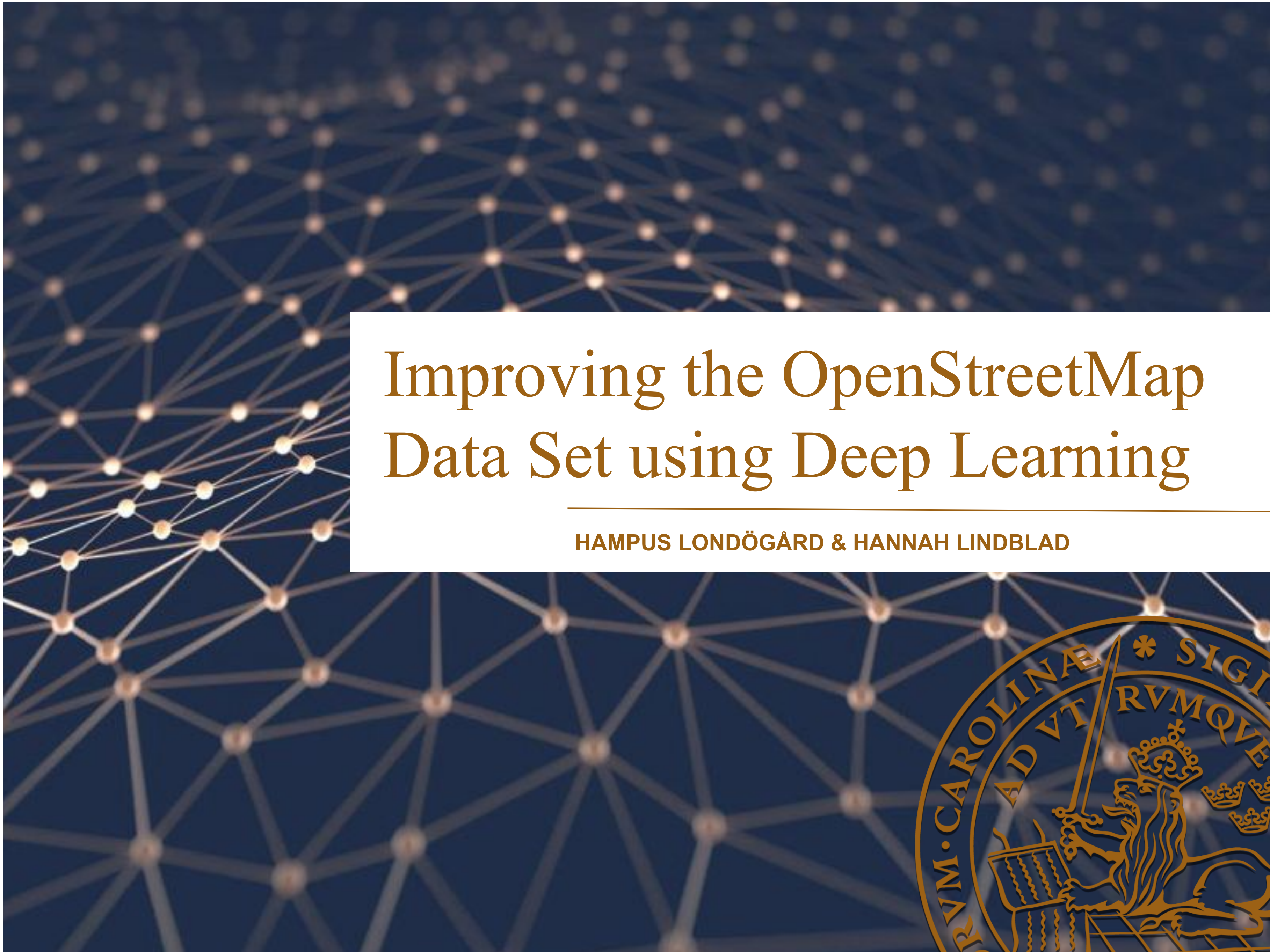


Improving the OpenStreetMap Data Set using Deep Learning

HAMPUS LONDÖGÅRD & HANNAH LINDBLAD

Agenda

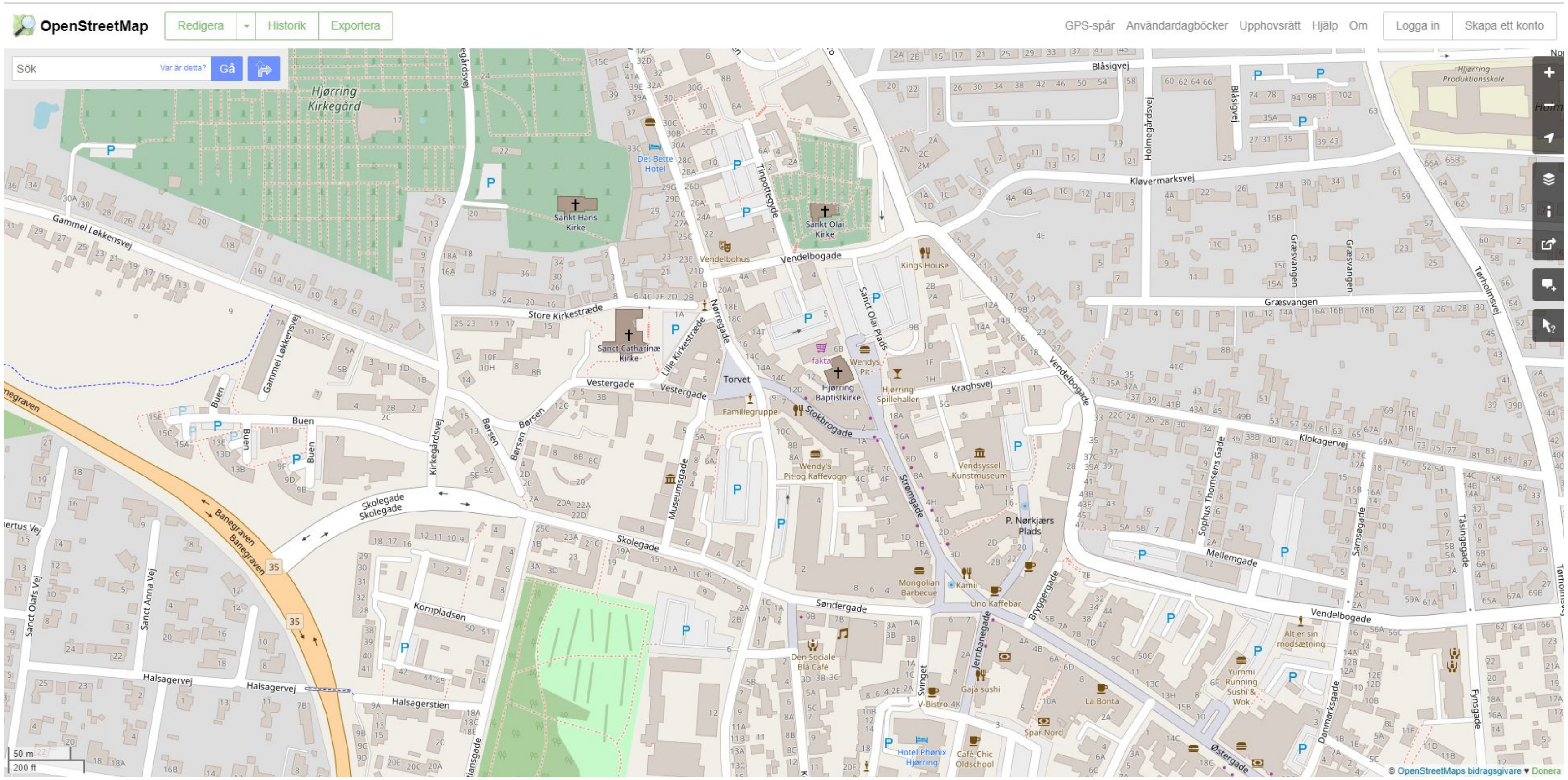
1. Introduction
2. Theory
3. Implementation & Results
4. Future Work

Introduction

- OpenStreetMap (OSM) is an **Open Source** of geographical data
- Anyone can **add**, **edit,** and **delete** data
  → **varying data quality**

- Improve quality by correcting **Way names** in Denmark

OpenStreetMap



OpenStreetMap Data Format

Data

- Components
  - e.g. ***Way***, *Node,* and *Relation*
- Tags (key=value)
  - e.g. ***Name***, *Surface,* and *Highway*

OSM: Example of a Way component

Sträcka: Lille Kirkestræde (95745230) ×

Added footways based on SDFE around Hjørring

Redigerades för 10 månader sedan av hyggemap
Version #4 · Ändringsset #49978828

Etiketter

highway	residential
name	Lille Kirkestræde

Noder

461167737 (del av sträckorna Vestergade (93206057) och Vestergade (95745232))
1109781395
4946655758 (del av sträcka 504516286)
461167740
461167742
447560774 (del av sträcka Nørregade (25477990))

Research Questions

1. How to identify misspelled names for Way components?
2. How to generate suggestions of a misspelled name?
3. How to generate suggestions for ways with missing names?
4. Is it possible to find anomalies in the data from names in combination with other tags like max speed?

