DSMER: A Deep Semantic Matching based Framework for Named Entity Recognition

No Author Given

No Institute Given

Abstract. The task of named entitiy recognition(NER) is normally regarded as a sequence labeling problem. However, this kind of NER framework does not utilize any prior knowledge. In this paper, we propose a novel framework called **DSMER**, which stands for **D**eep **S**emantic **M**atching based Framework for Named **E**ntity **R**ecognition. DSMER is a two-phase framework: 1) detect the boundary and extract candidate span, 2) calculate the distance between candidates and entity type. Meanwhile, the representation of each entity type is encoded from its corresponding annotation rules and example set. Since the combination of various textual data, DSMER has the ability to integrate informative prior knowledge. Additionally, we introduce the Word Mover's Distance to measure the similarity between sequences of different lengths. We conduct experiments on CoNLL 2003 and OntoNotes 5.0 dataset. Experimental result shows our approach achieve state of the art performance, and demonstrates the effectiveness of the proposed framework.

Keywords: Named entity recognition · Semantic matching · Entity boundary detection

1 Introduction

Named entity recognition(NER) is a subtask of information extraction, which refers to a task of detecting spans from text and classifying their types. Among mainstream research methods, the NER task is commonly considered as a sequence labeling problem[6, 1, 12, 24, 3]: for each token of the input sequence, predict a class label assigned to it. The sequence labeling framework solves NER with an end-to-end way, and has achieved effective results on various datasets. However, this formalization of NER is quite different from the recognition process of humans. Figure 1 shows human conventions when annotating entity labels. The annotation rules should first be summarized according to human experience and background knowledge. Then the annotator would try to annotate a few examples according to the rules and adjust the rules based on example set. Finally, the annotation rule and the example set are combined together as prior knowledge to carry out the complete data annotation process.

Inspired by human convention, we propose a new framework that is capable of integrating knowledge from annotation rules and example set. Instead of treating

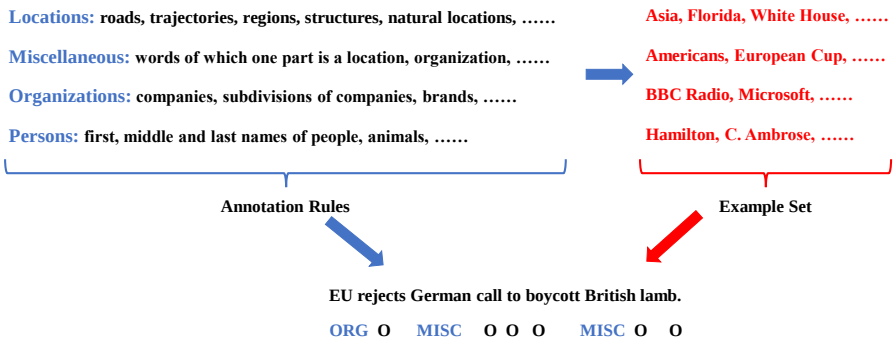

Fig. 1. Human annotation process of named entity extraction and recognition. The annotation rules and example set are chosen from CoNLL 2003 dataset.

NER as a sequence labeling problem, we formulate it as a deep semantic matching task[5, 22, 14]. Following the principle of two-phase framework[10], we design three sub-modules: 1) Prior Knowledge Encoding: encode the representation of entity types from annotation rules and example set, 2) Boundary Detection: predict the start and end index of candidate entities and extract the representation of them, 3) Semantic Matching: calculate the similarity between candidate span and different types. The input sentence is first sent to the boundary detection module to extract a set of candidates. At the same time, we combine the annotation rules and example set corresponding to each entity type, and encode them to obtain the representation vector of the entity type. In the second phase, we input the representation vector of each candidate span and entity types into the semantic matching module. The label of candidate span is determined by the similarity of semantic representation between them. In order to measure the similarities between spans and entity types with different lengths, we introduce Word Mover's Distance(WMD)[7], which is a novel distance function based on Earth Mover's Distance(EMD)[20].

