Why?

- Optical character recognition
- Speech recognition
- Noisy user texts
- Spelling correction

Spelling correction

Types of spelling errors:

1. non-word spelling errors (OOV)
2. real-word errors
    - typographical errors
    - orthographic errors
    - cognitive errors
    - grammatical errors
    - intentional incorrect writing

Today

1. Traditional methods
   - Edit distance
   - Deep Levenshtein
2. Noisy channel model
   - Improved noisy channel
3. Deep learning approaches
   - Spelling correction
   - Punctuation and capitalization restoration
   - Sentence boundary detection

Edit distance

following SLP book

1. The minimum edit distance between two strings
2. Is the minimum number of editing operations needed to transform one into the other:
   - Insertion
   - Deletion
   - Substitution

I	N	T	E	*	N	T	I	O	N
*	E	X	E	C	U	T	I	O	N
d	s	s		i	s				

Table: Two strings and their alignment

- 5 operations
- cost(s) = cost(d) = cost(i) = 1, dist = 5
- cost(s) = 2, cost(d) = cost(i) = 1, dist = 8

Min Edit Distance

following SLP book

- Two strings: $a = |n|, b = |m|$
- $D(i, j) =$ edit distance$(a_{1:i}, b_{1:j})$, $D(n, m) =$ edit distance(a, b)

Dynamic algorithm

$$d_{i0} = \sum_{k=1}^{i} w_{\text{del}}(b_k), \qquad \text{for } 1 \leq i \leq m$$

$$d_{0j} = \sum_{k=1}^{j} w_{\text{ins}}(a_k), \qquad \text{for } 1 \leq j \leq n$$

$$d_{ij} = \begin{cases} d_{i-1,j-1} & \text{for } a_i = b_j \\ \min \begin{cases} d_{i-1,j} + w_{\text{del}}(a_i) \\ d_{i,j-1} + w_{\text{ins}}(b_j) \\ d_{i-1,j-1} + w_{\text{sub}}(a_i, b_j) \end{cases} & \text{for } a_i \neq b_j \end{cases} \qquad \text{for } 1 \leq i \leq m, 1 \leq j \leq n.$$

Time and space complexity: $O(mn)$

Edit Distance

		А	Р	Е	С	Т	А	Н	Т
	0	1	2	3	4	5	6	7	8
Д	1	1	2	3	4	5	6	7	8
А	2	1	2	3	4	5	5	6	7
Г	3	2	2	3	4	5	6	6	7
Е	4	3	3	2	3	4	5	6	7
С	5	4	4	3	2	3	4	5	6
Т	6	5	5	4	3	2	3	4	5
А	7	6	6	5	4	3	2	3	4
Н	8	7	7	6	5	4	3	2	3

Figure: Edit distance computation

Image source: habr

Weighted Edit Distance

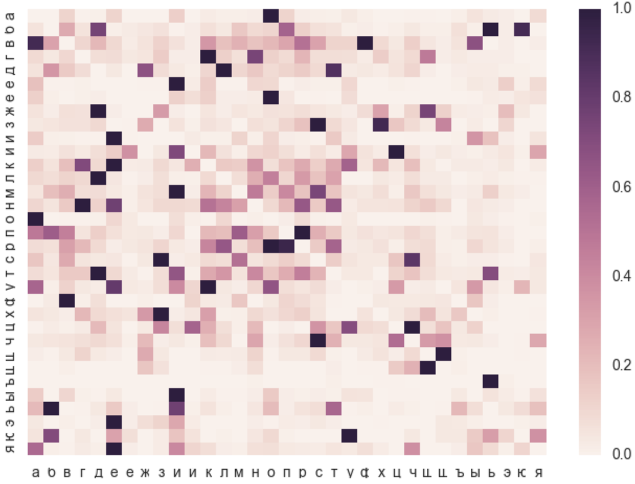


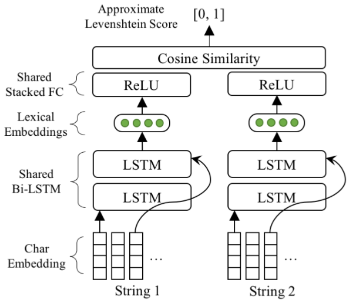
Figure: Confusion matrix for spelling errors

Image source: kartaslov

Deep Levenshtein [1]

Deep neural network is used to compute approximate Levenshtein distance which we call Deep Levenshtein composed of a shared bi-directional character LSTM, shared character embedding matrix, fully connected layers, and a dot product merge operation layer.
The objective:

$$||\frac{1}{2}(\cos(x_c, x_c^{'}) + 1) - \mathrm{sim}(x_c, x_c^{'})||$$



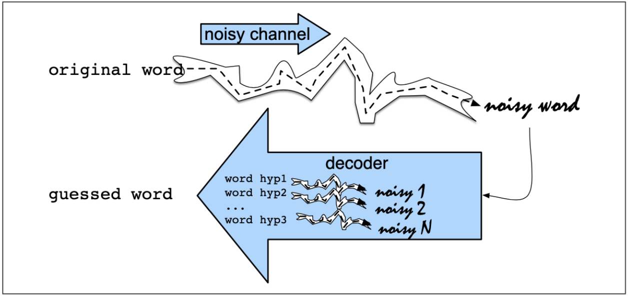
Today

1. Traditional methods
   - Edit distance
   - Deep Levenshtein
2. Noisy channel model
   - Improved noisy channel
3. Deep learning approaches
   - Spelling correction
   - Punctuation and capitalization restoration
   - Sentence boundary detection

