A Co-guided Neural Network for Person Name Recognition in Academic Homepages

Anonymous submission

Abstract

Academic homepages are important channels for learning researchers' profiles. Knowing the person names in academic homepages is essential to the extraction of other entities such as contacts, publications, and biography. Traditional NER models are trained on newswire corpora such as CoNLL-2003, which contain well-formed names in consistent and complete syntax. However, academic homepages often contain text with incomplete syntax and names of various forms. Few studies address person name recognition in this context. To fill this gap, we start with proposing a fine-grained name annotation scheme. This scheme further labels detailed name forms including first, middle, or last names, and name words in full or name initials. We then propose a *Co-guided Neural Network* (CogNN) model to learn from homepages labelled with our fine-grained annotations. CogNN uses co-attention mechanisms to co-guide two jointly trained neural networks, each focusing on different dimensions of the name forms. It thus takes full advantage of our annotation scheme and can accurately recognise person names in academic homepages. Experimental results on real datasets show that CogNN significantly outperforms state-of-the-art NER models in extracting person names from academic homepages, while achieving comparable performance on a traditional NER dataset.

1 Introduction

Academic homepages are an important source for learning researchers' profiles, including names, contact details, bibliography, working experience, and publications. Among these, person names are basic and yet important. Knowing the person names is essential to extracting other entities in academic homepages and also provides valuable insights on researcher collaboration networks.

We focus on extracting person names from academic homepages using detailed name form information, such as whether a token is a first, middle, or last name, and whether the token is a full name word or a name initial. Figure 1 shows an example of person name recognition in academic homepages. Given the text content of an academic homepage, we aim to recognise all person names as highlighted in the example.

The challenges of recognising person names in academic homepages lie in the diversity of text and name forms. The text in academic homepages is free-form and may have incomplete syntax. For example, in Figure 1, the biography section consists of complete sentences while the students section is simply a table. The person names may be in different forms as well. Figure 1 contains well-formed full name of the researcher '`John Doe`' in the page header and abbreviated names in the publications section. Further, the abbreviated names may have different abbreviation forms, e.g., '`B.B. Doe`' vs. '`Doe, J.`'.

Recent NER models (Huang et al., 2015; Chiu and Nichols, 2016; Ma and Hovy, 2016) are trained on well-formed texts such as news articles. These models do not solve our problem since news articles and academic homepages have substantially different underlying text distributions. Figure 2 shows an example, where Stanford Named Entity Tagger is applied to recognise the person names from a publication string in an academic homepage. The tokens in italics are name tokens, while the tokens in bold are those recognised as name tokens. We can see that not all the names have been recognised.

To address person name recognition in academic homepages, we propose a fine-grained name annotation scheme that annotates detailed name forms including first, middle, or last name, and a full name word or a name initial (cf. Fig-

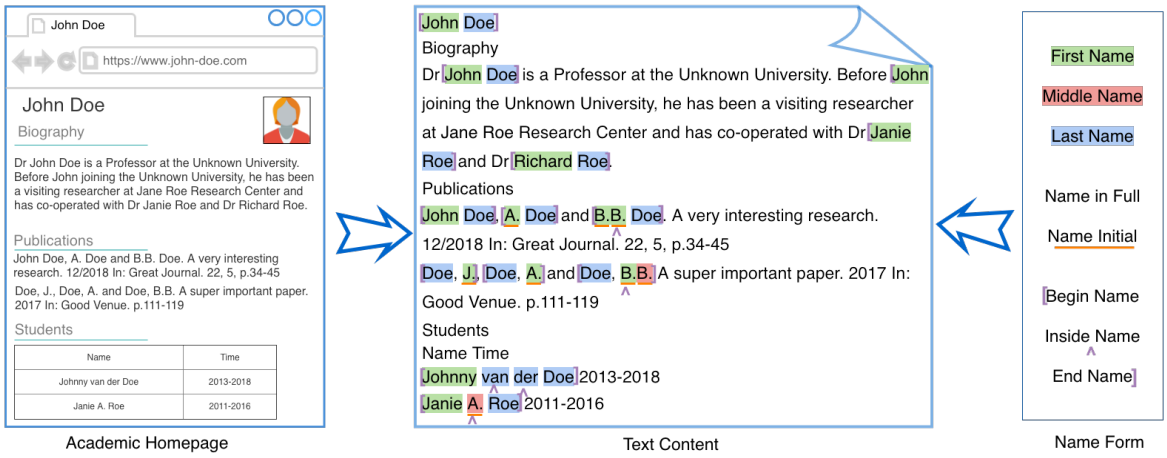

Figure 1: An example of person name recognition in academic homepages (best view in color).

Kime *C* , *Sakaki-Yumoto M* , ***Goodrich*** *L* , ***Hayashi*** *Y* , ***Sami S*** , ***Derynck R*** , *Asahi M* , *Panning B* , ***Yamanaka*** *S* , ***Tomoda*** *K*

Figure 2: An example of applying Stanford Named Entity Tagger on the text from an academic homepage. The bold tokens are recognised as names.

ure 1). Such detailed annotations offer richer training signals to NER models to learn the patterns of person names in free-form text. However, detailed annotations also bring challenges because more label classes need to be learned.

To take advantage of the detailed name annotations and to address their challenges, we propose a *Co-guided Neural Network* (CogNN) model for person name recognition. CogNN consists of two sub-neural networks (Bi-LSTM-CRF variants). One of the sub-network focuses on predicting whether a token is a name token, while the other focuses on predicting the name form class of the token. We add a co-attention layer to connect the two Bi-LSTM-CRF based sub-networks. This way, the two sub-networks can share the learning signals to reinforce their label prediction confidence. The two sub-networks are trained simultaneously by minimising their total loss.

