In-Memory Distributed Training
of
Linear-Chain Conditional Random Fields
with an Application to
Fine-Grained Named Entity Recognition

Robert Schwarzenberg, Leonhard Hennig, Holmer Hemsen

Motivation: Fine-Grained Named Entity Recognition

Types: **Location, City, Route, Street, Stop, Distance, Other (O)**

Output

Die/**O** U1/**Route** in/**O** Berlin/**City** ist/**O** sehr/**O** laut/**O** ab/**O** der/**O** Warschauer/**Stop** Str./**Stop**

Output

A1/**Street** ,/**O** Seevetal/**Location** Richtung/**O** Bremen/**Location** ,/**O** 5/**Distance** KM/**Distance** Stau/**O** ./**O**

Motivation: CRFs, Big Data and Scalability

- Fine-Grained NER **improves performance** on several tasks [10, 7, 4] but amplifies **data sparsity problem** that
- can be tackled w/ **distant supervision** [14, 1] which, however, introduces **scalability issues** with linear chain CRFs
- because training is **time-consuming**
  - 1 million tokens, 45 labels, around 500k parameters, **more than 3 days of training** (POS task) on 2.4 GHz Machine [15]

Approach: Distribution w/ MapReduce

- MapReduce [3] is an established programming model for distributed computing, supported by several frameworks.

MapReduce Example: Maximum token length

myMapOp: token $\rightarrow$ **len**(token)
myReduceOp: (x,y) $\rightarrow$ **max**(x,y)

distributedDataSet.**map**(**myMapOp**)
    .**reduce**(**myReduceOp**)

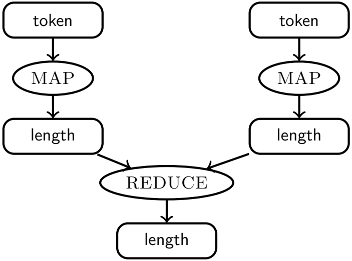

Figure 1: MapReduce.

Approach: Notation

- $O = o_1 \dots o_T$: sequence of observations (i.e. tokens),
- $L = l_1 \dots l_T$: sequence of labels (i.e. NE tags),
- $D = \{O^{(i)}, L^{(i)}\}_{i=1}^{N}$: training data.
- f_k denotes one of K binary feature functions weighted by $\theta_k \in \mathbb{R}$ in a linear chain CRF [8]

$$p(L|O) = \frac{1}{Z(O)} \prod_{t=1}^{T} \exp\left(\sum_{k}^{K} \theta_k f_k(l_{t-1}, l_t, o_t)\right) \qquad (1)$$

  where $Z(O)$ is a normalization term.
- Parameters θ_k are estimated s.t. conditional log-likelihood $\mathcal{L}$ of the training labels is maximized.

Approach: Data-Parallel Gradient Computation

Partially deriving the cond. log-likelihood $\mathcal{L}$ by θ_k yields [15, 9]

$$\frac{\partial \mathcal{L}}{\partial \theta_k} = \mathbb{E}(f_k) - \mathbb{E}_\theta(f_k) \tag{2}$$

with

$$\mathbb{E}(f_k) = \sum_{i=1}^{N} \mathbb{E}^{(i)}(f_k) \tag{3}$$

and

$$\mathbb{E}_\theta(f_k) = \sum_{i=1}^{N} \mathbb{E}_\theta^{(i)}(f_k). \tag{4}$$

Thus

$$\frac{\partial \mathcal{L}}{\partial \theta_k} = \sum_{i=1}^{N} (\mathbb{E}^{(i)}(f_k) - \mathbb{E}_\theta^{(i)}(f_k)). \tag{5}$$

Approach: Gradient Computation w/ MapReduce

- Partition and distribute disjoint data chunks of size p and
- perform gradient computation within MapReduce:

$$
\left.
\begin{array}{l}
\left.\sum_{i=1}^{p}(\mathbb{E}^{(i)}(f_k) - \mathbb{E}_\theta^{(i)}(f_k))\right\} \text{ map} \\
\left.\sum_{i=p+1}^{2p}(\mathbb{E}^{(i)}(f_k) - \mathbb{E}_\theta^{(i)}(f_k))\right\} \text{ map}
\end{array}
\right\} (+) \text{ reduce}
$$

$$\vdots$$

Approach: Framework

- Li et al. (2015) [9] implemented distributed training w/ Hadoop but
- for each iteration a new Hadoop job is submitted, which is costly due to
  - JVM startup times and
  - disk IO for re-reading the training data.

- Apache Flink [2] provides primitives for massively parallel iterations and
- identifies iteration-invariant parts and caches them to prevent unnecessary recomputations [5].

