Automated ICD-9-CM medical coding of diabetic patient’s clinical reports

Vitor Pereira
DETI/IEETA, Universidade de Aveiro
Aveiro, Portugal

Sérgio Matos
DETI/IEETA, Universidade de Aveiro
Aveiro, Portugal

José Luís Oliveira
DETI/IEETA, Universidade de Aveiro
Aveiro, Portugal

ABSTRACT
The assignment of ICD-9-CM codes to patient’s clinical reports is a costly and laborious process, manually done by medical personnel, estimated to cost about $25 billion per year in the United States. To develop a system that automates this process has been an ambition of researchers but is still an unsolved problem due to the inherent difficulties in processing unstructured clinical text.

This problem is here formulated as a multi-label supervised learning one where the independent variable is the report’s text and the dependent the several assigned ICD-9-CM labels. Different variations of two neural network based models, the Bag-of-Tricks and the Convolutional Neural Network (CNN) are investigated. The models are trained on the diabetic patient subset of the freely available MIMIC-III dataset.

The results show that a CNN with three parallel convolutional layers achieves F1 scores of 44.51% for five digit codes and 51.73% for three digit, rolled up, codes. Although fully automated coding is not achievable, these results suggest that automated classification could be used to aid clinical staff by selecting the most probable codes.

CCS CONCEPTS
• Social and professional topics → Medical records; • Computing methodologies → Machine learning; Supervised learning by classification;

KEYWORDS
Electronic Health Records, Classification, Supervised Learning

ACM Reference Format:
Vitor Pereira, Sérgio Matos, and José Luís Oliveira. 2018. Automated ICD-9-CM medical coding of diabetic patient’s clinical reports. In International Conference on Data Science, E-learning and Information Systems 2018 (DATA ’18), October 1–2, 2018, Madrid, Spain. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3279996.3280019

1 INTRODUCTION
It is in Electronic Health Records (EHR), written by health professionals, where most of patient data resides. These records capture the state of a patient across time and are invaluable in diagnosis. Besides this, they play a significant part in the reimbursement process of health institutions by insurance companies. Medical codes, like ICD-9-CM, assigned to patient’s reports after treatment serve as a justification for carrying out the said treatment. Failure in correctly assigning these codes is a liability for healthcare institutions: a missing code represents a loss of revenue and an extra one can constitute fraud. Assigning and correcting these codes is done manually by specialized medical personnel, called a medical coders, and is estimated to cost about $25 billion per year in the United States [3].

In this paper a system to automate the assignment of ICD-9-CM codes to clinical reports is described. This is achieved by developing a statistical classification model which approximates a function parameterizing the relation between a report’s textual contents and its assigned codes. The used classifiers are based on artificial neural networks, which have shown to be capable of handling large volumes of relatively messy data [12].

The classifier is developed on the diabetic patient subset of the MIMIC-III dataset. By constraining the system to this subset, its diseases, terminology and patterns, it is made less versatile but the lesser variability sets it up to produce more accurate results.

2 BACKGROUND

2.1 Clinical Text
Text mining clinical text presents some unique challenges when compared to other types of usually mined text such as biomedical literature. Meystr et al. sum up these challenges [9].

First, a single patient’s record has several types of report with no single neatly defined structure. These can be the patient’s description, his and his family’s medical history, remarks made by the physician during an examination, etc. The length of the reports also varies greatly e.g. writing a couple of vital signs during a routine check-up vs. the patient’s full medical history during the first consultation. Related to length is the structure of the reports. EHR software is usually form-based (“symptoms”, “suggested causes”, “suggested treatment”, …) which could result in thematically well defined short sentences for each of the form’s fields. However, the clinician might opt to ignore the form’s structure and use the free form data entry field, if it is available, or, worst case, misuse one of the form entry forms, if it is not, to introduce the report’s data.

