GermEval 2014 NER: Evaluation plan

May 20, 2014

This document describes the evaluation plan for the GermEval 2014 Shared Task.

1 Data

1.1 Phenomena

The data of the shared task uses the standard "big four" NER classes (LOC, ORG, PER, OTH). It differs from previous NER shared tasks in two regards:

- it contains *nested* markables (i.e., NEs within NEs)
- it subtypes the NE classes to provide a fine-grained description of *sublexical NEs* (i.e., NEs that span parts of tokens). These are helpful in describing compounding and derivation, frequent morphological processes in German. Sublexical NE labels are constructed by suffixing the NE class with *-part* (NE part of a compound) and *-deriv* (word derived from a NE).

The following sentence contains nested markables and a sublexical compound markable:

> Aufgrund seiner Initiative fand 2001/2002 in [$\text{Stuttgart}_{\text{LOC}}$], [$\text{Braunschweig}_{\text{LOC}}$] und [Bonn_{LOC}] eine große und publizistisch vielbeachtete [$\text{Troia-Ausstellung}_{\text{LOCpart}}$] statt, " [[$\text{Troia}_{\text{LOC}}$] - Traum und $\text{Wirklichkeit}_{\text{OTH}}$] ".

And here is an example for an NE derivation:

> [$\text{norddeutsche}_{\text{LOCderiv}}$] Stämme

1.2 Phenomena

Analysis of the corpus showed that having two levels of NEs (i.e., "one level of embedding") is sufficient to cover almost all cases. We therefore decided, for the sake of simplicity, to represent the data non-recursively, adopting a CoNLL-style one-token-per-line format that uses two columns (columns 3 and 4) for the NE annotation and represents them in the established BIO annotation. For the first example from above:

```
1 Aufgrund O O
2 seiner O O
3 Initiative O O
4 fand O O
5 2001/2002 O O
6 in O O
7 Stuttgart B-LOC O
8 , O O
9 Braunschweig B-LOC O
10 und O O
11 Bonn B-LOC O
12 eine O O
13 große O O
14 und O O
15 publizistisch O O
16 vielbeachtete O O
17 Troia-Ausstellung B-LOCpart O
18 statt O O
19 , O O
20 " O O
21 Troia B-OTH B-LOC
22 - I-OTH O
23 Traum I-OTH O
24 und I-OTH O
25 Wirklichkeit I-OTH O
26 " O O
27 . O O
```

2 Evaluation

2.1 Previous evaluation

Previous evaluation in CoNLL shared tasks on non-embedded annotation used the standard precision, recall and F1 score metrics, using each markable individually as a datapoint in the P/R calculation. Let M denote the set of markables on which P, R, and F_1 is computed. Then true positives are correctly predicted markables m where there is exact match on spans and on labels: $correct(m)$ iff $goldLabel(span(m)) == label(m)$.

We want to keep precision and recall but need to potentially redefine our datapoints to account for the nature of the data. Let M_1 denote the set of all "first-level"/"outer" NEs (column 3) and M_2 denote the set of all "second-level"/"inner" NEs (column 4). We will conduct three evaluations:

2.2 Evaluation Metrics

2.2.1 Metric 1: Strict, Combined Evaluation

The most simple evaluation treats first-level and second-level NEs individually and independently: $M = M_1 \cup M_2$ and uses all 12 labels (4 NE types, each in base, deriv and part varieties). This metric treats all markables on a par. **This metric is used to determine the overall ranking of the system.**

2.2.2 Metric 2: Loose, Combined Evaluation

As a variant of metric one, we again treat each NE individually but we collapse the label subtypes (base, deriv, part) so that a match on the base NE class is sufficient. For example, `PER` matches `PERderiv`. In other words, $M = M_1 \cup M_2$ and $correct(m)$ iff $goldNEclass(span(m)) == NEclass(m)$. This metric is useful to quantify the quality of systems on a more coarse-grained level. **This metric is provided for information only.**

2.2.3 Metric 3: Strict, Separate Evaluation

This evaluation computes two sets of P/R/F_1 values, once for $M = M_1$ and once for $M = M_2$. This metric considers the first-level and second-level markables separately which allows analysts to see how well systems do on first-level vs. second-level markables individually. It uses strict matching of labels. **This metric is provided for information only.**

Methodical note: In line with the split between first and second level performed by M3, we will consistently evaluate the two levels separately. That is, predictions only receive credit if they are placed in the correct column, with the first ("outer") level being the default.

2.2.4 Metric 4: Accuracy

The final metric that the script outputs is Accuracy. Note that in keeping with the previous CoNLL evaluations, accuracy is computed on a *per-token* basis – i.e., accuracy counts the number of correctly labeled tokens. This is contrast to Metrics 1–3 which are computed on a per-chunk basis (again, in keeping with previous CoNLL evaluations). **This metric is provided for information only.**

3 Evaluation Script

Our evaluation script computes all three metrics and outputs all three metrics. In addition, it provides per-class breakdowns for M1.

3.1 Input

It assumes that its input file has six tab-separated columns:

1. Index
2. Tokens
3. First-level NEs (gold)
4. Second-level NEs (gold)
5. First-level NEs (prediction)
6. Second-level NEs (prediction)

3.2 Parameters

-l Generate latex table output

-d Alternative delimiter

-v Verbose output

See also the documentation in the header of the script.

3.3 Participant responsibilities

1. Participants are strongly encouraged to run the evaluation script on their prediction files before submitting to ensure that the prediction files pass the evaluation script's well-formedness checks.
   When no gold labels are available (as in the case of the test data), we recommend duplicating the two prediction columns to produce the six-column file format for the evaluation script. Alternatively, you can fill columns 3 and 4 consistently with the label `O`. Clearly, the script cannot produce interesting results in this situation – the performance should be perfect in the first case and zero in the second case.

2. Participants are strongly encouraged to test that their prediction files show a line-by-line match with the test data file as provided by the Shared Task (**NER-de-test.tsv**), that is, contain neither extraneous lines nor miss lines. On UNIXish systems, a check can be performed by calling

   ```
   cut -f1,2 [YourPredictionsFile] | cmp /dev/stdin NER-de-test.tsv
   ```

   Also, the evaluation script makes an effort to detect invalid files (e.g., uninterpretable chunk labels). However, we cannot guarantee that all possible errors are detected automatically.

3. As specified in 2.2, gold standard and predictions are compared on a per-level basis. Please ensure that your markables are predicted at the correct level (larger/outer NEs on the first level, smaller/inner NEs on the second level).