

WG1 Dresden

2017-09-06

DeepSpell

**FTS token auto-suggest using
neural networks**



- Currently token completion suggestion can only be done using the FTS5 vocab table
 - All tokens are stored with their number of occurrence there
 - No relations
 - No classification of tokens
- Doing an auto-suggest for tokens based on more than the available information takes a lot of time consuming algorithm and data crunching
- Idea: Let the Artificial Intelligence find out what the best suggestion is

- FTS5 vocab table example
- Even if we would add priorities or other „importance factors“ to tokens we still would need to come up with a certain idea „how“
- Still the tokens are/would not be related to each other → context has no influence on suggestion

Term	Occurences
10th	10.000
new	1000
york	50
main	800
st	100.000
...	...

St L[ouis]

vs

St Loui[siana]

→ Subtoken violation

California Los A[ngeles]

vs

California Los A[lamos]

→ Hierarchy violation

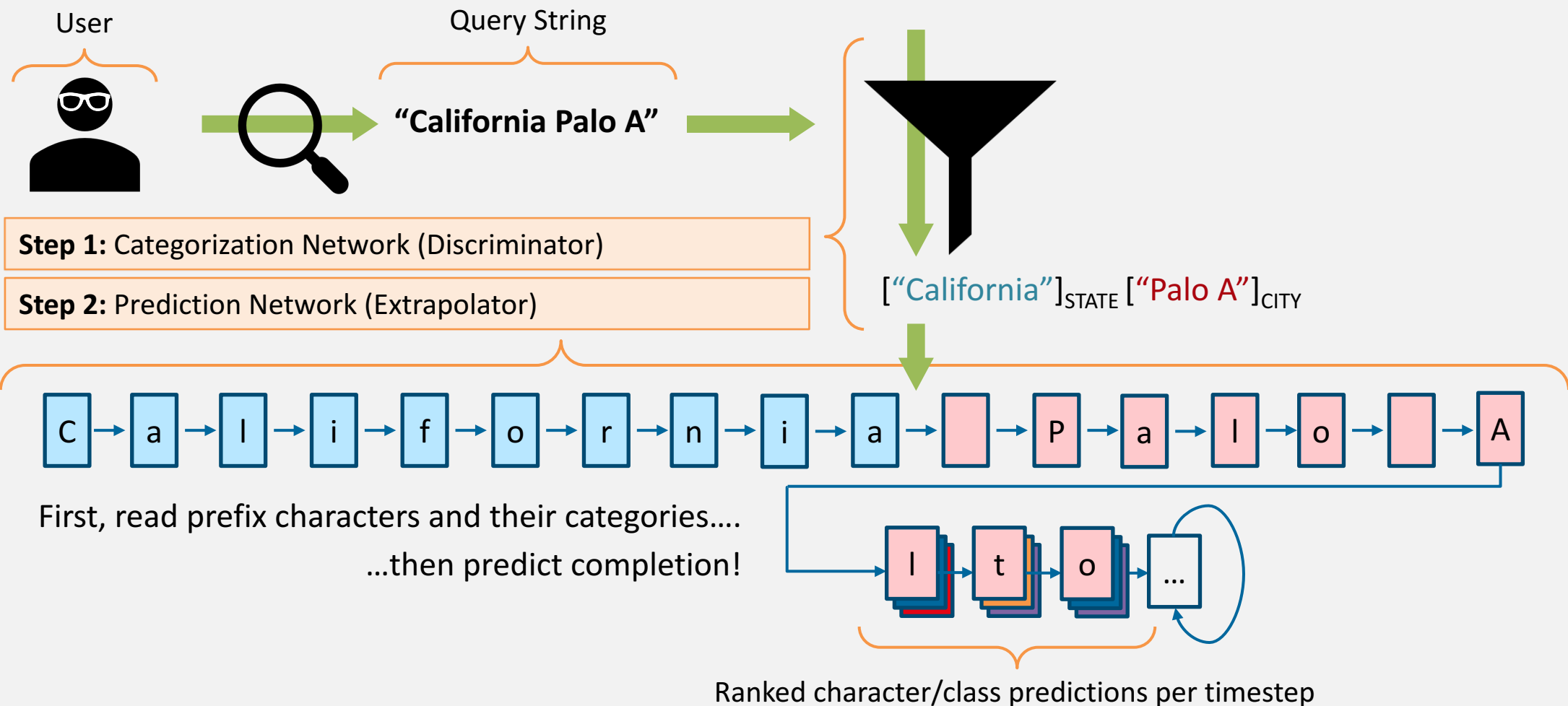
Virginia V[ictoria]

vs

Virginia V[irginia]

→ Redundancy violation

- Leave query string processing to Neural Network (NN):
 - Tokenization
 - NN input is full user string, not just last term:
 - NN Input: "New York C"
 - Old FTS Input: "C"
 - Categorization
 - NN categorizing hierarchy classes before completion, so actually:
 - User Input: "California Palo A"
 - NN Input: "California Palo A" \rightarrow ["California"]_{STATE} ["Palo A"]_{CITY}
 - Suggestion
 - NN returns probability distribution for following characters:
 - $\text{suggest}([\text{"California"}]_{\text{STATE}} [\text{"Palo A"}]_{\text{CITY}}) = \left(\begin{bmatrix} l & p = 0.3 \\ n & p = 0.1 \\ \dots & \dots \end{bmatrix} \begin{bmatrix} t & p = 0.5 \\ a & p = 0.1 \\ \dots & \dots \end{bmatrix} \begin{bmatrix} o & p = 0.3 \\ e & p = 0.2 \\ \dots & \dots \end{bmatrix} \right)$



- Road FTS North America database with full road coverage
 - Using classification of COUNTRY, STATE, CITY, ROAD
 - About 44,000 cities and total of 9,448,382 roads
- Tensorflow in Python to train neural network models
 - Classifier: 2-way LSTM-RNN (forward, backward), ~ 400K neurons
 - Completer/Suggester: 1 LSTM-RNN, ~600K neurons
 - Trained model size: ~ 10 MB
- Python Webservice to showcase prediction

Deep Spell: Neural FTS Suggestions for NDS

deepsp_discr-v2_na_lr003_dec50_bat3072_fw128-128_bw128 / deepsp_extra-v2_na_lr003_dec50_bat2048_256-256

Enter your query

California Los

California Los **Angeles**

Predicted Categories

C a l i f o r n i a L o s
ST ST ST ST ST ST ST ST ST ST ST RO RO RO RO
CI CI CI CI CI CI CI CI CI CI CI CI CI CI CI

Completion

A n g e l e s D r
t C l d u s i r \$ S e \$
s P r t o e t s R o i
e R v a i p a y i C a a
a S u n a n s z a A u c
h M g i l a l c - P i e
u B m c h m o n e L u
k L d s r q v v B d o

- Performance in an embedded/mobile environment?
- Combine with database content to improve suggestions?
- How to implement an interface for this in NDS?
 - Virtual Table in SQLite?
 - Support different frameworks for DeepLearning?