The clinician also appropriates his writing style and language depending on if the text is going to be read by other practitioners. A report, written by a radiologist, which accompanies an x-ray is composed for clear communication. In contrast, notes written by a physician during a consultation are not. These can be ungrammatical, ignoring capitalization and punctuation rules, and composed of short, telegraphic phrases where shorthand abounds - homonyms,
acronyms, abbreviations and local dialectal shorthand phrases. Furthermore, the shorthand can be overloaded i.e. the same shorthand unit takes a different meaning depending on surrounding context. According to the 2015AA version of UMLS, the abbreviation “RA” can stand for 25 meanings as distant as “rheumatoid arthritis”, “radioactive” and “ragweed antigen” [11]. Liu et al. show that 33% of the time acronyms in UMLS are highly ambiguous even in context [8]. Lexical, morphological and spelling errors also proliferate in clinical text. Ruch and Gaudinat report that about up to one in every five sentences in a corpus of discharge summaries, surgical reports and laboratory results has a spelling error [14]. It is not uncommon to find the acronyms themselves misspelled [9]. Text can also be interrupted by values copy-pasted from laboratory results, vital signs or drug dosages which can confuse standard text mining algorithms like sentence segmentation.

2.2 Prior Work

Assigning medical codes to health records is usually formulated as a supervised, multi-label classification task where the features are the report’s text and the outputs the assigned medical codes. Therefore, most techniques, such as the support vector machine (SVM), generally used in classification tasks apply. Increasingly, neural networks are preferred to other machine learning methods. These have proven to be able to generalize to large volumes of relatively messy data, like clinical text [12].

Due to the mentioned heterogeneity in clinical records, researchers usually choose to focus on a restricted subset of data with shared characteristics. Baumel et al. and Ayyar et al. both restrict their classifiers to the Discharge Summary category of MIMIC-III [1, 2]. Baumel et al. use an SVM based one-vs-all model, a continuous bag-of-words model, a convolutional neural network model and a bidirectional gated recurrent unit with attention mechanism. Ayyar et al. use a recurrent neural network with LSTM cells.

3 DATASET

The Medical Information Mart for Intensive Care III (MIMIC-III) is a large, freely available database of anonymized health-related data associated with patients of the intensive care unit (ICU) of the Beth Israel Deaconess Medical Center between 2001 and 2012 [5].

MIMIC-III is composed of several tables which host several types of data from admissions and transfers to patient information and test results. For the purposes of this thesis only two are used: the notes table (NOTEEVENTS) which contains reports input by clinicians and nurses into the EHR at each patient admission and the diagnosis table (DIAGNOSES_ICD) which contains ICD-9-CM codes, associated with each admission, generated for billing purposes at the end of each patient’s hospital stay. Furthermore, diagnosis entries, and their associated notes, are only kept for patients who have been diagnosed with diabetes. This filtering adds up to a total of 406 190 reports which will be further reduced during preprocessing.

Each report in the notes table is assigned to one of fifteen categories which are heterogeneous in their structure and language much like already described in Section 2.1.

The disease labels associated with each report use the ICD-9-CM taxonomy. The International Statistical Classification of Diseases and Related Health Problems (ICD) is a disease classification system maintained by the World Health Organization. These codes are composed of three primary digits and either one or two optional digits which convey extra information. The Clinical Modification (CM) augments ICD-9 with additional morbidity detail and with codes for surgical, diagnostic and therapeutic procedures.

For the classification task, two different representations of the codes are considered. A regular one with codes which range from three to five digits and a rolled up version where all codes are limited to their first three (primary) digits.

The used MIMIC-III subset has 4006 different regular codes and 779 different rolled up codes. Each of the codes is used in labeling at least one report. Procedure codes are not used. The dataset has a very low label cardinality - defined as the average number of codes per report - of 17 for regular and 15 for rolled up labels. Label densities - defined as the number of codes per report divided by the number of unique codes, averaged over the full dataset - are 0.0042 and 0.0198.

4 PREPROCESSING

Preprocessing starts with a simple four step process that transforms each report into a sequence of tokens:

1. Punctuation characters, except for the apostrophe, are converted to whitespace.
2. Digits are replaced by the character ‘d’.
3. All characters are converted to lowercase.
4. The report is split at whitespace characters.

As previously noticed, clinical text is prodigal of misspellings that can interfere with a classifier’s correctness. Baumel et al. suggest a simple vocabulary reduction step as a partial remedy [2]. Tokens that occur less than five times in the whole corpus are eliminated. These eliminated tokens are then mapped to their most similar token of the ones which were kept. The most similar token is defined as the one which minimizes the Levenshtein distance string metric.

