Error Detection and Correction using Record Embeddings

Manjunath Shettar

Meghana Moorthy Bhat

Yogesh Chockalingam

Agenda

- Motivation
- Problem Statement
- Background
- Approach
- Results
- Conclusion

Motivation

- Error detection and correction of large datasets is laborious.
- Current solutions use graphical models for databases which are computationally expensive and memory hungry to perform any analysis in the ecosystem.

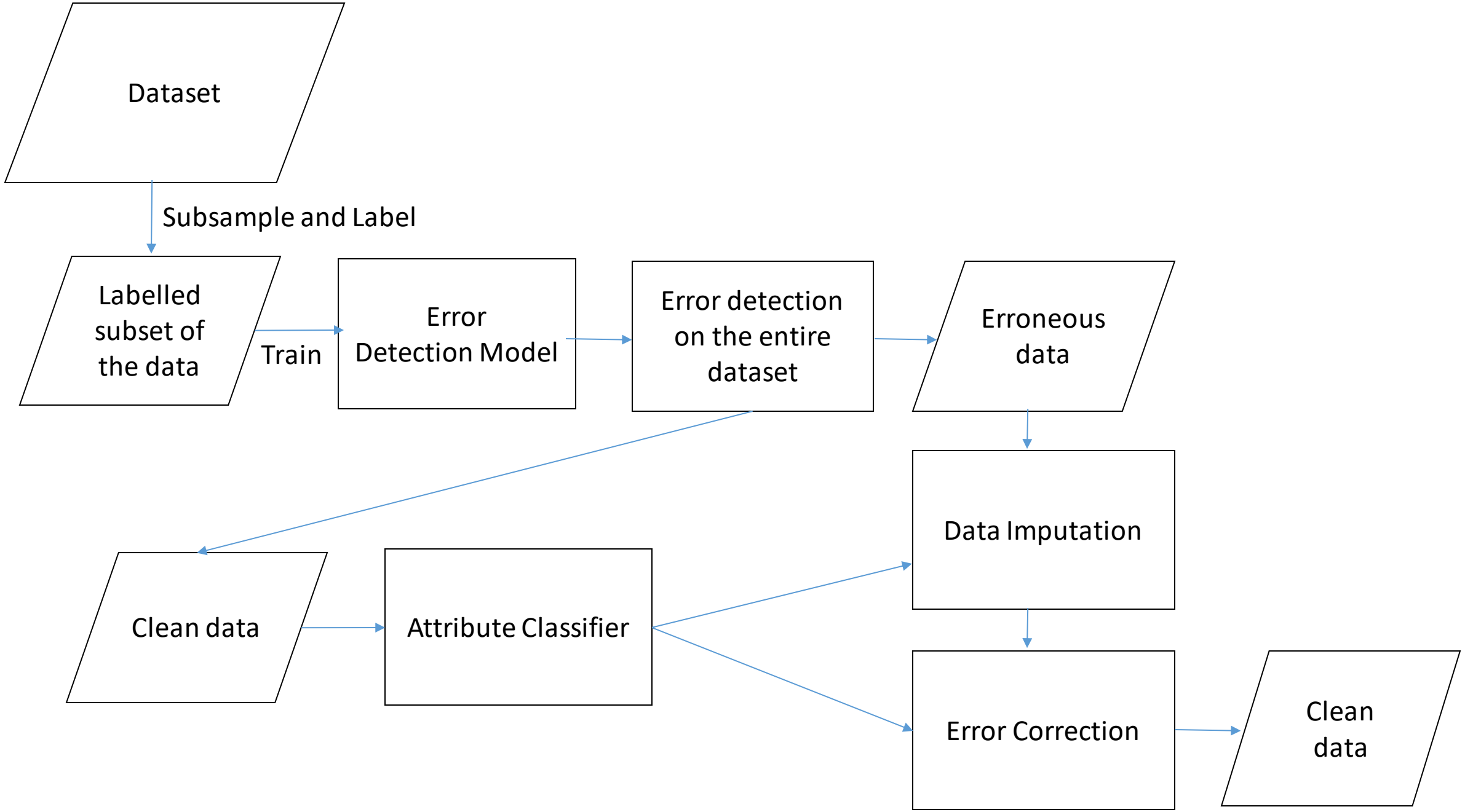
Background

- Structure2vec
  - Embedding latent variable models into feature spaces.
  - Performs sequence of function mappings similar to graphical model inference procedures.
- Fast.ai
  - Deep Learning applied on structured data.

