 2008. ParsCit: an Open-source CRF Reference String Parsing Package.. In *LREC*. 661–667.

[5] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*. 363–370.

[6] Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al. 2017. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. In *EMNLP*. 1923–1933.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[8] Ching Hoi Andy Hong, Jesse Prabawa Gozali, and Min-Yen Kan. 2009. FireCite: Lightweight real-time reference string extraction from webpages. In *NLPIR4DL*. 71–79.

[9] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[10] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.

[11] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[12] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*. 1378–1387.

[13] Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint Learning for Targeted Sentiment Analysis. In *EMNLP*. 4737–4742.

[14] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *ACL*. 1064–1074.

[15] Simone Marinai. 2009. Metadata extraction from PDF papers for digital library ingest. In *ICDAR*. 251–255.

[16] Alexander G Ororbia II, Jian Wu, Madian Khabsa, Kyle Williams, and Clyde Lee Giles. 2015. Big scholarly data in CiteSeerX: Information extraction from the web. In *WWW*. 597–602.

[17] Juan Pavez, Hector Allende, and Hector Allende-Cid. 2018. Working Memory Networks: Augmenting Memory Networks with a Relational Reasoning Module. In *ACL*. 1000–1009.

[18] Fuchun Peng and Andrew McCallum. 2004. Accurate Information Extraction from Research Papers using Conditional Random Fields. In *HLT-NAACL*. 329–336.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.

[20] Vassilis Plachouras, Matthieu Rivière, and Michalis Vazirgiannis. 2012. Named entity recognition and identification for finding the owner of a home page. In *PAKDD*. 554–565.

[21] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *NAACL*. 464–468.

[22] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *NeurIPS*. 2440–2448.

[23] Jie Tang, Alvis CM Fong, Bo Wang, and Jing Zhang. 2011. A unified probabilistic framework for name disambiguation in digital library. *IEEE TKDE* 24, 6 (2011), 975–987.

[24] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *KDD*. 990–998.

[25] Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. 2015. CERMINE: automatic extraction of structured metadata from scientific literature. *IJDAR* 18, 4 (2015), 317–335.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.

[27] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* (2014).

[28] Zhaohui Wu, Jian Wu, Madian Khabsa, Kyle Williams, Hung-Hsuan Chen, Wenyi Huang, Suppawong Tuarob, Sagnik Ray Choudhury, Alexander Ororbia, Prasenjit Mitra, et al. 2014. Towards building a scholarly big data platform: Challenges, lessons and opportunities. In *JCDL*. 117–126.

[29] Baosong Yang, Zhaopeng Tu, Derek F Wong, Fandong Meng, Lidia S Chao, and Tong Zhang. 2018. Modeling Localness for Self-Attention Networks. In *EMNLP*. 4449–4458.

[30] Kai-Hsiang Yang and Jan-Ming Ho. 2010. Parsing publication lists on the web. In *2010 IEEE/WIC/ACM WI-IAT*, Vol. 1. 444–447.

[31] Yiqing Zhang, Jianzhong Qi, Rui Zhang, and Chuandong Yin. 2018. PubSE: A Hierarchical Model for Publication Extraction from Academic Homepages. In *EMNLP*. 1005–1010.

[32] Qi Zhu, Xiang Ren, Jingbo Shang, Yu Zhang, Ahmed El-Kishky, and Jiawei Han. 2019. Integrating local context and global cohesiveness for open information extraction. In *WSDM*. 42–50.