GIS search queries correction

Vadim Fomin, NSU
Ivan Bondarenko, MIPT

Candidate search

String metric + dictionary lookup
Problems:

1. Too many unusual tokens:
   хоум кредит банк → ухом кредит банк
2. How bad can a typo be?
   ditroitigers → detroit tigers, log wood → dog food

Candidate search

Iterative spelling correction as “unsupervised” candidate search:
ditroitigers → ditroittigers → detroittigers → detroit tigers

Cucerzan S., Brill E. Spelling correction as an iterative process that exploits the collective knowledge of web users //Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. – 2004.

Candidate selection

Which suggestion is better? Why?

сабака → собака / кабака
зодорнов → задорнов / лингвистика
еда для кта → еда для кита / еда для кота

Candidate selection

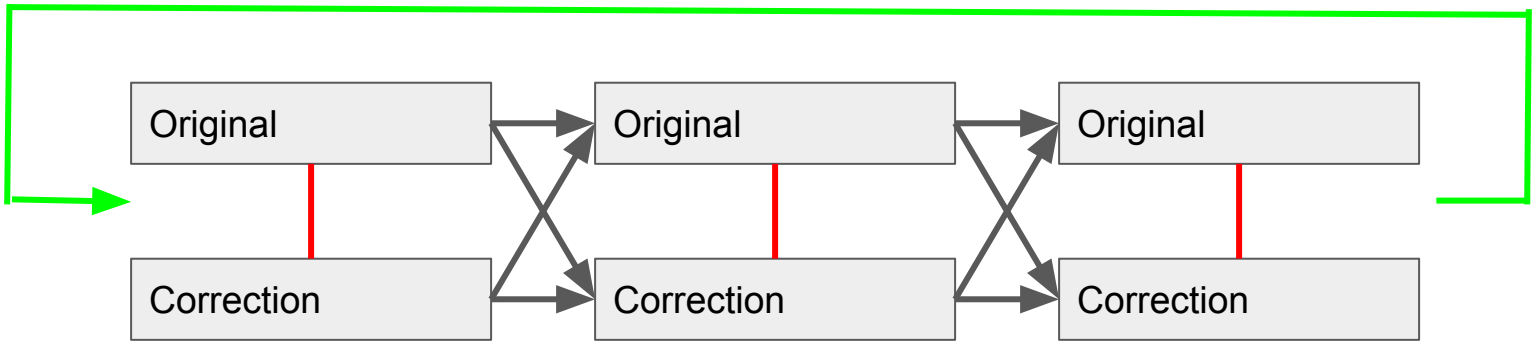
Kernighan M. D., Church K. W., and Gale W. A. (1990) A spelling correction program based on a noisy channel model. In Proceedings of the 13th conference on Computational linguistics, pp. 205–210. Association for Computational Linguistics.

Candidate selection

Sorokin A. A., Shavrina T. O. Automatic spelling correction for Russian social media texts //Proceedings of the International Conference “Dialog”(Moscow. – 2016. – С. 688-701.

Reranking: models as features

Overall architecture



Training

Logistic regression

Winner – loser → 1

Loser – winner → 0

Dataset

"аверкеева,22",аверкеева 22,аверкиева 22,1 × 14 400 — supervised

сорокина 24/б, сорокина 24 б × 949 000 — unsupervised

1	Correction length	18	*а* → *о*, *о* → *а*, *е* → *и*, or *и* → *е*
2	Simple Levenshtein distance	19	*ы* → *и*, *ё* → *о*, *ю* → *у* after *ж*, *ч*, *ш*, *щ*
3	Ngram-model score	20	*цы* → *ци* or *ци* → *цы*
4	OOV in corrections	21	*ыва* → *ова*
5	Vocabulary → OOV	22	*аро* → *оро* or *ало* → *оло*
6	OOV → vocabulary	23	*э* → *е*
7	Corrections that are more frequent than the originals	24	*ща* → *ще*
8	Simple Levenshtein distance, only OOV originals	25	*пре* → *при* or *при* → *пре*
9	Simple Levenshtein distance, originals in vocabulary only	26	*э* → *и*
10	1-operation corrections	27	*ё* → *йо* or *е* → *йо*
11	Space deletions	28	unvoiced → voiced, or voiced → unvoiced
12	Space insertions	29	*зн* → *здн*, *сн* → *стн*, *сл* → *стл*, *нст* → *нтст*, *здн* → *зн*, *стн* → *сн*, *стл* → *сл*, or *нтст* → *нст*
13	OOV that can be split into two vocabulary words	30	*хк* → *гк*
14	Weighted keyboard layout Levenshtein distance	31	*н* → *нн*, *с* → *сс*, *м* → *мм*, *ф* → *фф*, or vice versa
15	Weighted Levenshtein distance with insertion weight 10	32	*ь* → *ъ* or *ъ* → *ь*
16	Weighted Levenshtein distance with deletion weight 10	33	insertion of *ь* as the fourth-to-last letter
17	Simple Levenshtein distance between phonetic codes	34	*тся* → *ться* or *ться* → *тся*

