Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings

Nanyun Peng and **Mark Dredze**
Human Language Technology Center of Excellence
Center for Language and Speech Processing
Johns Hopkins University, Baltimore, MD, 21218
npeng1@jhu.edu, mdredze@cs.jhu.edu

Abstract

We consider the task of named entity recognition for Chinese social media. The long line of work in Chinese NER has focused on formal domains, and NER for social media has been largely restricted to English. We present a new corpus of Weibo messages annotated for both name and nominal mentions. Additionally, we evaluate three types of neural embeddings for representing Chinese text. Finally, we propose a joint training objective for the embeddings that makes use of both (NER) labeled and unlabeled raw text. Our methods yield a 9% improvement over a state-of-the-art baseline.

1 Introduction

Named entity recognition (NER), and more generally the task of mention detection[1], is an essential component of information extraction technologies: the first step before tasks such as relation extraction (Bunescu and Mooney, 2005) and entity linking (Dredze et al., 2010; Ratinov et al., 2011). A long line of work has focused on NER in both formal and informal domains (Collins and Singer, 1999; McCallum and Li, 2003; Nadeau and Sekine, 2007; Jin and Chen, 2008; He et al., 2012a), with recent efforts turning towards social media (Finin et al., 2010; Liu et al., 2011; Ritter et al., 2011; Fromreide et al., 2014; Li et al., 2012; Liu et al., 2012). While NER has included work on several languages, work on social media NER has largely focused on English language data.[2]

We consider NER on Chinese social media from the popular Sina Weibo service, both because of the popularity of the service (comparable in size to Twitter and previously used in NLP research (Ling et al., 2013)) and the challenges faced in processing Chinese language data. One approach is to utilize lexical embeddings to improve NER systems (Collobert and Weston, 2008; Turian et al., 2010; Passos et al., 2014), including for Twitter (Cherry and Guo, 2015). However, the use of embeddings for Chinese remains a challenge. Unlike most languages, we cannot easily assign an embedding to each Chinese word without automated segmentation, which may be unreliable, especially when we want to model informal text.[3] For this reason, state-of-the-art NER systems for Chinese do not tag words; they instead tag characters directly (Mao et al., 2008). While work has explored different embeddings for Chinese (Liu et al., 2014; Sun et al., 2014; Qiu et al., 2014; Chen et al., 2015), their inclusion in downstream tasks, such as NER, remains untested.

We explore several types of embeddings for Chinese text and their effect on Chinese social media NER. Specifically, we make the following contributions. 1) We present the first system for NER on Chinese social media using a new corpus based on Weibo messages. We consider both name and nominal mentions, with the goal of supporting downstream systems, such as coreference resolution. Notably, our results reveal that the gap between social media and traditional text for Chinese is much larger than similar corpora for English, suggesting this task as an interesting area of future work.[4] 2) We evaluate three types of embeddings for Chinese text based on their inclusion in a downstream task. We include results with and without fine-tuning. 3) We present a joint ob-

[1] Since we consider name and nominals, our work is closer to mention detection. For simplicity, we use the term NER.

[2] Etter et al. (2013) considered Spanish Twitter, which is quite similar to English from the standpoint of building models and features.

[3] Word segmentation performance is much worse on social media compared to formal text (Duan et al., 2012).

[4] Consider the overall F1 scores from Ritter et al. (2011), Cherry and Guo (2015) and Fromreide et al. (2014) compared to our best results in Table 2. This is despite the fact that Chinese NER performance on formal texts is similar to English.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 548–554,
Lisbon, Portugal, 17-21 September 2015.

jective that trains embeddings simultaneously for both NER and language modeling. Joint training yields better results than post-hoc fine-tuning.

2 NER for Chinese Social Media

Several SIGHAN shared tasks have focused on Chinese NER (Zhang et al., 2006; Jin and Chen, 2008; He et al., 2012b; Zhu et al., 2003; Fang et al., 2004; Zhang et al., 2006), though they have been restricted to formal text, e.g. news. NER for Chinese social media remains unexplored.[5]

As is the case for other languages, social media informality introduces numerous problems for NLP systems, such as spelling errors, novel words, and ungrammatical constructions. Chinese presents additional challenges, since it uses logograms instead of alphabets, and lacks many of the clues that a word is a name, e.g. capitalization and punctuation marks. The lack of explicit word boundaries further confuses NER systems. These problems are worse in social media, which has worse word segmentation. Additionally, typical Chinese corpora use exclusively traditional or simplified characters, whereas social media mixes them. Figure 1 demonstrates some challenges.

