Extracting Information from Unstructured Data

Anzor Gozalishvili & Ioseb Khutsishvili @ MaxinAI

Part 1

Outsourcing forces for AI projects

Our experience

- Many different projects
- Great experiences
- Knowledge sharing
- Community

Demand in different domains of AI

- Natural Language Processing
- Computer Vision
- Data Analysis and Prediction
- Big Data

Most Exciting Projects

- Meaningful number extraction from market news
- Extracting merchant names from bank transactions
- Menu recommendation system
- SalesForce plugin for predicting user conversion

Data collection and preparation

- Web scraping
- Data by customers
- Data labeling, tools
- Categorizing and storing datasets

Data Augmentation

- Creating more data from little
- Creating data from nowhere (OCR case)
- Creating proper datasets

Community & School of AI

- Data Science Tbilisi meetups
- Access to AI experts
- Young people around us
- School of AI

Part 2

Extracting Data

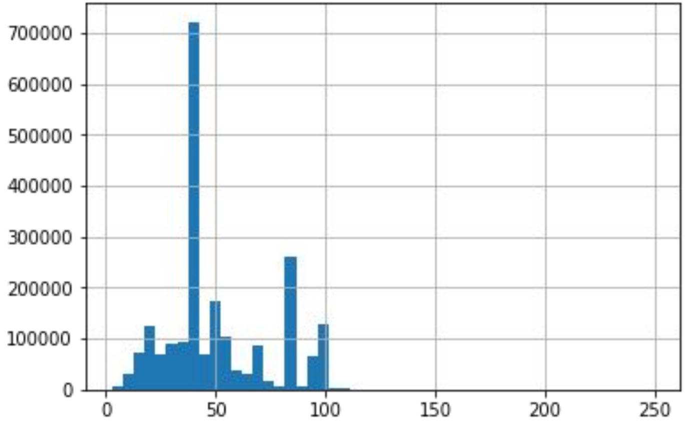
Task Description

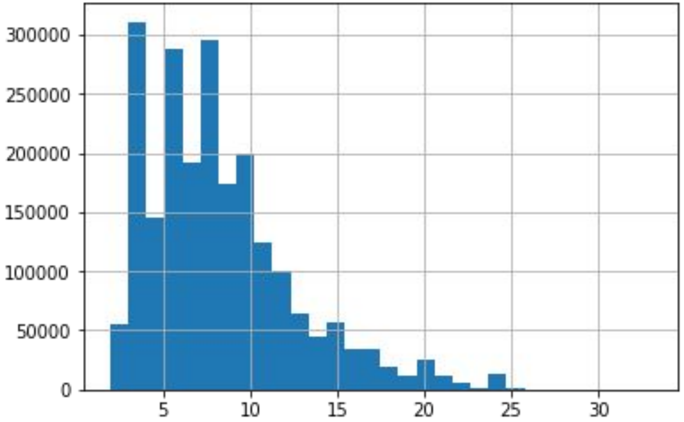
CHECKCARD 1228 DEBIT CARD PURCHASE NETFLIX.COM 8005858131 CA
ATLANTA GA 24755423363153630626348

- **Transaction keywords**: CHECKCARD, PURCHASE
- **Location**: CA ATLANTA
- **Some numbers :** 1228, 8005858131, 24755423363153630626348
- **Merchant Name**: NETFLIX.COM

Data

- 1 million transaction records
- Transaction and merchant name lengths





Goal

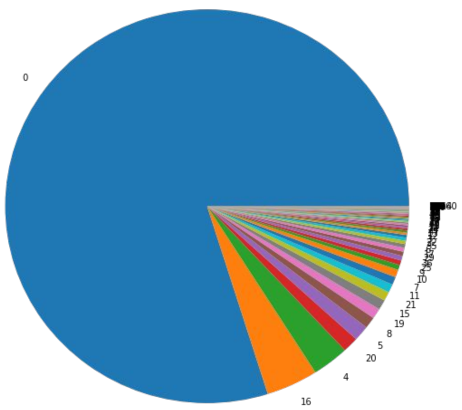
- Create model to extract merchant names from transactions
- Get high accuracy on predictions
- Make model able to detect previously unseen merchant names

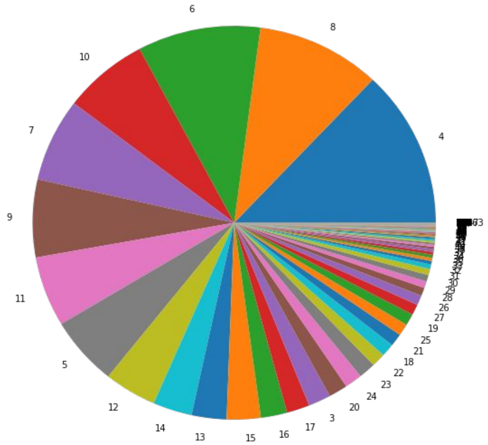
Possible solutions

- Regex
- Sequence to sequence RNN
- Deep Convolutional Neural Network
- Others.