Approach: Implementation

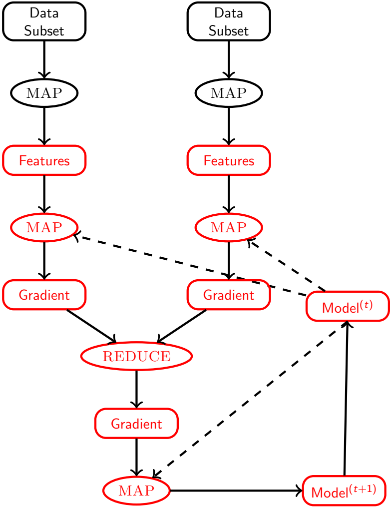

Figure 2: Distributed in-memory iteration step.

- Implemented using FACTORIE [12],
- constant step-size optimizer.

Experiments: Outline

- Accuracy: Fine-Grained NER (parameter validation)
- Scalability

Accuracy Experiments: Sources and Datasets

Dataset	Tokens	Noise
RSS	20152	35.6%
Twitter	12606	45.3%

Table 1: Sources, size and noise where noise refers to the tokens the Enchant Myspell dictionary did not recognize.

Experiment setup

- Seven fine-grained geospatial entities,
- over 100k parameters (task-specific and general features),
- distributed 10-fold experiments conducted w/ level of parallelism fixed at four,
- sanity checks involving local sequential counter part (w/o Flink directives) and
- 10-fold experiments also conducted with state-of-the-art reference model: Stanford NER [6] in standard configuration.

Accuracy Experiments: Results

- Sanity checks passed: Very similar parameters in place after distributed and local training.

System	Dataset	P	R	F1
Locator	RSS	80.7	75.8	75.2
Stanford	RSS	82.8	78.8	**80.5**
Locator	Twitter	57.0	50.4	**51.7**
Stanford	Twitter	79.0	35.9	47.2

Table 2: Results of 10-fold NER experiments (micro averages).

Scalability Experiments: Setup

- Distributed and local experiments (w/o Flink directives).
- Cluster consisting of four physical machines (+ master node)
  - three 1.80GHz CPUs w/ 8 cores, 16 threads, 20 MB cache,
  - two 2.40GHz CPUs w/ 8 cores, 16 threads, 20 MB cache.
- Local experiments ran on master node.
- Each YARN task manager was assigned 8 GB of memory
  - 30% reserved for Flink,
  - master node memory reduced to 8 GB.
- Data distribution and feature extraction considered part of the training.

Scalability Experiments: Results

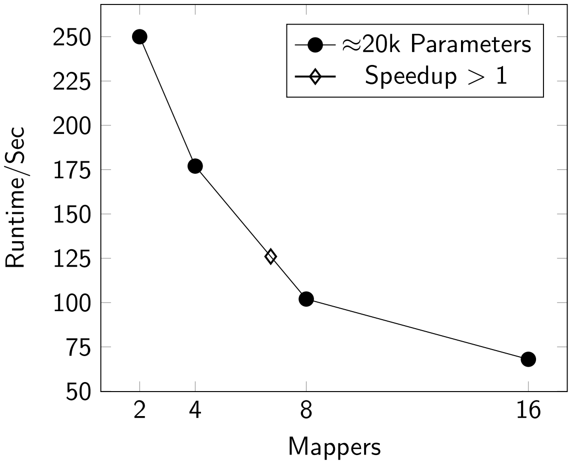

Figure 3: Execution times for increasing numbers of mappers (tokens: ≈ 100k, iterations: 25).

Scalability Experiments: Results

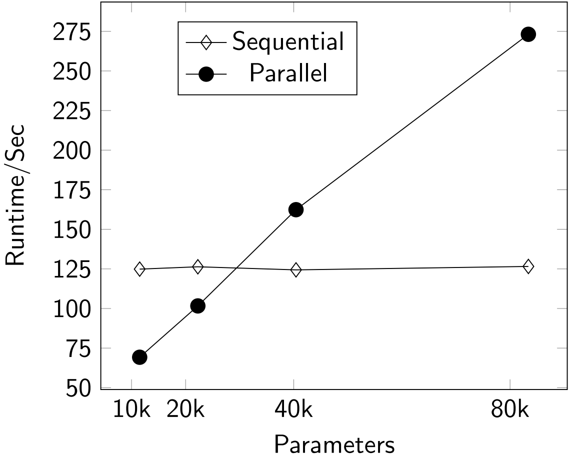

Figure 4: Execution times for increasing numbers of parameters (tokens: $\approx$ 100k, iterations: 25, parallelism: 8).