The baseline system for our task is our own implementation of Mao et al. (2008), which is the current state-of-the-art on the SIGHAN 2008 shared task (Jin and Chen, 2008). They use a CRF tagger with a BIOSE (begin, inside, outside, singleton, end) encoding that tags individual characters, not words, since word segmentation errors are especially problematic for NER (Zhang et al., 2006). Features include many common English NER features, e.g. character unigrams and bigrams, with context windows of size 5. See Mao et al. (2008) for complete details on their system.

Mao et al. (2008) use a two pass approach, training a CRF first for mention detection and using the resulting predictions as a feature for an NER system. Furthermore, they make extensive use of gazetteer features. For simplicity, we exclude the first pass mention detection and the gazetteer features, which make only small improvements to their overall performance. We note that other implementations of this system (Zhang et al., 2013) have been unable to match the performance reported in Mao et al. (2008). Similarly, our implementation yields results on SIGHAN 2008 similar to those reported in Zhang et al. (2013).[6] Overall, we take this tagger as representative of state-of-the-art for Chinese NER.

有好多好多的话想对你说李巾凡想要瘦瘦瘦成李帆我是想切开云朵的心
Have many many words to say to you Jinfan Li wanna thin thin thin to Fan Li I am a heart that want to cut the cloud

美得呀～顾天池苦逼青年杨素晗闵日记肖立伟嘻嘻嘻嘻嘻嘻美啊
Beautiful Tianchi Gu bitter youth Suhan Yang Riji Min Liwei Xiao hahahahahaha beautiful

看见前女友和她的新欢走在一起的时候，已经无处可躲了，只好硬着 头皮上去打招呼哎呀，好久不见，你儿子都这么高了。
When saw ex-girl friend and her new partner coming across, nowhere to hide, have to say hello, long time no see, your son grown up.

Figure 1: Examples of Weibos messages and translations with named (red) and nominal (blue) mentions.

3 Embeddings for Chinese Text

Lexical embeddings represent words in a continuous low dimensional space, which can capture semantic or syntactic properties of the lexicon: similar words would have similar low dimensional vector representations. Embeddings have been used to gain improvements in a variety of NLP tasks. In NER specifically, several papers have shown improvements by using pre-trained neural embeddings as features in standard NER systems (Collobert and Weston, 2008; Turian et al., 2010; Passos et al., 2014). More recently, these improvements have been demonstrated on Twitter data (Cherry and Guo, 2015). Embeddings are especially helpful when there is little training data, since they can be trained on a large amount of unlabeled data. This is the case for new languages and domains, the task we face in this paper.

However, training embeddings for Chinese is not straightforward: Chinese is not word segmented, so embeddings for each word cannot be trained on a raw corpus. Additionally, the state-of-the-art systems for downstream Chinese tasks, such as NER, may not use words.

We present three types of Chinese embeddings that will be trained on a large corpus of Weibo messages. These embeddings will be used as features in the NER system by adding a (real valued) feature for each dimension of the embedding for the current word/character.

Word Embeddings We train an embedding for each word type, the standard approach in other languages. We run a Chinese word segmentation

[5] Yang et al. (2014) consider a related problem of identifying product mentions in Weibo messages.

[6] Our implementation obtains an F1 of 88.63%.

system[7] over the raw corpus of Weibo messages. To create features, we first segment the NER data, and then lookup the embedding that matches the segmented word. Since the NER system tags characters, we add the same word embedding features to each character in the word.

Character Embeddings We learn an embedding for each character in the training corpus (Sun et al., 2014; Liu et al., 2014).This removes the dependency on pre-processing the text, and better fits our intended use case: NER tagging over characters. Since there are many fewer characters than words, we learn many fewer embeddings. On the one hand, this means fewer parameters and less over-fitting. However, the reduction in parameters comes with a loss of specificity, where we may be unable to learn different behaviors of a character in different settings. We explore a compromise approach in the next section. These embeddings are directly incorporated into the NER system by adding embedding features for each character.

