Learning Effective Embeddings from Medical Notes

Sebastien Dubois
Center for Biomedical Informatics Research
Stanford University
Stanford, CA
sdubois@stanford.edu

Nathanael Romano
Center for Biomedical Informatics Research
Stanford University
Stanford, CA
naromano@stanford.edu

Abstract

With the large amount of available data and the variety of features they offer, electronic health records (EHR) have gotten a lot of interest over recent years, and start to be widely used by the machine learning and bioinformatics communities. While typical numerical fields such as demographics, vitals, lab measurements, diagnoses and procedures, are natural to use in machine learning models, there is no consensus yet on how to use the free-text clinical notes. We show how embeddings can be learned from patients' history of notes, at the word, note and patient level, using simple neural and sequence models. We show on various relevant evaluation tasks that these embeddings are easily transferable to smaller problems, where they enable accurate predictions using only clinical notes.

1 Introduction

With the rise of electronic health records (EHR), applying machine learning methods to patient data has raised a lot of attention over the past years, for problems such as survival analysis, causal effect inference, mortality prediction, etc. However, whereas the EHR data warehouses are usually enormous, a common setback often met by bioinformatics researchers is that populations quickly become scarce when restricted to cohorts of interest, rendering almost impossible using the raw high-dimensional data. To that extent, several attempts have been made in learning general-purpose representations from patient records, to make downstream predictions easier and more accurate.

These records usually contain a large set of numerical features such as patient demographics (e.g. age, gender, ethnicity), lab measurements (e.g. blood gases, fluid balances, vital signs), binary indicators for diseases and medical procedures, as well as free-text clinical notes. While numerical fields can be straightforward to use in most machine learning models, there is no common ground on how to process clinical notes from physicians and nurses. Researchers often use general-purpose off-the-shelf techniques, such as fitting topic models or learning direct word representations. These techniques, besides being somewhat impractical from a computational standpoint, are very generic and do not leverage the particular structure of these notes. They are, for each patients, a sequence of un-ordered collections of words[1].

In this paper, we use a combination of natural language processing and deep learning techniques to develop models that can learn embeddings of clinical terms and notes. We propose multiple models, and compare them on a wide range of evaluation tasks. Because of the particular structure described above, our model learns embeddings on three scales: word-level embeddings are pooled to create note-level embeddings, which are aggregated to create patient-level embeddings, representing a patient's medical history through his medical notes. We compare (1) models where embeddings are first learned on the word level using count-based and neural models, and then aggregated across notes, and (2) models where all three embeddings are learned jointly in a recurrent neural network.

As we believe work on representation learning and embedding extraction is only as valuable as its evaluations, we also focused our efforts on designing relevant and thorough quantitative evaluations of our embeddings. Because of the three-scale structure of these embeddings, it is natural to evaluate them on three levels.
Word embeddings are evaluated through the *Medical Relatedness Property* developed in [1] and detailed in 5.1. This is an intrinsic evaluation similar to the analogy task and aims at assessing how well the embeddings capture semantic relations. Note-level representations are evaluated through the prediction of 15 phenotypes such as *Asthma* and *Obesity*, using a publicly available dataset, as detailed in 5.2. Patient-level representations are evaluated through multiple classification tasks of different

[1]The words are always un-ordered in the notes as a consequence of the anonymization and annotation pipeline in the annotator used in our dataset, which is quite widely spread.

complexity. These were designed to mimic real-world settings and tackle problems of interest for health-care providers. In particular we consider the problem of *transfer learning* and train our models on a small number of patients (500 or 2,000), since labeling is complex and expensive, and thus bioinformaticians usually have relatively small sample sizes when looking at particular cohorts.

We use these evaluations to compare various approaches at extracting these note embeddings, and also discuss some particular design choices involved in those models.

2 Related work

Several popular papers offer extremely powerful techniques to extract representations from words or documents, such as word2vec [2], GloVe [3], or latent Dirichlet allocation [4]. Those techniques provide good representations for general-purpose tasks, and offer interesting analogy properties, but they do not utilize the full structure of clinical notes. In [5], the authors propose a way to use similar models to learn document-level embeddings, which is similar to our goal, but this work has been found quite difficult to reproduce ([6]).