The assumption for this step is that tokens which occur infrequently are misspellings of other tokens which occur more frequently. This does not always hold true but good results were observed, with the added advantage that the vocabulary was reduced from 162 784 to 53 229 unique tokens.

Most text classification tasks opt to use word embeddings to represent tokens. Word2vec’s skip-gram neural network model was input the corpus and used to generate three-hundredth dimensions word embeddings. The concrete implementation of word2vec used was the one from the gensim library1. Pre-trained word embeddings were also experimented with but produced worse results due, most likely, to not containing enough domain specific terminology.

Based on an analysis of the data, it was chosen to keep only reports which have more than 9 and less than 2200 tokens. By doing this, 98.38% of the reports are kept unaltered and can fit in working memory while 6 567 are eliminated. The average length for the reports after this step is 309 tokens.

Table 1 summarizes the preprocessed dataset.

---

1https://radimrehurek.com/gensim/models/word2vec.html
Table 1: Preprocessed dataset description.

| Category                        | Value         |
|---------------------------------|---------------|
| Num. of used records            | 399,623       |
| Num. of regular labels          | 4,006         |
| Num. of rolled up labels        | 779           |
| Regular label cardinality       | 16.99         |
| Rolled up label cardinality     | 15.39         |
| Regular label density           | 0.0042        |
| Rolled up label density         | 0.0198        |
| Num. of unique tokens           | 53,229        |
| Avg. num. of tokens per report  | 309.06        |

5 METHODS
To better understand how this problem can be solved and determine the most advantageous form of classification, two different methods are experimented with:

- Multi-label classification of the reports. Here, the classifier must learn to predict all ICD-9-CM labels assigned to each report. Separate classifiers are trained to predict labels in their regular and rolled up forms. For both forms, two variations of these models are considered. One where the model uses just the tokenized reports and another where, besides using the reports, it also takes as input the category of the report.
- Binary classification of reports for each of the most common rolled up labels (see Table 2). Here, each classifier is trained to predict a single label, contrasting with the previous task where a classifier predicts multiple labels. Both baseline and category augmented classifiers are also used.

Reports are encoded as a sequence of indexes which map to tokens in the known vocabulary; categories are encoded as a 1-hot vector where the category assigned to the report is positive; labels are encoded as a k-hot vector, where all labels assigned to the report are positive.

The models are based on two different types of classifier: the Bag of Tricks and the Convolutional Neural Network.

5.1 Bag of Tricks
The first method is a simple linear model based on Armand et al.’s Bag of Tricks (BoT) [6]. BoT uses a very similar architecture to Mikolov et al.’s Continuous Bag-of-Words [10] but is not limited to words and can classify arbitrary type labels.

The baseline version of this classifier is composed of an embedding layer that transforms the report’s tokens into high dimension word embeddings. These embeddings are then averaged into a fixed dimension vector before being fed into a dense/fully-connected layer. The size of the dense layer varies according to the task: 4,006 for regular multi-label classification, 779 for rolled up label classification and 1 for binary classification.

The significant departure from Armand et al. is in the usage of the sigmoid activation function at the dense layer’s outputs instead of the softmax function. Softmax squashes the layer’s inputs into a probability distribution which would not be appropriate in a multi-label scenario as labels are not mutually exclusive e.g. both “diabetes” and “heart arrhythmia” can be assigned to a single report. The sigmoid function, instead, squashes each neuron’s input into a value in the range $[0, 1]$ so that each neuron outputs a Bernoulli probability distribution independent of all other neurons. To determine if the neuron’s label is assigned, a fixed threshold of 0.5 is used.

The classifier is trained by minimizing the binary/binomial cross-entropy loss function as is usually the case in multi-label and binary classification scenarios.

The Adam optimizer [7] is used and examples are fed into the classifier in batches of size 32. The classifier uses no other regularization except for early stopping with a minimum delta of 0.0001 and a patience of 2 epochs. Choosing the Adam optimizer, instead of the more traditional Stochastic Gradient Descent, enables the adaptive tuning of the classifiers hyperparameters during training [4].