We conduct experiments on public NER datasets to show the effectiveness of our approach. Experimental results show that our deep semantic matching based framework outperforms both sequence labeling and machine reading comprehension based frameworks. In addition, we also conducted ablation experiments to verify the influence of different prior knowledge on our method. Our main contributions are summarized as follows:

- We propose a novel deep semantic matching based NER framework which exploits prior knowledge and is closer to human annotation behavior.
- Our boundary detection module overcomes the problem of excessive sample size and imbalance between positive and negative samples in previous entity classification methods.
- We first introduce the Word Mover's Distance into semantic modeling to directly measure the similarity of unequal length sequences.

2 Related Work

Named Entity Recognition(NER). Traditional entity recognition methods treat NER task as a sequence labeling problem and use CRFs as the backbone[8, 25]. More recently, neural models was introduced for NER under the sequence labeling framework. Collobert et al.[2] presented a CNN-CRF structure, Huang et al.[6] first applied BiLSTM-CRF model to NER, Lample et al.[9] proposed a BiLSTM-CRF model with character-based word representations, Ma and Hovy[12] and Chiu and Nchols[1] extend the BiLSTM-CRF structure with a character CNN to extract features, Sturbell et al.[24] proposed a iterated dilated convolutions NER model to accelerate the parallel computing on GPU. With the rise of large-scale pre-trained language models[16, 3, 18, 19], sequence labeling style NER models achieved state of the art performance. In addition to the recognition of flat entities, there are also some studies on nested entities. Previous work was mainly based on the two-phase framework, which first enumerated all possible spans, and then predicted entity type. According to this idea, Sohrab et al.[23] proposed a deep exhaustive model which limited all the regions within a specified maximum length. Zheng et al.[28] leveraged the entity boundaries to improve the performance of identifying entities. Moreover, Li et al.[11] migrate the NER task to machine reading comprehension framework and make the model compatible with recognizing both flat and nested entities.

Semantic Textual Matching. Huang et al.[5] first proposed the deep structured semantic model(DSSM) in web search area to map a query to its relevant documents at semantic level. The principle is that the query and documents are embedded to semantic vectors, and the distance between them is calculated by cosine distance, and finally the semantic matching model is trained. Aiming at the shortcoming of the bag-of-words model used by DSSM, Shen et al.[22] replaced the DNN with CNN, so that the model can make up for the loss of context. Since the CNN based model can not capture the feature from long term context, Palang et al.[14] introduced the LSTM to overcome the problem.

Word Mover's Distance. Kusner et al.[7] proposed the document distance matrix called Word Mover's Distance(WMD), which can be cast as an instance of the Earth Mover's Distance(EMD). In statistics, the EMD is a measure of the distance between two probability distributions over a region D. If the distributions are interpreted as two different ways of piling up a certain amount of dirt over the region D, the EMD is the minimum cost of turning one pile into the other, where the cost is assumed to be the amount of dirt moved times the distance by which it is moved. The concept of EMD was first introduced by Gaspard Monge[13] in the context of transportation theory. The use of the EMD as a distance measure for monochromatic images was described by Peleg et al[15]. Stolfi et al.[20] first proposed the name "Earth Mover's Distance". Rubner et al.[20] first used it on image retrieval task to measure the distance between images.