This paper makes the following contributions:

• *New angle*: We study person name recognition in academic homepages, for which we propose a fine-grained annotation scheme that provides information on various forms of names.

• *New dataset*: We create a dataset of diverse academic homepages where the person names are fully annotated with detailed name forms. This dataset will be released upon paper publication.

• *New model*: We propose a *Co-guided Neural Network* (CogNN) model to recognise person names using the fine-grained annotations. It learns the different name form classes with two neural networks while fusing the learned signals through co-attention mechanism. Experimental results show that CogNN outperforms state-of-the-art NER models in the accuracy of extracting person names from academic homepages.

The rest of this paper is organised as follows. Section 2 summarises related studies. Section 3 describes our name annotation scheme and dataset. Section 4 detials the proposed model. Section 5 presents experimental results. Section 6 concludes the paper.

2 Related Work

Named entity recognition (NER) aims to identify proper names in text and classify them into different types, such as person, organisation, and location (Nadeau and Sekine, 2007). Neural NER models have shown excellent performance on long texts which follow strict syntactic rules, such as newswire and Wikipedia articles (Huang et al., 2015; Chiu and Nichols, 2016; Ma and Hovy, 2016). However, these NER models are less attractive when applied to short texts which may not have consistent and complete syntax (Li et al., 2015; Dugas and Nichols, 2016). Recent studies also consider user-generated short texts from social media platforms such as Twitter and Snapchat. Since social media texts are usually posted with images, researchers propose to make use of both textual and visual context to recognise the named entities (Lu et al., 2018; Moon et al., 2018). Such models are less relevant because academic homepages do not contain images consistently except for a photo of the page owner.

NER studies on academic homepages usually

treat the text content of a webpage as a document, upon which traditional NER techniques are applied. For example, Zhang et al. (2018) use a Bi-LSTM-CRF based hierarchical model to extract all the publication strings from a given academic homepage. Tang et al. (2010) assume that the page owner's full name is given and use a Tree-structured CRF model to extract multiple types of entities from a given academic homepage. This technique does not apply to our problem as we assume no pre-knowledge about the page owners.

Person names are often recognised together with other named entities, such as locations and organisations (Huang et al., 2015; Ma and Hovy, 2016; Chiu and Nichols, 2016). There are a few studies focusing only on person names. Dozier and Haschart (2000) extract the attorney and judge names in legal texts using a semantic parser. Packer et al.(2010) focus on extracting name from noisy OCR data, which may include spelling errors and incomplete texts. They combine rule based methods, the Maximum Entropy Markov Model, and the CRF model using a simple voting-based ensemble. Minkov et al. (2005) extract person names from emails using the CRF model. They design email specific structural features and exploit in-document repetition to improve the extraction accuracy. Shaalan and Raza (2007), Elsebai et al. (2009), Bidhend et al. (2012), and Aboaoga and Ab Aziz (2013) study person name recognition in Arabic. They use rule-based methods based on gazetteer name lists and regular expressions. To the best of our knowledge, no existing work has used detailed name forms for extracting person names.

3 Proposed Name Annotation Scheme

We first present our name annotation scheme with detailed name forms and our *HomeName* dataset that is annotated under this scheme.

Detailed name form annotations are done to better capture the person name form features in free-form texts. Both well-formed names written in full and various forms of abbreviated names may appear in academic homepages. Annotating the name tokens with detailed forms offers more direct training signals to NER models to learn the patterns of person names. This also allows an NER model to be trained with fewer data.

Thus, unlike traditional NER datasets, which only label a name token with a *PER* (person) label, we further provide detailed name form information for each name token. We label each name token using a three-dimensional annotation scheme:

- *BIE*: *Begin*, *Inside*, or *End* of name, indicating the position of a token in a person name,
- *FML*: *First*, *Middle*, or *Last* name, indicating whether a name token is used as the first, middle, or last name, and
- *FI*: *Full* or *Initial*, indicating whether a name token is a full name word or an initial.

Using the three-dimensional annotation scheme above, we can describe the detailed name form of a name token. For example, in Figure 1, '`John Doe`' can be labelled as *Begin_First_Full End_Last_Full*, while '`Johnny van der Doe`' can be labelled as *Begin_First_Full Inside_Last_Full Inside_Last_Full End_Last_Full*.

HomeName is a collection of academic homepages with person names fully annotated using the proposed annotation scheme. We construct our dataset based on the *HomePub* dataset[1]. The *HomePub* dataset contains 2,500 web pages from different universities and research institutes, among which 2,087 are academic homepages and 413 are non-academic homepages (such as staff directory pages). We keep only the academic homepages, which are from 286 institutes.

We annotate all the person names in the academic homepages using our proposed name annotation scheme. Each academic homepage is annotated by two annotators, with inter-annotator agreement measured at Cohens $\kappa = 0.63$ for names and $\kappa = 0.41$ for detailed name forms. Disagreement is resolved by a third annotator. We provide an annotation scheme, a semi-automatic annotation tool and a one-hour training to each annotator. More details on the annotation are given in the supplementary material.

The fine-grained annotations offer more direct training signals to NER models but also bring challenges because more label classes need to be learned. Next, we present our CogNN model that takes advantages of the detailed name form annotations to recognise person names.

