

Person Name Recognition with Fine-grained Annotation

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

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

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

ABSTRACT

Person names are essential and important entities in the Named Entity Recognition (NER) task. Traditional NER models have shown success in recognising well-formed person names from text with consistent and complete syntax, such as news articles. However, user-generated text such as academic homepages, academic resumes, articles in online forums and social media may contain lots of free-form text with incomplete syntax including person names with various forms. This brings significant challenges for the NER task. In this paper, we address person name recognition in this context by proposing a fine-grained annotation scheme based on anthonymy together with a new machine learning model to perform the task of person name recognition. Specifically, our proposed name annotation scheme labels fine-grained name forms including first, middle, or last names, and whether the name is a full name or initial. Such fine-grained annotations offer richer training signals for models to learn person name patterns in free-form text. We then propose a *Co-guided Neural Network* (CogNN) model to take full advantage of the fine-grained annotations. CogNN uses co-attention and gated fusion to co-guide two jointly trained neural networks, each focusing on different dimensions of the name forms. Experiments on academic homepages and news articles demonstrate that our annotation scheme together with the CogNN model outperforms state-of-the-art significantly.

ACM Reference Format:

Yimeng Dai, Rui Zhang, and Jianzhong Qi. 2020. Person Name Recognition with Fine-grained Annotation. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20)*, August 1–5, 2020, Virtual Event, China. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3383583.3398515>

1 INTRODUCTION

Person names are basic yet important entities in the Named Entity Recognition (NER) task. Recognising person names from unstructured text has become an important process for many online academic mining systems, such as AMiner [26] and CiteSeerX [21]. Person name recognition plays an important role in learning the relationships between people and provides valuable insights for analysing their collaboration networks [2, 14].

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

JCDL '20, August 1–5, 2020, Virtual Event, China

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7585-6/20/06...\$15.00

<https://doi.org/10.1145/3383583.3398515>

BBC NEWS


Steve Jobs, co-founder and former chief executive of US technology giant Apple, has died at the age of 56. Apple said he had been "the source of countless innovations that enrich and improve all of our lives" and had made the world "immeasurably better". Mr Jobs had announced he was suffering from pancreatic cancer in 2004. Tributes have been made by technology company bosses and world leaders, with US President Barack Obama saying the world had "lost a visionary". "Steve was among the greatest of American innovators - brave enough to think differently, bold enough to believe he could change the world, and talented enough to do it," said Mr Obama. A statement from Mr Jobs's family said they were with him when he died peacefully on Wednesday. "In his public life, Steve was known as a visionary; in his private life, he cherished his family," they said, requesting privacy and thanking those who had "shared their wishes and prayers" during his final year.

Apple said the company had "lost a visionary and creative genius, and the world has lost an amazing human being". Tim Cook, who was made Apple's CEO after Mr Jobs stood down in August, said his predecessor had left behind "a company that only he could have built, and his spirit will forever be the foundation of Apple". Flags are being flown at half mast outside the Apple headquarters in Cupertino, California, while fans of the company have left tributes outside Apple shops around the world. "What he's done for us as a culture, it resonates uniquely in every person," said Cory Moll, an Apple employee in San Francisco. "Even if they never use an Apple product, the impact they have had is so far-reaching." At the company's Shanghai shop, customer Jin Yi said Mr Jobs had created gadgets which had "changed people's perceptions of machines".

Source: <https://www.bbc.com/news/world-us-canada-15193922>

(a) News Articles

Rui Zhang Professor
School of Computing and Information Systems
The University of Melbourne
Email: (my first name) (dot) zhang@unimelb.edu.au
Mailing Address: School of Computing & Information Systems
Level 8, Doug McDonnell Building (Building 168)
University of Melbourne, Parkville, Victoria, Australia 3052



Brief Biography:
Dr Rui Zhang is currently a Professor at the School of Computing and Information Systems of the University of Melbourne. Professor Zhang's research interests include big data and AI, particularly in areas of recommendation systems, chatbot, knowledge bases, spatial and temporal data analytics, moving object management and data streams. Professor Zhang has won several awards including the prestigious Future Fellowship by the Australian Research Council in 2012 and Chris Wallace Award for Outstanding Research by the Computing Research and Education Association of Australasia in 2015.

Selected Publications:

- 1) Y. Dai, R. Zhang, J. Qi. Person Name Recognition with Fine-grained Name Annotation. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020)*, 2020.
- 2) B. D. Trisedya, J. Qi, R. Zhang. Sentence Generation for Entity Description with Content-plan Attention. *Proceedings of the 33th AAAI Conference on Artificial Intelligence (AAAI 2020)*, 2020
- 3) X. Huang, J. Qi, Y. Sun, R. Zhang. MALA: Cross-Domain Dialogue Generation with Action Learning. *Proceedings of the 33th AAAI Conference on Artificial Intelligence (AAAI 2020)*, 2020.
- 4) K. Ramamohanarao, S. Karunasekera, L. Kulik, E. Tanin, R. Zhang, H. Xie, E. B. Khunayn. SMARTS: Scalable Microscopic Adaptive Road Traffic Simulator. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2): 26:1-26:22, 2017
- 5) F. G. D. Ward, Z. He, R. Zhang, J. Qi. Real-time Continuous Intersection Joins over Large Sets of Moving Objects using Graphic Processing Units. *VLDB Journal*, 23(6): 965-985, 2014.