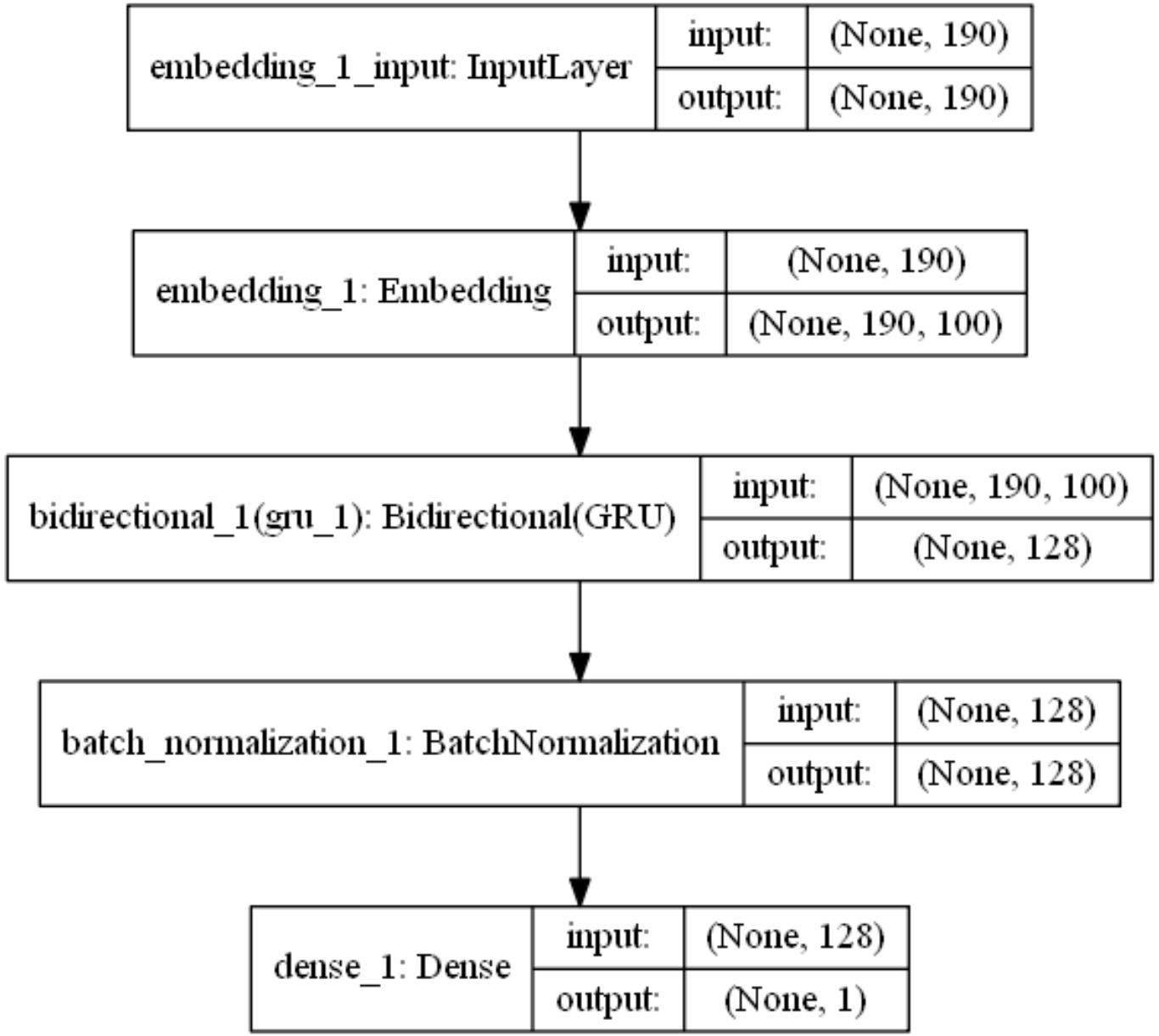
Problem Statement

Devise an end to end pipeline incorporating embeddings to perform error detection and error correction.

Approach


Dataset
Subsample and Label
Labelled subset of the data
Train
Error Detection Model
Error detection on the entire dataset
Erroneous data
Data Imputation
Clean data
Attribute Classifier
Error Correction
Clean data

Step 1



Error Detection

- Randomly sample 30% of the total records for labeling.
- Convert the record to a comma separated string and feed it into the model.

