Supplementary Materials for EMNLP 2018 Paper: Joint Multilingual Supervision for Cross-lingual Entity Linking

Shyam Upadhyay
University of Pennsylvania
Philadelphia, PA
shyamupa@seas.upenn.edu

Nitish Gupta
University of Pennsylvania
Philadelphia, PA
nitishg@seas.upenn.edu

Dan Roth
University of Pennsylvania
Philadelphia, PA
danroth@seas.upenn.edu

A Candidate Generation Details

We compute $\mathrm{P}_{\text{prior}}(e \mid m)$ using a surface-title index by counting how often the surface of m links to entity e in Wikipedia hyperlinks. Redirect title surfaces are also treated as hyperlink surfaces and added to the surface to link counts. To generate candidates from mentions in another language (e.g., Chinese), we first generate candidates as described above in the Wikipedia of that language (Chinese Wikipedia), and then use inter-language links to map candidate titles to English titles.

To improve recall, we also keep counts of a version of the hyperlink surface with its non-ascii characters replaced (e.g., “Algebre” vs “Algèbre”). For Chinese, we keep counts of a version converted to simplified Chinese. For foreign languages with a latin script (e.g., Spanish, Turkish), we also use a BACKOFF-TO-ENGLISH strategy – if querying the target language surface-to-link index for the mention surface does not generate any candidates, we query the English surface-to-link index.

B Implemention and Tuning Details.

All models were implemented using PyTorch.[1] We used ADAM (Kingma and Ba, 2014) optimizer with a learning rate of 1e-3 in all our experiments. For all experiments, we limit the candidate generator to output the top-20 candidates. Local context window was set to $W = 25$ tokens. The convolutional filter width was set to $k = 5$. The mention surface vocabulary V was limited to size 1M for both monolingual and joint training. The multilingual embeddings (d=300) were scaled to a fixed norm R (=5.0), and were not updated during training. Dropout (Srivastava et al., 2014) was separately applied to local context and document context feature, each being tuned over $\{0.4, 0.45, \cdots, 0.7\}$. The size of entity, type and context vectors was fixed to $h = 100$. Batch size was tuned over $\{128, 256, 512, 1024\}$.

The Wikipedia dumps were parsed using the WikiExtractor script.[2] Stanford segmenter was used for Arabic (Monroe et al., 2014) and Chinese segmentation (Tseng et al., 2005).

References

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*.

Will Monroe, Spence Green, and Christopher D. Manning. 2014. Word Segmentation of Informal Arabic with Domain Adaptation. In *Proc. of ACL*.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research*, volume 15.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A Conditional Random Field Word Segmenter for Sighan Bakeoff 2005. In *Proc. of SIGHAN*.

[1] github.com/pytorch

[2] github.com/attardi/wikiextractor