Source: <http://www.ruizhang.info/>

(b) Academic Homepage

Figure 1: An example of a news article with well-formed text and an example of an academic resume with free-form text. All the person names are highlighted.

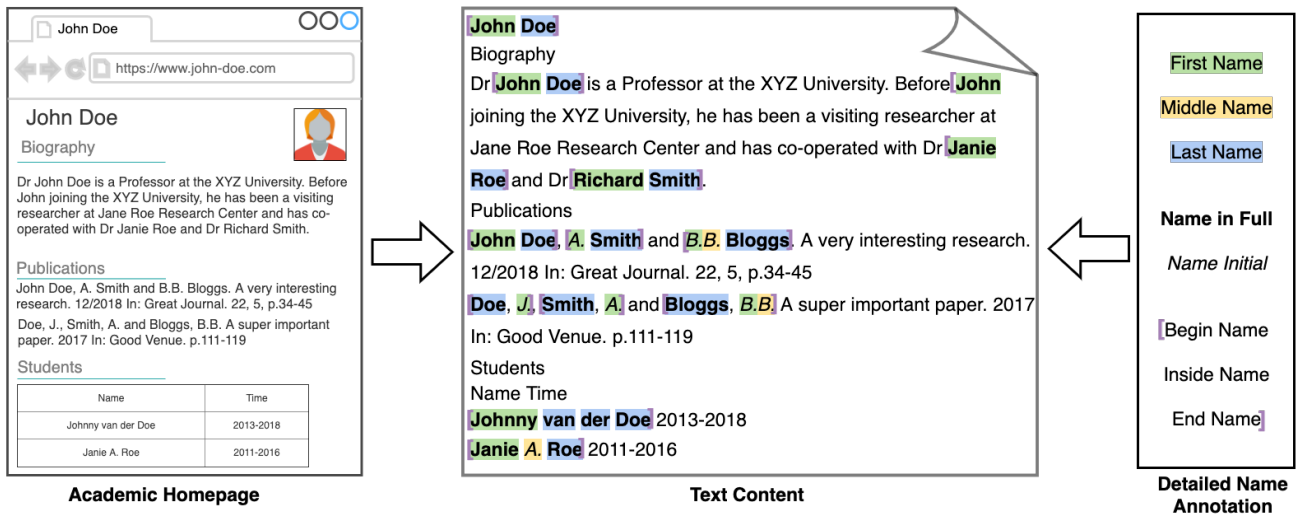


Figure 2: An example of person name recognition in academic pages. All person names are highlighted (best viewed in color).

Traditional NER models [5, 12, 17] have shown success in recognising person names from well-formed text, such as news articles (cf. Figure 1(a)). Such text often has consistent and complete syntax, which provides textual contexts for recognising person names. Person names in such text are often well-formed with straightforward patterns such as first name followed by last name in full. However, challenges remain for recognising person names from free form text such as user-generated text. These may appear in many applications, e.g., user-generated academic homepages, academic resumes, articles in online forums and social media (cf. Figure 1(b)). They often contain person names of various forms with incomplete syntax.

Figure 2 shows an example of person name recognition in academic homepages. The biography section consists of complete sentences while the students section simply lists information in a line. The person names may be in different forms. Figure 2 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. Bloggs’ vs. ‘Doe, J.’.

To better recognize person names from such a free-form text, we exploit knowledge from *anthroponymy* [9] (i.e., the study of the names of human beings) and propose a fine-grained annotation scheme that labels detailed name forms including first, middle, or last name, and a full name word or a name initial (cf. Figure 2). Such fine-grained annotations offer richer training signals to NER models to learn the patterns of person names in free-form text. However, fine-grained annotations also bring challenges because more label classes need to be learned.

To take full advantage of the fine-grained annotations, 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 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. The intuition is that knowing whether

a token is part of a name helps recognise the fine-grained 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. However, the underlying correlation between different annotations cannot be captured well when the two sub-networks are trained together by simply minimizing the total loss. The reason is that the learning signals of the two sub-neural networks are not shared well when training. To better capture the underlying correlation between different annotations, we share the learning signals of two sub-neural networks through a co-attention layer. Further, we use a gated-fusion layer to balance the information learned from two sub-networks. This way, neither sub-network is overwhelmed by misleading signals that may be learned by the other sub-network.