4 Proposed Model

Given a sequence of input tokens $\boldsymbol{X}$, where $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}]$ and n is the length of the sequence, our aim is to predict for each token $\boldsymbol{x_i}$

[1] http://www.ruizhang.info/data/homepub.html

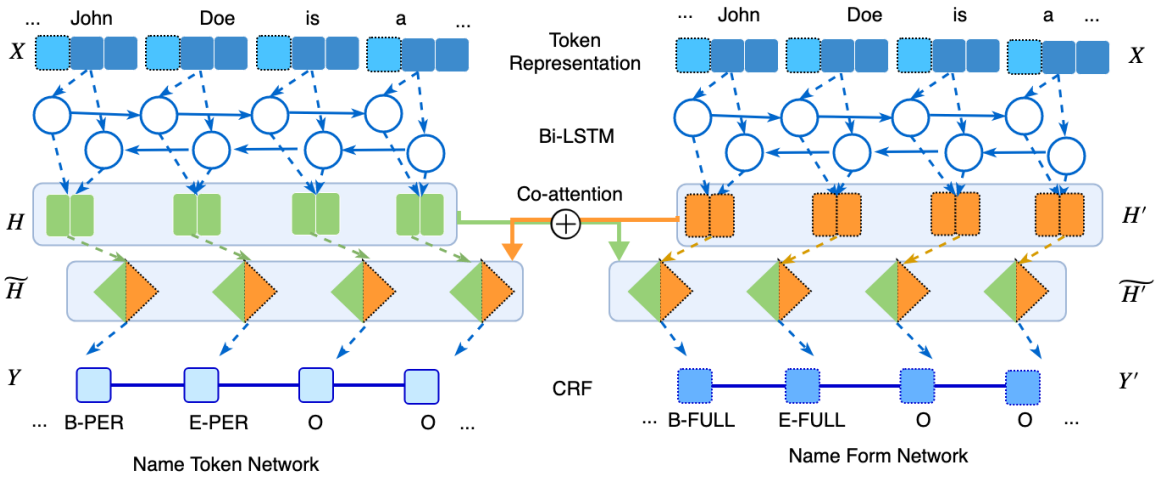

Figure 3: CogNN network structure.

whether it is a name token.[2]

Our proposed model CogNN achieves this aim with the help of two Bi-LSTM-CRF based sub-networks: the *name token network* and the *name form network*, as illustrated in Figure 3. The name token network focuses on predicting whether a token is part of a name (the *BIE* dimension), while the name form network focuses on predicting the detailed name form of the token (*FML* or *FI* dimensions). Co-attention layers are added to the Bi-LSTM-CRF based sub-networks, so that they can share the learning signals and reinforce the prediction confidence. The intuition here is that knowing whether a token is part of a name helps recognise the detailed name form class of the token, and vice versa. For example, if a token is not considered as part of a name, then even if it is a word initial, it should not be labelled as a name initial.

In particular, we start with concatenating the word embedding and a letter case vector for an input token to produce its representation. We feed the input under this representation into Bi-LSTM to learn its hidden representation matrix (Section 4.1). Then, we use co-attention mechanism to co-guide the two jointly trained sub-networks. Our co-attention mechanism is designed to share the training signals between the two sub-networks. It updates the importance of each token learned from the two sub-networks and records their correlations (Section 4.2). The two sub-networks are trained simultaneously by minimising their total loss (Section 4.4). Next, we detail each layer of our CogNN model.

[2]We use $\boldsymbol{x_i}$ to denote both a token and its embedding vector as long as the context is clear.

4.1 Capture: Hidden Feature Extraction

Both the name token network (denoted as N_Y) and the name form network (denoted as $N_{Y'}$) share the same layer structure. They only differ in the target labels $\boldsymbol{Y}$ and $\boldsymbol{Y'}$. Here, $\boldsymbol{Y}$ denotes the label sequence that records whether an input token is part of name, and $\boldsymbol{Y'}$ denotes the label sequence that records the form class of each input token. For simplicity, we focus on the name token network N_Y in the following discussion.

We start with concatenating the word embedding $\boldsymbol{e_i}$ and letter case vector $\boldsymbol{s_i}$ for an input token $\boldsymbol{x_i} \in \boldsymbol{X}$ to produce its vector representation. We use GloVe (Pennington et al., 2014) computed on our *HomeName* corpus for the word embeddings $\boldsymbol{e_i}$. The letter case vector $\boldsymbol{s_i}$ records the letter case information of $\boldsymbol{x_i}$, which is an important hint for recognising names. For example, the first letter of a name token is often in uppercase, and a name initial is often formed by an uppercase letter plus a dot. Our letter case vector is a three-dimensional binary vector where each dimension represents: (i) the first character in the token is in uppercase, (ii) all characters in the token are in uppercase, and (iii) any character in the token is in uppercase.

We then use Bi-LSTM (Dyer et al., 2015) to capture the hidden features from the input sequence. The output hidden representation, denoted as $\boldsymbol{h_i}$, summarises the context information of $\boldsymbol{x_i}$ in $\boldsymbol{X}$. Our hidden representation matrix $\boldsymbol{H}$ in N_Y can be written as $[\boldsymbol{h_1}, \boldsymbol{h_2}, ..., \boldsymbol{h_n}]$, where $\boldsymbol{h_i} \in \mathcal{R}^d$ and d is the number of dimensions of the hidden representation. Similarly, $\boldsymbol{H'}$ in $N_{Y'}$ can be written as $[\boldsymbol{h'_1}, \boldsymbol{h'_2}, ..., \boldsymbol{h'_n}]$.

4.2 Share: Co-attention Mechanism

Next, we share the learning signals in the hidden representation matrices $\boldsymbol{H}$ and $\boldsymbol{H'}$, and obtain new hidden representation matrices $\tilde{\boldsymbol{H}}$ and $\tilde{\boldsymbol{H'}}$ for the two sub-networks, respectively.