Learning representations from medical data has been an area of active research recently, either by using free-text notes and converting them to patient-feature matrices [7], or by using a combination of EHR features and free-text notes. [8] for instance uses stacked auto-encoders to learn a general-purpose representation of EHRs, but they interestingly use a latent Dirichlet allocation model to fist process the medical notes. We have tried fitting a model with similar parameters to our medical notes, and it either does not converge or takes weeks to run.
[9] is very relevant to our work since it learns distributed representations of medical codes at different levels (code-level and visit-level). However they do not consider clinical notes and do not consider patient-level representations.

Representing patients' states in a sequence model has been explored as well, using different feature extraction techniques than learning representations from free-text notes. DoctorAI [10] and [11] use a recurrent neural network trained en electronic health records to predict future occurrences of diseases, and [12] uses a recurrent model to extract phenotypes from medical data, regularizing it with prior medical knowledge.

There has been various relevant attempts to learn embeddings from medical words or concepts. While [13], [14] and [15] are direct application of classic word-embeddings algorithms on various medical corpus, [1] learns embeddings using various techniques and compares them with relevant quantitative metrics, some of which we use in this work. However only [15] consider the use of such embeddings in a real-world application. Its approach is still very generic and does not apply to many problems of interest in the medical community such as utilization, cost, etc.

3 Data

3.1 Presentation

Throughout this work, we use the STRIDE dataset, which is an anonymized medical data extract used by the *Stanford Shah Lab* (BMIR[2].). This dataset is extracted from the much larger *Stanford Translational Research Integrated Database Environment* ([16]), a clinical data warehouse.

This extract contains about 27 million notes, corresponding to 1,2 million patients, most of them being between 1998 and 2014. There are about 49 million visits, each them having multiple procedures and diagnoses.

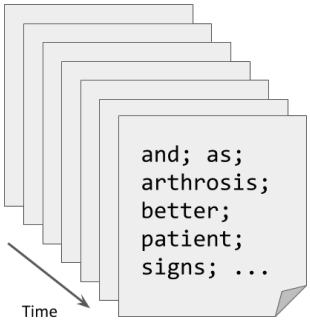

Figure 1: Notes are collections of terms, ordered in a patient's history

[2]Stanford Center for Biomedical Informatics

The clinical notes in this particular extract are processed and annotated using the Open Biomedical Annotator ([7]), which is a pretty standard practice. This process anonymizes the notes (rendering the words un-ordered), and converts each term to concept IDs (CUI), which we will sill refer to as words in the remaining of the paper. The annotated notes have the following structure:

- each patient has a number of clinical notes, written sequentially and timestamped
- each note is a (un-ordered) collection of terms
- terms have been processed, such that it is annotated whether they are negated and/or refer to the patient's family history
- terms can be mapped (in a many-to-one relation) to medical concepts

Each notes can also be associated with a *visit*, during which various diagnoses are made (through ICD9 codes). Those diagnoses can be mapped to more general diseases categories through the *Clinical Classification Software* (CCS)[3]. Thus, for each note, we can obtain a collection of CCS codes diagnosed at the same time as the note.

3.2 Setup and pre-processing

Vocabulary

While it is common practice to discard all negated words (such as in [8]), we decided to include them as extra words in the vocabulary, as we believe negations have a particular meaning in clinical notes. This means that each word in our vocabulary also has its negated counterpart.

This gave us about 265k words, with an extremely long tail, so we restricted our vocabulary to keep relevant words only. We removed words appearing in fewer than 50 notes, and those appearing in more than 10 million notes. This led us to a vocabulary size of 107,388.

Patient eras

As most of our extrinsic evaluations consist of prediction on patients, we split each patient's records in two eras: the input era, and the target era. Records in the input era will be used to train our embeddings as well as inputs in evaluation tasks, whereas records in the target era will be used as prediction targets in evaluation tasks.

For each patient, we isolated a 6-months span of interest for the target era, and we used the previous year as input era. For instance, a year of medical notes and their embeddings could be used to predict the development of various diseases for a patient in the 6 months following this year of data.