Our contributions are summarized as follows:

- *New setting and annotation:* We propose a fine-grained annotation scheme based on anthroponymy. This fine-grained annotation scheme provides information on various forms of person names. Experimental results show that our annotations can be utilised in different ways to improve the recognition performance.

- *The first dataset under the fine-grained annotation scheme:* We create the first dataset consisting of diverse academic homepages where the person names are fully annotated under our posed fine-grained annotation scheme, called *FinegrainedName*, for the research of name recognition and new articles.

- *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 and gated fusion mechanisms. Experimental results show that CogNN outperforms state-of-the-art NER models and multi-task models by utilising the fine-grained annotations, and improve the recognition performance on academic homepages.

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 [20]. Neural NER models have shown excellent performance on long texts which follow strict syntactic rules, such as newswire and Wikipedia articles [5, 12, 17]. However, these NER models are less attractive when applied to texts which may not have consistent and complete syntax [7, 13]. Recent studies consider user-generated short texts from social media platforms such as Twitter and Snapchat [15, 19]. However, there are few NER studies on free-form text with incomplete syntax including person names with various forms, such as academic homepages, academic resumes, articles in online forums and social media.

BIO and BIEO tagging schemes [3] are often used for named entity recognition in well-formed text, such as news articles in CoNLL-2003 [24] and wikipedia articles in WiNER [11]. However, such annotations for name spans cannot reflect the patterns of various name forms and brings challenges for recognising persons names in free-form text. To the best of our knowledge, no existing work has utilised anthroponymy [9] and fine-grained annotations to help recognise person names.

Information Extraction (IE) studies on academic homepages and resumes usually treat the text content as a document, upon which traditional NER techniques are applied. For example, Zhang et al. [27] use a Bi-LSTM-CRF based hierarchical model to extract all the publication strings from the text content of a given academic homepage. Dai et al. [6] capture the relationship between publication strings and person names in academic homepages, and extract them simultaneously. This technique does not apply to our problem as we assume no pre-knowledge about the publication strings.

Person names are often recognised together with other entities, such as locations and organisations [5, 7, 12, 17]. Packer et al. [22] focus on extracting name from noisy OCR data by combining rule based methods, the Maximum Entropy Markov Model, and the CRF model using a simple voting-based ensemble. Minkov et al. [18] extract person names from emails using CRF. They design email specific structural features and exploit in-document repetition to improve the extraction accuracy. Aboaga and Ab Aziz [1] study person name recognition in Arabic using rule-based methods. To the best of our knowledge, this paper is the first study on person name recognition problem that takes into account name forms and uses deep learning based models.

Multi-task learning models, which train tasks in parallel and share representations between related tasks, have been proposed to handle many NLP tasks. Caruana [4] propose to share the hidden layers between tasks, while keeping several task-specific output layers. Søgaard and Goldberg [25] jointly learn POS tagging, chunking and CCG supertagging by using successively deeper layers. Ma et al. [16] propose a model for sentiment analysis by jointly learn the character features and long distance dependencies through concatenation procedures. Rather than directly sharing the representations or concatenating the representations of different tasks, our co-attention and gated fusion mechanisms allow our model to co-guide the jointly trained tasks without overwhelmed by the misleading signals.

3 PROPOSED ANNOTATION SCHEME

We first present our fine-grained annotation scheme and introduce our FinegrainedName dataset annotated under this scheme.

3.1 Fine-grained Annotations

Fine-grained annotations are done to better capture the person name form features in free-form texts. Annotating the name tokens with fine-grained forms offers more direct training signals to NER models to learn the patterns of person names. Thus, unlike traditional NER datasets, which only label a name token with a *PER* (person) label, we further provide fine-grained name form information for each name token based on anthroponymy [9].

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 fine-grained name form of a name token. For example, in Figure 2, ‘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*.

3.2 The FinegrainedName Dataset

FinegrainedName¹ is a collection of academic homepages with person names fully annotated using the proposed annotation scheme. We use Selenium², an open-source automated rendering software, to render the webpages and collect visible texts from the webpages. We download academic homepages from universities and research institutes around the world and focus on English homepages.

FinegrainedName contains 2,087 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 Selenium³ package.
- A JSON file containing name annotations. Figure 3 shows the example format of the JSON files.

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

- *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*’.