The main advantage of this method, when compared to the CNN, is that, due to its simplicity, it experiences very fast training. However, its disadvantage is significant: linear classifiers do not share parameters among features and labels. This is particularly consequential in textual data as it means word order is ignored and multi-word expressions are not established.

5.2 Convolutional Neural Network
The Convolutional Neural Network (CNN), by using parameter sharing, is able to address BoT’s main problem. This comes at the cost of significantly slower training.

This classifier’s baseline architecture is very similar to BoT as can be seen in Figure 1a. Instead of being averaged, the word embeddings are input into a typical CNN three-stage computation. First, a one-dimensional convolution layer with filter size 250 and kernel size 3 outputs a set of linear activations. Next, each linear activation is run through a (non-linear) rectified linear unit. A final max pooling layer helps to make the obtained representation invariant to small translations in the input [4].

Another architecture with three parallel convolutional layers, as shown in Figure 1b, is also experimented with. Its filters sizes, from left to right, are 2, 3 and 4. The “concat” operation merely concatenates its input vectors. All other hyperparameters are shared with the BoT classifier.

Figure 2 shows the category augmented architecture which is used with both BoT and CNN models. An additional input is added for the category 1-hot vectors which is then concatenated with the classifier specific layer’s output (see yellow color coded layers in Figures 1a and 1b). Four variations of the category augmented architecture were tested, the shallower uses no additional dense layers and the deeper uses three additional ones. All the added dense layers, colored in blue, have output size 64 and a rectified linear unit activation.

6 RESULTS
6.1 Multi-Label Classification Task
The displayed results were obtained using 5-fold cross validation. In each fold, 10% of the training subsamples were used for validation.

Table 3 shows the results for the multi-labeling sub-task for both regular and rolled ICD-9-CM codes. The CNN with three
Table 2: The eleven most common rolled up ICD-9-CM codes.

| ICD-9-CM Code | Description                                             | % Reports |
|---------------|---------------------------------------------------------|-----------|
| 250           | Diabetes mellitus                                       | 100.00%   |
| 427           | Cardiac dysrhythmias                                    | 46.64%    |
| 518           | Other diseases of lung                                  | 45.53%    |
| 428           | Heart failure                                           | 45.19%    |
| 584           | Acute kidney failure                                    | 44.27%    |
| 401           | Essential hypertension                                  | 43.47%    |
| 276           | Disorders of fluid electrolyte and acid-base balance    | 38.74%    |
| 414           | Other forms of chronic ischemic heart disease           | 37.40%    |
| 285           | Other and unspecified anemias                           | 35.10%    |
| 585           | Disorders of lipoid metabolism                          | 34.52%    |

Figure 1: Convolutional Neural Network models.

Table 3: Results for the multi-label classification sub-task.

| Model                      | Precision | Recall | \(F_1\) | Precision | Recall | \(F_1\) |
|----------------------------|-----------|--------|---------|-----------|--------|---------|
| BoT Baseline               | 66.25     | 8.61   | 15.24   | 75.91     | 18.89  | 30.25   |
| BoT Baseline w/ Cat. 0 Dense | 66.50   | 8.59   | 15.21   | 75.82     | 19.13  | 30.54   |
| BoT Baseline w/ Cat. 1 Dense | 67.99   | 11.17  | 19.18   | 74.72     | 22.24  | 34.28   |
| BoT Baseline w/ Cat. 2 Dense | 69.27   | 12.46  | 21.12   | 73.67     | 24.09  | 36.31   |
| BoT Baseline w/ Cat. 3 Dense | 69.03   | 13.08  | 21.99   | 75.02     | 24.21  | 36.59   |
| CNN Baseline               | 73.97     | 25.88  | 38.13   | 77.73     | 35.13  | 48.38   |
| CNN Baseline w/ Cat. 0 Dense | 74.59   | 25.53  | 38.03   | 77.60     | 35.34  | 48.56   |
| CNN Baseline w/ Cat. 1 Dense | 71.96   | 17.66  | 28.35   | 76.06     | 26.50  | 39.28   |
| CNN Baseline w/ Cat. 2 Dense | 73.98   | 18.10  | 29.08   | 75.60     | 29.19  | 42.11   |
| CNN Baseline w/ Cat. 3 Dense | 73.07   | 19.62  | 30.93   | 76.34     | 30.02  | 43.07   |
| CNN 3-Conv1D               | **76.07** | **31.46** | **44.51** | **79.82** | **38.26** | **51.73** |
parallel convolutional layers achieved best precision, recall and, consequently, F₁ score in both code formats.