Character and Position Embeddings Character embeddings cannot distinguish between uses of the same character in different contexts, whereas word embeddings fail to make use of characters or character n-grams that are part of many words. A compromise is to use character embeddings that are sensitive to the character's position in the word (Chen et al., 2015). We first word segment the corpus. For each character in each word, we add a positional tag, e.g. the first/second/etc. character in the word, yielding multiple embeddings per character. We learn separate embeddings for each positionally tagged character. To use these embeddings as features, we segment the NER text, obtain position tags for each character, and add features for the corresponding embedding.

These three methods lead to 179,809 word embeddings, 10,912 character embeddings, and 24,818 character with position embeddings.

3.1 Fine Tuning

For each of the embeddings, we fine-tune pre-trained embeddings in the context of the NER task. This corresponds to initializing the embeddings parameters using a pre-trained model, and then modifying the parameters during gradient updates of the NER model by back-propogating gradients. This is a standard method that has been previously explored in sequential and structured prediction problem (Collobert et al., 2011; Zheng et al., 2013; Yao et al., 2014; Pei et al., 2014).

3.2 Joint Training Objectives

Fine-tuning has a disadvantage: it can arbitrarily deviate from the settings obtained from training on large amounts of raw text. Recent work has instead tuned embeddings for a specific task, while maintaining information learned from raw text. Yu and Dredze (2014) use multi-part objectives that include both standard unlabeled objectives, such as skip-gram models in word2vec, and task specific objectives. Jointly training the embeddings with the multi-part objectives allows the fine-tuned embeddings to further influence other embeddings, even those that do not appear in the labeled training data. This type of training can help improve OOVs (Yu and Dredze, 2015), an important aspect of improving social media NER.

We propose to jointly learn embeddings for both language models and the NER task. The modified objective function (log-likelihood) for the CRF is given by:

$$\begin{aligned}&\mathcal{L}_s(\boldsymbol{\lambda}, \boldsymbol{e_w}) \\ =&\frac{1}{K}\sum_k \left[\log \frac{1}{Z(x)^k} + \sum_j \lambda_j F_j(\boldsymbol{y}^k, \boldsymbol{x}^k, \boldsymbol{e_w})\right],\end{aligned}$$

where K is the number of instances, $\boldsymbol{\lambda}$ is the weight vector, $\boldsymbol{x}^k$ and $\boldsymbol{y}^k$ are the words and labels sequence for each instance, $\boldsymbol{e_w}$ is the embedding for a word/character/character-position representation w, $Z(x)^k$ is the normalization factor for each instance, and $F_j(\boldsymbol{y}^k, \boldsymbol{x}^k, \boldsymbol{e_w}) = \sum_{i=1}^n f_j(y_{i-1}^k, y_i^k, \boldsymbol{x}^k, \boldsymbol{e_w}, i)$ represents the feature function in which j denotes different feature templates and i denotes the position index in a sentence. This differs from a traditional CRF in that the feature function depends on the additional variables $\boldsymbol{e_w}$, which are the embeddings (as defined above). As a result, the objective is no longer log-linear, but log-bilinear [8].

[7] We use Jieba for segmentation: `https://github.com/fxsjy/jieba`

[8] It is log-bilinear because the log-likelihood takes the form $f(x, y) = axy + bx + cy$, where x, y are variables and a, b, c are coefficients. In this case, x is the feature weight and y is the embedding; both of them are vectors. Taking the partial derivative with respect to any one of the variables, one gets a constant (wrt that variable). This satisfies the definition of log-bilinear functions.

The second term is the standard skip-gram language model objective (Mikolov et al., 2013):

$$\mathcal{L}_u(\boldsymbol{e_w}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t), \quad (1)$$

where

$$p(w_i|w_j) = \frac{\exp\left(\boldsymbol{e}_{\boldsymbol{w_i}}^T \boldsymbol{e}_{\boldsymbol{w_j}}\right)}{\sum_{i'}\exp\left(\boldsymbol{e}_{\boldsymbol{w_{i'}}}^T \boldsymbol{e}_{\boldsymbol{w_j}}\right)}.$$

The first objective is notated $\mathcal{L}_s$ for "supervised" (trained on labeled NER data), and the second is $\mathcal{L}_u$, "unsupervised" (trained on raw text.) Both objectives share the same variables $\boldsymbol{e_w}$. The overall goal is to maximize their weighted sum:

$$\arg\max_{e_w} = \mathcal{L}_s(\boldsymbol{\lambda}, \boldsymbol{e_w}) + C\mathcal{L}_u(\boldsymbol{e_w}) \quad (2)$$

where C is a tradeoff parameter.