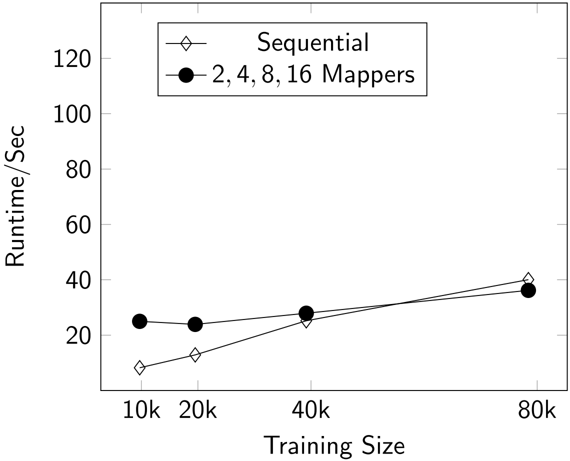

Figure 5: Scalability of the distributed model (parameters: $\approx$ 20k, iterations: ten).

Conclusion

Contributions

- Proof-of-concept distributed, iteration-aware training of a linear chain CRF.
- Experimental validation of the parameters learned during distributed training in a fine-grained NER task.
- Experimental demonstration of the scalability of our approach w/ an analysis of the communication overhead trade offs.

Future work

- Implementation of more sophisticated optimizers (Adagrad, LBFGS).
- Work w/ sparse tensors.
- Distribution of general factor graph training.

References I

[1] Abhishek, A., Anand, A., and Awekar, A.
Fine-Grained Entity Type Classification by Jointly Learning Representations and Label Embeddings.
In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (Valencia, Spain, Apr. 2017), Association for Computational Linguistics, pp. 797–807.

[2] Alexandrov, A., Bergmann, R., Ewen, S., Freytag, J.-C., Hueske, F., Heise, A., Kao, O., Leich, M., Leser, U., Markl, V., et al.
The stratosphere platform for big data analytics.
The VLDB Journal 23, 6 (2014), 939–964.

[3] Dean, J., and Ghemawat, S.
MapReduce: simplified data processing on large clusters.
Communications of the ACM 51, 1 (2008), 107–113.

References II

[4] Dong, L., Wei, F., Sun, H., Zhou, M., and Xu, K.

A Hybrid Neural Model for Type Classification of Entity Mentions.

In *Proceedings of the 24th International Conference on Artificial Intelligence* (Buenos Aires, Argentina, 2015), IJCAI'15, AAAI Press, pp. 1243–1249.

[5] Ewen, S., Schelter, S., Tzoumas, K., Warneke, D., and Markl, V.

Iterative parallel data processing with stratosphere: an inside look.

In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data* (2013), ACM, pp. 1053–1056.

[6] Finkel, J. R., Grenager, T., and Manning, C.

Incorporating non-local information into information extraction systems by gibbs sampling.

In *Proceedings of the 43rd annual meeting on association for computational linguistics* (2005), Association for Computational Linguistics, pp. 363–370.

References III

[7] Koch, M., Gilmer, J., Soderland, S., and Weld, D. S.
Type-Aware Distantly Supervised Relation Extraction with Linked Arguments.
In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, Oct. 2014), Association for Computational Linguistics, pp. 1891–1901.

[8] Lafferty, J., McCallum, A., and Pereira, F.
Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
In *Proc. of the ICML* (2001), vol. 1, pp. 282–289.

[9] Li, K., Ai, W., Tang, Z., Zhang, F., Jiang, L., Li, K., and Hwang, K.
Hadoop recognition of biomedical named entity using conditional random fields.
IEEE Transactions on Parallel and Distributed Systems 26, 11 (2015), 3040–3051.

References IV

[10] Ling, X., and Weld, D.

Fine-Grained Entity Recognition.

In *Proc. of AAAI '12* (2012).

[11] Liu, Z., Tang, B., Wang, X., and Chen, Q.

De-identification of clinical notes via recurrent neural network and conditional random field.

Journal of Biomedical Informatics (2017).

[12] McCallum, A., Schultz, K., and Singh, S.

Factorie: Probabilistic programming via imperatively defined factor graphs.

In *Advances in Neural Information Processing Systems* (2009), pp. 1249–1257.

References V

[13] Peng, J., Bo, L., and Xu, J.
Conditional neural fields.
In *Advances in neural information processing systems* (2009), pp. 1419–1427.

[14] Plank, B., Hovy, D., McDonald, R. T., and Søgaard, A.
Adapting taggers to twitter with not-so-distant supervision.
In *COLING* (2014), pp. 1783–1792.

[15] Sutton, C., and McCallum, A.
An introduction to conditional random fields.
Foundation and Trends in Machine Learning 4, 4 (2011), 267–373.

[16] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H.
Conditional random fields as recurrent neural networks.
In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1529–1537.

Thank you!

(Backup) Motivation: CRFs in the Neural Era

Conditional Neural Fields

Jian Peng and Liefeng Bo and Xu, Jinbo, Advances in Neural Information Processing Systems 22 (2009) [13]

Recurrent Conditional Random Fields

Yao, Kaisheng, et al., IEEE International Conference on Acoustics, Speech and Signal Processing (2014) [16]

Ensemble learning w/ Conditional Random Fields

Liu, Zengjian, et al., Journal of Biomedical Informatics (2017) [11]