¹Dataset will be available at <http://www.ruizhang.info/namerec/>

²<https://www.seleniumhq.org/>

³<https://selenium-python.readthedocs.io/>

```

{
  "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_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 3: Screenshot of an example JSON file

- *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 particles⁴, e.g., 'van', 'zu' and 'de', should be annotated as last names.

Each academic homepage is annotated by two annotators using our annotation tool. Any pages with uncertain name labels is noted

⁴https://en.wikipedia.org/wiki/Nobiliary_particle/

Summary of Annotation		
Confidence	Uncertain pages	3.64%
Inter-annotator agreement (κ)	Accuracy on uncertain names	78.08%
	Names	0.63
	Name 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 1: Summary of annotation and dataset. κ is the Cohens Kappa measurement.

down in the comment field (cf. Figure 3). After their annotations, we make a decision on the disagreement between annotators and also check the uncertain pages and names. We send feedback when they annotate every 230 homepages. ur

Annotation Analysis We summarise the disagreement between annotators. Table 1 reports the annotation summary:

- **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 Cohen's Kappa measurement. The annotators have higher agreement on name strings ($\kappa = 0.63$) and lower agreement on fine-grained 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 FinegrainedName contains 2,087 English 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 1 summarises the annotation results and the dataset.

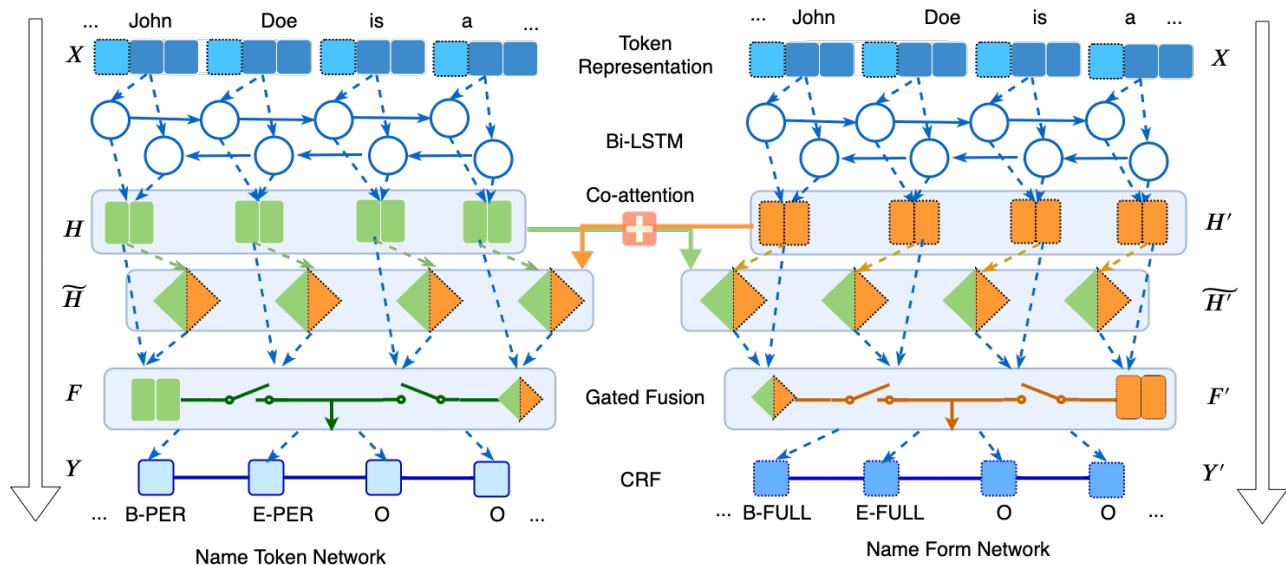


Figure 4: CogNN network structure.

4 PROPOSED MODEL

The fine-grained annotations offer more direct training signals to NER models but also bring challenges because more label classes need to be learned. In this section, we present our CogNN model that takes advantages of the fine-grained annotations to recognise person names⁵.

Given a sequence of input tokens X , where $X = [x_1, x_2, \dots, x_n]$ and n is the length of the sequence, our aim is to predict for each token x_i whether it is a name token.⁶

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 4. 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 fine-grained name form class of the token (*FML* or *FI* dimensions). The intuition is that knowing whether a token is part of a name helps recognise the fine-grained 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. To better capture the underlying correlation between different annotations, we share the learning signals of two sub-neural networks through a co-attention layer. To avoid being overwhelmed by possible misleading signals, we further add a gated-fusion layer to balance the information learned from two sub-networks.

In particular, an input token is represented by concatenating its word embedding and its letter case vector. We feed such representation of the input into Bi-LSTM to learn its hidden representation matrix, which is detailed in Section 4.1. Then, we use co-attention and gated fusion mechanisms to co-guide the two jointly trained

sub-networks. Our co-attention mechanism updates the importance of each token learned from the two sub-networks and records their correlations (Section 4.2). Our gated fusion mechanism helps decide whether and how much to accept new signals from the other sub-network (Section 4.3). The two sub-networks are trained simultaneously by minimising their total loss (Section 4.4).

4.1 Capture: Hidden Feature Extraction

The name token network (denoted as N_Y) and the name form network (denoted as $N_{Y'}$) have a similar structure. They only differ in the target labels Y and Y' . Here, Y denotes the label sequence that indicates whether an input token is part of a name, and Y' denotes the label sequence that indicates the form class of each input token. We focus our explanation on the name token network N_Y in the following discussion while the name form network works in a similar way.

An input token $x_i \in X$ is represented by concatenating its word embedding e_i and its letter case vector s_i . We use GloVe [23] computed on our FinegrainedName corpus for the word embeddings e_i . The letter case vector s_i indicates the letter case information of 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) whether the first character in the token is in uppercase, (ii) whether all characters in the token are in uppercase, and (iii) whether any character in the token is in uppercase.

We then use Bi-LSTM [8] to capture the hidden features from the input sequence. The output hidden representation, denoted as h_i , summarises the context information of x_i in X . Our hidden representation matrix H in N_Y can be written as $[h_1, h_2, \dots, h_n]$, where $h_i \in \mathcal{R}^d$ and d is the number of dimensions of the hidden representation. Similarly, H' in $N_{Y'}$ can be written as $[h'_1, h'_2, \dots, h'_n]$.

⁵A demonstration of our model will be available at <http://www.ruizhang.info/namerec/>

⁶We use x_i to denote both a token and its embedding vector as long as the context is clear.

4.2 Share: Co-attention Mechanism

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

Specifically, we use co-attention to take the learning signals from two hidden representations into account by:

