Neural Language Models for Spelling Correction

Master's thesis

Matthias Hertel

Chair of Algorithms and Data Structures
University of Freiburg

December 19, 2019

Contents

1. Introduction
2. Methods
3. Evaluation

Motivation

- Electronic messaging

Motivation

- Electronic messaging
- Spellchecking documents

Motivation

- Electronic messaging
- Spellchecking documents
- Natural language processing systems

Motivation

- Electronic messaging
- Spellchecking documents
- Natural language processing systems

Google image search[1]:

Your search - **cute liitle catpi ctures** - did not match any documents.

[1] images.google.com

Motivation

- Electronic messaging
- Spellchecking documents
- Natural language processing systems

Google image search[1]:

Your search - **cute liitle catpi ctures** - did not match any documents.

DeepL machine translation[2]:

Cute liitle catpi ctures	Niedliche kleine Katpi-Katzenfiguren mit süßem Titel

[1] images.google.com

[2] deepl.com

Spelling Correction

- Task definition:

  Given a misspelled text S_{input}

  "S he isa Austran competer sceintist."

  predict the intended text S_{true}.

  "She is an Austrian computer scientist."

Contents

1 Introduction

2 Methods

3 Evaluation

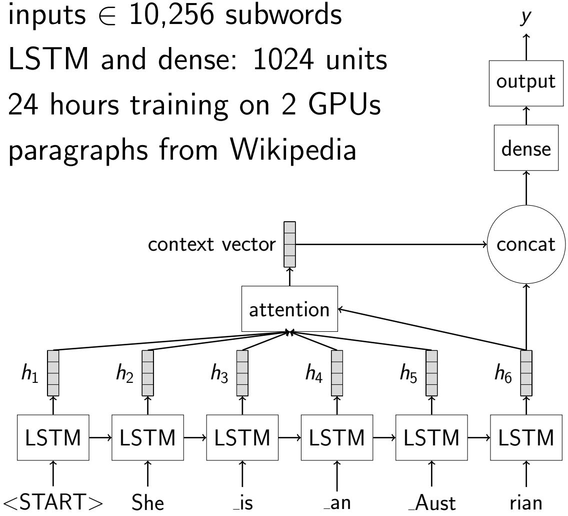
Language Model 1/3

- Language models
  estimate the probability $p(w_i|w_1, ..., w_{i-1})$
  that a word w_i follows the words w_1 to w_{i-1}.
  - Example: *She is an ...*

expert	5.4 %
active	4.7 %
author	3.1 %
...	...
Austrian	0.1 %
...	...
Austran	$2.5 \cdot 10^{-7}$ %

Language Model 2/3

- Recurrent neural network with attention
  - inputs $\in$ 10,256 subwords
  - LSTM and dense: 1024 units
  - 24 hours training on 2 GPUs
  - paragraphs from Wikipedia



Language Model 3/3

- From subwords to words
  - ▁Austrian = [▁Aust, rian]
  - $p(\text{Austrian}|\text{She is an}) =$
    $p(\text{\_Aust}|\text{She, \_is, \_an}) \cdot p(\text{rian}|\text{She, \_is, \_an, \_Aust})$

… for Spelling Correction 1/3

Input: *S he isa Austran competeter sceintist.*

- Candidate corrections
  - Vocabulary V containing 100,000 correctly spelled words
  - Edit operations
    - character insertion: *Astran* $\rightarrow$ *Austrian*
    - character deletion: *isa* $\rightarrow$ *is*
    - character replacement: *competer* $\rightarrow$ *computer*
    - character transposition: *sceintist* $\rightarrow$ *scientist*
    - split: *isa* $\rightarrow$ *is a*
    - merge: *S he* $\rightarrow$ *She*
  - Combination of up to two operations: *isa* $\rightarrow$ *is an*

... for Spelling Correction 2/3

Input: *S he isa Austran competer sceintist.*

1. Procedure maintains k partial solutions:

1.	*She is a*	likely solution
2.	*She is an*	less likely

2. Generate candidate corrections:
   {*Austran, Austrian*}

3. Append candidate corrections and rescore:

1.	*She is an Austrian*	likely solution
2.	*She is a Austrian*	less likely
3.	*She is an Austran*	very unlikely
4.	*She is a Austran*	very unlikely

4. Keep the k best solutions.

... for Spelling Correction 3/3

- Sequence rescoring
  - Candidate score depending on the previous words
  - reflects likelihood of candidate c being correct
    1. How well does c fit into the context?
    $\rightarrow$ probability $p(c|w_1, ...w_{i-1})$
    2. How similar is c to the input?
    $\rightarrow$ number of edit operations ed

$$\mathrm{score}(c) = \underbrace{-\log(p(c|w_1, ...w_{i-1}))}_{\text{log likelihood}} + \underbrace{\lambda \cdot \mathrm{ed}}_{\text{similarity}}$$