3.3 Parameter Estimation

We pre-trained embeddings using word2vec (Mikolov et al., 2013) with the skip-gram training objective and NEC negative sampling. Unless otherwise stated, we used word2vec's default parameter settings. All embeddings were 100-dimensional, and we used the same embeddings for the input and output parameters in the skip-gram objective. We optimized the joint objective (2) using an alternative optimization strategy: we alternated 30 iterations of CRF training on the NE labeled data and 5 multi-threaded passes through both the labeled and unlabeled data for the skip-gram objective. We avoided over-fitting using early-stopping. For simplicity, we set $C = 1$ for (2). The CRF was trained using stochastic gradient descent with an L2 regularizer. All model hyper-parameters were tuned on dev data.

We use the off-the-shelf tool word2vec (Mikolov et al., 2013) to do skip-gram training for language model, and implement our own CRF model to modify the embeddings. We optimize (2) by alternating the optimzation of each of the two objectives.

4 Weibo NER Corpus

We constructed a corpus of Weibo messages annotated for NER. We followed the DEFT ERE (Linguistics Data Consortium, 2014) [9] annotation guidelines for entities, which includes four major semantic types: person, organization, location and geo-political entity. We annotated both name and nominal mentions. Chinese pronoun mentions can be easily recognized with a regular expression. We used Amazon Mechanical Turk, using standard methods of multiple annotators and including gold examples to ensure high quality annotations (Callison-Burch and Dredze, 2010).

	Mentions		
Entity Type	**Name**	**Nominal**	**Total**
Geo-political	243	0	243
Location	88	38	126
Organization	224	31	255
Person	721	636	1,357

Table 1: Mention statistics for the Weibo NER corpus.

Our corpus includes 1,890 messages sampled from Weibo between November 2013 and December 2014. Rather than selecting messages at random, which would yield a small number of messages with entities, we selected messages that contained three or more (segmented) words that were not in a fixed vocabulary of common Chinese words. Initial experiments showed this gave messages more likely to contain entities.

Table 1 shows statistics of the final corpus. We divided the corpus into 7 folds, each with 127 messages, where each message corresponds to a single instance. We use the first 5 folds for train, the 6th for development, and the 7th for test. We make our code and the annotated corpus available.[10]

We constructed an additional corpus of unlabeled messages for training the embeddings. We randomly selected 2,259,434 messages from the same time period as above.

5 Experiments

We evaluate our methods under two settings: training on only name mentions, and training on both name and nominal mentions. We re-train the Stanford NER system (Finkel et al., 2005) as a baseline; besides, we also evaluate our implementation of the CRF from Mao et al. (2008) as described in §2 as *Baseline Features*. To this baseline, we add each of our three embedding models: *word*, *character*, *character+position* (as described in §3), and report results on the modified

[9] See Aguilar et al. (2014) for a comparison of DEFT ERE with other common standards.

[10] https://github.com/hltcoe/golden-horse

Method	Dev						Test					
	Without Fine Tuning			With Fine Tuning			Without Fine Tuning			With Fine Tuning		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Stanford	63.51	23.27	34.06				55.70	22.86	33.06			
Baseline Features	63.51	27.17	38.06				56.98	25.26	35.00			
+ word	65.71	26.59	37.86	70.97	25.43	37.45	56.82	25.77	35.46	64.94	25.77	36.90
+ character	53.54	30.64	38.97	58.76	32.95	42.22	56.48	31.44	40.40	57.89	34.02	42.86
+ character+position	60.87	32.37	42.26	61.76	36.42	**45.82**	61.90	33.51	43.48	57.26	34.53	43.09
Joint (cp)				57.41	35.84	44.13				57.98	35.57	**44.09**
Stanford	72.39	31.80	44.19				63.96	22.19	32.95			
Baseline Features	71.94	33.22	45.45				60.16	23.87	34.18			
+ word	69.66	33.55	45.29	70.67	35.22	47.01	59.40	25.48	35.67	60.68	22.90	33.26
+ character	58.76	32.95	42.22	66.88	35.55	46.42	58.28	28.39	38.18	55.15	29.35	38.32
+ character+position	73.43	34.88	47.30	69.38	36.88	48.16	65.91	28.06	39.37	62.33	29.35	39.91
Joint (cp)				72.55	36.88	**48.90**				63.84	29.45	**40.38**

Table 2: NER results for name mentions (top) and name + nominal mentions (bottom).