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

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, \tanh is the activation function to scale into the range of $(-1,1)$, and $\mathbf{P} \in \mathcal{R}^{2k \times n}$.

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

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

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

The new hidden representation $\tilde{\mathbf{h}}_i$ can be computed by:

$$\tilde{\mathbf{h}}_i = \mathbf{a}_i \mathbf{h}_i, \mathbf{a}_i \in \mathbf{A}, \mathbf{h}_i \in \mathbf{H}$$

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

4.3 Balance: Gated Fusion Mechanism

To avoid being overwhelmed by misleading learning signals from the other sub-network $N_{Y'}$ during co-attention, we dynamically balance the information learned from the (independent) hidden representation \mathbf{H} and the corresponding new (dependent) hidden representation $\tilde{\mathbf{H}}$ for N_Y (also \mathbf{H}' and $\tilde{\mathbf{H}}'$ for $N_{Y'}$), and obtain a fused representation matrix \mathbf{F} (\mathbf{F}' for $N_{Y'}$).

Inspired by the study on multi-modal fusion for images and text [15], we add a gated fusion layer to balance the information from \mathbf{H} and $\tilde{\mathbf{H}}$ to obtain better representations.

We first transform each item in \mathbf{H} and $\tilde{\mathbf{H}}$ by:

$$\mathbf{h}_{\tilde{h}_i} = \tanh(W_{\tilde{h}_i} \tilde{\mathbf{h}}_i + b_{\tilde{h}_i})$$

$$\mathbf{h}_{h_i} = \tanh(W_{h_i} \mathbf{h}_i + b_{h_i})$$

where $W_{\tilde{h}_i}$ and W_{h_i} are trainable parameters.

Then, our fusion gate, which decides whether and how much to accept the new information, is computed as:

$$\mathbf{g}_t = \sigma(W_{g_t} (\mathbf{h}_{\tilde{h}_i} \oplus \mathbf{h}_{h_i}))$$

where σ is the element-wise sigmoid function to scale into the range of $(0,1)$ and W_{g_t} are trainable parameters.

We fuse the two representations using the fusion gate through:

$$\mathbf{f}_i = \mathbf{g}_t \mathbf{h}_{h_i} + (1 - \mathbf{g}_t) \mathbf{h}_{\tilde{h}_i}$$

The fused representation sequence $\mathbf{F} = [f_1, f_2, \dots, f_n]$ is trained to produce a label sequence Y . To enforce the structural correlations between labels, 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 F . Then, the the probability of the label sequence Y for a given fused representation sequence F can be written as :

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

where $\psi_t(y', y; F)$ 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 the fused representation sequence F' and $p(Y'|F', W_{Y'})$.

4.4 Joint Training

The remaining question is how to train two networks simultaneously to produce label sequences Y and 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{Y} and \hat{Y}' for the input sequences respectively, which are computed by:

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

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

5 EXPERIMENTAL STUDY

We explore the following three aspects of our approach by a comprehensive experimental study:

- The impact of using fine-grained annotations in different ways for recognising person names from academic homepages.
- The performance of the CogNN model against baseline joint models and variants of the CogNN model on recognising person names from academic homepages.
- The applicability of the proposed annotation scheme together with the CogNN model on recognising person names from news articles.

5.1 Effectiveness on Academic Homepages

In this subsection, we study the performance of our proposed annotation scheme together with the CogNN model on academic homepages.

Dataset We use the FinegrainedName with the proposed fine-grained annotation scheme (Section 3), where 1,677 homepages are used for training and developing and 410 homepages are used for testing.

Evaluation Recall (**R**), Precision (**P**) and F1-scores (**F**) are used to measure the performance. 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

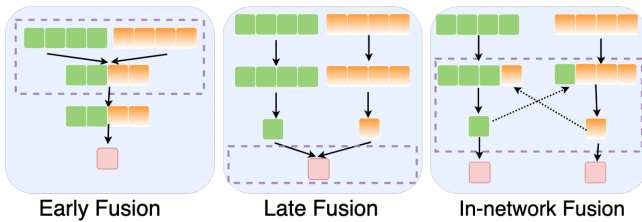


Figure 5: Early, late, and in-network fusion.

missing any token. The reported improvements are statistically significant with $p < 0.05$ as calculated using McNemar’s test.

Preprocessing We focus on English webpages and first convert any text in Unicode to ASCII using Unidecode⁷. 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 use GloVe⁸ to learn word embeddings, although other pre-training methods may be used here without loss of generality. For experiments on academic homepages, we train 100-dimensional word embeddings using GloVe on FinegrainedName, with a window size of 15, minimum vocabulary count of 5, full passes through cooccurrence matrix of 15, and an initial learning rate of 0.05. For experiments on newswire articles, we initialise word embeddings with GloVe pretrained 100-dimensional embeddings which are pretrained on English Gigaword Fifth Edition⁹ containing a comprehensive archive of newswire text data.

5.1.1 Effectiveness of the Annotation Scheme. We first study the impact of using our fine-grained annotations with different fusion strategies (Figure 5) and different models. Specifically, four fusion strategies are tested:

- **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.