Data split

Our dataset consists of 69,416 patients for training, 11,289 for validation and 34,524 for testing. These patients were filtered from the STRIDE database to satisfy practical constraints such as having at least a couple visits during the input / target eras and having at least one clinical note containing a word form our restricted vocabulary.

4 Methods

We explore three types of methods, which we explain here. They are, respectively, learning word-level embeddings and aggregating them to a fixed-length representation, fitting a sequence model to patients' notes to learn all embeddings jointly, and learning a cross-channel encoder from medical notes to diagnoses.

4.1 Embed and aggregate

We first propose a method to learn word embeddings from clinical notes, and then discuss how these can be aggregated to represent a note or a patient state.

4.1.1 Learning word embeddings from clinical notes

It is natural to think about common methods in Natural Language Processing to learn word embeddings, such as `GloVe`[3] and `Word2Vec`[2]. However, as explained above, the Open Biomedical Annotator removes the order of words in the notes as part of the de-indentification process. We decided to tackle this challenge by simply re-generating notes with a random word

[3] https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp

order. Such approach was already proposed (e.g. in [10] and [1]) as an easy trick to deal with the unordered medical codes and did not seem to alter the models' quality. However since clinical notes can be quite long, we suspect that it might be too hard for our models to capture the context of a word. Therefore, we also propose to juxtapose a couple times duplicates of a same note, but with different random orders.

This makes our data look exactly like text data, so that we can learn embeddings with an efficient C implementation of GloVe[4].

4.1.2 Learning word embeddings from external data (MCEMJ)

In [14] the authors learn embeddings from 350,000 medical journal abstracts in the OHSUMED dataset. They first map the free text to UMLS[5] concept ID's and then learn the word embeddings using Word2Vecs skip-gram model with hierarchical softmax. Their best results were achieved with embeddings of dimension 200 and a window size of 5. 52,100 such embeddings were shared publicly by the authors [6] and approximately 28,000 of them belonged to our vocabulary. We will refer to them as MCEMJ (Medical Concept Embeddings from Medical Journal).

4.1.3 Fixed-length aggregation at the note and patient level

Although it seems to be a simple approach, it is still very common in medical applications to aggregate a patient's data over time by summing or averaging counts of medical codes. Similarly, we propose to aggregate (mean, max, min, or sum) the word embeddings per note and then per patient. We explored these various ways of pooling the embeddings both on the note and the patient level.

In addition, we investigate the utility of concatenating multiple of these patient-level aggregations, especially some derived by different sources of embeddings (e.g. randomly ordered notes and medical journals).

4.2 Recurrent neural network

Our methods so far do not rely on the sequential nature of the notes in a patient's history. For a number of prediction tasks, it is important to capture the progressive evolution of a patient's state in its representation. Thus, it is a natural idea to use recurrent neural networks to learn those patient-level embeddings.

4.2.1 Supervision

We supervise our network with diagnoses associated with the notes (from the corresponding *visits*, as explained in 3.1). Since these descriptors are extremely high-dimensional, we map them to more general disease categories using the Clinical Classification Software (CCS)[7], leaving a label space of dimension 254.

At this point we have a design choice, training the network as a multi-task sequence classification problem (aggregating the CCS labels on a per-patient basis), or as a sequence labeling problem (supervising each timestep with its corresponding CCS codes). Experiments showed that the former gave considerably higher scores on the evaluation task, which is a probable consequence of the labels' sparsity (the latter providing very scarce supervision signal).

4.2.2 Architecture

Pooled embedding layer

Since this un-ordered structure in the input is quite uncommon, we use an unusual layer as input to our neural network. This layer computes trainable embeddings from the collections of words in an input note, and pools them before feeding them to the rest of the network. More precisely, we use max-pooling aggregation, so the input to our network for a note x is actually given by:

$$\phi_L(x) = \max_1(x \odot L)$$

where $x = [0, 0, 1, ..., 0, 1, 0]$ is the note bag-of-words encoding of shape $(V, 1)$, L is the word embedding matrix of shape (V, D_x) (V being the vocabulary size and D_x the word embedding size), and $\max_1$ performing a max over the first dimension.