Data Imputation using Word2vec

Etta	Casey	11/17/1987	19	Austin	TX	11285	ec@mail.com

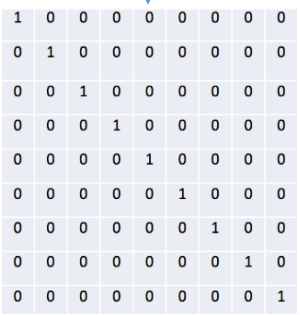
"Etta,Casey,11/17/1987,19,Austin,TX,11825,ec@mail.com"

Word2vec model

[1.23,-3.20,2.01,0.02,-1.34,……]

Clean Dataset

Word2vec model

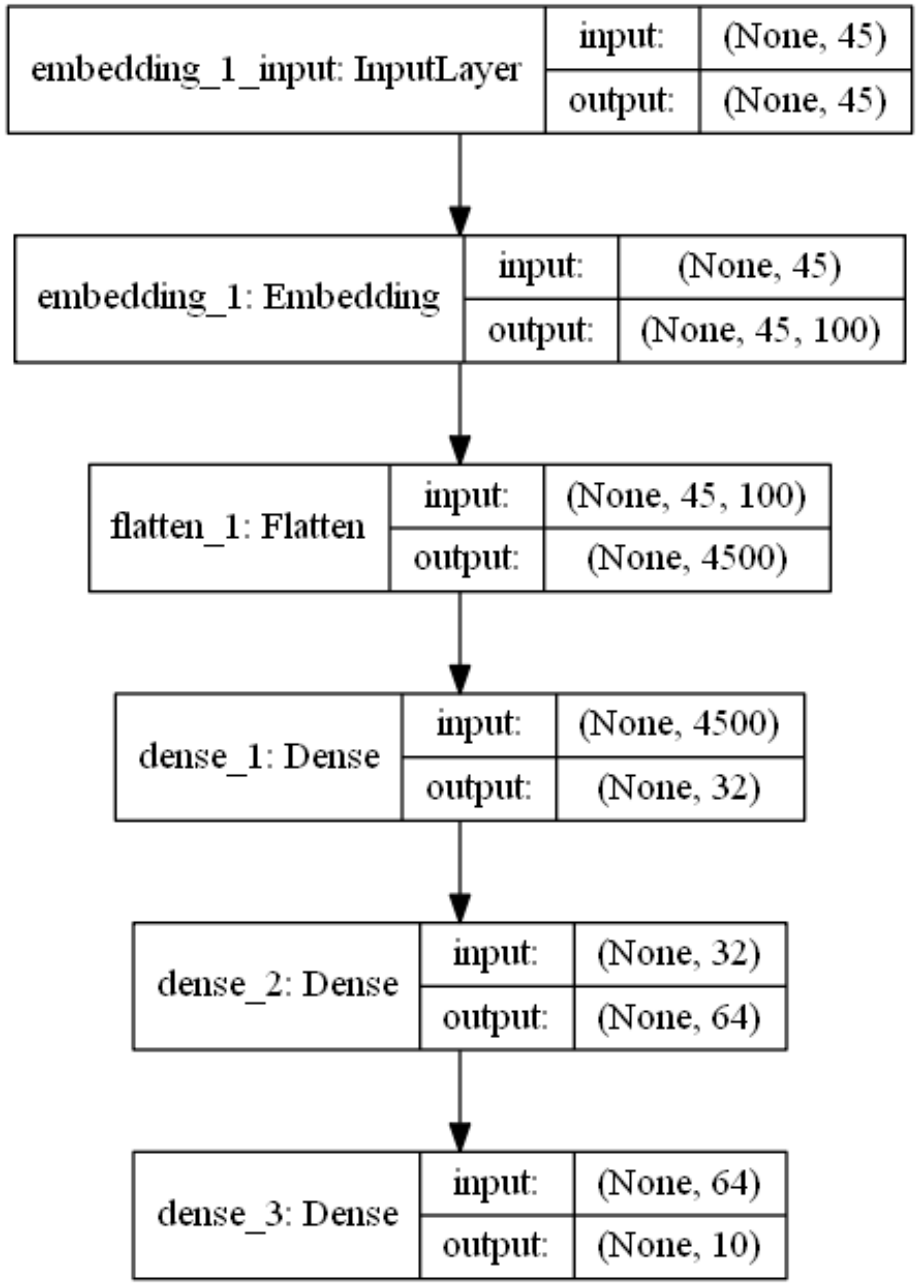


10022	CHEROKEE MEDICAL CENTER	400 NORTHWOOD DR	CENTRE	35960	CHEROKEE	2569275531	Voluntary non-profit - Private	Pneumonia	20 patients
10022	CHEROKEE MEDICAL CENTER	400 NORTHWOOD DR	CENTRE	35960	CHEROKEE	2569275531	Voluntary non-profit - Private	Pneumonia	31 patients
10022	CHEROKEE MEDICAL CENTER	400 NORTHWOOD DR	CENTRE	35960	CHEROKEE	2569275531	Voluntary non-profit - Private	Pneumonia	33 patients
10022	CHEROKEE MEDICAL CENTER	400 NORTHWOOD DR	CENTRE	35960	CHEROKEE	2569275531	Voluntary non-profit - Private	Heart Failure	35 patients

```
model.most_similar("CHEROKEE")
```

```
('2569275531', 0.762696385383606),
 ('35960', 0.757010817527771),
 ('CENTRE', 0.756468356656702),
 ('400 NORTHWOOD DR', 0.7488652467727661),
('CHEROKEE MEDICAL CENTER', 0.6274447441101074),
 ('10022', 0.5803501605987549),
 ('20 patients', 0.5598365068435669),
 ('35 patients', 0.4954341650009155),
 ('31 patients', 0.4659847915172577),
 ('CLANTON', 0.46123188734054565)
```

Word2vec captures information across the row!



Attribute Classifier

"Madison" -> AttributeClassifier -> City

"abc@xyz.com" -> AttributeClassifier -> Email

Algorithm for Imputation

1. Feed in each attribute of the row with one or more missing values to the word2vec model.
2. For each match returned, use the attribute classifier to detect only values of the same attribute as the missing cell.
3. Pick the match with the highest confidence, if multiple matches are returned.

