Supplementary Materials
Attentive Multiview Text Representation for Differential Diagnosis

Hadi Amiri^{a,c}, **Mitra Mohatarami**^{b}, **Isaac S. Kohane**^{c}
^{a}Department of Computer Science, University of Massachusetts, Lowell
^{b}MIT Computer Science and Artificial Intelligence Laboratory
^{c}Department of Biomedical Informatics, Harvard University
Massachusetts, USA
hadi_amiri@uml.edu, mitram@mit.edu, isaac_kohane@harvard.edu

1 Detailed Model Performance

We illustrate the Interpolated Precision Recall and P@K, $\forall$K $\in \{5, 10, 15, 20, 30, 100\}$, performance of all competing models for further analysis. The results in Figures 1 and 2 show that BERT outperforms other baselines at almost all recall levels and across both text and code views. In addition, BERT performs better on text view than code view; this is perhaps due to the richer context information that is available in the text view which allows BERT to incorporate salient features (such as negation, hedges, etc., that are important for differential diagnosis) in its learning process. The results also show that SVM^{rank} can effectively combine predictions of baseline models and it often maintains higher P@K than other models in case of text view, see Figure 1. However, it doesn't perform as well on the code view; this is perhaps due to the lower performance of the best-performing model (BERT) on the code view which, in turn, affects the performance of SVM^{rank}.

Figures 3 and 4 show the performance of our model (AMNM) across different views and fusion functions. All versions of AMNM except AMNM$_{bert\text{-}svms}$ (g^{conv}) lead to significant improvement against the best performing baseline (BERT). Furthermore, the fusion functions g^{dot} outperforms g^{outer} and g^{conv} at most recall and P@K levels. Figure 5 compare the best AMNM model (AMNM$_{bert\text{-}svms}$ (g^{dot})) against PhenoTips.

2 Implementation Details

We plan to make our code publicly available to the research community. In what follows, we describe detail settings of different models that we investigated in this research. All competing models were developed on the same computing infrastructure (GPU server with 8 Titan GPUs and 90GB of memory).

BM25: We employed Lucene toolkit with default parameter settings for our BM25 experiments. We also experimented with other ranking models including Classic TF/IDF Similarity and Divergence From Randomness (DFR) similarity introduced in (Amati et al., 2002) in conjunction with different normalization approaches which take into account (a): uniform distribution of term frequency, (b): term frequency density inversely related to length, (c): term frequency normalization provided by Dirichlet prior, (d): term frequency normalization provided by a Zipfian relation, and (e): no normalization, all implemented as part of Lucene toolkit. BM25 lead to the highest performance on our dataset.

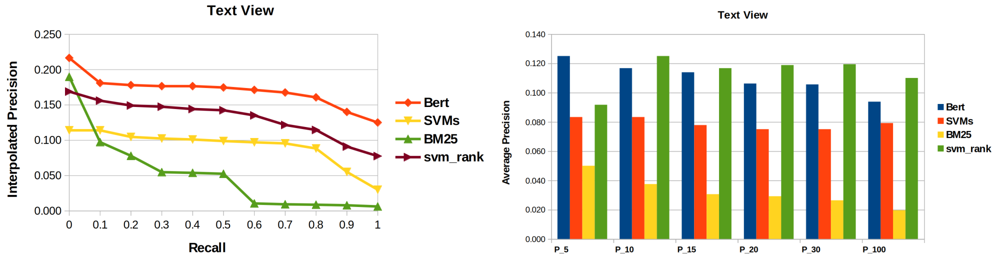
SVMs: We developed this classifier using sklearn toolkit;[1] we applied grid search over the following parameters for tunning the classifier and used Precision as the metric for optimization and re-fitting the model (note that some of these parameters can't be used simultaneously for optimization).

```
tuned_parameters = {’penalty’:  [’l1’,
’l2’], ’C’: [0.0001, 0.001, .01, .1,
1, 2], ’class_weight’:  [’balanced’],
’dual’:  [False, True], ’loss’:
[’hinge’,’squared_hinge’]}
```

As for TF/IDF weighting, we set the max number of features to 20K, max document frequency to 50% (ignoring terms that occur in more than 50% of inputs) and ngrams to $n = [1\text{–}2]$ for both medical histories and disease descriptions.

BERT: We used default parameter settings of BERT except for the number of epochs which was set to a maximum value of 32, and batch size which was set to 32; BERT obtained its best model/performance on validation data at iteration

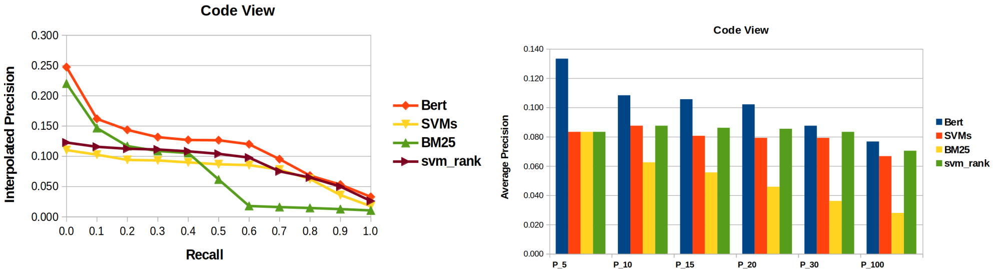
[1] https://scikit-learn.org



(a) Interpolated Precision-Recall of baselines (text view)