We also explored with aggregating the word embeddings by taking their average, but this performed extremely poorly on the evaluation tasks. It makes sense to use particularly activated features in the learned word representations rather than their average, as those activated features will capture specific semantic indicators from the note.

[4] `https://github.com/stanfordnlp/GloVe`
[5] Unifieid Medical Language System, the same as the one we use.
[6] `https://github.com/clinicalml/embeddings`
[7] https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp

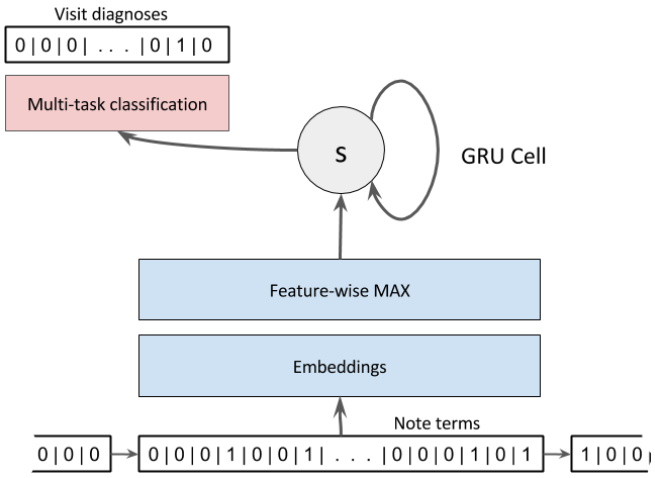

Figure 2: Model architecture

RNN cells

We explored both using Long Short-Term Memory ([17]) and Gated Recurrent Units ([18]) as cell structure for our recurrent network. We found that LSTM learned a poor representation of the input, leaving a number of columns at 0 and performing poorly on evaluation tasks, which confirms the findings of [10]. Thus, most of our final experiments were conducted with GRUs, and this is what we strongly recommend for researchers working with similar problems.

Other details

We used `tanh` activations, dropout (with drop probability 0.1 or 0.2 depending on models), and a dense layer with binary cross-entropies for the multi-task supervision. We trained the networks for 100 epochs, using RMSProp ([19]) for loss optimization. The models were implemented in Keras ([20]) and trained on a NVIDIA GeForce GTX TITAN X GPU.

4.2.3 Embeddings initializations

We explored initializing our embeddings in various ways. In particular, we tried initializing them randomly, with word embeddings pre-trained on our notes, and with word embeddings trained on journal abstract (MCEMJ) (see 4.1.2 for details). While the results for random initialization and MCEMJ initialization are reported below, we found that initializing the weights with embeddings pre-trained on the clinical notes themselves gave very poor results on the evaluation tasks.

4.3 Note-level cross-channel autoencoder

Finally, we experimented with a flat version of the sequential model presented in the previous section, *ie.* a two-hidden-layer neural network where the first layer is a randomly initialized embedding layer with max-pooling. For this model, the inputs are the clinical notes and the non-exclusive targets are CCS labels derived from the corresponding visit data.

This model seems appealing since it leverages our data to provide supervision in order to learn word embeddings. In addition, it is simpler to implement and deploy than a sequential model. However we found that in practice it mostly combined the drawbacks of both previous models: it does not provide a clear patient-level representation, it requires a lot of training data (and labels!), and is also very slow to run.

5 Experiments

A key component in describing novel embeddings is a thorough evaluation of their relative effectiveness. The multi-scale nature of thes embeddings (words, notes, patients) imposes a multi-scale evaluation. As we will see in the results, it is not clear whether or not the results in the intrinsic and note-level evaluation have an impact on the final extrinsic evaluation at the patient level.

5.1 Word-level evaluation

We evaluate our word embeddings through the *Medical Relatedness Property* developed in [1] [8].