CRF model with and without fine-tuning. We also report results for the joint method trained with the character+position model (cp), which performed the best on dev data for joint training.

General Results Table 2 shows results for both dev (tuned) and test (held out) splits. First, we observe that the results for the baseline are significantly below those for SIGHAN shared tasks as well as the reported results on Twitter NER, showing the difficulty of this task. In particular, recall is especially challenging. Second, all embeddings improve the baseline on test data, but the character + position model gets the best results. Fine-tuning improves embedding results, but seems to overfit on dev data. Finally, our joint model does the best in both conditions (name and name+nominal) on test data, improving over fine-tuning, yielding up to a 9% (absolute) improvement over a strong baseline.

Effect of Embeddings We expect improvements from embeddings to be larger when there is less training data. Figure 2 shows F1 on dev data for different amounts of training data, from 200 instances up to 1400, for the character + position embeddings versus the baseline model. We see that for both settings, we see larger improvements from embeddings for smaller training sets.

Error Analysis Since the results are relatively low, we conducted an error analysis by randomly sampling 150 error items and manually looking through them. Among the 150 examples, 65 are annotation errors, majorly cause by annotators neglecting some mentions, this contributes 43% of the errors. The second largest error source are the person names: Chinese person names are very flexible and nearly every character can be used in given names, this makes recognizing person names challenging and contributes to 9% of our errors. The following largest source of error are transliterated foreign names, which contributes to 7% of the errors. Other sources including boundary error, type error, name abbreviation, nicknames, etc.

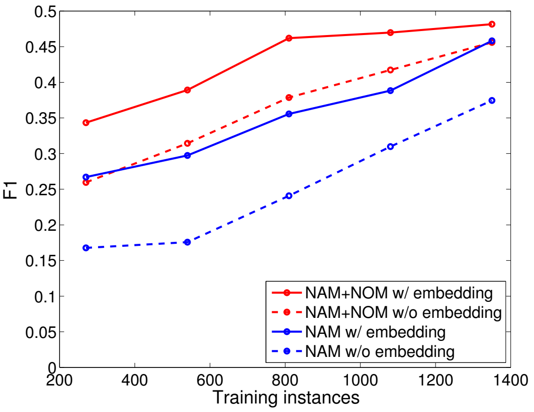

Figure 2: Dev F1 for varying number of training instances.

6 Conclusion

Our results show that NER for Chinese social media remains a challenging task, results lag behind both formal Chinese text and English Twitter. Nevertheless, our embeddings, combined with our joint training objective, provide a large improvement over a state-of-the-art model.

Acknowledgement We would like to thank the reviewers for their helpful comments and perspectives. We thank Mo Yu, Kevin Duh, Jiang Guo and Wenzhe Pei for the insightful discussions and Xuezhe Ma for help running the Stanford baseline.

References

Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. 2014. A comparison of the events and relations across ACE, ERE, TAC-KBP, and FrameNet annotation standards. In *ACL Workshop: EVENTS*.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 724–731. Association for Computational Linguistics.

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with Amazon's Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Mechanical Turk*.

Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015. Joint learning of character and word embeddings. In *International Joint Conference on Artificial Intelligence (IJCAI'15)*.

Colin Cherry and Hongyu Guo. 2015. The unreasonable effectiveness of word representations for twitter named entity recognition. In *North America Chapter of Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 100–110. Citeseer.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning (ICML)*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Conference on Computational Linguistics (Coling)*.

Huiming Duan, Zhifang Sui, Ye Tian, and Wenjie Li. 2012. The cips-sighan clp 2012 chineseword segmentation onmicroblog corpora bakeoff. In *Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 35–40, Tianjin, China, December. Association for Computational Linguistics.

David Etter, Francis Ferraro, Ryan Cotterell, Olivia Buzek, and Benjamin Van Durme. 2013. Nerit: Named entity recognition for informal text. Technical report, Technical Report 11, Human Language Technology Center of Excellence, Johns Hopkins University, July.

Xiaoshan Fang, Jianfeng Gao, and Huanye Sheng. 2004. A semi-supervised approach to build annotated corpus for chinese named entity recognition. In Oliver Streiter and Qin Lu, editors, *ACL SIGHAN Workshop 2004*, pages 129–133, Barcelona, Spain, July. Association for Computational Linguistics.

Tim Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in twitter data with crowdsourcing. In *NAACL Workshop on Creating Speech and Language Data With Mechanical Turk*.

