Intermediate Championship!

Naoaki Okazaki (okazaki at ecei.tohoku.ac.jp),
Kentaro Inui (inui at ecei.tohoku.ac.jp)

Please let us know…

- Please report F1 scores on both of the development and test sets
- Do not forget to update feature files for development and test sets when you apply a new trained model
- Training

```
$ ./feature.py < train > train.f
$ crfsuite learn -a ap -p max_iterations=20 -m ner.model train.f
```

- Performance on the development set

```
$ ./feature.py < dev > dev.f
$ crfsuite tag –r -m ner.model < dev.f | conlleval.py
```

- Performance on the test set

```
$ ./feature.py < test > test.f
$ crfsuite tag –r -m ner.model < test.f | conlleval.py
```

Please explain…

- Feature set (your strategy for improving NER)
  - What kinds of features did you implement?
  - How did you implement the features (show us your code!)
  - Originality of the features (or external references if any)
- Performance
  - The performance value of the current system
  - Which features were effective?
  - Why do you think the features were effective?
- Thoughts and difficulties in developing your NER

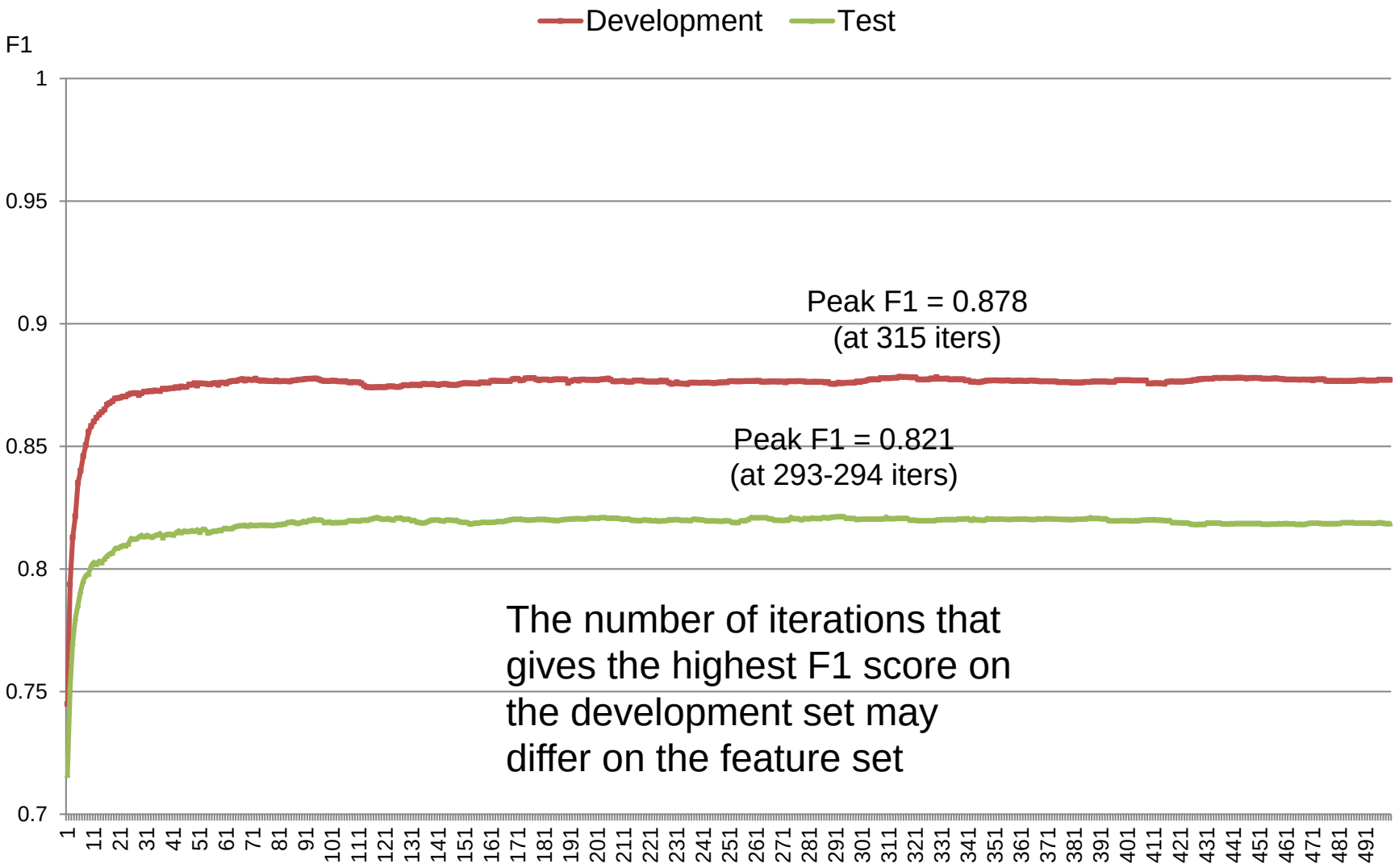
How good are our implementations?