The National Drug File Reference Terminology (NDF-RT) provides sets of relations (such as May-Treat and May-Prevent) between drug and diseases. The objective is equivalent to the analogy task used in NLP. Suppose we have two diseases d_1, d_2 and medications m_1, m_2 such that the relation r holds for both pairs (d_1, m_1) and (d_2, m_2). The goal is to check if the following arithmetic holds in the word embeddings space:

$$e_{d_1} - e_{m_1} + e_{m_2} \approx e_{d_2}$$

Precisely, the metric is computed by the ratio of words for which one of the top-40 neighbors of $e_{d_1} - e_{m_1} + e_{m_2}$ satisfies the relation r with m_2.

Embedding method	May-Treat (%)	May-Prevent (%)
GloVe300-W10-R1	7.83	8.51
GloVe-100-W7-R2	6.01	08.09
GloVe-300-W4-R2	8.81	**10.64**
GloVe-300-W7-R2	8.25	10.21
GloVe-500-W7-R2	9.23	10.21
GloVe-300-W10-R2	**10.49**	9.79
GloVe-300-W4-R3	6.57	7.23
MCEMJ	8.25	6.81
Cross-channel	5.45	2.55
MaxGRU200-MCEMJ	7.27	5.53
MaxGRU300	1.26	0.43

Table 1: Evaluation of the Medical Relatedness Property for two relations: May-Treat and May-Prevent. "GloVe-300-W10-R2" corresponds to GloVe embeddings trained on our clinical notes, with length 300, window size 10, and clinical notes repeated twice in the corpus. All GloVe models were run for 25 iterations. "MaxGRU200-MCEMJ" is a RNN with GRU cells, 200 cells, initialized with MCEMJ. This evaluation was computed with the code provided by [1]

Interestingly, we notice that many of our GloVe embeddings perform better than the MCEMJ ones (learned from abstracts of medical journals), despite randomized word orders and a vocabulary twice as big. In addition, embeddings from randomly initialized neural models perform worse, especially the RNN.

5.2 Note-level evaluation

To evaluate our note representations, we used the `i2b2` (Informatics for Integrating Biology to the Bedside) 2008 Obesity Challenge [21]. The publicly available dataset contains approximately 1,230 clinical notes and 15 phenotype targets such as *Asthma* and *Obesity*. The dataset is divided in two: the *textual task* for which labels are attributed from explicit mention in the notes; for the *intuitive task* however, classification was done by using doctors' intuition and judgment.

To align with this competition, we consider the micro-averaged F1[9] as our main metric. For simplicity, we will only report the average performance across the 15 targets. More details about this dataset and our approach is provided in Appendix A. We used a raw bag-of-words encoding of the notes as baseline.

Embedding method	Textual	Intuitive
Baseline	**0,822**	**0,802**
GloVe-300-W7-R2	0,784	0,773
GloVe-500-W7-R2	0,779	0,774
MCEMJ	0,776	0,765
MaxGRU300	**0,793**	**0,778**
MaxGru200-MCEMJ	0,781	0,769

Table 2: Micro-averaged F1 scores averaged over the 15 targets of the i2b2 Obesity Challenge test set. All embeddings where aggregated among the notes by component-wise maximum.

Overall, we did not notice much disparity in the accuracies yielded by the different representations, although the baseline performed a bit better than all our dense features. We also observed that in general, aggregating the word representation in a

[8] We used their code to compute this metrics, which is available at `https://github.com/clinicalml/embeddings`

[9] Which corresponds to summing counts (true positives, ...) across classes, see [21] for details.

note with the `max` operator yielded the best results (compared to `mean, min, sum`). Finally, we notice that the baseline had a larger drop in performance between the *textual* task and the *intuitive* one, compared to the models based on the word embeddings. This could mean that word embeddings become more robust when the target gets more complex.

5.3 Patient-level evaluation

The final evaluation - at the patient level - consists of various real-world applications in which such embeddings could be used, and thus it is our extrinsic evaluation. We look at the performance of various prediction tasks, using only the learned embeddings as input. Moreover, to replicate the common drawback of prediction tasks in healthcare in actual settings, where sample sizes for actual problems of interest are often relatively small, we only use small training sets (respectively 500 and 2,000 patients) to evaluate those prediction tasks (the results are then averaged over 20 trials).

