Sentence-Level Grammatical Error Identification as Sequence-to-Sequence Correction

Allen Schmaltz Yoon Kim Alexander M. Rush Stuart M. Shieber
Harvard University, Cambridge, MA

HARVARD
John A. Paulson School of Engineering and Applied Sciences

The Binary Classification Task

The Automated Evaluation of Scientific Writing (AESW) **Shared Task** 2016: Given a sentence, determine whether it needs to be edited (i.e., contains a grammatical error, broadly construed).

Data

- The AESW dataset is the first large-scale, publicly available professionally edited dataset of academic, scientific writing
- A collection of nearly 10,000 scientific journal articles (Engineering, Computer Science, Mathematics, Chemistry, Physics, etc.)
- Training set consists of about 500,000 sentences with errors and an additional 700,000 that are error-free
- Errors are described at the token level with insert and delete tags (see diagram at right)

Approach

- To establish a strong baseline on this new dataset, we utilize a CNN for binary classification, experimenting with word2vec:
  - Keeping the word vectors static (CNN-STATIC)
  - Fine-tuning the vectors (CNN-NONSTATIC)
- We propose two encoder-decoder architectures for this task, recasting the problem as translation (incorrect $\rightarrow$ correct) in order to train at the lowest granularity of annotations:
  - A word-based model (WORD)
  - A character-based model (CHAR)
- Evaluation is via F_1 at the sentence level
- On the final run on test, we use an ensemble of multiple encoder-decoder models and a CNN classifier (COMBINATION)

Acknowledgements

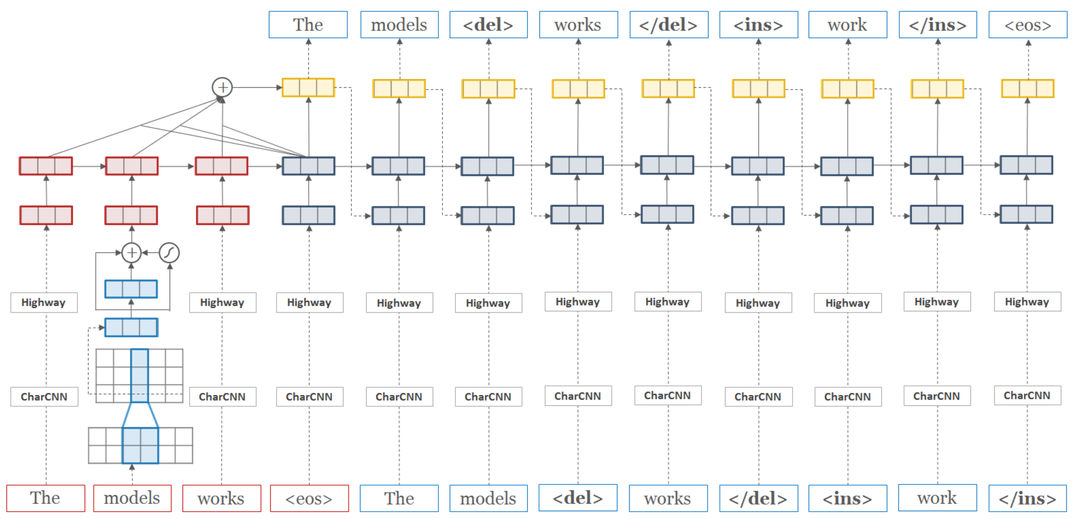
We would like to thank the Institute for Quantitative Social Science (IQSS) and the Harvard Initiative for Learning and Teaching (HILT) for support and Jeffrey Ling for software development.

Encoder-Decoder vs. CNN

Model	Data	Precision	Recall	F_1
RANDOM	N/A	0.3885	0.4992	0.4369
CNN-STATIC	Training+word2vec	0.5349	0.7586	0.6274
CNN-NONSTATIC	Training+word2vec	0.5365	0.7758	0.6343
WORD+ALL	Training	0.5399	0.7882	0.6408
WORD+SAMPLE	Training	0.5394	0.8024	0.6451
CHAR+ALL	Training	0.5400	0.8048	0.6463
CHAR+SAMPLE	Training	0.5526	0.8126	0.6579

Table 1: Experimental results on the development set excluding the held-out 10k tuning subset.

- The encoder-decoder models (and CHAR in particular) improve over the CNN models, at the expense of training/testing time.
- The +SAMPLE models are given a random sample of 200,000 sentences without edits and perform better than those given all error-free sentences (+ALL). See also Figure 1.



Character-aware Encoder-Decoder Architecture (Char)

Illustration (above) of the CHAR model architecture translating an example source sentence into the corrected target

- A CNN is applied over character embeddings to obtain a fixed dimensional representation of a word, which is given to a highway network (in light blue, above).
- Output from the highway network is used as input to a LSTM encoder-decoder.
- At each step of the decoder, its hidden state is interacted with the hidden states of the encoder to produce attention weights (for each word in the encoder), which are used to obtain the context vector via a convex combination.
- The context vector is combined with the decoder hidden state through a one layer MLP (yellow), after which an affine transformation followed by a softmax is applied to obtain a distribution over the next word/tag.
- The MLP layer (yellow) is used as additional input (via concatenation) for the next time step.
- An equality check between the source and the highest scoring output sentence (via beam search) determines the binary label.

Contributions

- Highest performing system (ensemble, as well as CHAR separately) on the binary classification Shared Task
- Demonstrated utility of a neural attention-based model for sentence-level grammatical error identification
- Our end-to-end approach does not have separate components for candidate generation or re-ranking that make use of hand-tuned rules or explicit syntax, nor do we employ separate classifiers for human-differentiated subsets of errors
- Evidence to suggest modeling at the sub-word level is beneficial

Tuning

- Post-hoc tuning was necessary to avoid under-prediction
  - **CNN models**: tuned the decision boundary to maximize the F_1-score on the held-out tuning set
  - **Encoder-decoder models**: tuned the bias weights (given as input to the final softmax layer generating the words/tags distribution) associated with the four annotation tags via a coarse grid search by iteratively running beam search on the tuning set

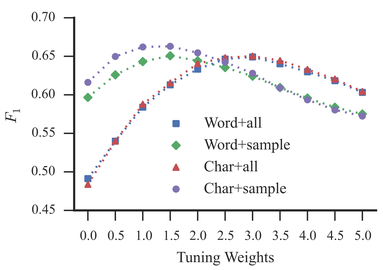


Figure 1: F_1 scores for varying values applied additively to the bias weights of the four annotation tags on the held-out 10k tuning subset.

Final Results

Model	Precision	Recall	F_1
RANDOM	0.3607	0.6004	0.4507
KNOWLET	0.6241	0.3685	0.4634
NTNU-YZU	0.6717	0.3805	0.4858
HITS	0.3765	0.948	0.5389
UW-SU	0.4145	0.8201	0.5507
NTNU-YZU	0.5025	0.7785	0.6108
CHAR+SAMPLE	0.5112	0.7841	0.6189
COMBINATION	0.5444	0.7413	0.6278

Table 2: Our final system results on test (143,802 sentences evaluated on the Shared Task CodaLab server) are highlighted.

Conclusion

- Demonstrated comparatively strong results on the Shared Task, but many areas remain to be explored
- Future work to examine, among others:
  - An end-to-end approach for languages such as Japanese
  - Approaches for incorporating additional error-free data
  - Performance on the correction task
  - User studies to assess the utility of correction vs. identification

{schmaltz@fas,yoonkim@seas,srush@seas,shieber@seas}.harvard.edu