  - Candidate score is added to solution score

Approaches

1. NLMspell
   - neural language model spelling corrector
   - $k = 10$ partial solutions
2. TranslationSpell
   - machine translation model
   - input: misspelled English
   - output: correct English
   - encoder-decoder recurrent neural network

Baselines

1. UnigramSpell: context-free baseline
   - if word not in V, replace by most frequent candidate
   - preference for candidates with less edits
2. NgramSpell: context-dependent baseline
   - same as NLMspell
   - trigram language model
3. Google: commercial baseline
   - copy text into Google document
   - apply all suggested edits

Contents

1 Introduction

2 Methods

3 Evaluation

Evaluation: Language Models

$$\text{perplexity}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{p_(w_i | w_1, \ldots, w_{i-1})}}$$

model	perplexity
LSTM	157.0
LSTM+attention	103.3
Transformer	106.5
GPT 117M [Radford et al., 2019]	78.7

Evaluation metric 1/2

- Comparison of three sequences
  - S_{true} = The cute cat eats delicious fish.
  - S_{input} = Te cute cteats delicious fi sh.
  - $S_{\text{predicted}}$ = The cute act eats delicate fi sh.

Evaluation metric 1/2

- Comparison of three sequences
  - $S_{\text{true}} =$ The cute cat eats delicious fish.
  - $S_{\text{input}} =$ Te cute cteats delicious fi sh.
  - $S_{\text{predicted}} =$ The cute act eats delicate fi sh.
- Cases
  - True positives TP: a misspelled word is restored.

Evaluation metric 1/2

- Comparison of three sequences
  - $S_{\mathrm{true}} =$ The cute cat eats delicious fish.
  - $S_{\mathrm{input}} =$ Te cute cteats delicious fi sh.
  - $S_{\mathrm{predicted}} =$ The cute act eats delicate fi sh.
- Cases
  - True positives TP: a misspelled word is restored.
  - False negatives FN: a misspelled word is not restored.

Evaluation metric 1/2

- Comparison of three sequences
  - $S_{\mathrm{true}} =$ The cute cat eats delicious fish.
  - $S_{\mathrm{input}} =$ Te cute cteats delicious fi sh.
  - $S_{\mathrm{predicted}} =$ The cute act eats delicate fi sh.
- Cases
  - True positives TP: a misspelled word is restored.
  - False negatives FN: a misspelled word is not restored.
  - False positives FP: an input word is changed incorrectly.

Evaluation metric 2/2

- Metric

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Benchmarks

1,000 paragraphs from Wikipedia
every word misspelled with 20 % probability

- artificial benchmark
  up to two randomly sampled operations out of {insertion, deletion, replacement, transposition, merge, split}
- realistic benchmark
  typo collection by Peter Norvig
  39,709 misspellings for 7,841 words

Results 1/2

- Artificial benchmark

corrector	precision	recall	F-score	sequence acc.
UnigramSpell	67.4 %	60.8 %	63.9 %	17.3 %
NgramSpell	89.3 %	87.0 %	88.1 %	43.1 %
commercial	75.3 %	58.6 %	65.9 %	22.8 %
NLMspell	92.5 %	90.6 %	**91.5 %**	**49.5 %**
TranslationSpell	75.1 %	77.0 %	76.0 %	28.2 %

- Realistic benchmark

corrector	precision	recall	F-score	sequence acc.
UnigramSpell	50.5 %	44.7 %	47.4 %	22.0 %
NgramSpell	82.7 %	79.8 %	81.2 %	45.7 %
commercial	85.9 %	56.0 %	67.8 %	35.0 %
NLMspell	88.2 %	88.7 %	**88.4 %**	**57.4 %**
TranslationSpell	61.2 %	58.9 %	60.0 %	30.8 %

Results 2/2

- NLMspell on different types of artificial misspellings

error type	precision	recall	F-score
nonword	93.7 %	91.2 %	92.4 %
real-word	87.0 %	87.4 %	87.2 %
single-edit	93.0 %	91.6 %	92.3 %
multi-edit	81.0 %	81.3 %	81.2 %
split	95.4 %	95.2 %	95.3 %
merge	99.7 %	92.3 %	95.8 %
mixed	90.5 %	88.4 %	89.4 %

Conclusion

- The attention mechanism improves language models.
- Context helps to correct spelling:

  unigrams $<$ ngrams $<$ neural model
- Difficult cases: multi-edits and real-word errors.
- Language models worked better than the translation approach.

The end

Questions?