Note that training the two sub-networks separately is suboptimal, since the underlying correlation among the name label dimensions is lost. For example, a token recognised as `Inside` in N_Y is more possible to be `Middle` in $N_{Y'}$. To address this issue, we use co-attention to take the learning signals from two hidden representations into account by:

$$\boldsymbol{P} = tanh(W_h \boldsymbol{H} \oplus (W_{h'} \boldsymbol{H'} + b_{h'}))$$

where W_h and $W_{h'} \in \mathcal{R}^{k \times d}$ are trainable parameters, k is dimensionality of the parameters, $\oplus$ is the concatenating operation, and $P \in \mathcal{R}^{2k \times n}$.

A related technique is used by Yang et al. (2016). However, they only consider the token importance in a single hidden representation sequence of a document, while we consider the token importance in two hidden representation sequences simultaneously.

The co-attention distribution that records the importance of each token after examining two hidden representation sequences can be obtained as:

$$\boldsymbol{A} = softmax(W_p \boldsymbol{P} + b_p)$$

where $W_p \in \mathcal{R}^{1 \times 2k}$ are trainable parameters and $A \in \mathcal{R}^n$ is an importance weight matrix.

The new hidden representation $\tilde{\boldsymbol{h_i}}$ can be computed by:

$$\tilde{\boldsymbol{h_i}} = \sum \boldsymbol{a_i} \boldsymbol{h_i}, \boldsymbol{a_i} \in \boldsymbol{A}, \boldsymbol{h_i} \in \boldsymbol{H}$$

We thus obtain the new hidden representation sequences $\tilde{\boldsymbol{H}} = [\tilde{\boldsymbol{h_1}}, \tilde{\boldsymbol{h_2}}, ..., \tilde{\boldsymbol{h_n}}]$ and $\tilde{\boldsymbol{H'}} = [\tilde{\boldsymbol{h'_1}}, \tilde{\boldsymbol{h'_2}}, ..., \tilde{\boldsymbol{h'_n}}]$ for the two sub-networks.

4.3 Output Layer

The new hidden representation sequences $\tilde{\boldsymbol{H}} = [\tilde{\boldsymbol{h_1}}, \tilde{\boldsymbol{h_2}}, ..., \tilde{\boldsymbol{h_n}}]$ is trained to produce a label sequence $\boldsymbol{Y}$. To enforce the structural correlations between labels, $\boldsymbol{Y}$ is passed to a CRF layer to learn the correlations of the labels in neighborhood. Let $\mathcal{Y}$ denotes the set of all possible label sequences for $\tilde{\boldsymbol{H}}$. Then, the the probability of the label sequence $\boldsymbol{Y}$ for a given representation sequence $\tilde{\boldsymbol{H}}$ can be written as :

$$p(\boldsymbol{Y}|\tilde{\boldsymbol{H}}, W_Y) = \frac{\prod_t \psi_t(y_{t-1}, y_t; \tilde{\boldsymbol{H}})}{\sum_{Y' \in \mathcal{Y}} \prod_t \psi_t(y'_{t-1}, y'_t; \tilde{\boldsymbol{H}})}$$

Algorithm 1: CogNN Forward Computation

Input : $\boldsymbol{x_i}$: the i-th input token, N: the number of input tokens

Output: $\boldsymbol{y_i}$: the BIE label of $\boldsymbol{x_i}$, $\boldsymbol{y'_i}$: the FML or FI label of $\boldsymbol{x_i}$

for $i \leftarrow 1$ **to** N **do**
- *// Capture*
  - $\boldsymbol{e_i}$ = GloVe($\boldsymbol{x_i}$)
  - $\boldsymbol{s_i}$ = computeCaseVector($\boldsymbol{x_i}$)
  - $\boldsymbol{h_i}$ = Bi-LSTM($\boldsymbol{e_i} \oplus \boldsymbol{s_i}$)
  - $\boldsymbol{h'_i}$ = Bi-LSTM($\boldsymbol{e_i} \oplus \boldsymbol{s_i}$)
- *// Share*
  - $\boldsymbol{a_i}$ = getCoAttention($\boldsymbol{h_i}$, $\boldsymbol{h'_i}$)
  - $\tilde{\boldsymbol{h_i}}$ = getGuided($\boldsymbol{h_i}$, $\boldsymbol{a_i}$)
  - $\boldsymbol{a'_i}$ = getCoAttention($\boldsymbol{h'_i}$, $\boldsymbol{h_i}$)
  - $\tilde{\boldsymbol{h'_i}}$ = getGuided($\boldsymbol{h'_i}$, $\boldsymbol{a'_i}$)

for $i \leftarrow 1$ **to** N **do**
- *// Predict*
  - $\boldsymbol{y_i}$ = CRF($\tilde{\boldsymbol{h_i}}$)
  - $\boldsymbol{y'_i}$ = CRF($\tilde{\boldsymbol{h'_i}}$)
  - **return** $\boldsymbol{y_i}$, $\boldsymbol{y'_i}$

where $\psi_t(y', y; \tilde{\boldsymbol{H}})$ is a potential function, W_Y is a set of parameters that defines the weight vector and bias corresponding to label pair (y', y).

Similarly, we can also compute $p(\boldsymbol{Y'}|\tilde{\boldsymbol{H'}}, W_{Y'})$.