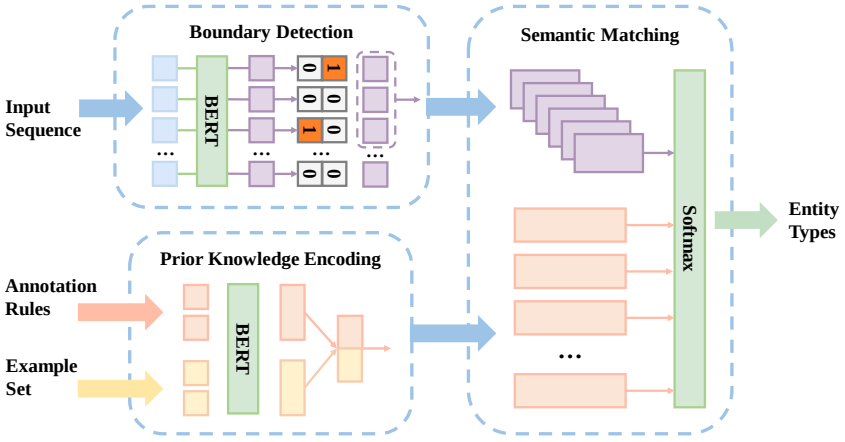

Fig. 2. Overview of Deep Semantic Matching Entity Recognition Framework(DSMER).

3 NER as Semantic Matching

Figure 2 shows the architecture of DSMER. Given an input sequence $X = \{x_1, x_2, ..., x_l\}$, where l denotes the length of the sequence, we need to extract every candidate entity span from X, and then assign a label $t \in T$ to it through semantic matching model, where T is the set of all entity types. The framework is a two-phase model composed of three modules. In the first phase, the representations of candidate spans are extracted, and entity types are encoded through prior knowledge like annotation rules, example set, etc. In the second phase, we separately measure the similarity of each candidate span and all entity types through the semantic matching module. BERT[3] is used as the encoder in each module of the first phase. The following subsections will describe the detail of different modules in DSMER.

3.1 Prior Knowledge Encoding

The prior knowledge encoding procedure is important for DSMER since the external text like annotation rules contains informative semantics and has a significant impact on the final result. Seyler et al[21] discussed the importance of different categories of external knowledge for performing NER task, including Name-based, Knowledge-Base-based and Entity-based. Besides, Li et al[11] encoded annotation guideline notes as reference queries and achieved a vast amount of performance boost over current SOTA models. In this paper, we take both annotation rules and example set of entity mentions as prior knowledge. Annotation rules are not only the guidelines provided to the annotators of the dataset but the Wikipedia definition and synonyms of entity type.

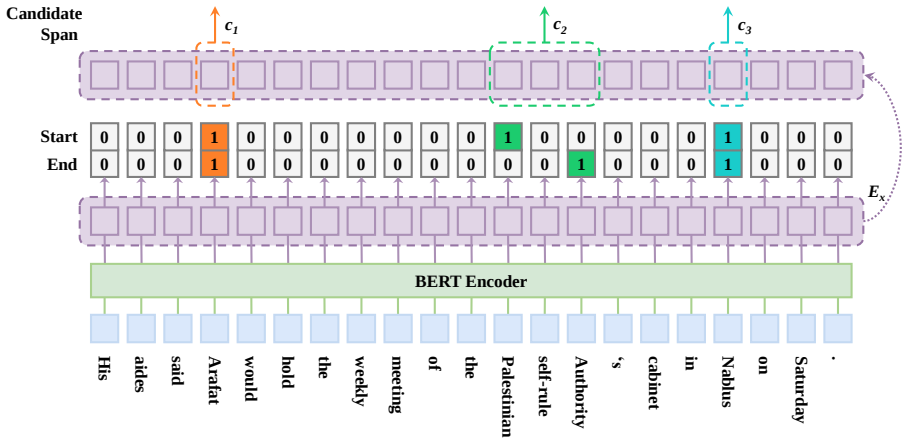

Fig. 3. The workflow of Boundary Detection module.