NgramSpell

- N-gram language model
  - Trigram Markov assumption:

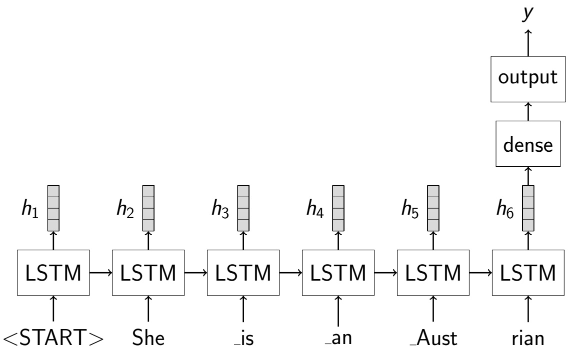
$$p(w_i|w_1, ...w_{i-1}) = p(w_i|w_{i-2}, w_{i-1})$$

  - Interpolation of trigram, bigram and unigram probabilities:

$$\begin{aligned} p(\mathsf{Austrian}|\mathsf{is}, \mathsf{an}) = &\alpha \cdot \frac{\mathrm{count}(\mathsf{is}, \mathsf{an}, \mathsf{Austrian})}{\mathrm{count}(\mathsf{is}, \mathsf{an})} \\ &+ (1-\alpha) \cdot \alpha \cdot \frac{\mathrm{count}(\mathsf{an}, \mathsf{Austrian})}{\mathrm{count}(\mathsf{an})} \\ &+ (1-\alpha)^2 \cdot \frac{\mathrm{count}(\mathsf{Austrian})}{N} \end{aligned}$$

Neural language model

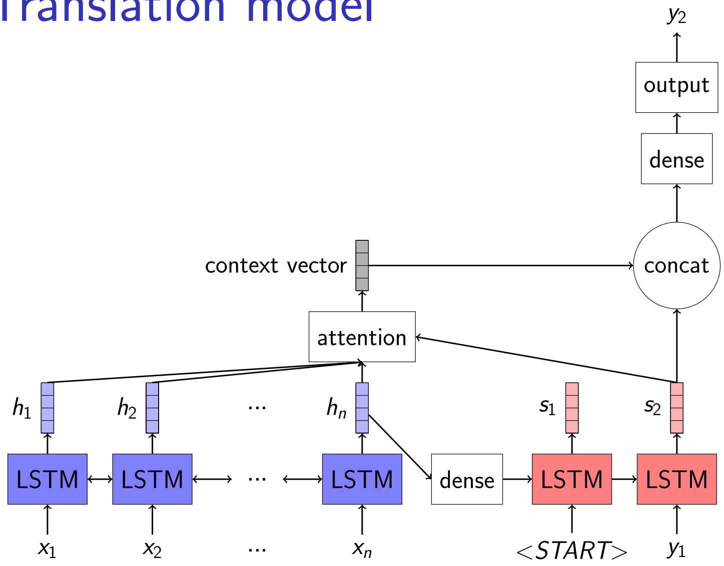
- Model without attention



Transformer

- Transformer language model
  - no recurrent network
  - multiple attention mechanisms
  - deep model

Translation model



Benchmark statistics

- error types in the two benchmarks

error type	artificial	realistic
single-edit	5348	3520
multi-edit	1015	564
split	651	7
merge	1266	4
mixed	493	7
nonword	7294	2448
real-word	1479	1654
total	8773	4102

Runtimes

- Total runtimes in seconds

corrector	artificial	realistic
UnigramSpell	5.5	2.5
NgramSpell	4,790.0	4,967.2
NLMspell	17,150.5	18,458.7
TranslationSpell	3,134.1	2,308.8

Perplexity

$$PP(W) = p(w_1, ..., w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{p(w_1, ..., w_N)}}$$

$$= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{p(w_i|w_1, ..., w_{i-1})}}$$

$$= \exp(-\frac{1}{N}\sum_{i=1}^{N} log(p(w_i|w_1, ..., w_{i-1})))$$

Edit Distance

	ϵ	a		l	i	f	e
ϵ	0	1	2	3	4	5	6
a	1	$\searrow$ equal 0	$\xrightarrow{\text{insert}}$ 1	2	3	4	5
l	2	2	3	$\searrow$ equal 1	2	3	4
i	3	3	4	3	$\searrow$ equal 1	2	3
k	4	4	5	4	3	$\searrow$ replace 2	3
e	5	5	6	5	4	4	$\searrow$ equal 2

Candidate generation

- Word stump index
  - word stumps = all substrings with up to 2 characters removed
  - their: their, heir, teir, thir, thei, eir, ..., **ther**, ...
  - there: there, here, tere, thre, thee, ther, ..., **ther**, ...
  - if no common stump $\rightarrow$ edit distance > 2

References

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.