4.4 Joint Training

The remaining question is how to train two networks simultaneously to produce label sequences $\boldsymbol{Y}$ and $\boldsymbol{Y'}$. We achieve this by joint optimisation. Specifically, we train the CogNN model end-to-end by minimising loss $\mathcal{L}$, which is the sum of the loss of the two sub-networks:

$$\mathcal{L} = \mathcal{L}(W_Y) + \mathcal{L}(W_{Y'})$$

where $\mathcal{L}(W_Y)$ and $\mathcal{L}(W_Y)$ are the negative log-likelihood of the ground truth label sequences $\hat{\boldsymbol{Y}}$ and $\hat{\boldsymbol{Y'}}$ for the input sequences respectively, which are computed by:

$$\mathcal{L}(W_Y) = -\sum_i \sum_{\boldsymbol{Y_i}} \delta(\boldsymbol{Y_i} = \hat{\boldsymbol{Y}}) \log p(\boldsymbol{Y_i}|\tilde{\boldsymbol{H}})$$

$$\mathcal{L}(W_{Y'}) = -\sum_j \sum_{\boldsymbol{Y'_j}} \delta(\boldsymbol{Y'_j} = \hat{\boldsymbol{Y'}}) \log p(\boldsymbol{Y'_j}|\tilde{\boldsymbol{H'}})$$

5 Experimental Study

We evaluate our proposed CogNN model on our *HomeName* dataset and the *CoNLL-2003*

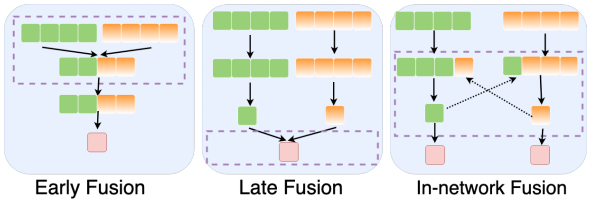

Figure 4: Early, late, and in-network fusion.

NER shared task dataset (Tjong Kim Sang and De Meulder, 2003). Recall (**R**), Precision (**P**) and F1-scores (**F**) are used to measure the performance. The experiments are run with an NVIDIA GeForce GTX 1080 GPU. We verify the following two aspects of our approach:

1) Effectiveness. We compare the CogNN model with baseline NER models and variants of the CogNN model on extracting person names from academic homepages (Section 5.1).

2) Applicability. We further explore whether our model and labelling scheme can improve person name recognition on news articles (Section 5.2).

5.1 Effectiveness on Academic Homepages

We first study the performance of CogNN on academic homepages.

Dataset and annotations We use the *HomeName* dataset with the proposed detailed name form annotation scheme (Section 3). We use 1,677 homepages for training and developing, and 410 homepages for testing.

Models and hyperparameters Four models are tested:

- **CRF** : The Stanford NER system (Finkel et al., 2005) with parameter tuned.
- **Bi-LSTM-CRF**: The Bi-LSTM-CRF model (Huang et al., 2015) with a 100-dimensional hidden layer, a dropout layer with probability 0.5, a batch size of 32, and an initial learning rate of 0.01 with a decay rate of 0.05. The model is optimised with stochastic gradient descent and we stop if the accuracy does not improve in 10 epochs.
- **CogNN**: Our proposed model.

We use GloVe trained on *HomeName* to get a 100-dimensional word embedding. For CogNN, we use the same hyperparameters and training parameters as those in Bi-LSTM-CRF. The optimal hyperparameters are obtained with a standard grid search on the developing dataset. More Details of the preprocessing, word embedding training, model implementation, and example output are given in the supplemental material.

Fusion strategies We also study the impact of using the detailed name form annotation scheme in different ways (cf. Figure 4):

- **No fusion**: Training an independent model that learns to label the input sequence with the BIE, FML, or FI label types but not a combination of any two types of the labels.
- **Early fusion**: Training an independent model that learns to label the input sequence with a cartesian product of the BIE, FML, and FI label types, e.g., to label '`John Doe`' with *Begin_First_Full End_Last_Full*.
- **Late fusion**: Training sub-models each focusing on one label type and merging all the predicted labels afterwards to yield the final prediction by using every span of tokens with name label as a name (so we do not report the token level performance).
- **In-network fusion**: Training two sub-models each focusing on one label type and sharing the learning signals in the intermediate levels of the sub-models. Our CogNN model use this strategy.

Results We report the token level performance, which reflects the model capability to recognise each person name token. We also report the name level performance, which reflects the model capability to recognise a whole person name without missing any token.

As Table 1 shows, overall, the neural models perform much better than the non-neural models, especially on the name level. When examining the performance of Bi-LSTM-CRF, we find that early fusion is much worse than no fusion. This is expected as early fusion of different name form types leads to too many classes to be predicted. Even for a two-token name, it may have $(3 \times 3 \times 2)^2 = 324$ possible name form combinations. Late fusion offers better performance, which indicates that the separately trained networks on different annotations have their own focuses. However, it does not take advantage of the correlations between name form types and is not as good as our in-network fusion strategy.

Our proposed model CogNN outperform the baseline models. CogNN outperforms the best baseline results by up to 5.64% and 5.35% in terms of F1-score in token level and name level, respectively. The reason is that our model can capture the underlying relationships among different name forms when training and gain higher prediction

Fusion Methods	Models	Annotations	Token Level			Name Level
			R	P	F	F
No fusion	CRF	BIE	64.94	94.68	77.04	41.15
		FML	60.93	94.48	74.08	54.98
		FI	61.31	95.13	74.57	50.32
	Bi-LSTM-CRF	BIE	87.97	89.64	88.79	80.89
		FML	84.96	87.12	86.03	82.11
		FI	86.69	88.79	87.72	81.71
Early fusion	Bi-LSTM-CRF	BIE × FML × FI	67.37	79.28	72.84	62.65
Late fusion	Bi-LSTM-CRF	BIE ∪ FML	–	–	–	83.12
		BIE ∪ FI	–	–	–	83.08
		FML ∪ FI	–	–	–	83.29
		BIE ∪ FML ∪ FI	–	–	–	83.45
In-network fusion	CogNN (proposed)	[BIE, FML]	89.23	90.54	**89.88**	**84.26**
			85.99	87.20	**86.75**	**84.30**
		[BIE, FI]	93.06	92.85	**92.95**	**85.85**
			86.70	89.33	**88.00**	**85.92**
		[FML, FI]	85.79	87.50	**86.64**	**83.50**
			86.90	88.01	**87.45**	**83.47**

Table 1: Model performance on *HomeName*. The difference between the bold and the non-bold numbers is statistically significant with $p < 0.05$ as calculated using McNemar's test.

confidence. Also, CogNN yields best results when the models are jointly trained with input annotations BIE and FI. This is consistent with the observation on the no-fusion models where a model trained with either BIE or FI annotations outperforms a model trained with FML annotations.