Approach

Improving Way name data

1. Missing Names
2. Anomalies
3. Misspelled Names

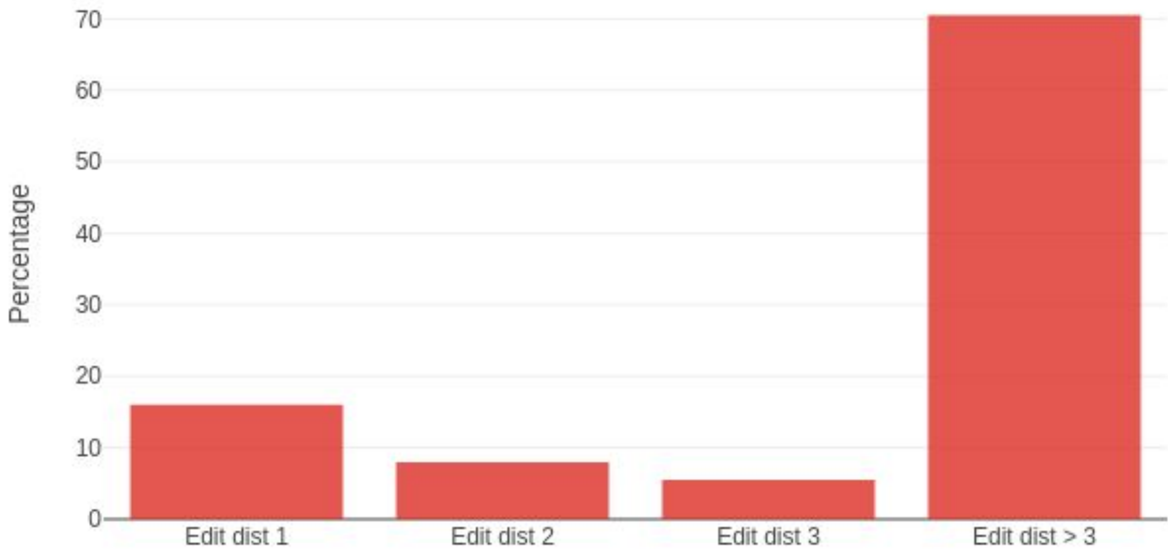
Theory

- Data Understanding
- Evaluation
- Machine Learning

Data Understanding

- Collect and analyze the data (2015-2018)
- Become familiar with data format

Data Understanding: Edit distance



Data Understanding: Problems

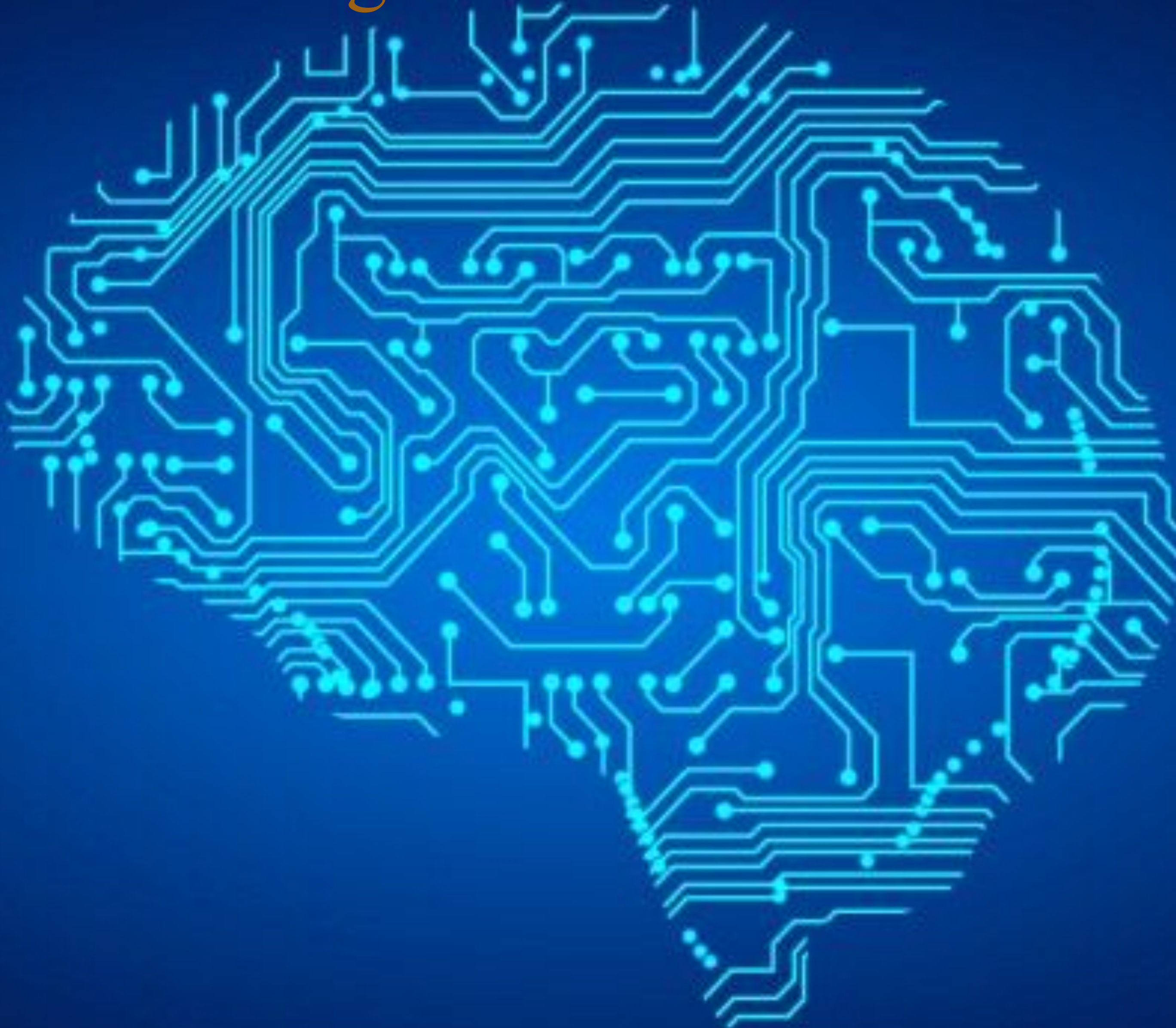
- 300/52 000 Way components edited (2015-2018)
  - Less within one edit distance
- Machine Learning requires large amounts of data
- Supervised machine learning requires both true and false cases