- **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 (This is what our proposed CogNN model does).

Three models are tested:

- **CRF:** Finkel et al. [10]. We use the Java implementation provided by the Stanford NLP group¹⁰. The software provides a generic implementation of linear chain CRF model.

⁷<https://pypi.org/project/Unidecode/>

⁸<https://nlp.stanford.edu/projects/glove/>

⁹<https://catalog.ldc.upenn.edu/LDC2011T07/>

¹⁰<https://nlp.stanford.edu/software/CRF-NER.html>

Fusion Strategies	Models	Annotations	F
No Fusion	CRF	BIE	41.15
		FML	54.98
No Fusion	Bi-LSTM-CRF	FI	50.32
		BIE	80.89
Early Fusion	Bi-LSTM-CRF	FML	82.11
		FI	81.71
Early Fusion	CRF	BIE × FML × FI	28.14
		BIE × FML × FI	62.65
Late Fusion	CRF	BIE ∪ FML	56.23
		BIE ∪ FI	56.01
		FML ∪ FI	56.38
		BIE ∪ FML ∪ FI	57.10
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]	87.04
		[BIE, FI]	87.34
In-network Fusion	CogNN (proposed)	[BIE, FML, FI]	88.26
In-network Fusion	CogNN (proposed)	[BIE, FI]	88.80

Table 2: Name level F1 score of using the fine-grained annotations in different ways with different models on FinegrainedName.

- **Bi-LSTM-CRF:** Huang et al. [12]. Specifically, 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. Specifically, it has 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.
- **CogNN:** Our proposed model is implemented 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 on the developing dataset. 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] and has a batch size of 32. The optimal hyperparameters are highlighted above in bold.

All above deep learning models are implemented using Theano¹¹ and Lasagne¹². All the optimal hyperparameters of above deep learning models are obtained with standard grid search on the same developing dataset with stochastic gradient descent. We stop training if the accuracy does not improve in 10 epochs.

Table 2 shows the results. CRF achieves an up to 13.83% improvement on F1 score when the fine-grained FML and FI annotations are provided without any fusion. The same trend can also be observed for Bi-LSTM-CRF. The reason is that academic homepages contain

¹¹<http://deeplearning.net/software/theano/>

¹²<https://lasagne.readthedocs.io/>

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* , *Pan-*
ning 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* , *Pan-*
ning 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* , *Pan-*
ning 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* , *Pan-*
ning 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.

person names with various forms and simple BIE annotations cannot reflect the patterns of name forms well. When examining the performance of using early fusion, we find that it is much worse than no fusion. This is expected as early fusion of different dimensions of name form 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 than no fusion with up to 15.95% and 2.56% improvements in F1 score for Bi-LSTM-CRF and CRF, respectively, which indicate that the separately trained models on different annotations have their own focuses. However, the underlying relationships among different name forms are not captured using late fusion. Our CogNN model, which uses in-network fusion, outperforms the best late

fusion baseline by up to 5.35% in F1-score. The reason is that our model can take advantage of the correlations between name form types when training and gain higher prediction confidence.

Overall, the fine-grained annotations can improve the performance of person name recognition on academic homepages using no fusion, late fusion and in-network fusion strategies. The neural-based models perform better than non-neural models and the in-network fusion can achieve the best results.

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 recognises all the names.

5.1.2 Effectiveness of the CogNN Model. Next, we study the performance of using the in-network fusion strategy with different multi-task learning models. Since there are no existing multi-task learning models that jointly learn the person name span and the person name form, we compare with the following joint models and variants of our proposed model:

- **JointNN:** We share the hidden layers between two tasks while keeping two task-specific output layers, which is similar to Caruana [4]. Specifically, a Bi-LSTM layer is used to get the hidden representations of the input. The hidden representations are passed two task-specific output layers to predict name forms and name span respectively. All the output is further fed into a linear chain CRF layer before perform predictions. And the two tasks are trained jointly to minimize the total loss. Bi-LSTM has 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.
- **DeepNN:** We use two successively deeper layers similar to Søgaard and Goldberg [25] for predicting the name form classes and the name spans respectively. Specifically, two Bi-LSTM layers are stacked for predicting the name form classes and the name spans respectively. The output of the first Bi-LSTM becomes the input of the second Bi-LSTM. All the output is further fed into a linear chain CRF layer to predict the tokens labels. And the two tasks are trained jointly to minimize the total loss. Each Bi-LSTM has 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.