5.2 Applicability on Newswire Articles

We further show the applicability of our CogNN model and detailed name form annotation scheme on traditional newswire texts.

Dataset We use the CoNLL-2003 dataset which contains 1,393 annotated English newswire articles that focus on four types of named entities: person (PER), location (LOC), organisation (ORG) and miscellaneous entity (MISC). This dataset does not come with detailed name form annotations. We add annotations using the same method described in Section 3.

Annotations We compare the following combinations of annotations:

- **PER**: Using only PER labels.
- **FI**: Using only FI labels.
- **FML**: Using only FML labels.
- **CoNLL**: Using all original labels in CoNLL-2003.
- **FI + CoNLL**: Replacing PER by FI labels in CoNLL-2003.
- **FML + CoNLL**: Replacing PER by FML labels in CoNLL-2003.

Since the detailed name form labels are necessary for training CogNN, we use the following four pairs of input annotations for CogNN: [PER, FI], [PER, FML], [CoNLL, FI + CoNLL], and [CoNLL, FML + CoNLL].

Models	Annotations	R	P	F
CRF	PER	85.29	94.75	89.77
	FI	85.00	94.73	89.60
	FML	83.66	93.36	88.25
	CoNLL	92.43	89.96	91.18
	FI+CoNLL	92.40	89.93	91.14
	FML+CoNLL	90.19	89.04	89.61
Bi-LSTM-CRF	PER	96.25	96.98	96.62
	FI	96.32	96.71	96.51
	FML	94.74	95.15	94.94
	CoNLL	96.43	96.74	96.59
	FI+CoNLL	**96.54**	96.12	96.33
	FML+CoNLL	95.17	94.49	94.83
CogNN	[PER, FI]	94.93	98.37	96.62
	[PER, FML]	94.84	97.57	96.18
	[CoNLL, FI+CoNLL]	94.99	**98.43**	**96.68**
	[CoNLL, FML+CoNLL]	94.93	97.78	96.33

Table 2: Token level performance of person name recognition on CoNLL-2003. The difference between the bold and the non-bold numbers is statistically significant with $p<0.05$ as calculated using McNemar test.

Models and Hyperparameters We compare **CRF**, **Bi-LSTM-CRF**, and **CogNN**. We use the same model hyperparameters and training parameters as described in Section 5.1. We initialise word embeddings with GloVe pretrained 100-dimensional embeddings. Unknown words are randomly initialised. All the above models are trained, developed, and tested on the training, developing, and testing datasets in CoNLL-2003.

Results From Table2, we see that neural models perform better than the non-neural model, which is consistent with the results in Section 5.1. When providing extra ORG, LOC, and MISC annotations apart from PER to CRF and Bi-LSTM-CRF, the recall increases while the precision decreases. This indicates that the extra annotations

help recognise more named entity tokens but may also misguide the model. In comparison, CogNN is less impacted.

When providing extra FI or FML annotations apart from PER to CRF and Bi-LSTM-CRF, the performance of both models does not improve while that of CogNN improves. Our improvements mainly lie in the precision, which indicates that CogNN can well distinguish person name tokens from others. These results also indicate that only applying the detailed name form annotations on newswire data for the existing models is not enough. Our CogNN model is essential to make use of the extra detailed name form information.

Although, the advantage of CogNN on formal English newswire articles is smaller than that on the HomePub dataset (Section 5.1). The main reason is that the name forms in newswire articles are less flexible compared with those in academic homepages, which impinges the importance of adding extra name form information.

6 Conclusion

We studied the person name recognition problem in academic homepages. We propose a new name annotation scheme which provides more detailed information for various forms of names. We constructed a new dataset *HomeName*. To take advantages of the detailed name form information, we proposed the CogNN model that makes use of two sub-networks to learn whether a token is part of a name and its forms, respectively. Through co-attention, the two sub-networks help each other to boost the overall name recognition accuracy. We performed an experimental study to evaluate model effectiveness and applicability. The results showed that CogNN outperforms state-of-the-art NER models by up to 5.64% and 5.35% in terms of F1-score in token and name levels on academic homepages, while achieving comparable performance on a traditional NER benchmark dataset CoNLL-2013.

We will release *HomeName* and the annotation tools for public use upon paper publication. For future work, we plan to investigate the CogNN model on other tasks that exhibit dependencies between each other, such as POS tagging and Chunking. We also plan to investigate using three sub-networks in CogNN, one for each of the BIE, FML, and FI dimensions, with a dual decomposition in the fusion steps. Late fusion would be another interesting direction to explore.

References

Mohammed Aboaoga and Mohd Juzaiddin Ab Aziz. 2013. Arabic person names recognition by using a rule based approach. *Journal of Computer Science*, 9(7):922–927.

Majidi Bidhend, Behrouz Minaei-Bidgoli, and Hosein Jouzi. 2012. Extracting person names from ancient islamic arabic texts. In *Proceedings of Language Resources and Evaluation for Religious Texts (LRE-Rel) Workshop, International Conference on Language Resources and Evaluation (LREC)*, pages 1–6.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association of Computational Linguistics*, 4(1):357–370.

Christopher Dozier and Robert Haschart. 2000. Automatic extraction and linking of person names in legal text. In *Content-Based Multimedia Information Access*, pages 1305–1321.