Evaluation

- F1 score
  - Harmonized average of precision and recall
  - **Precision**: measure of exactness or quality whereas **Recall**: measure of completeness or quantity
- Accurate Correction Rate (**ACR**)
  - **CI**: Corrections introduced
  - **EI**: Errors introduced

$$\mathrm{ACR} = \frac{\mathrm{CI}}{\mathrm{EI} + \mathrm{CI}}$$

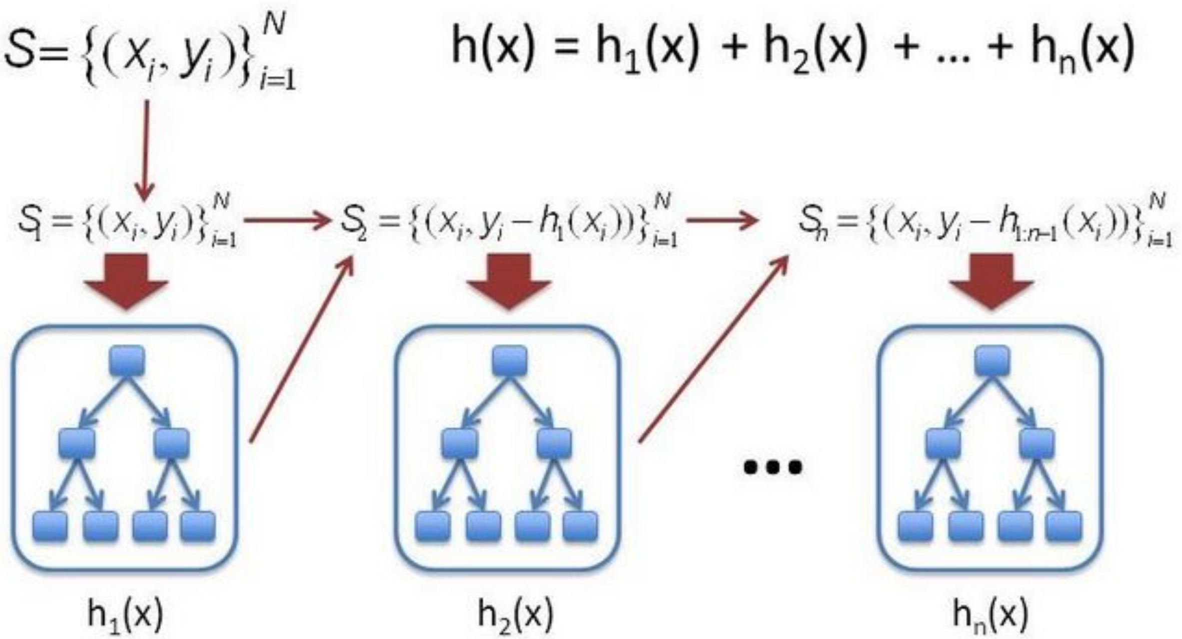
Machine Learning



https://becominghuman.ai/machine-learning-for-dummies-explained-in-2-mins-e83fbc55ac6d

XGBoost

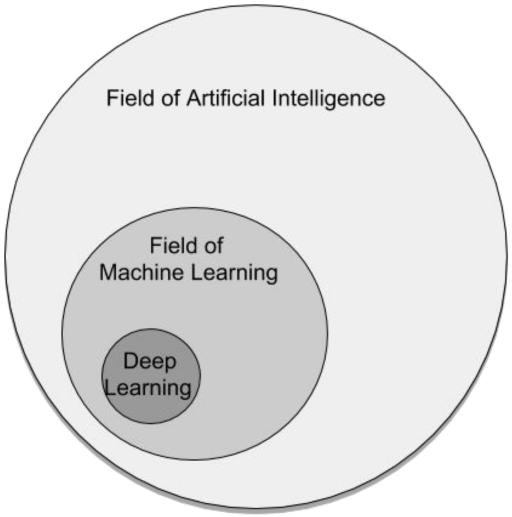
- **Ensemble Learning**
- Multiple Decision Trees built together **sequentially**