Data is important!

- Merchant names are at the beginning (78.9%)
- Model overfitting





Data Augmentation

- Sliding merchant name

```
"KROGER #442 0000004486661119799 999999942838666111979"
```

```
" #442 KROGER  000000448666111979 999999942838666111979"
" #442 000000448666111979 KROGER  999999942838666111979"
```

- Padding transactions

```
"USAA.COM PMT - THANK YOU SAN ANTONIO TX"
"APL* ITUNES.COM/BILL 866-712-7753 CA"
```

```
"USAA.COM PMT - APL* ITUNES.COM/BILL 866-712-7753 CA SAN ANTONIO TX"
"APL* USAA.COM PMT - THANK YOU SAN ANTONIO TX.COM/BILL 866-712-7753 CA"
```

- Exchanging merchant names

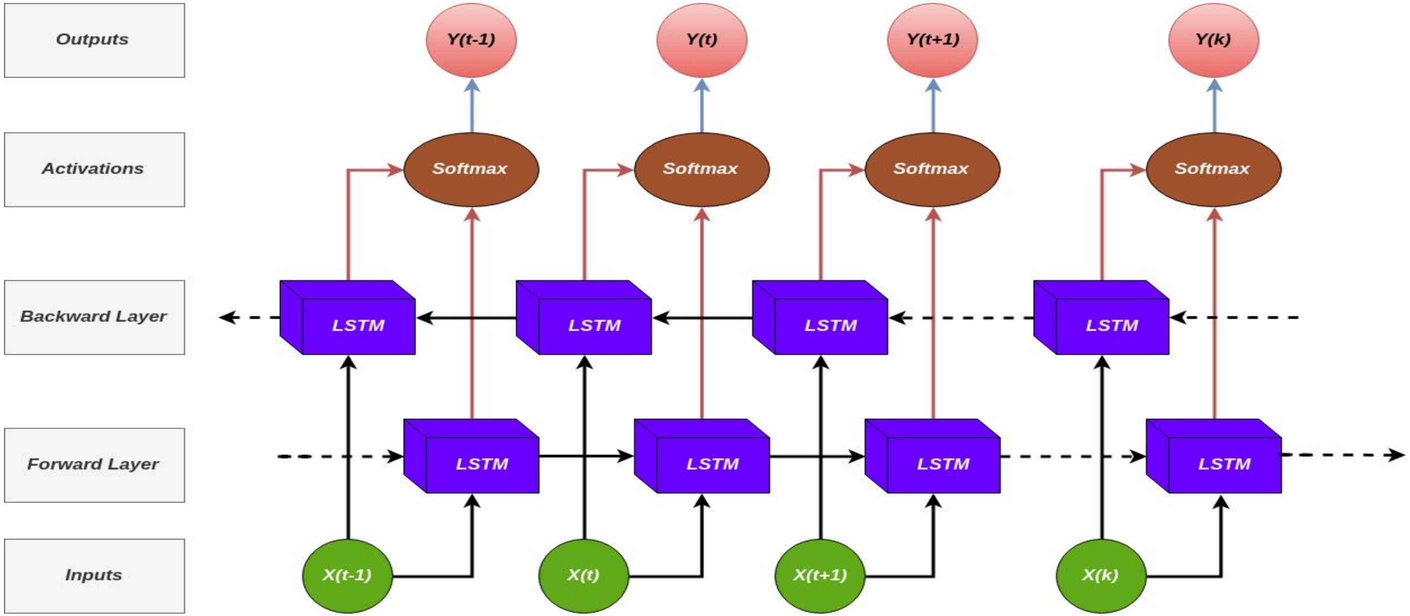
```
"KROGER #442 000000448666111979 999999942838666111979"
"CVS/PHARMACY #03818 BOCA RATON FL"
```

```
"CVS/PHARMACY #442 000000448666111979 999999942838666111979"
"KROGER #03818 BOCA RATON FL"
```