Fabrice Dugas and Eric Nichols. 2016. Deepnnner: Applying blstm-cnns and extended lexicons to named entity recognition in tweets. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 178–187.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*.

Ali Elsebai, Farid Meziane, Fatma Zohra Belkredim, et al. 2009. A rule based persons names arabic extraction system. *Communications of the IBIMA*, 11(6):53–59.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 363–370.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Chenliang Li, Aixin Sun, Jianshu Weng, and Qi He. 2015. Tweet segmentation and its application to named entity recognition. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 27(2):558–570.

Di Lu, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. 2018. Visual attention model for name tagging in multimodal social media. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1990–1999.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In

Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL), pages 1064–1074.

Einat Minkov, Richard C Wang, and William W Cohen. 2005. Extracting personal names from email: Applying named entity recognition to informal text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 443–450.

Seungwhan Moon, Leonardo Neves, and Vitor Carvalho. 2018. Multimodal named entity disambiguation for noisy social media posts. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2000–2008.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Thomas L Packer, Joshua F Lutes, Aaron P Stewart, David W Embley, Eric K Ringger, Kevin D Seppi, and Lee S Jensen. 2010. Extracting person names from diverse and noisy ocr text. In *Proceedings of the 4th Workshop on Analytics for Noisy Unstructured Text Data*, pages 19–26.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Khaled Shaalan and Hafsa Raza. 2007. Person name entity recognition for arabic. In *Proceedings of Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 17–24.

Jie Tang, Limin Yao, Duo Zhang, and Jing Zhang. 2010. A combination approach to web user profiling. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(1):2.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 142–147.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1480–1489.

Yiqing Zhang, Jianzhong Qi, Rui Zhang, and Chuandong Yin. 2018. Pubse: A hierarchical model for publication extraction from academic homepages. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1005–1010.

A Supplemental Material

In this supplementary section, we provide additional details for the Dataset (Section 3) and the Experiments (Section 5).

A.1 HomeName Dataset

File Structure Our dataset is constructed based on the *HomePub* dataset[3], which contains academic homepages with publication strings manually labelled. It contains 2,500 subfolders and each subfolder contains three files for a webpage:

- An HTML file containing the page source.
- A TXT file containing the visible text of the webpage, which is rendered by python's Selenium[4] package.
- A JSON file containing publication annotations.

We edit the JSON files to add name annotations using the following format:

```
{
  "filename": "270eddc7-bffc-4425-9733-6a202f6ab08a",
  // a unique id of the txt file
  "is_personal_homepage": "T",
  // whether the webpage is an academic homepage, e.g., T or F
  "comment": "Uncertain",
  // leave any comment if needed, e.g., N/A or Uncertain
  "names": [
    {
      "form": "Begin_First_Full End_Last_Full",
      // form of a name
      "index": [
        [204,207]
      ], // position indices of the name
      "text": "Doe" // surface of a name
    },
    {
      "form": "Begin_Last_Full End_First_Initial",
      "index": [
        [2331,2336],[2557,2562],[2802,2807]
      ],
      "text": "Doe J"
    },
    {
      "form": "Begin_Last_Full End_First_Initial",
      "index": [
        [10053,10062]
      ],
      "text": "Joon-gi L"
    },
    {
      "form": "Begin_Last_Full Inside_Last_Full End_First_Initial",
      "index": [
        [3027,3037]
      ],
      "text": "van Laar J"
    }
  ]
}
```

Figure 5: Screenshot of an example JSON file

Annotation Tool Annotation of homepages is time-consuming, especially when a homepage contains many names in complex forms. We developed a semi-automatic tool[5] to assist the annotation, which has five main functionalities:

[3] http://www.ruizhang.info/data/homepub.html

[4] https://selenium-python.readthedocs.io/

[5] The tool will be released with the *HomeName* dataset upon paper publication.

- *Group_label*: This functionality helps annotate a group of names of the same form. For example, *'Doe J'* and *'Joon-gi L'* have the same forms and can be annotated at once.
- *Index*: This functionality helps find all positions of a given name string in the TXT file.
- *Mask*: This functionality helps annotators to proofread the text and find unlabelled names. It replaces all the names already annotated with a special token *'ANNOTATED'*.
- *Validate*: This functionality runs a simple automated quality check of the annotations. It checks: (1) whether the position indices of the names annotated in the JSON file are consistent with the names appeared in the TXT file; and (2) whether each annotated name comes with the name form under the three-dimensional annotation schemes.
- *Compare*: This functionality locates disagreement between two annotators' labels on the same homepage. It identifies the list of names with inter-annotator disagreement.

Annotators There are 6 annotators to annotate the dataset. The annotators are postgraduate students who have taken machine learning subjects. We provide a one-hour training to each annotator.

We provide the annotators with an annotation scheme and two example pages that are already annotated. We ask each annotator to annotate six pages. We examine the results and provide guidance on how to improve the annotation quality.

We highlighted the following at training:

- Any named entities such as places, buildings, organizations, prizes, honored titles or books, which are named after a person, should not be annotated as a person's name.
- Words connected with a hyphen or an apostrophe should not be split into multiple tokens. For example, both *'Joon-gi'* and *'O'Keeffe'* both have only one token.
- Nobiliary particles[6], e.g., *'van'*, *'zu'* and *'de'*, should be annotated as last names.

Each academic homepage is annotated by two annotators. We ask all the annotators to annotate using our annotation tool and also note down any pages with uncertain name labels in the `comment` field. After their annotations, we summarise the disagreement between annotators. We make a decision on the disagreement and also check the uncertain pages and names. We send feedback when they annotate every 230 homepages.