Rolled up F₁ scores are an average of 12% higher than regular ones. This is to be expected as, by truncating codes to three digits, label sparsity is reduced and each code has more examples. All CNN based models are better than BoT based ones. The worst performing CNN, CNN Baseline w/ Categories 2 Dense, achieved a 7% higher F₁ score, in the regular codes, when compared to the best performing BoT, BoT Baseline w/ Categories 3 Dense. For rolled up codes, the difference is less significant at a 3% improvement.

Category augmented architectures managed to improve the baseline BoT model’s F₁ score by a maximum of 6%, mostly through increased recall. It can be observed that deeper models marginally improved results. This once again confirms the empirical observation that deeper models result in better generalization [4]. Interestingly, not only did category augmentation fail to improve the baseline CNN model, it decreased its F₁ score by a maximum of 9%.

Another observation is that category augmented models with no additional dense layers produced no change when compared to the baseline ones. This can be justified as the model having the category information but not being able to learn how to use it.

Even though CNN based models performed the best, it came at the cost of a remarkable increase in training time, when compared to BoT. The baseline CNN took roughly 5x more time training than the BoT baseline. The CNN with triple convolution was more than twice slower than the baseline CNN at training.

As far as the author is aware, no other publication exists where the classification of the MIMIC-III diabetic subset (or any other subset of similar magnitude) is attempted. Other authors opt to limit themselves to a single category of report. Baumel et al. use 52 139 reports belonging to the Discharge Summary category with 6 527 regular and 1 047 rolled up unique labels [2]. Their CNN’s F₁ scores are similar to the here obtained, which are only 2.59% and 4.26% lower for regular and rolled up labels. Curiously, Baumel et al.’s BoT model (CBOW in the paper) performed much better with 14.78% and 13.05% higher F₁ scores.

6.2 Binary Classification Task
This task restricts classification to the ten most common labels (250 - diabetes mellitus - is ignored). A binary classifier is trained for each different label and all of them are then combined, using the binary relevance method, to make multi-label predictions [13]. This method simply consists of combining the ten distinct binary models into a single multi-label classifier. To establish a comparison, the already trained multi-label classifiers are used as is except all labels which are not in the most common are ignored.

Table 4 shows the results for this task. Multi-label and binary BoT based models performed similarly with the category augmented binary version achieving a 3% higher F₁ score due to a 4% increase in recall. CNN based models once more were the most accurate. Binary CNN models performed better than multi-label ones with F₁ scores 5.73% higher for the baseline model and 6.94% for the category augmented version. Noticeably, the category augmented
CNN binary model shows an almost 9% increase in precision when compared to the multi-label one.

A final observation was made when developing this experiment. It can be seen in the multi-label CNN results of Table 4 that the category augmented version performed marginally better than the baseline one with a 0.45% increase in $F_1$ score. What is surprising about this is that it contradicts what is observed in Table 3 where augmenting the CNN baseline with category information produces much worse results. This leads to the conclusion that training multi-label category augmented CNNs on less sparse labels (see Table 2) could be fruitful.

7 CONCLUSION

The problem of automatically assigning ICD-9-CM codes to patient reports was formulated as a multi-label supervised learning task to be run on the diabetic patient subset of the MIMIC-III dataset using artificial neural network based models. $F_1$ scores of 44.51% and 51.73% were obtained for regular and rolled up labels, respectively. As far as the author knows, no other publications address a similar overarching problem but when compared to other, more localized, experiments, the results appear to be good.

Furthermore, it was explored whether using a collection of binary models, instead of a single multi-label one was a better alternative. This experimentation was conducted on the restricted subset of the ten most common labels. It was shown that the best binary classifiers produced an almost 7% improvement in $F_1$ score over its multi-label equivalent.