- **ConcatNN:** We use the concatenating procedure similar to Ma et al. [16] to fuse the output of one task-specific layer with the input of another task-specific layer. Specifically, a Bi-LSTM layer is used to get the hidden representations of the input. The hidden representations are passed to the first task-specific output layer to predict name form classes. Then the output of the name form prediction layer as well as the initial hidden representations from Bi-LSTM layer are concatenated as the input of the second task-specific output layer to predict name spans. All the output is further fed into a linear chain CRF layer before perform predictions. And the two tasks are trained jointly to minimize the total loss. Bi-LSTM has a 100-dimensional hidden layer, a dropout layer

Models	Annotations	Token			Name
		R	P	F	F
JointNN	[BIE, FML]	88.02	89.82	88.91	81.04
		85.54	87.20	86.36	82.57
	[BIE, FI]	88.09	89.85	88.96	81.22
		86.69	88.93	87.80	82.74
DeepNN	[BIE, FML]	87.84	89.44	88.63	80.12
		84.39	86.86	85.61	81.24
	[BIE, FI]	87.88	89.54	88.70	80.40
		86.64	88.26	87.44	81.46
ConcatNN	[BIE, FML]	87.79	89.44	88.61	80.12
		84.35	86.83	85.57	81.22
	[BIE, FI]	87.81	89.51	88.65	80.31
		86.56	88.19	87.37	81.34
CoAttNN (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
CogNN (proposed)	[BIE, FML]	91.08	91.74	91.41	87.04
		85.66	88.73	87.17	87.34
	[BIE, FI]	94.63	94.23	94.43	88.26
		86.71	91.93	89.24	88.80

Table 3: Name level and token level performance of using different in-network fusion models on FinegrainedName.

with probability 0.5, a batch size of 32, and an initial learning rate of 0.01 with a decay rate of 0.05.

- **CoAttNN**: A variant of our proposed CogNN model with co-attention mechanism but not gated-fusion mechanism. The implementation, training procedures and hyperparameters are the same as those for CogNN.
- **CogNN**: Our proposed model.

All the optimal hyperparameters of above deep learning models are obtained with standard grid search on the same developing dataset with stochastic gradient descent. We stop training if the accuracy does not improve in 10 epochs.

Table 3 shows the results. All the models yield better results when jointly trained with input annotations BIE and FI on both the token level and the name level. JointNN achieves slightly better results at the name level compared with the no fusion Bi-LSTM-CRF model in Table 2. However, DeepNN and ConcatNN are worse than no fusion Bi-LSTM-CRF. The reason is that DeepNN and ConcatNN do not have a good mechanism to filter the learning signals. DeepNN uses successively deeper layers to connect two tasks and ConcatNN utilises straightforward concatenating procedures to share the informations. Noisy signals may be introduced into the training process and the propagation of error may reduce the effectiveness of our annotations. Both our proposed model CogNN and its variant CoAttNN outperform these multi-task models. CogNN outperforms the best baseline by up to 5.47% and 6.06% in F1-score at the token level and name level, respectively. This verifies the effectiveness of our co-attention and gated fusion mechanisms for utilising the fine-grained annotations. CogNN performs better than

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 (proposed)	[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 4: Token level performance of person name recognition on CoNLL-2003 using different models. For CogNN, we report the performance of the sub-networks that use fine-grained annotations.

CoAttNN since the sub-networks can share the learning signals while neither sub-network is overwhelmed by misleading signals.

5.2 Applicability on Newswire Articles

While not the focus of this study, we further show the applicability of our CogNN model and fine-grained annotation scheme on traditional newswire texts, which are different from academic homepages and mostly have consistent and complete syntax.

We use the CoNLL-2003 dataset which contains 1,393 annotated English newswire articles that focus on four types of named entities: person, location, organisation and miscellaneous entity. We use the training, developing, and testing datasets in CoNLL-2003 to train CRF, Bi-LSTM-CRF, and CogNN models with different annotations. The reported improvements are statistically significant with $p < 0.05$ as calculated using McNemar’s test.

This dataset does not come with fine-grained annotations. We add annotations using the same method described in Section 3 and compare the following combinations of annotations:

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

Since the fine-grained 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].

From Table 4, we see that neural models perform better than the non-neural model, which is consistent with the results in Section 5.1.1. 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. This can be explained by the share (Section 4.2) and balance (Section 4.3) procedures in CogNN, which reduce the possibility for wrong positive predictions during learning. 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, with a 1.43% improvement at the token level compared with the best baseline, which indicates that CogNN can well distinguish person name tokens from others. These results also indicate that only applying the fine-grained name form annotations on newswire data for the existing models is not enough. Our CogNN model is essential to make use of the extra name form information.