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

Hege Fromreide, Dirk Hovy, and Anders Søgaard. 2014. Crowdsourcing and annotating NER for Twitter# drift. In *LREC*.

Zhengyan He, Houfeng Wang, and Sujian Li. 2012a. The task 2 of cips-sighan 2012 named entity recognition and disambiguation in chinese bakeoff. In *Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 108–114, Tianjin, China, December. Association for Computational Linguistics.

Zhengyan He, Houfeng Wang, and Sujian Li. 2012b. The task 2 of cips-sighan 2012 named entity recognition and disambiguation in chinese bakeoff. In *Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 108–114, Tianjin, China, December. Association for Computational Linguistics.

Guangjin Jin and Xiao Chen. 2008. The fourth international chinese language processing bakeoff: Chinese word segmentation, named entity recognition and chinese pos tagging. In *Sixth SIGHAN Workshop on Chinese Language Processing*, page 69. Citeseer.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012. Twiner: Named entity recognition in targeted twitter stream. In *SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 721–730, New York, NY, USA. ACM.

Wang Ling, Guang Xiang, Chris Dyer, Alan Black, and Isabel Trancoso. 2013. Microblogs as parallel corpora. In *Association for Computational Linguistics (ACL)*. Association for Computational Linguistics.

Linguistics Data Consortium. 2014. DEFT ERE Annotation Guidelines: Entities.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Association for Computational Linguistics (ACL)*, pages 359–367. Association for Computational Linguistics.

Xiaohua Liu, Ming Zhou, Furu Wei, Zhongyang Fu, and Xiangyang Zhou. 2012. Joint inference of named entity recognition and normalization for tweets. In *Association for Computational Linguistics (ACL)*, ACL '12, pages 526–535, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaodong Liu, Kevin Duh, Yuji Matsumoto, and Tomoya Iwakura. 2014. Learning character representations for chinese word segmentation. In *NIPS 2014 Workshop on Modern Machine Learning and Natural Language Processing*.

Xinnian Mao, Yuan Dong, Saike He, Sencheng Bao, and Haila Wang. 2008. Chinese word segmentation and named entity recognition based on conditional random fields. In *IJCNLP*, pages 90–93.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *North America Chapter of Association for Computational Linguistics (NAACL)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems (NIPS)*, pages 3111–3119.

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

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. *CoRR*, abs/1404.5367.

Wenzhe Pei, Tao Ge, and Chang Baobao. 2014. Max-margin tensor neural network for chinese word segmentation. In *Association for Computational Linguistics (ACL)*.

Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Co-learning of word representations and morpheme representations. In *Conference on Computational Linguistics (Coling)*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Association for Computational Linguistics (ACL)*, pages 1375–1384. Association for Computational Linguistics.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1524–1534. Association for Computational Linguistics.

Yaming Sun, Lei Lin, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-enhanced chinese character embedding. In *Neural Information Processing Systems (NIPS)*, pages 279–286. Springer.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Association for Computational Linguistics (ACL)*, pages 384–394. Association for Computational Linguistics.

Xianxiang Yang, Heyan Huang, Xin Xin, Quanchao Liu, and Xiaochi Wei. 2014. Domain-specific product named entity recognition from chinese microblog. In *Computational Intelligence and Security (CIS), 2014 Tenth International Conference on*, pages 218–222. IEEE.

Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. 2014. Recurrent conditional random field for language understanding. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4077–4081. IEEE.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Association for Computational Linguistics (ACL)*, pages 545–550.

Mo Yu and Mark Dredze. 2015. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242.

Suxiang Zhang, Ying Qin, Juan Wen, and Xiaojie Wang. 2006. Word segmentation and named entity recognition for sighan bakeoff3. In *Fifth SIGHAN Workshop on Chinese Language Processing*, pages 158–161, Sydney, Australia, July. Association for Computational Linguistics.

Longkai Zhang, Houfeng Wang, Xu Sun, and Mairgup Mansur. 2013. Exploring representations from unlabeled data with co-training for chinese word segmentation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for chinese word segmentation and pos tagging. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 647–657.

Xiaodan Zhu, Mu Li, Jianfeng Gao, and Chang-Ning Huang. 2003. Single character chinese named entity recognition. In *Second SIGHAN Workshop on Chinese Language Processing*, pages 125–132, Sapporo, Japan, July. Association for Computational Linguistics.