Data Cleaning using fastText

Etta	Casey	11/17/1987	19	Austin	TX	11285	ec@mail.com

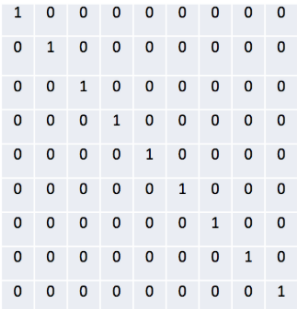
"Etta,Casey,11/17/1987,19,Austin,TX,11825,ec@mail.com"

fastText model

[1.23,-3.20,2.01,0.02,-1.34,……]

Clean Dataset

fastText model



```
model.most_similar("BUNTERVILLE")
```

Erroneous cell value

```
[('GUNTERSVILLE', 0.9831987619400024),
 ('THOMASVILLE', 0.7798212766647339),
 ('HUNTSVILLE', 0.7334578037261963),
 ('RUSSELLVILLE', 0.7136012315750122),
 ('PRATTVILLE', 0.7130258083343506),
 ('HARTSELLE', 0.4626261591911316),
 ('2565718000', 0.3078015148639679),
 ('36116', 0.2716265916824341),
 ('36467', 0.2487492710351944),
 ('BOAZ', 0.24434436857700348)]
```

"GUNTERSVILLE" -> AttributeClassifier ->'CountyName'

FastText top 10 predictions -> Captures syntactically closer words!

Algorithm for data cleaning using fastText

- Detect misspelt cell and column index by checking against ground truth.
- Run fastText algorithm to predict top 10 predictions.
- For each match, use attribute classifier to detect values of the same attribute as the misspelt word.
- Pick the match with highest confidence.

Results

Hospital Dataset

Schema

- ProviderNumber : Integer
- HopsitalName : String
- Address1 : String
- City : String
- ZipCode : String
- CountyName : String
- PhoneNumber : Integer
- HospitalOwner : String
- Condition : String
- Sample : Integer

Type	Count
Positive Samples	827
Negative Samples	825
Total data size	1652

Errors	Count
Empty cell rows	275
Misspelt cell rows	275
Column wise shuffled rows	275

Error Detection

Negative Samples detected- 748/825

	Precision	Recall	F1-Score	Support
Negative samples	0.95	0.86	0.91	825
Positive samples	0.88	0.96	0.92	827
Total	0.91	0.91	0.91	1652