Assuming E_t is the representation of entity type t. Given a list of annotation rules $R = [r_1, r_2, ..., r_n]$ and a set of example mentions $S = s_1, s_2, ..., s_m$, where n and m denote the number of rules and mentions. We first encode the annotation rules and the example set separately, and then concatenate the hidden representations of them as E_t:

$$E_t = tanh(W_t[E_R, E_S] + b_t) \tag{1}$$

where E_R and E_S are both encoded by BERT, W_t and b_t is the trainable weight and bias:

$$\begin{aligned} E_R &= \frac{1}{n}\sum_{i=1}^{n} BERT(r_i) \\ E_S &= \frac{1}{m}\sum_{j=1}^{m} BERT(s_j) \end{aligned} \tag{2}$$

In particular, we only take the output context representation of [CLS] position to calculate the average representation of rules and mentions with different lengths.

3.2 Boundary Detection

The boundary detection module is designed to recognize all possible candidate span in the input sentence X. Previous work[23, 28] simply set a maximum length of entity, and enumerated all possible spans as a candidate set, which caused the imbalance of positive and negative samples and the problem that the number of

samples increased exponentially with the length of the input sequence. To tackle this problem, we use two binary classifiers: one to predict whether each token is the start index or not, the other to predict the end index. Figure 3 shows the architecture of boundary detection module.

Given the representation matrix E_X output from BERT,

$$E_X = BERT(X), \quad E \in R^{n \times d} \tag{3}$$

where d is the dimension size of the output layer of BERT. The module adopts two fully-connected layers to detect the start and end position indexed respectively by assigning each token a binary tag (0/1).

$$P^i_{start} = \sigma(W_{start} E_{x_i} + b_{start}) \tag{4}$$

$$P^i_{end} = \sigma(W_{end} E_{x_i} + b_{end}) \tag{5}$$

where P_i^{start} and P_i^{end} represent the probability of identifying the i-th token in the input sequence X as the start and end position of a candidate span.

After predicting the start and end positions, we combine start index and each end index greater than it as a candidate span c, and extract the representation $E_c = \{E_{x_{start}}, E_{x_{end}}\}$ for semantic matching in next phase.

3.3 Semantic Matching

The semantic matching module is a deep neural network following DSSM[5] and CLSM[22]. Figure 4 shows the structure of this module. Considering the ground truth type $t^+ \in T$, which is closer to candidate span than other types in semantic space. We can simply use the deep semantic model to calculate the relevance of each pair of (c, t).

To directly measure the difference between two sequences of different lengths, we introduce the Word Mover's Distance. Considering the embedding of entity span E_c and the embedding of entity type E_t, the cost of WMD can be calculated by:

$$\begin{aligned} &\min_{d_{i,j} \geq 0} \sum_{i,j} d_{i,j} \left\| e_i - e'_j \right\| \\ &\text{s.t.} \sum_i d_{i,j} = \frac{1}{l_c}, \sum_j d_{i,j} = \frac{1}{l_t} \end{aligned} \tag{6}$$

where l_c and l_t are the length of candidate span and entity type vector, e_i and $e_j^{'}$ are i-th and j-th embedding vector in E_c and E_t. The semantic relevance score between a candidate c and a entity type t is then measured as:

$$M(c, t) = WMD(E_c, E_t) \tag{7}$$

After obtaining the semantic relevance score, we compute the posterior probability through a softmax function:

$$P(t|c) = \frac{exp(M(c, t))}{\sum_{t' \in T} exp(M(c, t'))} \tag{8}$$

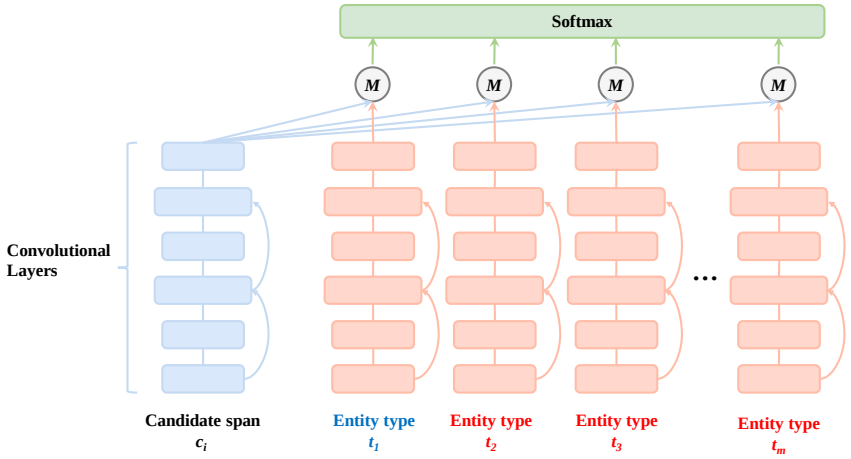

Fig. 4. The structure of deep semantic matching module. Let t_1 be the matched entity type of candidate span c_i, and all others are negative examples. Send their representations into the model, calculate the similarity of each pair, and finally output the posterior probability through softmax layer.

In particularly, we adopt shortcut connections every other layer parallel to linear transformation before the activation function, as in ResNet[4]. This helps the training of a deep neural network.

3.4 Loss Function

At the training time, X is paired with two label sequences Y_{start} and Y_{end} that represent the ground-truth label of each token x_i. We use the binary cross-entropy loss for the prediction of start and end index:

$$L_{start} = BCE(P_{start}, Y_{start}) \tag{9}$$

$$L_{end} = BCE(P_{end}, Y_{end}) \tag{10}$$

The parameters of semantic matching module are estimated to maximize the likelihood of t^+. Equivalently, we need to minimize the following loss function:

$$L_{match} = -log \prod_{(c,t^+)} P(t^+|c) \tag{11}$$

The overall training objective to be minimized is as follows:

$$L = \alpha L_{start} + \beta L_{end} + \gamma L_{match} \tag{12}$$

where $\alpha, \beta, \gamma \in [0, 1]$ are the hyper-parameters to control the contributions of different modules. The three losses from two phrase of DSMER are jointly trained with parameters shared at BERT.

At the test time, candidate spans are first extracted based on boundary detection module. Then the semantic matching model is used to measure the similarity of candidate span and entity types, leading to the final answers.

4 Experiments and Discussions

In this section, we conduct experiments on several public datasets and compare DSMER with models of different NER framework. The following subsections will describe the implementation details and ablation analysis in detail.

4.1 Datasets and Preprocessing

Datasets. We use corpora provided by CoNLL 2003 Shared Task[26] and OntoNotes 5.0[17] to evaluate the model presented in this paper. CoNLL2003 is an English dataset with four types of named entities: Location, Organization, Person and Miscellaneous. And Ontonotes 5.0 includes 18 types of named entity, consisting of 11 types(Person, Organization, etc) and 7 values(Date, Percent, etc).

Data Reconstruction. Most NER corpora provide the labeled data for sequence labeling framework. Different from other NER frameworks, the DSMER needs to extract the rules from annotation document and random sampling part of entities for each type from raw dataset. For each train set, we random choose 10% annotated entities as example set, and remain 90% as train set as usual. The statistical details are listed in Table 1. To further experiment, we also test the ratio of 5%, 15%, 20% and 40% in following experiments.

Corpus	Example set	Train set	Dev set	Test set
CoNLL 2003[26]	2,350	21,149	5,942	5,648
OntoNotes 5.0[17]	8,183	73,645	11,066	11,257

Table 1. The entity statistics of preprocessed datasets.

As for the boundary detection module, training data requires binary label for start and end indexes. The ground truth label of entities is converted into two lists for start and end, which are set to 1 only when the token belongs to the boundary of the entity.

4.2 Implementation Details

We use fastNLP[1] to implement the model and evaluate all experiments on datasets. The DSMER model uses BERT as the skeleton. In order to ensure the effectiveness of the semantic matching method, we only use BERT-base as a semantic encoder in all the comparison experiments below. All experiments are run on Nvidia Tesla V100 GPU, which has 32GB memory to accommodate larger batch size.