(b) P@K of baselines (text view)

Figure 1: Baseline Interpolated Precision-Recall and P@K Performance on Text view



(a) Interpolated Precision-Recall of baselines (code view)

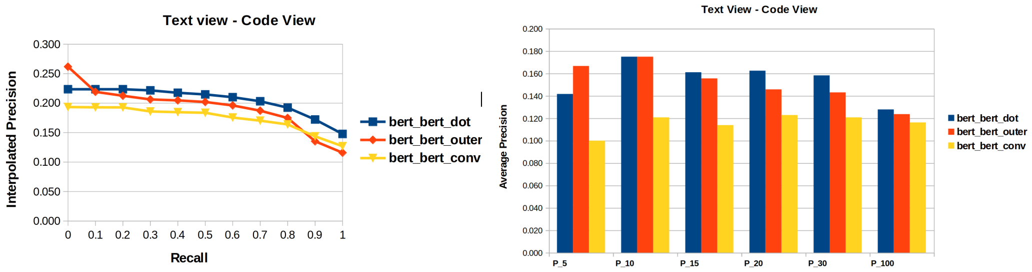
(b) P@K of baselines (code view)

Figure 2: Baseline Interpolated Precision-Recall and P@K Performance on Code view

10 for text view and iteration 7 for code view, which we used for testing the model. We use BERT models developed for clinical text (Alsentzer et al., 2019).

SVM^{rank}: All parameters were set to their default values except for the parameter that controls the trade-off between training error and margin (c) which was chosen from range $[0.01–0.96]$ with step size of 0.05. The model obtained it's best performance with $c = .86$ on both text and code views.

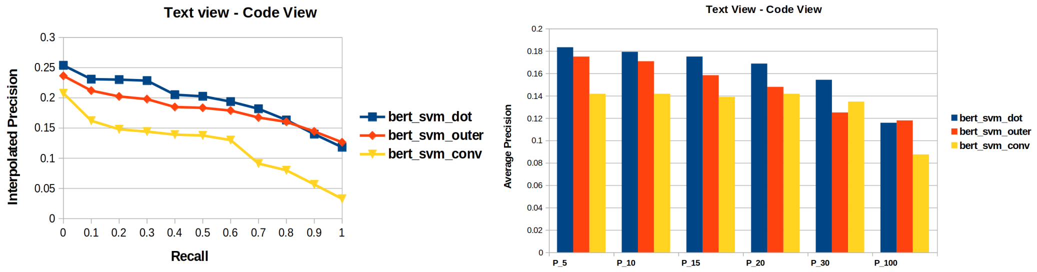
AMNM: Similar to BERT, we set the number of epochs to a maximum value of 32 and batch size to 32; Our model obtained its best performance on validation data at iteration 15 for text view and iteration 10 for code view, which we used for testing the model. We used Adam as optimizer with a smaller learning rate than that of BERT; we set it to $1e$-5. In terms of the fusion network, our CNN employed average pooling with 250 filters and kernel size of 3. We note that Max pooling led to slightly lower performance than average pooling for both AMNM$_{bert\text{-}bert}$ and AMNM$_{bert\text{-}svm}$ (g^{dot}).



(a) Interpolated Precision-Recall of AMNM$_{bert\text{-}bert}$

(b) P@K of AMNM$_{bert\text{-}bert}$

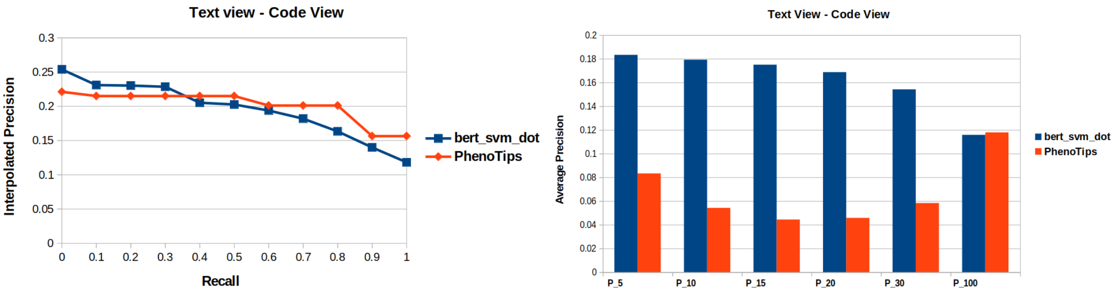
Figure 3: Interpolated Precision-Recall and P@K Performance of our Attentive Multiview Neural Model AMNM$_{bert\text{-}bert}$ across different fusion strategies.



(a) Interpolated Precision-Recall of AMNM$_{bert\text{-}svms}$

(b) P@K of AMNM$_{bert\text{-}svms}$

Figure 4: Interpolated Precision-Recall and P@K Performance of our Attentive Multiview Neural Model AMNM$_{bert\text{-}svms}$ across different fusion strategies.



(a) Interpolated Precision-Recall of AMNM$_{bert\text{-}svms}$ (g^{dot}) vs PhenoTips

(b) P@K of AMNM$_{bert\text{-}svms}$ (g^{dot}) vs PhenoTips

Figure 5: Interpolated Precision-Recall and P@K Performance of our Attentive Multiview Neural Model AMNM$_{bert\text{-}svms}$ with g^{dot} (dot product between attentive embeddings of each input view) versus vs PhenoTips.