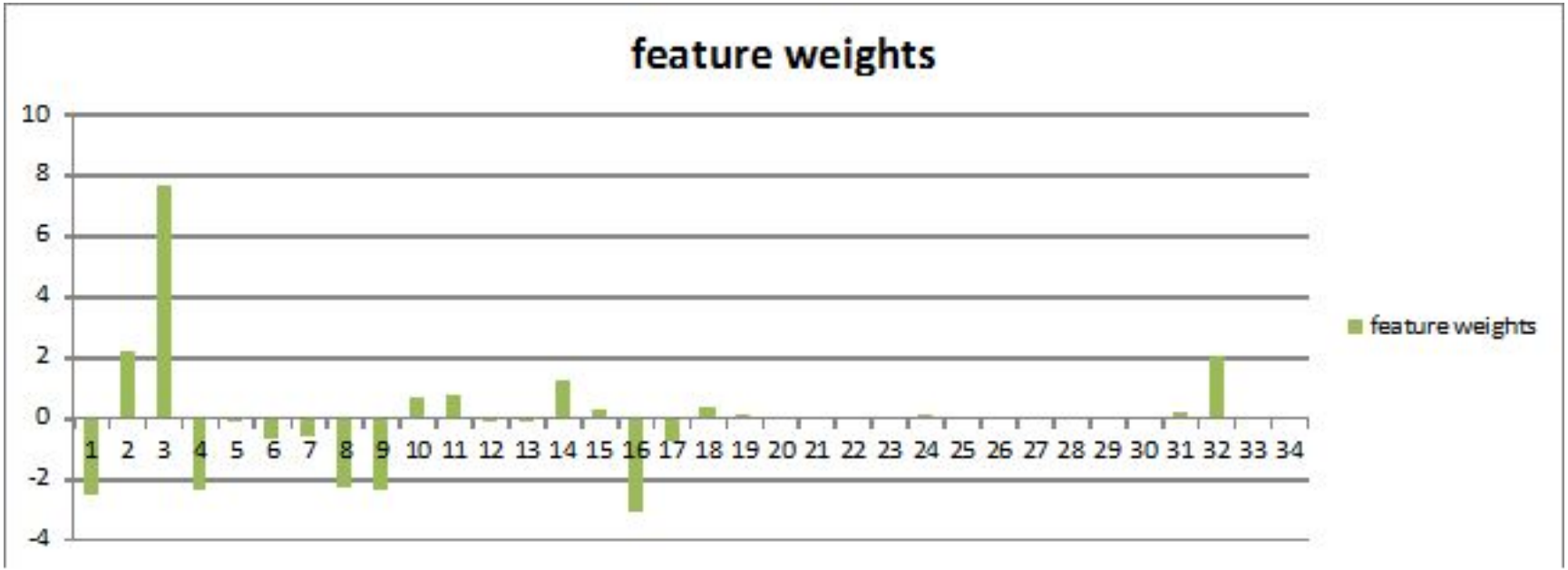
Features

семёновская 8 → семёновская <NUMBER>

Features

Rejected features:
Word2vec as a probabilistic language model
Various settings of ngram-models
Ngram-model of morpho-tags
Ngram-model of prepositions and morpho-tags
Ngram-model of top-words and morpho-tags
Corrections that are more frequent than the originals etc.