The tasks we use are particularly relevant to bioinformatics scientists and healthcare providers. We use the embeddings to predict various events in the patient's "target era": mortality. future admission, future emergency visits (ER), future development of the 6 densest diseases in our training set (thyroid disorder, cardiac dysrhythmias, diabetes, spondylosis, disorder of lipid metabolism, and essential hypertension).

All of our evaluation models were trained using Ridge logistic regression, cross-validated over 5 folds (using `glmnet`), on a per-training group basis. The results were further averaged over 20 sampled training groups. All GloVe methods reported here use a window size of 7 and 2 resampling of the notes. In addition all aggregations of word embeddings (GloVe and MCEMJ) are concatenations of aggregations by maximum, minimum and average since this always improved the performance.

The baseline is a Ridge logistic regression fit on a raw bag-of-words encoding of the patient history of notes, where the vocabulary is restricted to the most frequent words appearing in the small training set[10]. We only kept the results when the baselines converged, but this does not happen all the time, and would even more rarely with longer eras and more patients.

Results with 500 patients

For each model, we report the resulting average and standard error of Area Under Receiver-Operator Curve (AUROC):

Method	**Death**	**Admission**	**ER Visit**
Baseline	0.801 (0.019)	0.711 (0.019)	0.698 (0.005)
GloVe-300 + MCEMJ	0.811 (0.005)	0.758 (0.002)	0.756 (0.002)
GloVe-500 + MCEMJ	0.809 (0.006)	0.757 (0.003)	0.755 (0.003)
MCEMJ	0.795 (0.004)	0.74 (0.003)	0.729 (0.004)
GloVe-300	0.797 (0.005)	0.752 (0.003)	0.752 (0.002)
GloVe-500	0.805 (0.005)	0.752 (0.004)	0.752 (0.003)
MaxGRU300	**0.842 (0.003)**	0.723 (0.022)	0.745 (0.005)
MaxGRU600	0.829 (0.017)	**0.765 (0.004)**	**0.759 (0.002)**
MaxGRU500-MCEMJ	0.812 (0.017)	0.746 (0.006)	0.747 (0.004)
MaxGRU200-MCEMJ	0.817 (0.017)	0.75 (0.006)	0.754 (0.002)

Table 3: Evaluation on mortality and utilization prediction tasks. Reported metrics are mean and standard error of AUROC and across 20 small evaluations.

Methods	**Thyroid**	**Dysrhythmias**	**Diabetes**	**Spondylosis**	**LMD**[11]	**Hypertension**
Baseline	0.610 (0.009)	0.687 (0.01)	0.68 (0.005)	0.667 (0.017)	0.635 (0.011)	0.664 (0.009)
GloVe-300 + MCEMJ	0.632 (0.009)	0.69 (0.004)	0.689 (0.004)	0.696 (0.004)	0.648 (0.009)	0.679 (0.005)
GloVe-500 + MCEMJ	0.648 (0.006)	0.689 (0.005)	0.691 (0.003)	0.701 (0.004)	0.649 (0.009)	0.667 (0.01)
MCEMJ	0.642 (0.01)	0.688 (0.004)	0.69 (0.004)	0.688 (0.003)	0.651 (0.005)	0.666 (0.01)
Glove-300	0.618 (0.008)	0.683 (0.005)	0.683 (0.004)	0.689 (0.003)	0.649 (0.005)	0.672 (0.005)
Glove-500	0.618 (0.01)	0.686 (0.004)	0.683 (0.004)	0.693 (0.003)	0.642 (0.008)	0.67 (0.004)
MaxGRU300	0.701 (0.011)	**0.709 (0.017)**	0.737 (0.013)	0.732 (0.012)	0.7 (0.015)	**0.745 (0.003)**
MaxGRU600	**0.721 (0.004)**	0.673 (0.023)	**0.744 (0.013)**	**0.738 (0.013)**	0.702 (0.016)	0.735 (0.013)
MaxGRU500-MCEMJ	0.623 (0.013)	0.673 (0.02)	0.718 (0.012)	0.725 (0.004)	**0.706 (0.004)**	0.739 (0.005)
MaxGRU200-MCEMJ	0.662 (0.01)	0.697 (0.016)	0.705 (0.02)	0.728 (0.012)	0.703 (0.011)	0.741 (0.003)

Table 4: Evaluation on prediction of future diseases. Reported metrics are mean and standard error of AUROC and across 20 small evaluations.