Work	Dev F1	Test F1	Comment
Florian+ 2003	93.87	88.76	The best system in the CoNLL 2003 shared task. Gazetters. Classifier combination.
Kazama+ 2007	92.29	88.02	Gazetteers extracted from Wikipedia
Suzuki+ 2008	94.48	89.92	Semi-supervised learning with giga-scale unlabeled text
Ratinov+ 2009	93.50	90.57	Non-local information (prediction history, context aggregation). External knowledge (gazetteers, word clusters)
Passos+ 2014	94.46	90.90	Stacked CRF with lexicon infused embeddings
Lample+ 2016	N/A	90.94	Bi-LSTM-CRF with word embeddings computed by Bi-LSTM on characters
Chiu+ 2016	N/A	91.62	Bi-LSTM-CNN with word embeddings and lexicon features

Common voice from participants: “performance drops after adding a feature”

- Over-fitting to the training data
  - Features effective in the training set are not reliable on the test set
  - Even if we *memorize* the answers of a practice exam (training set), the final exam (test set) may not have the same problem!
- Parameters are not tuned for the new feature space
  - We need to tune training parameter(s) on the development set
  - More specifically, we find the parameter value(s) such that the performance on the development set is maximized
- The performance decrease is not significant
  - The performance decrease happens by chance
- We are not sure…
  - Machine learning is “black-box”

F1 scores with different number of iterations



How to improve the NLP system

- Parameter tuning (including ML algorithms)
  - E.g., fitting parameter (number of iterations)
  - E.g., changing structured perceptron to CRF
- Increasing the number of training instances

- False analysis
  - Collect *false instances* for which the NER makes wrong prediction
  - Generalize the false instances to classify a small set of the possible causes of failures
  - (Design heuristic rules, features, etc based on the false analysis)
- Observations and ideas for the target problem

Report

- Submission due: 19th January 2016
- How to submit:
  - Mail to Kentaro Inui inui@ecei.tohoku.ac.jp
- What to submit (via email):
  - The source code of your feature extractor
  - A report (no longer than two pages, in PDF or Word format)
- Contents of a report
  - Explanation of feature sets
  - Performance of feature sets
  - False analysis
    - Collect false instances on the test sets
    - Categorize (generalize) false instances into several types, and count the number of false instances in each category

False analysis

- Merging a CRFsuite output with the test set
  - crfsuite tag -m ner.model -r < test.f | merge.py test

- Extracting false instances only
  - crfsuite tag -m ner.model -r < test.f | merge.py test | false_instance.py

References

- J. P.C. Chiu, E. Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. TACL 2016.
- R. Florian, A. Ittycheriah, H. Jing and T. Zhang. 2003. Named Entity Recognition through Classifier Combination. CoNLL 2003.
- J. Kazama and K. Torisawa. 2007. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. EMNLP 2007.
- P. Koehn. Statistical Significance Tests for Machine Translation Evaluation. EMNLP 2004.
- G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer. 2016. Neural Architectures for Named Entity Recognition. NAACL 2016.
- A. Passos, V. Kumar, A. McCallum. 2014. Lexicon Infused Phrase Embeddings for Named Entity Resolution. CoNLL 2014.
- L. Ratinov and D. Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. CoNLL 2009.
- J. Suzuki and H. Isozaki. 2008. Semi-Supervised Sequential Labeling and Segmentation using Giga-word Scale Unlabeled Data. ACL 2008.