Attribute Classifier

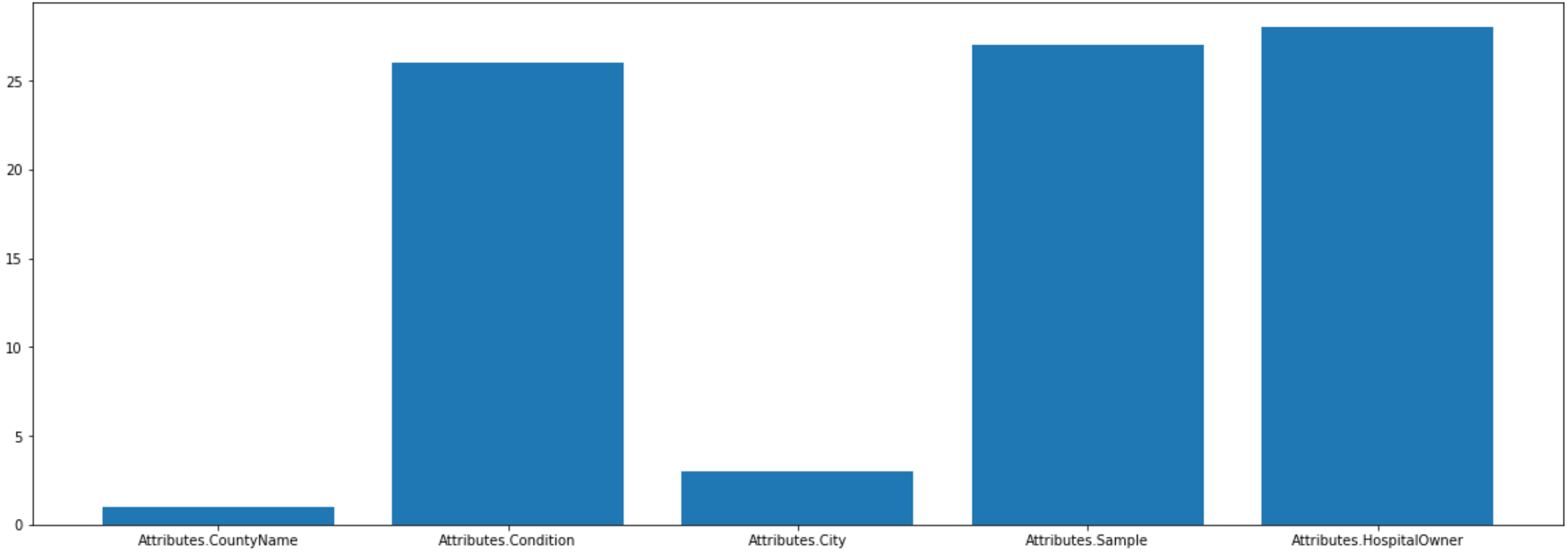
- model.predict("10022") -> 'ProviderNumber'
- model.predict("36278") -> 'ZipCode'

Accuracy	98.68%

Data Imputation

Detected	275/275
Imputed	190/275
Accuracy	69.09%

Error Analysis of Data Imputation



```
df['HospitalOwner'].value_counts()
```

```
Proprietary                                    384
Voluntary non-profit - Private                 372
Government - Hospital District or Authority    360
Voluntary non-profit - Other                   195
Voluntary non-profit - Church                  140
Government - State                              81
Government - Federal                            61
Government - Local                              28
```

```
df['Condition'].value_counts()
```

```
Surgical Infection Prevention    458
Pneumonia                        444
Heart Attack                     406
Heart Failure                    312
Children s Asthma Care             6
```

```
df['Sample'].value_counts()
```

```
0 patients       126
1 patients        31
3 patients        27
2 patients        25
4 patients        25
5 patients        22
10 patients       20
15 patients       19
6 patients        18
14 patients       18
8 patients        16
19 patients       16
16 patients       16
85 patients       15
125 patients      14
101 patients      12
34 patients       12
77 patients       12
25 patients       12
7 patients        12
37 patients       12
69 patients       12
```

Error Cleaning using FastText

Precision	96.89%
Corrected	187/193

Error Analysis of Data Cleaning

- Incorrect Predictions
  - fastTextModel.most_similar("MONTOMARY")

```
[('MONTGOMERY', 0.9795540571212769),
 ('1300 SOUTH MONTGOMERY AVENUE', 0.4560515582561493),
 ('2105 EAST SOUTH BOULEVARD', 0.42764002084732056),
 ('JACKSON HOSPITAL & CLINIC INC', 0.36702218651771545),
 ('36106', 0.31643322110176086),
 ('ONEONTA', 0.2884874641895294),
 ('3342882100', 0.26676562428474426),
 ('36116', 0.26303714513778687),
 ('BUTLER', 0.2594197392463684),
 ('3342938000', 0.25250470638275146)]
```

Error Analysis (contd.)

City	ZipCode	CountyName
MONTGOMERY	36116	MONTGOMERY

- `predictAttribute("MONTGOMERY") ->'City’`
- Ambiguous attribute values made it hard to pick the right match.

Future Work

- Improvise error detector module to categorize different kinds of errors.
- Data cleaning module to handle errors due to attribute shuffling.
- Overall improvement of the workflow to handle less correlated attributes or complex relationships.