http://statweb.stanford.edu/~jhf/ftp/trebst.pdf

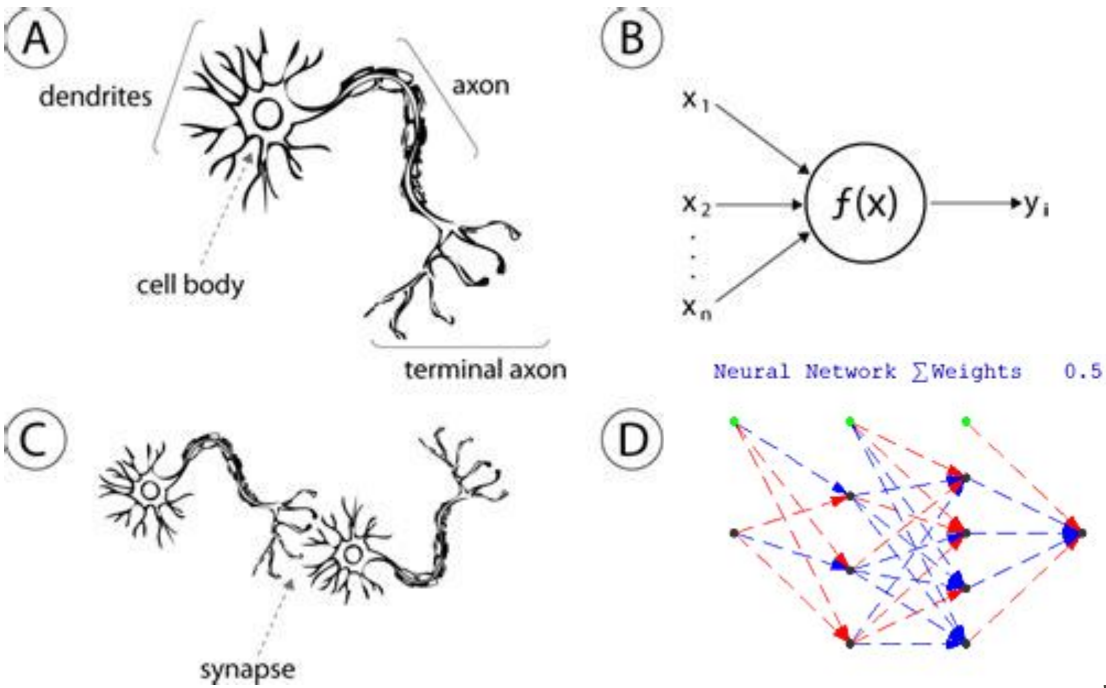
Deep Learning

- Neural Networks
- Autoencoder
- Sequence-2-Sequence
- Bidirectional Recurrent Neural Networks



Neural Network (NN)

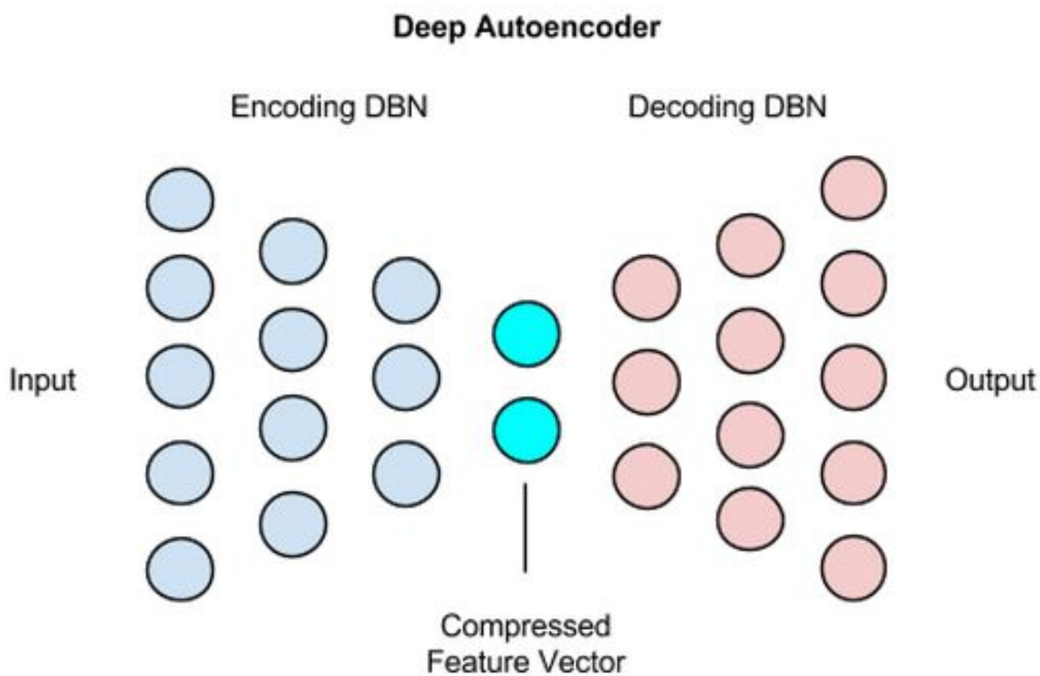
- Tries to mimic the mammalian neurons
- Can either pass forward signal or "keep closed"
- Weights itself during training



s://medium.com/@ivanlilje qvist/the-essence-of-artificial-neural-networks-5de300c995d6

Autoencoder

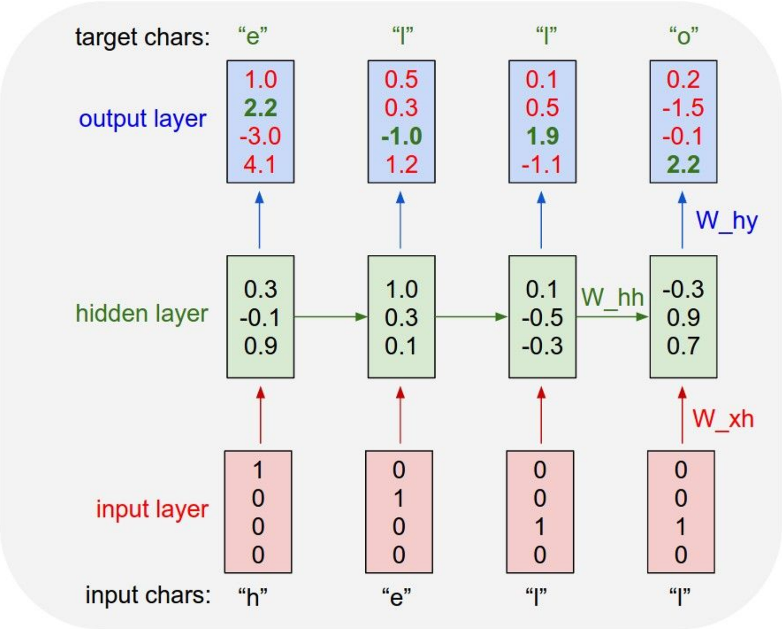
- Finds a structure in the data
- Unsupervised