[10]This is the only way to have this model converge in a majority of cases.

We can observe most models beat the baseline quite easily (which often did not converge when fitting these evaluation models), except on mortality prediction where it appears to perform particularly well. The sequence models achieve the best results, however it is unclear whether it is due to the training supervision or to the fact it captures the sequential semantic information in the notes.

Results with 2,000 patients

The results with 2,000 patients are reported in the appendix B. The baseline is not included since it could never converge (2,000 patients per training groups means more word features, at a level where `glmnet` could not converge).

6 Conclusion

We showed in this work that is possible to learn embeddings from patients' clinical notes, which can easily be transfered to considerably smaller populations for specific prediction and classification problems, using a particularly relevant and thorough set of evaluation tasks. We showed that such learned representations not only render those smaller problems computationally more tractable, they also have the ability to improve the accuracy of such predictions, by capturing more of the semantic information contained in the notes than baseline models.

Furthermore, we proved that recurrent neural network can learn representations that yield particularly good improvements on these evaluation tasks, by leveraging the sequential arrangements of the notes in a patient's history, provided the correct design choices are made. However, it is interesting to note that while our *"embed and aggregate"* methods perform less well than our sequence models, they are way quicker to train, demand less tweaking, and can be transfered across institutions and sub-problems in a more flexible manner. It is also interesting to note that models that perform the best on our extrinsic tasks (which are realistically the most important) are not necessarily the ones yielding the best word embeddings according to our word-level evaluation.

There are future developments that could be made to this model. First, it would be interesting to develop a new expression of the `GloVe` or `word2vec` objective function that takes into account the specific structure of these notes (sample negated words during negative sampling, sample windows from the set of words at each iteration, etc.). Secondly, it would be worth exploring other options for the RNN supervision. Ideally, the labels would be less sparse and would only use the notes' content; so perfect labels would be some very high-level aggregation of the concepts present in the note.

Acknowledgments

We would like to warmly thank Kenneth Jung for his mentorship on the project, providing constant, relevant and involved guidance, as well as pointers to useful literature and datasets. Weekly discussions with him have profoundly shaped this work. Furthermore, we want to thank Prof. Nigam Shah for the resources offered by the Shah Lab, in terms of data and computing power, as well as Dave Kale (USC) for his advices in the early stages of the project.

Contributions

Sebastien implemented and investigated the *"embed and aggregate"* models as well as the cross-channel encoding. He also implemented and ran the word-level and note-level evaluations. Nathanael implemented and investigated the sequence models. He also did most of the data preprocessing. The patient-level evaluation tasks, this report and the poster were done jointly by Sebastien and Nathanael.

References

[1] Youngduck Choi, Chill Yi-I Chiu, and David Sontag. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41, 2016.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[3] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[5] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[6] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016.

[7] Paea LePendu, Srinivasan V Iyer, Anna Bauer-Mehren, Rave Harpaz, Jonathan M Mortensen, Tanya Podchiyska, Todd A Ferris, and Nigam H Shah. Pharmacovigilance using clinical notes. *Clinical pharmacology & therapeutics*, 93(6):547–555, 2013.

[8] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep Patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6, 2016.

[9] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1495–1504. ACM, 2016.

[10] Edward Choi, Mohammad Taha Bahadori, and Jimeng Sun. DoctorAI: Predicting clinical events via recurrent neural networks. *arXiv preprint arXiv:1511.05942*, 2015.

[11] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzell. Learning to diagnose with LSTM recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[12] Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 507–516. ACM, 2015.

[13] José Antonio Minarro-Giménez, Oscar Marín-Alonso, and Matthias Samwald. Exploring the application of deep learning techniques on medical text corpora. *Studies in health technology and informatics*, 205:584–588, 2013.