Feature importance



1	Correction length	18	*а* → *о*, *о* → *а*, *е* → *и*, or *и* → *е*
2	Simple Levenshtein distance	19	*ы* → *и*, *ё* → *о*, *ю* → *у* after *ж*, *ч*, *ш*, *щ*
3	Ngram-model score	20	*цы* → *ци* or *ци* → *цы*
4	OOV in corrections	21	*ыва* → *ова*
5	Vocabulary → OOV	22	*аро* → *оро* or *ало* → *оло*
6	OOV → vocabulary	23	*э* → *е*
7	Corrections that are more frequent than the originals	24	*ща* → *ще*
8	Simple Levenshtein distance, only OOV originals	25	*пре* → *при* or *при* → *пре*
9	Simple Levenshtein distance, originals in vocabulary only	26	*э* → *и*
10	1-operation corrections	27	*ё* → *йо* or *е* → *йо*
11	Space deletions	28	unvoiced → voiced, or voiced → unvoiced
12	Space insertions	29	*зн* → *здн*, *сн* → *стн*, *сл* → *стл*, *нст* → *нтст*, *здн* → *зн*, *стн* → *сн*, *стл* → *сл*, or *нтст* → *нст*
13	OOV that can be split into two vocabulary words	30	*хк* → *гк*
14	Weighted keyboard layout Levenshtein distance	31	*н* → *нн*, *с* → *сс*, *м* → *мм*, *ф* → *фф*, or vice versa
15	Weighted Levenshtein distance with insertion weight 10	32	*ь* → *ъ* or *ъ* → *ь*
16	Weighted Levenshtein distance with deletion weight 10	33	insertion of *ь* as the fourth-to-last letter
17	Simple Levenshtein distance between phonetic codes	34	*тся* → *ться* or *ться* → *тся*

Ngram-model score	Vocabulary → OOV
Weighted Levenshtein distance with deletion weight 10	Space insertions
Correction length	*ща* → *ще*
Simple Levenshtein distance, originals in vocabulary only	OOV that can be split into two vocabulary words
OOV in corrections	*ы* → *и*, *ё* → *о*, *ю* → *у* after *ж*, *ч*, *ш*, *щ*
Simple Levenshtein distance, only OOV originals	insertion of *ь* as the fourth-to-last letter
Simple Levenshtein distance	*э* → *е*
ь → *ъ* or *ъ* → *ь*	unvoiced → voiced, or voiced → unvoiced
Weighted keyboard layout Levenshtein distance	*пре* → *при* or *при* → *пре*
Simple Levenshtein distance between phonetic codes	*цы* → *ци* or *ци* → *цы*
Space deletions	*аро* → *оро* or *ало* → *оло*
OOV → vocabulary	*зн* → *здн*, *сн* → *стн*, *сл* → *стл*, *нст* → *нтст*, *здн* → *зн*, *стн* → *сн*, *стл* → *сл*, or *нтст* → *нст*
1-operation corrections	*ыва* → *ова*
Corrections that are more frequent than the originals	*э* → *и*
а → *о*, *о* → *а*, *е* → *и*, or *и* → *е*	*ё* → *йо* or *е* → *йо*
Weighted Levenshtein distance with insertion weight 10	*хк* → *гк*
н → *нн*, *с* → *сс*, *м* → *мм*, *ф* → *фф*, or vice versa	*тся* → *ться* or *ться* → *тся*

Feature importance

Ngram-model is incredibly helpful in picking a good candidate
Weighted string metrics are good at rejecting the bad ones
(Weighted Levenshtein distance with a high deletion weight is the best)

Morphological models and misspelling type features are not
(but ь → ъ!)

Evaluation

original:	мастер фиш	на	улленина
suggestion:	мастер фиш	над	ул ленина
expected:	мастерфиш	на	ул ленина
	FN	FP	TP

Evaluation

	Precision	Recall	F1
Model in question	98.4%	58.6%	73.5%
Hunspell	19.6%	22.2%	20.8%
2GIS algorithm	54.6%	31.5%	40.0%

Evaluation: HunSpell

An extended dictionary
First candidate, if any

Result: **20.8%**

Evaluation: 2GIS algorithm

No spelling correction unit

City-dependent data

Top-1 candidate query = output

Evaluation: SpellRuEval

SpellRuEval	82.0%	69.3%	75.0%

Model in question	98.4%	58.6%	73.5%

Results

— An annotated corpus
— Effectiveness of ngram-models and reranking tested on a quite different corpus
— Feature engineering
— Feature importance analyzed
— An open-source system

Acknowledgements

We are grateful to Botan Investments for supporting students’ interest to machine learning.