Overall, our approach is also applicable on formal English newswire articles and is especially helpful for improving the precision. However, the advantage of our approach is smaller than that on the academic homepages. The main reason is that the name forms in newswire articles are less flexible compared with those in academic homepages, which reduces the benefits of adding extra name form information.

6 CONCLUSION

We studied the person name recognition problem in user-generated free-form text. We propose a new name annotation scheme which gives fine-grained name annotations and a new model called CogNN to take advantage of the fine-grained annotation scheme via co-attention and gated fusion. We have also created the first dataset under the fine-grained annotation scheme, called FinegrainedName, for the research of name recognition.

Experiments on FinegrainedName dataset show that our annotations can be utilised in different ways to improve the person name recognition performance. Our CogNN model outperforms state-of-the-art NER models and multi-tasks models on utilising the fine-grained name form information. CogNN outperforms the best baseline by 5.35% in F1 score at the name level. We also study the applicability of our approach on well-formed newswire articles. CogNN outperforms state-of-the-art NER models and is especially advantageous in having high precision. CogNN improve the best baseline by 1.43% in precision at the token level.

For future work, we plan to investigate the fine-grained annotations and the CogNN model on other languages rather than English. We also plan to evaluate the performance of our approach on other types of datasets such as online forums and social media.

ACKNOWLEDGMENTS

This work is supported by the China Scholarship Council.

REFERENCES

- [1] Mohammed Aboagga and Mohd Juzaidin Ab Aziz. 2013. Arabic person names recognition by using a rule based approach. *Journal of Computer Science* 9, 7 (2013), 922–927.
- [2] Pablo Barrio, Gonalo Simões, Helena Galhardas, and Luis Gravano. 2014. REEL: A relation extraction learning framework. In *Proceedings of the 14th ACM/IEEE Joint Conference on Digital Libraries*. 455–456.
- [3] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. 1998. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Workshop on Very Large Corpora*.
- [4] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.
- [5] Jason Chiu and Eric Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association of Computational Linguistics* 4, 1 (2016), 357–370.
- [6] Yimeng Dai, Jianzhong Qi, and Rui Zhang. 2020. Joint Recognition of Names and Publications in Academic Homepages. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 133–141.
- [7] Fabrice Dugas and Eric Nichols. 2016. DeepNNER: Applying BLSTM-CNNs and Extended Lexicons to Named Entity Recognition in Tweets. In *Proceedings of the Workshop on Noisy User-generated Text*. 178–187.
- [8] 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* (2015).
- [9] Oliviu Felecan. 2012. *Name and naming: synchronic and diachronic perspectives*. Cambridge Scholars Publishing.
- [10] 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*. 363–370.
- [11] Abbas Ghaddar and Philippe Langlais. 2017. Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the International Joint Conference on Natural Language Processing*. 413–422.
- [12] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).
- [13] 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* 27, 2 (2015), 558–570.
- [14] Xiaozhong Liu, Yingying Yu, Chun Guo, Yizhou Sun, and Liangcai Gao. 2014. Full-text based context-rich heterogeneous network mining approach for citation recommendation. In *Proceedings of the 14th IEEE/ACM Joint Conference on Digital Libraries*. 361–370.
- [15] 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*. 1990–1999.
- [16] Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint Learning for Targeted Sentiment Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 4737–4742.
- [17] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*. 1064–1074.
- [18] Einat Minkov, Richard C Wang, and William W Cohen. 2005. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 443–450.
- [19] 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*. 2000–2008.
- [20] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes* 30, 1 (2007), 3–26.
- [21] 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 *Proceedings of the 24th International Conference on World Wide Web*. 597–602.
- [22] 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 *Workshop on Analytics for Noisy Unstructured Text Data*. 19–26.
- [23] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1532–1543.
- [24] Erik F Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050* (2003).
- [25] Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*. 231–235.
- [26] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 990–998.
- [27] Yiqing Zhang, Jianzhong Qi, Rui Zhang, and Chuandong Yin. 2018. PubSE: A Hierarchical Model for Publication Extraction from Academic Homepages. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1005–1010.