Noisy channel model

for spelling correction

Task: find and correct a misspelled word (OOV)



$$\hat{w} = \arg\max_{w \in V} P(w|x) = \arg\max_{w \in V} \frac{P(x|w)P(w)}{P(x)} \propto \arg\max_{w \in C} P(x|w)P(w)$$

Noisy channel model

for spelling correction

1. **Candidates:** at Damerau-Levenshtein distance 1 (insertions, deletions, substitutions, transpositions)
2. $P(w)$: probability of the word w in context, which can be computed using any language model
3. **Channel model**: weighted Damerau-Levenshtein distance based on confusion matrix

Noisy channel model

for real-world spelling correction

1. **Candidates:** take the input sentence $X = x_1, x_2, \ldots, x_k, \ldots, x_n$, generate a large set of candidate correction sentences $C(X)$, then picks the sentence with the highest language model probability
2. Noisy channel model:

$$\arg\max_{w \in C} P(X|W)P(W)$$

3. Estimate $P(W)$ using language model
4. Channel probability:

$$p(x|w) = \begin{cases} \alpha, \text{if } x = w \\ \frac{1-\alpha}{|C(x)|}, \text{if } x \in C(x) \\ 0 \end{cases}$$

Improved noisy channel model

1. Incorporate word embeddings for semantics
2. Weighted model: $\arg\max_{w \in C} P(x|w)P(w)^{\lambda}$
3. Brill and Moore (2000) propose partition model:
   $P(x|w) \approx \max_{R,T} \sum P(T_i|R_i, position)$
   Example: P(fisikle—physical)
4. Incorporate pronunciation using letter-to-sound or grapheme-to-phoneme models

Today

1. Traditional methods
   - Edit distance
   - Deep Levenshtein
2. Noisy channel model
   - Improved noisy channel
3. Deep learning approaches
   - Spelling correction
   - Punctuation and capitalization restoration
   - Sentence boundary detection

Neural Language Correction with Character-Based Attention [2]

- Trained on a parallel corpus of "good" (x) and "bad" (y) sentences
- Encoder has a pyramid structure:
  $f_t^{(j)} = \mathrm{GRU}(f_{t-1}^{(j-1)}, c_t^{(j-1)})$
  $b_t^{(j)} = \mathrm{GRU}(b_{t+1}^{(j-1)}, c_t^{(j-1)})$
  $h_t^{(j)} = f_t^{(j)} + b_t^{(j)}$

$$c_t^{(j)} = \tanh(W_{pyr}^{(j)}[h_{2t}^{(j-1)}, h_{2t+1}^{(j-1)}]^\top + b_{pyr}^{(j)})$$

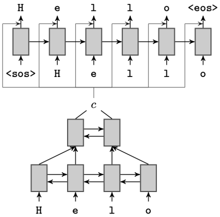


Figure: An encoder-decoder neural network model with two encoder hidden layers and one decoder hidden layer

Neural Language Correction with Character-Based Attention [2]

- Decoder network:
  $d_t^{(j)} = \mathrm{GRU}(d_{t-1}^{(j-1)}, {}_t^{(j-1)})$
- Attention mechanism:
  $u_{tk} = \phi_1(d^{(M)})^\top \phi_2(c_k), \phi : tahn(W \times \cdot)$
  $\alpha_{tk} = \frac{u_{tk}}{\sum_j u_{tj}}$
  $a_t = \sum_j \alpha_{tj} c_j$
- Loss:
  $L(x, y) = \sum_{t=1}^{T} logP(y_t | x, y_{<t})$

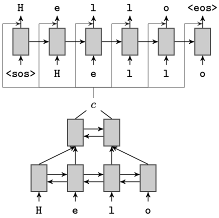


Figure: An encoder-decoder neural network model with two encoder hidden layers and one decoder hidden layer

Neural Language Correction with Character-Based Attention [2]

- Beam search for decoding: $s_k(y_{1:k}|x) = \log P_{NN}(y_{1:k}|x) + \lambda log P_{LM}(y_{1:k})$
- Synthezing errors: article or determiner errors (ArtOrDet) and noun number errors (Nn)

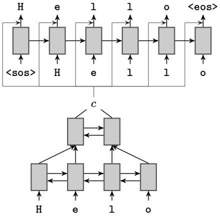


Figure: An encoder-decoder neural network model with two encoder hidden layers and one decoder hidden layer

Reference I

S. Moon, L. Neves, and V. Carvalho, "Multimodal named entity disambiguation for noisy social media posts," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2000–2008.

Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky, and A. Y. Ng, *Neural language correction with character-based attention*, 2016. arXiv: 1603.09727 [cs.CL].