Eclipse Deeplearning4j
Development Team

Sequence-2-Sequence

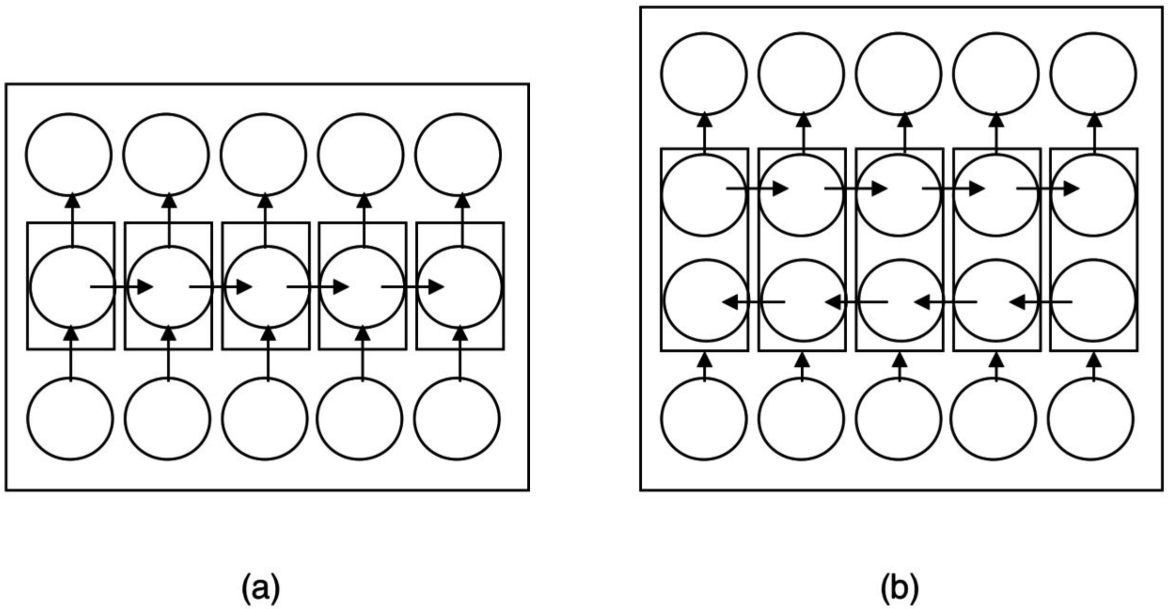
- Makes undefined sequences possible
- Great for **natural language**
- Recurrent Neural Network (RNN) architecture



http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Bidirectional RNN

- Gives future context



Structure overview
(a) unidirectional RNN
(b) bidirectional RNN

Wikimedia commons (2017)

Implementation

1. Missing Names
2. Anomalies
3. Misspelled Names

Missing Names

- Simple algorithmic solution
  - Graph and Label continuity
- Two cases caught
  - Small Tail
  - In-Between

Small Tail

In Between

1
2
Toyota
205
Københavnsvej
Københavnsvej
156
Københavnsvej
Navervej
200
192
194
196B
196C
198
132 kV
Hedegård

Missing Names: Results

- # instances = 6,066,646
- # name suggestions = 756,245
- ~12.5 %

- Results when manually inspecting 50 random instances:

  Error: 34
  No Error: 16

Anomalies

- Name + Tags = True or False
- End system makes use of heuristic
- Anders**gade** + 130 km/h = True (most likely)
    - Expand upon this concept with more tags

Results

XG-Boost achieves 0.91 F1-score
RNN achieves 0.9 F1-score

Why tokenize?

T	**3**	81,802	81,805
F	2	**81,728**	81,730
	5	163,530	**163,535**

T	**72,212**	9,593	81,805
F	6,339	**75,391**	81,730
	78,551	84,984	**163,535**

On todays OSM:
~20k/330k (6%)
~6k/330k (2%)

RNN is more selective

Misspelled Names

- We believed the problem to be a translation problem
    - Take misspelled words to well spelled
- Translation tasks have had great success using Sequence-2-Sequence
- Character-2-Character