Parameters	Values
Optimizer	AdamW
Initial learning rate	2e-5
Gradient Clipping value	1.0
Global Dropout rate	0.5
Warmup rate	0.1
Batch size	64
Training epoch	20
Layer of DSM	5
Hidden dim of DSM input	300

Table 2. Hyper-parameter settings.

We train the model using *AdamW* optimizer with an initial learning rate of 2e-5, and use warm-up mechanism with linear schedule to adjust the learning rate. To avoid gradient explosion problem, the gradient clip method is used as a callback in training. The semantic matching module of DSM follows the deep structured nerual network in [5], We use 5 fully connected layers, and the input dimension of candidate span and entity types is 300. All other details of hyperparameters are listed in Table 2.

4.3 Experimental Results

In order to verify the effectiveness of DSMER, we choose the classic and SOTA models under different NER frameworks for comparison. For sequence labeling framework, we change the encoder module connected to CRF in range of Bi-LSTM, IDCNN and Transformer. And BERT is also introduced for the pretrain+finetune framwork. Finally we use the MRC-BERT model to stand the machine reading comprehension framework. All comparison results on CoNLL2003 and Ontonotes 5.0 are listed in Table 3 and 4.

Because we use BERT-base as the model skeleton, we respectively give the experimental results without using the annotation rule and example set to verify the effectiveness of the semantic matching framework.

[1] https://github.com/fastnlp/fastNLP

Framework	Model	Precision	Recall	F1
Sequence Labeling	BiLSTM + CRF[6]	-	-	90.43
	IDCNN + CRF[24]	-	-	90.54
	TENER w/CNN-char [27]	-	-	91.45
	BERT-Tagger[3]	-	-	92.80
Reading Comprehension	MRC-BERT[11]	92.33	94.61	93.04
Semantic Matching	Ours w/o example set	91.75	90.13	90.93
	Ours w/o annotation rule	**92.75**	94.81	93.76
	Ours	92.74	**95.07**	**93.89**

Table 3. Comparison with other NER models on Conll2003.

Experimental results on CoNLL 2003 show a slight improvement by DSMER without example sets. However, significant improvement has been achieved under the conditions of only using the example set. At the same time, we observe that using example set and annotation rule can not improve all factors. This is because the example set can better represent the scope of the entity type in the semantic space, but the description text of the annotation rule may cause a certain offset, which makes the calculation of the semantic similarity also be affected.

	Model	Precision	Recall	F1
Sequence Labeling	LSTM + CRF[6]	-	-	86.99
	IDCNN + CRF[24]	-	-	86.84
	TENER w/CNN-char [27]	-	-	88.43
	BERT-Tagger[3]	-	-	89.16
Reading Comprehension	MRC-BERT[11]	**92.98**	89.95	91.11
Semantic Matching	Ours w/o example set	90.56	88.79	89.67
	Ours w/o annotation rule	92.90	90.27	91.57
	Ours	92.95	**90.47**	**91.69**

Table 4. Comparison with other NER models on OntoNotes 5.0.

Similar results are also observed in the experiment on the OnteNotes 5.0 dataset. However, the use of annotation rule can still improve F1 score, so we think it is effective prior knowledge. Comparative experiments show that DSMER can handle NER problems. We continue to conduct more ablation experiments in subsection 4.4 to analyze the impact of different model designs on performance.

4.4 Ablation Studies

The impact of example set. As shown in Table 3 and 4, whether to use example set has a great influence on model performance. In order to observe the impact of the size of the example set on the model, we split the data set according to the split ratio of subsection 4.1, and test it on the CoNLL 2003 dataset. The results are shown in Table 5:

Percentage	Precision	Recall	F1
5%	91.67	94.23	92.93
10%	**92.75**	94.81	**93.76**
15%	92.60	**94.95**	93.76
20%	91.83	93.79	92.80
40%	91.43	91.88	91.65

Table 5. The impact of the percentage of example set, experiments on CoNLL 2003.