Annotation Analysis

- Confidence: Only 3.64% of all the homepages contain annotations that are uncertain as flagged by the annotators, while 78.08% of these pages are actually correctly labelled. This indicates that the annotators have high confidence in their annotations.
- Inter-annotator Agreement: We compute the inter-annotator agreement on name strings and name forms using Cohens Kappa measurement. The annotators have higher agreement on name strings ($\kappa = 0.63$) and lower agreement on detailed name forms ($\kappa = 0.41$). The disagreement is mainly in homepages with a long string of consecutive name tokens such that different annotators may disagree on which tokens to form a name. The annotators may also disagree on whether a name token is a first name, middle name, or last name. This is difficult especially when the context is unclear.
- Time: On average, it takes 16 minutes to annotate an academic homepage with our tool.

Dataset Analysis In total, the *HomeName* dataset contains 2,087 academic homepages from 286 institutes, i.e., 7.29 pages per institute (standard deviation 7.27). A total of 34,880 names are annotated and 70,864 name position indices are recorded. On average, a name appears twice in an academic homepage. Most names begin with last names (64.73%) while the rest mostly begin with first names. Only 13 names start with middle names. Most names contain at least one initial (66.57%). The two most frequent name forms are *Begin_Last_Full End_First_Initial* and *Begin_First_Full End_Last_Full*. Table 3 summarises the annotation results and the dataset.

A.2 Experiment Details

Preprocessing We focus on English webpages and first convert any text in Unicode to ASCII using Unidecode [7]. We then split the text into sentences using the sentence tokenizer in NLTK. The sentences are further tokenized on whitespace and punctuations except for hyphens and apostrophes. Every punctuation is considered as a single token to retain the structural information.

Word Embedding We train 100 dimensional

[6] https://en.wikipedia.org/wiki/Nobiliary_particle

[7] https://pypi.org/project/Unidecode/

Summary of Annotation		
Confidence	Uncertain pages	3.64%
	Acc. on uncert. names	78.08%
Inter-annotator agreement (κ)	Names	0.63
	Names forms	0.41
Time		16 min

Summary of Dataset	
Total Homepages	2,087
Total Institutes	286
Average Institutes	7.29
STD. Institutes	7.27
Total Names Indexes	70,864
Total Names	34,880
Contain Initial	23,221
Begin with Last Name	22,581
Begin with Middle Name	13
Begin with First Name	12,286

Table 3: Summary of annotation and dataset: Acc. is accuracy, and κ is the Cohens Kappa measurement.

word embeddings using GloVe[8] on *HomeName*, with a window size of 15, minimun vocabulary count of 5, full passes through cooccurrence matrix of 15, and an initial learning rate of 0.05.

Implementation

- CRF: We use the Java implementation provided by the Stanford NLP group[9] with default parameter settings. The software provides a generic implementation of linear chain CRF model.
- Bi-LSTM-CRF: We implemente Bi-LSTM-CRF using Theano[10] and Lasagne[11]. The word embeddings are fed into a Bi-LSTM layer as input. Dropout is applied to the output of Bi-LSTM layer to avoid overfitting. The output is further fed into a linear chain CRF layer to predict the tokens labels.
- CogNN: Our Co-guided Neural Network is also implemented on Theano and Lasagne following the description in Section 4. Dropout is applied on the Bi-LSTM layers. We use a standard grid search to find the best hyperparameter values. We choose the initial learning rate among $[0.001, 0.01, 0.1]$, the decay rate among $[0.05, 0.1]$, the dimension of hidden layer among $[50, 100, 200]$, the dropout rate among $[0.2, 0.5]$

[8] https://nlp.stanford.edu/projects/glove/

[9] https://nlp.stanford.edu/software/CRF-NER.html

[10] http://deeplearning.net/software/theano/

[11] https://lasagne.readthedocs.io

A.3 Example Output

Figure 6 shows a sample output of different models. In the figure, tokens in italics are the ground truth, while tokens in bold are those predicted as names. We see that all the baseline models contain wrong predictions while the proposed CogNN model successfully recognise all the names.

Stanford NER (Newswire)

Proceedings of the National Academy of Sciences of the United States of America
Kime** C* , *Sakaki-Yumoto M* , ***Goodrich** L* , ***Hayashi** Y* , ***Sami S , ***Derynck R*** , *Asahi M* , *Panning B* , ***Yamanaka** S* , ***Tomoda** K*
Activators and repressors : A balancing act for X-inactivation .

CRF (BIE)

Proceedings of the National Academy of Sciences of the United States of America
Kime C , ***Sakaki-Yumoto** M* , *Goodrich L* , ***Hayashi Y*** , ***Sami S*** , ***Derynck R*** , ***Asahi M*** , *Panning B* , *Yamanaka S* , ***Tomoda** K*
Activators and repressors : A balancing act for X-inactivation .

Bi-LSTM-CRF (BIE)

Proceedings of the National Academy of Sciences of the United States of America
*Kime **C** , **Sakaki-Yumoto** M* , *Goodrich **L** , **Hayashi Y** , **Sami S** , **Derynck R** , **Asahi** M* , *Panning B*, *Yamanaka **S*** , ***Tomoda** K*
Activators and repressors : A balancing act for **X-inactivation** .

CogNN ([BIE, FI])

Proceedings of the National Academy of Sciences of the United States of America
Kime C , ***Sakaki-Yumoto M*** , ***Goodrich L*** , ***Hayashi Y*** , ***Sami S*** , ***Derynck R*** , ***Asahi M*** , ***Panning B*** , ***Yamanaka S*** , ***Tomoda K***
Activators and repressors : A balancing act for X-inactivation .

Figure 6: An example of applying different models on the text from an academic homepage. All the italic tokens except commas should be recognised as names while the bold tokens are actually recognised.