A Fast Randomized Algorithm for Massive Text Normalization

Nan Jiang, Chen Luo, Vihan Lakshman, Yesh Dattatreya, Yexiang Xue

Purdue University, Amazon Search

Lexical Normalization

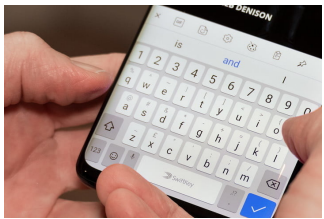
It is the process of transferring

- non-standard
- informal
- misspelled tokens

into their standardized counterparts as well as converting

- words of various tenses
- pluralization

into a consistent representation.



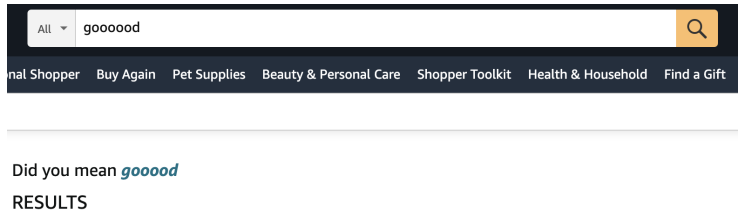


Figure: Two common scenarios where people make typos. (left) typing on phone. (right) search on web.

Current Bottlenecks

1. Mobile computing, Social networks, Web search contain huge amount of typos.
2. At the first step of deep learning models, typos will be all mapped to [UNK].
3. Human annotation is expensive.
4. Existing methods are slow for massive dataset.

Our Contribution:

1. is adaptive to diverse domains;
2. does not require annotation or supervised training;
3. is faster by using LSH to quickly compute the similarity for words;

Morphological[1] Similarity in Linguistics

If two words share the a lot of subwords, then they are likely to be the same words.

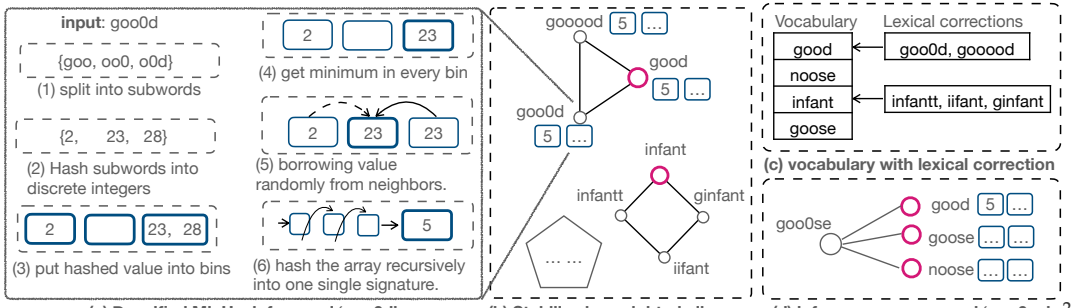
Examples:

`amazingg, amazinggg, mazing, mazinggg,amazinggggg, amazingggggggg, amazingggggggggggg, amazingggggggggggg, mazingggg, amazingggggggggg, soamazing, amazings, amazinggggggg`

`good, goose, noose`

[1]Morphology analyzes the structure of words: prefixes, stems and suffixes.

The architecture of the proposed algorithm



[2]Densified MinHash is a most recently proposed LSH algorithm.

Remarks

The defects

For million words of input, the probability for event "a non-relevant word is collided with a group of similar words" become large.

Our solution

We repeat the LSH for T times, and remove those edges with low weight.

Experiments

Examples from Twitter Dataset

Representative	Similar words
there	thereâ, therea, ithere, therer
night	gnight, nightï, nightâ, gnightâ, dnight, nighti
friends	friend, friendsss, friendz, friendss, friendzz, friendsssss, myfriends, friendssss, vfriends, myfriend, friendâ, friend1
feeling	feelin, feelingz, feelingg, feelinga, feelinf, feelinfg
morning	mornings, gmorning, morningg, gmornin, morningss, morningo, gmorningg , smorning, morningstar, morningâ, morningon

Empirical Running Time Comparison

Datasets	Methods	Indexing (Mins) Single	Indexing (Mins) Multi	Inference (Mins)
Twitter	FLAN ($\alpha = 0.2$)	**40•**	**3•**	**18•**
	Hunspell [1]	171	16	49
	Autocorrect[2]	510	41	154
	FAISS-Glove	408	25	83
	FAISS-Fasttext	44	6	29
Reddit	FLAN ($\alpha = 0.2$)	**59•**	**12•**	**26•**
	Hunspell [1]	520	46	71
	Autocorrect[2]	731	93	221
	FAISS-Glove	514	29	101
	FAISS-Fasttext	70	19	42

Twitter sentiment140: 1.6 million tweets; 0.7 million distinct words.

Reddit: 10 million sentences; 2.7 million unique words.

Quality of Correction Comparison

We send the corrected results of 100 sentences to AmazonMturk for human evaluation.

Datasets	Methods	Precision	Recall	F1-Score
Twitter	FLAN	60.45%	**41.76%**•	**49.39%**•
	Hunspell[1]	37.93%	35.71%	36.79%
	Autocorrect[2]	51.79%	28.57%	36.83%
	faiss-Glove	**71.43%**•	9.34%	16.52%
	faiss-Fasttext	65.28%	24.18%	35.28%
Reddit	FLAN	**84.85%**•	**34.33%**•	**48.88%**•
	Hunspell [1]	42.53%	**34.33%**•	37.99%
	Autocrrect [2]	66.00%	32.84%	43.85%
	faiss-Glove	63.64%	17.16%	27.04%
	faiss-Fasttext	75.71%	22.39%	34.56%

FLAN get good Recall and F1-scores compared with the baselines.

Impact to Downstream Applications - Perturbed GLUE benchmark

Subtask	Noise	Metrics	No corr.	Ours	Autocorrect	Hunspell	Glove	Fasttext
MRPC	20%	Acc.	78.67	**78.92**•	**78.92**•	74.26	78.18	78.18
		F1	84.26	84.83	84.07	82.98	**84.89**•	84.83
MRPC	40%	Acc.	76.22	**77.94**•	77.69	74.51	77.43	77.69
		F1	84.24	**85.09**•	84.17	83.38	84.71	84.49
MRPC	60%	Acc.	67.89	**69.11**•	67.11	65.44	67.64	67.89
		F1	74.10	**78.64**•	74.80	72.62	73.60	73.85

Impact to Downstream Applications - Twitter Sentiment Classification

Methods	Valid Accuracy	Test Accuracy
No Corr.	79.40%	79.39%
FLAN	**79.54**%●	**79.62**%●
Hunspell [1]	79.08%	79.16%
Autocorrect [2]	79.06%	79.18%
FAISS + Glove	79.42%	79.41%
FAISS + Fasttext	79.44%	79.41%

References

[1] Leena Al-Hussaini. Experience: Insights into the benchmarking data of hunspell and aspell spell checkers. *ACM J. Data Inf. Qual.*, 8(3-4):13:1–13:10, 2017.

[2] Peter Norvig. Natural language corpus data. *Beautiful data*, pages 219–242, 2009.

Q & A