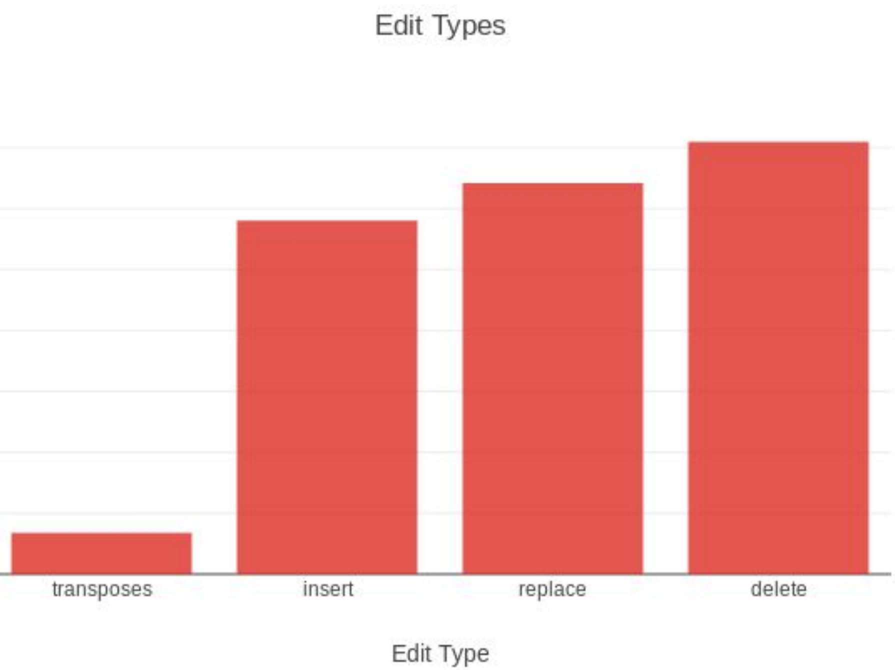
Misspelled Names

- We aimed for modularity
    - This was done by not including language dependant heuristics
- Corpus from *Det Danske Sprog- og Litteraturselskab*
    - Wikipedia dump also works
- OpenStreetMap data

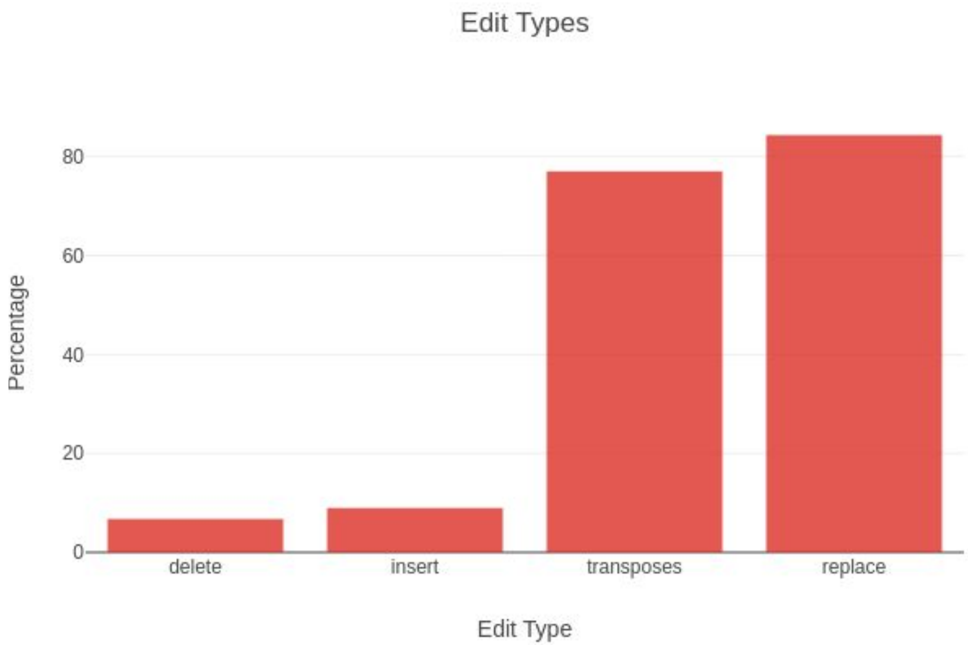
Problems

- No language context
  - I'm **goint** to school...
  - A **goint** venture is a business...
- Compounding language
  - Christian + vej → Christian**s**vej
- High quality data
  - Not nearly enough of data
  - Experiment for gathering misspellings

Comparison of edits



OSM change data



Experiment data

SymSpell

- Symmetric Delete Spell Correction Algorithm
- Language independent spell correction

Originally implemented by Wolf Garbe (https://github.com/wolfgarbe, C#)

Results on Test Data

	ACR
SymSpell	**0.1348**
RNN Baseline	**0.0086**
RNN Autoencoder	**0.5143**
BRNN Autoencoder	**0.6363**
BRNN Autoencoder + SymSpell	**0.6949**

$$\mathrm{ACR} = \frac{\mathrm{CI}}{\mathrm{EI} + \mathrm{CI}}$$

Results on Estonia

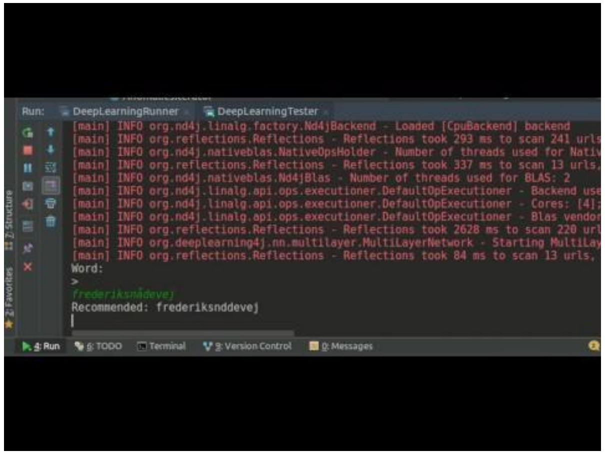
	ACR	F_1
SymSpell	0.36	-
BRNN Autoencoder (dk_alphabet)	0.85	0.983
BRNN Autoencoder (et_alphabet)	0.88	0.986