Using Deep Recurrent Neural Network

- Part of speech tagging
- Can detect several merchants separately
- Vocabulary dependent (can’t find out of vocabulary tokens)

Neural Network architecture



Data preparation

- Smallest unit: token
- Labeling transaction text tokens
- One hot encoding

Results

- Train: 98%
- Validation: 97.2%
- Test:96.4%

Idea of Using Convolutional Neural Networks

Daniel C. LaCombe, Jr   Home   About   Projects   Résumé   Feed

Deep Learning for RegEx

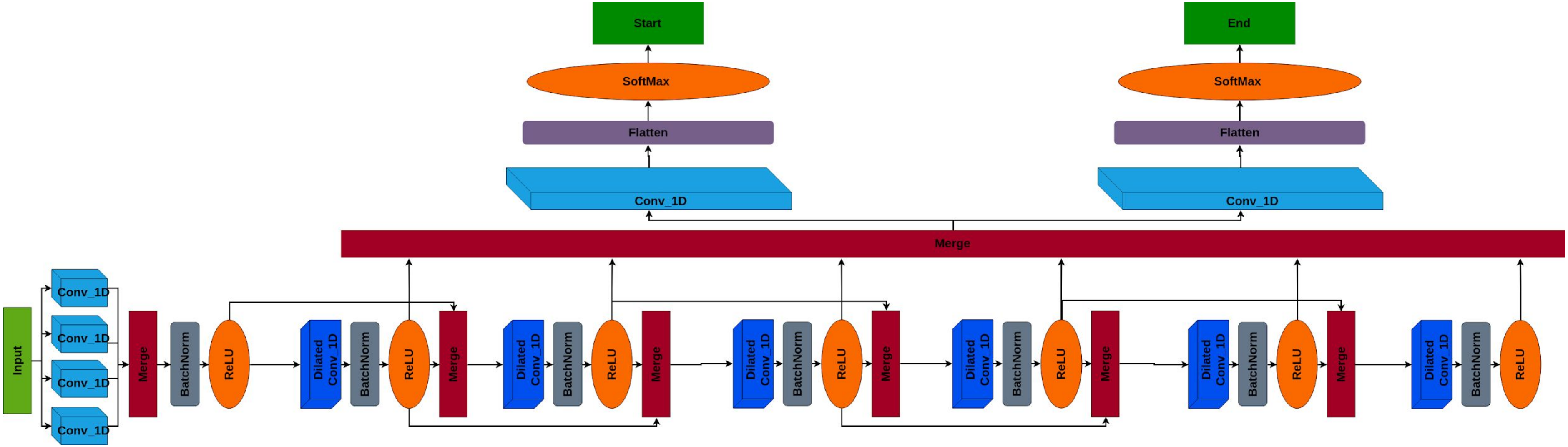
Nov 13, 2016

Recently I decided to try my hand at the Extraction of product attribute values competition hosted on CrowdAnalytix, a website that allows companies to outsource data science problems to people with the skills to solve them. I usually work with image or video data, so this was a refreshing exercise working with text data. The challenge was to extract the Manufacturer Part Number (MPN) from provided product titles and descriptions that were of varying length – a standard RegEx problem. After a cursory look at the data, I saw that there were ~54,000 training examples so I decided to give Deep Learning a chance. Here I describe my solution that landed me a 4th place position on the public leaderboard.

Using Convolutional Neural Network

- Rarely used in nlp tasks
- Can detect one merchant
- Can detect out of vocabulary merchant names

Neural Network architecture



Data preparation

- Smallest unit: character
- Setting start and end indices of merchant names
- Using binary encoding of characters for embedding

Results

- Train: 99.1%
- Validation: 98.4%
- Test:97.1%

Comparing Models (RNN vs CNN)

- **Pros:**
  - Faster to train (20 min/epoch)
  - Multiple merchant extraction
- **Cons:**
  - Less accurate
  - Vocabulary dependent

- **Pros:**
  - More accurate
  - Doesn’t require vocabulary
- **Cons:**
  - Slower to train (2 hour/epoch)
  - Single merchant extraction

Conclusions

- Many solutions
- Convolutional Neural Networks can be used in NLP
- Client chooses the Model

Datathon

- Tomorrow…
- Bring your notebooks with you
- We have some interesting data with us
- Let’s extract some useful information together...

Questions?!