The CLIN2017 Shared Task

Robert Östling
University of Helsinki

Eva Pettersson
University of Uppsala

Jörg Tiedemann
University of Helsinki

1 CNN

We used a plain 40-layer 1D convolutional neural network with residual connections, using character-level Levenshtein alignments. Decoding is done independently for each character at the bottom layer, so there is no language model involved except what is learned implicitly by the convolutional network. We use a dropout factor of 0.5 and the Adam algorithm for parameter optimization. The model was trained with the concatenated 1637 and 1657 Bible translations as source text, and the 1888 translation as target.

2 Lev

As training data, I use the provided historical-to-modern spelling dictionary, combined with automatic word alignments from the 1637 and 1657 versions of the bible to the 1888 version. I have further divided the training corpus into 90% training and 10% tuning.

In my approach to spelling normalisation, the historical word form is first matched against the training corpus of historical word forms mapped to their modern spelling. If the word form is found in this corpus, the most frequent corresponding modern spelling found in the mappings is chosen. Otherwise, Levenshtein edit distance calculations are performed, comparing the historical word form to word forms present in a modern corpus, i.e. in this case the 1888 and 2010 versions of the Statenvertaling bible. If the historical word form is found in a modern version of the bible as well, the original form is kept unchanged. Otherwise, the modern word form with the lowest edit distance to the original word form is chosen for normalisation, provided that the distance is below a certain threshold value (which is set based on the tuning part of the corpus). To adapt the Levenshtein comparisons to this specific task, I also include weights lower than one for certain commonly occurring edits observed in the training corpus.

For more details on the Levenshtein-based approach to spelling normalisation, refer to the paper by Pettersson et al [1].

3 SMT and SMT.word

The SMT system uses word-by-word translation using a character-level model trained on phrase-pairs extracted from the Bible corpora. We used both the translations from 1888 and 2010 as well as the small Blankaart text and trained a standard phrase-based SMT model with default settings and standard tools. For word alignment we used fast_align and symmetrisation using the grow-diag heuristics. From the resulting phrase-table, we then extracted reliable alphabetic phrase-pairs based on a Dice score filter. Those phrase pairs are then aligned on the character-level using Levenshtein and edit distance which we then feed in into the training procedures of the character-level phrase-based SMT model. This is finally tuned using MERT and applied to translate individual words in the test set.

The "SMT" variant is tuned on the Blankaart text, while the "SMT.word" variant is tuned on Blankaart and the 2010 Bible translation.

4 Ensembles

We compute the median string of each sentence by greedily generating one character at a time, minimizing the sum of the Levenshtein distances to each of the individual systems in the ensemble. All systems in the ensemble have equal weights, and the ensembling does not have any parameters to tune. As we are not sure which of the two SMT-based systems are best, we submit three different ensembles.

5 Runs

1. Lev
2. SMT
3. SMT.word
4. CNN
5. CNN-LEV-SMT
6. CNN-LEV-SMT-SMT.word
7. CNN-LEV-SMT.word

References

[1] Eva Pettersson, Beáta Megyesi, and Joakim Nivre. Normalisation of historical text using context-sensitive weighted levenshtein distance and compound splitting. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*. Linköping Electronic Conference Proceedings 85, 2013.