Results on Today's OSM

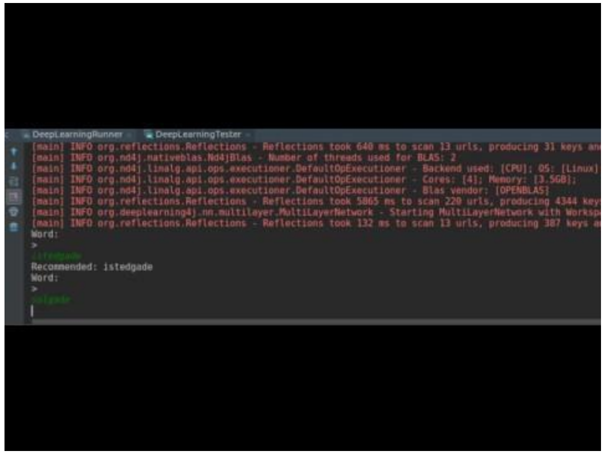
	F_1	ACR	EI	CI	FC
BiRNN Autoencoder 1	0.97	0.004	1653	7	73
BiRNN Autoencoder 2	0.98	0.011	93	1	42
BiRNN Autoencoder 3	0.99	0.167	5	1	3

BiRNN Autoencoder 1: 50% artificially noised names.
BiRNN Autoencoder 2: 0% artificially noised names.
BiRNN Autoencoder 3: 5% artificially noised names.

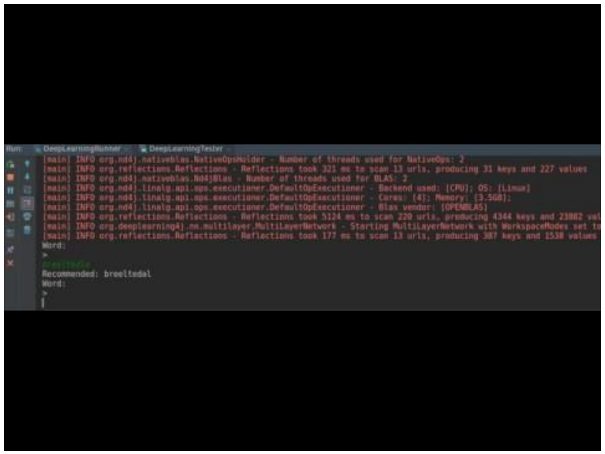
Demo



Run: DeepLearningRunner DeepLearningTester
[main] INFO org.nd4j.linalg.factory.Nd4jBackend - Loaded [CpuBackend] backend
[main] INFO org.reflections.Reflections - Reflections took 293 ms to scan 241 urls
[main] INFO org.nd4j.nativeblas.NativeOpsHolder - Number of threads used for Nativ
[main] INFO org.reflections.Reflections - Reflections took 337 ms to scan 13 urls,
[main] INFO org.nd4j.nativeblas.Nd4jBlas - Number of threads used for BLAS: 2
[main] INFO org.nd4j.linalg.api.ops.executioner.DefaultOpExecutioner - Backend use
[main] INFO org.nd4j.linalg.api.ops.executioner.DefaultOpExecutioner - Cores: [4];
[main] INFO org.nd4j.linalg.api.ops.executioner.DefaultOpExecutioner - Blas vendor
[main] INFO org.reflections.Reflections - Reflections took 2628 ms to scan 220 url
[main] INFO org.deeplearning4j.nn.multilayer.MultiLayerNetwork - Starting MultiLay
[main] INFO org.reflections.Reflections - Reflections took 84 ms to scan 13 urls,
Word:
frederiksnådevej
Recommended: frederiksnddevej
4: Run 6: TODO Terminal 9: Version Control 0: Messages



DeepLearningRunner DeepLearningTester
Word:
Recommended: istedgade
Word:



Run: DeepLearningRunner DeepLearningTester
Word:
Recommended: breeltedal
Word:

Future Work

- Generate or find a better data set to work from
- Add Meta information
  - Such as entity-recognition and geographical knowledge
- Make use of history

Conclusions

1. We have created an algorithm that fills ~9 % of all Missing Names correct

Conclusions

1. We have created an algorithm that fills ~9 % of all Missing Names correct
2. We have built a neural network that has learned to identify anomalies and can forward them to a manual editor with an F1-score of 90%
   a. Completely without natural data

Conclusions

1. We have created an algorithm that fills ~9 % of all Missing Names correct
2. We have built a neural network that has learned to identify anomalies and can forward them to a manual editor with an F1-score of 90%
   a. Completely without natural data
3. We have shown that neural networks:
   a. Can learn how to correct Way names without context
   b. Can learn a language model
   c. Data is a big problem but can be solved with time

Conclusions

1. We have created an algorithm that fills ~9 % of all Missing Names correct
2. We have built a neural network that has learned to identify anomalies and can forward them to a manual editor with an F1-score of 90%
   a. Completely without natural data
3. We have shown that neural networks:
   a. Can learn how to correct Way names without context
   b. Can learn a language model
   c. Data is a big problem but can be solved with time

Hard to do any real conclusions as data is a problem

Questions?