It can be seen that the 10% and 15% split ratios have the best effect. And as the proportion of the example set increases, the overall effect decreases since the lack of training data. Since all entities in the example set are phrases that can express their entity type, a large number of entity examples can better express the position of the entity type in the high-dimensional semantic space. In this way, the calculation of the distance between candidate span and entity type is more accurate. But with the increase of the example set, the decrease of training data makes the model easy overfitting on the training data. This is a trade-off process for dataset segmentation. Comparing with other models, we choose 10% as the segmentation ratio.

The impact of annotation rules. How to construct the annotation rule sentence also has a significant influence on the final results. In this subsection, we explore difference sources to construct annotation rules and their influence, including:

- **Annotation guideline:** the annotation rule from documents, like *"find organizations including companies, agencies and institutions"*.
- **Wikipedia:** the wikipedia definition of entity type, like *"an organization is an entity comprising multiple people, such as an institution or an association."*
- **Synonyms:** word or phrases that mean nearly the same as the entity type word from Dictionary, like *"association"*
- **All above:** encode above three concepts and use the average representation.

Table 6 shows the experimental results on CoNLL 2003. DSMER outperforms BERT-tagger by using different types of annotation rules. Among them, the effect of using annotation guideline is the best among the three categories,

Model	F1
BERT-Tagger	89.16
Annotation guideline	90.21(+1.05)
Wikipedia	89.65(+0.49)
Synonyms	89.90(+0.74)
All above	**90.93(+1.77)**

Table 6. Results of different types of annotation rules on CoNLL 2003.

because it is the closest text description to the entity annotation. At the same time, it can be seen that the combined usage of three different kind of rules can achieve better performance improvement.

5 Conclusion

In this paper, we introduce a novel framework for named entity recognition task which reflect the natural entity annotation process of human being. The proposed model obtain state of the art results on public datasets, which indicates the effectiveness of DSMER. The deep semantic matching based framework shows a possible new paradigm to tackle such problem. We would like to explore more variant of the framework in the future.

References

1. Chiu, J.P., Nichols, E.: Named Entity Recognition with Bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics **4**, 357–370 (2016), https://www.aclweb.org/anthology/Q16-1026
2. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural Language Processing (Almost) from Scratch. The Journal of Machine Learning Research **12**(null), 2493–2537 (Nov 2011)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://www.aclweb.org/anthology/N19-1423
4. He, K., Zhang, X., Ren, S., Sun, J.: Identity Mappings in Deep Residual Networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 630–645. Lecture Notes in Computer Science, Springer International Publishing, Cham (2016)
5. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2333–2338. CIKM '13, Association for Computing Machinery, New York, NY, USA (Oct 2013). https://doi.org/10.1145/2505515.2505665, https://doi.org/10.1145/2505515.2505665

6. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF Models for Sequence Tagging. arXiv:1508.01991 [cs] (Aug 2015), http://arxiv.org/abs/1508.01991, arXiv: 1508.01991
7. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From Word Embeddings To Document Distances. In: International Conference on Machine Learning. pp. 957–966 (Jun 2015), http://proceedings.mlr.press/v37/kusnerb15.html
8. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282–289. ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (Jun 2001)
9. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural Architectures for Named Entity Recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 260–270. Association for Computational Linguistics, San Diego, California (Jun 2016). https://doi.org/10.18653/v1/N16-1030, https://www.aclweb.org/anthology/N16-1030
10. Lee, K.J., Hwang, Y.S., Rim, H.C.: Two-Phase Biomedical NE Recognition based on SVMs. In: Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine. pp. 33–40. Association for Computational Linguistics, Sapporo, Japan (Jul 2003). https://doi.org/10.3115/1118958.1118963, https://www.aclweb.org/anthology/W03-1305
11. Li, X., Feng, J., Meng, Y., Han, Q., Wu, F., Li, J.: A Unified MRC Framework for Named Entity Recognition. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5849–5859. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.519, https://www.aclweb.org/anthology/2020.acl-main.519
12. Ma, X., Hovy, E.: End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1064–1074. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/P16-1101, https://www.aclweb.org/anthology/P16-1101
13. Monge, G.: Mémoire sur la théorie des déblais et des remblais. Histoire de l'Académie Royale des Sciences de Paris (1781)
14. Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., Ward, R.: Semantic Modelling with Long-Short-Term Memory for Information Retrieval. arXiv:1412.6629 [cs] (Feb 2015), http://arxiv.org/abs/1412.6629, arXiv: 1412.6629
15. Peleg, S., Werman, M., Rom, H.: A unified approach to the change of resolution: space and gray-level. IEEE Transactions on Pattern Analysis and Machine Intelligence **11**(7), 739–742 (Jul 1989). https://doi.org/10.1109/34.192468, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence
16. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep Contextualized Word Representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2227–2237. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-1202, https://www.aclweb.org/anthology/N18-1202
17. Pradhan, S., Moschitti, A., Xue, N., Ng, H.T., Björkelund, A., Uryupina, O., Zhang, Y., Zhong, Z.: Towards Robust Linguistic Analysis using OntoNotes. In:

Proceedings of the Seventeenth Conference on Computational Natural Language Learning. pp. 143–152. Association for Computational Linguistics, Sofia, Bulgaria (Aug 2013), https://www.aclweb.org/anthology/W13-3516
18. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving Language Understanding by Generative Pre-Training. OpenAI (2018)
19. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners p. 24
20. Rubner, Y., Tomasi, C., Guibas, L.J.: The Earth Mover's Distance as a Metric for Image Retrieval. International Journal of Computer Vision **40**(2), 99–121 (Nov 2000). https://doi.org/10.1023/A:1026543900054, https://doi.org/10.1023/A:1026543900054
21. Seyler, D., Dembelova, T., Del Corro, L., Hoffart, J., Weikum, G.: A Study of the Importance of External Knowledge in the Named Entity Recognition Task. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 241–246. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-2039, https://www.aclweb.org/anthology/P18-2039
22. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 101–110. CIKM '14, Association for Computing Machinery, New York, NY, USA (Nov 2014). https://doi.org/10.1145/2661829.2661935, https://doi.org/10.1145/2661829.2661935
23. Sohrab, M.G., Miwa, M.: Deep Exhaustive Model for Nested Named Entity Recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2843–2849. Association for Computational Linguistics, Brussels, Belgium (Oct 2018). https://doi.org/10.18653/v1/D18-1309, https://www.aclweb.org/anthology/D18-1309
24. Strubell, E., Verga, P., Belanger, D., McCallum, A.: Fast and Accurate Entity Recognition with Iterated Dilated Convolutions. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2670–2680. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). https://doi.org/10.18653/v1/D17-1283, https://www.aclweb.org/anthology/D17-1283
25. Sutton, C., McCallum, A., Rohanimanesh, K.: Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. Journal of Machine Learning Research **8**(Mar), 693–723 (2007), https://www.jmlr.org/papers/v8/sutton07a.html
26. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. pp. 142–147 (2003), https://www.aclweb.org/anthology/W03-0419
27. Yan, H., Deng, B., Li, X., Qiu, X.: TENER: Adapting Transformer Encoder for Named Entity Recognition. arXiv:1911.04474 [cs] (Dec 2019), http://arxiv.org/abs/1911.04474, arXiv: 1911.04474
28. Zheng, C., Cai, Y., Xu, J., Leung, H.f., Xu, G.: A Boundary-aware Neural Model for Nested Named Entity Recognition. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 357–366. Association for Computational Linguistics, Hong Kong, China (Nov 2019).

https://doi.org/10.18653/v1/D19-1034, https://www.aclweb.org/anthology/D19-1034