The results obtained indicate that such a method could be used to aid clinical coders. Also, various research direction can be taken to improve these results. First, the development of new classifiers. Despite the triple convolution classifier being the one with the best results, a category augmented version of it was not trained due to incredibly long training times. Doing so is certainly worthy. Given the improvements deeper architectures showed in the BoT classifier, deeper yet architectures should be experimented with. Besides these, whole new models, like an LSTM based neural network should be tried. All classifiers use early stopping as the single regularisation technique. Trying other methods such as the L1 Loss or more sophisticated ones, like Dropout, could improve results at the cost of even longer training times.

Second, the leveraging of more features of the MIMIC-III dataset. While the reports are used, the model does not have any information regarding their temporal relation. Adding this information can enable it to develop associations between diseases. For example, if a report is classified with diabetes, the model can discern that following reports have a higher probability of being classified with diseases which are complications of diabetes like kidney and foot damage. It was also mentioned that the codes associated with each report have a sequence number assigned. Finding a way to factor it into the models and the evaluation method could be useful in better appropriating them to the task at hand.

ACKNOWLEDGMENTS

This work was supported by the Integrated Programme of SR&TD “SOCA” (Ref. CENTRO-01-0145-FEDER-000010), co-funded by Centro 2020 program, Portugal 2020, European Union, through the European Regional Development Fund, and by the NETDIAMOND project (Ref. POCI-01-0145-FEDER-016385), co-funded by Lisbon 2020 Regional Operational Programme, through European Structural and Investment Funds (ESIF), and by National Funds through the FCT - Foundation for Science and Technology, Portugal.

REFERENCES

[1] Sandeep Ayyar. 2017. Tagging Patient Notes With ICD-9 Codes. (2017).
[2] Tal Baumel, Jamuna Nassour-Kassis, Michael Elhadad, and Noemie Elhadad. 2017. Multi-Label Classification of Patient Notes a Case Study on ICD Code Assignment. arXiv preprint arXiv:1709.09587 (2017).
[3] Richard Farkas and Gyorgy Szavars. 2008. Automatic construction of rule-based ICD-9-CM coding systems. In BMC bioinformatics, Vol. 9. BioMed Central, S10.
[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.
[5] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. Scientific data 3 (2016), 160035.
[6] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016).
[7] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
[8] Hongfang Liu, Yves A Lassier, and Carol Friedman. 2001. A study of abbreviation in the UMLS.. In Proceedings of the AMIA Symposium. American Medical Informatics Association, 393.
[9] Stéphane M Meystre, Guergana K Savova, Karin C Kipper-Schuler, John F Hurdle, etc. 2008. Extracting information from textual documents in the electronic health record: a review of recent research. Yearb Med Inform 35, 8 (2008), 128–144.
[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).
[11] Serguei Pakhomov, Ted Pedersen, and Christopher G Chute. 2005. Abbreviation and acronym disambiguation in clinical discourse. In AMIA Annual Symposium Proceedings. Vol. 2005. American Medical Informatics Association, 389.
[12] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Peter J Liu, Xiaoqing Liu, Mimi Sun, Patrik Sundberg, Hector Yee, et al. 2018. Scalable and accurate deep learning for electronic health records. arXiv preprint arXiv:1801.07860 (2018).
[13] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. Machine learning 85, 3 (2011), 333.
[14] Patrick Ruch and Arnaud Gaudinat. 2006. Comparing corpora and lexical ambiguity. In Proceedings of the workshop on Comparing corpora-Volume 9. Association for Computational Linguistics, 14–19.

Table 4: Multi-label vs. binary classifier results for the most common labels.

| Model                  | Precision | Recall | $F_1$  | Multi-Label | Precision | Recall | $F_1$  |
|------------------------|-----------|--------|--------|-------------|-----------|--------|--------|
| BoT Baseline           | 68.46     | 35.29  | 46.57  | 68.68       | 33.81     | 45.31  |
| BoT Baseline w/ Cats. 3 Dense | 68.47     | 42.85  | 52.71  | 68.68       | 46.94     | 55.77  |
| CNN Baseline           | 71.06     | 50.15  | 58.80  | 69.89       | 59.94     | 64.53  |
| CNN Baseline w/ Cats. 3 Dense | 66.78     | 53.25  | 59.25  | 75.70       | 58.80     | 66.19  |