[14] Lance De Vine, Guido Zuccon, Bevan Koopman, Laurianne Sitbon, and Peter Bruza. Medical semantic similarity with a neural language model. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1819–1822. ACM, 2014.

[15] Yun Liu, Collin M Stultz, John V Guttag, Kun-Ta Chuang, Fu-Wen Liang, and Huey-Jen Su. Transferring knowledge from text to predict disease onset. *arXiv preprint arXiv:1608.02071*, 2016.

[16] Henry J Lowe, Todd A Ferris, Penni M Hernandez, Susan C Weber, et al. Stride-an integrated standards-based translational research informatics platform. In *AMIA*, 2009.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[19] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

[20] François Chollet. Keras, 2015.

[21] Özlem Uzuner. Recognizing obesity and comorbidities in sparse data. *Journal of the American Medical Informatics Association*, 16(4):561–570, 2009.

Appendices

A Details on note-level evaluation

The publicly available dataset contains approximately 1,230 clinical notes (730 for training and 500 for testing) and 15 targets. The targets are phenotypes such as *Asthma, Obesity*, etc, and they all have 4 possible classes: Present, Absent, Questionable and Unmentioned.

The notes were processed with the Shah Lab annotator so that it had the exact same format as the notes in the STRIDE database. From the original training set, we randomly subsampled a validation set containing 20% of the data. We report only the classification models that yielded the highest accuracies: a Random Forest with 10 trees was used for the baseline representation (bag-of-words) and a k-Nearest Neighbors for all other inputs (k was chosen by cross-validation on the remaining training data).

Embedding method	Aggregation	Textual	Intuitive
Baseline	all	**0,807**	**0,803**
GloVe-300-W7-R2	max	0,784	**0,785**
GloVe-300-W7-R2	min	0,772	0,774
GloVe-300-W7-R2	mean	0,754	0,759
GloVe-300-W7-R2	sum	0,763	0,765
GloVe-500-W7-R2	max	**0,788**	0,776
GloVe-300-W10-R2	max	0,749	0,758
GloVe-300-W10-R1	max	0,77	0,764
GloVe-300-W4-R3	max	0,769	0,783
MCEMJ	max	0,775	0,761
Cross-channel	max	0,775	0,759
MaxGRU300	max	0,780	0,783
MaxGRU200-MCEMJ	max	0,773	0,769

Table 5: Micro-averaged F1 scores on the *validation* set for several word embeddings

B Patient-level evaluation with 2,000 patients

We report, for reference, the results of some of our models when the evaluation task is trained on 2,000 patients instead of 500. As stated earlier, the baseline could not converge in this setting. What's reported is still the average AUROC across 20 training groups, as well as the standard error.

Method	Death	Admission	ER Visit
Baseline	NA	NA	NA
GloVe-300-W7-R2 + MCEMJ	0.864 (0.001)	0.783 (0.002)	**0.78 (0.001)**
MaxGRU300	**0.866 (0.001)**	**0.791 (0.001)**	0.768 (0.002)
MaxGRU600	0.864 (0.001)	0.789 (0.001)	0.775 (0.001)
MaxGRU500-MCEMJ	0.849 (0.001)	0.775 (0.002)	0.766 (0.001)
MaxGRU200-MCEMJ	0.854 (0.001)	0.782 (0.002)	0.767 (0.001)

Table 6: Evaluation on prediction of mortality and hospital utilization.

Method	Thyroid	Dysrhythmias	Diabetes	Spondylosis	LMD	Hypertension
Baseline	NA	NA	NA	NA	NA	NA
GloVe + MCEMJ	0.704 (0.003)	0.728 (0.002)	0.735 (0.002)	0.73 (0.001)	0.706 (0.002)	0.719 (0.002)
MaxGRU300	0.744 (0.002)	0.738 (0.013)	0.77 (0.002)	0.761 (0.001)	0.749 (0.001)	0.768 (0.001)
MaxGRU600	**0.759 (0.002)**	0.738 (0.013)	**0.785 (0.002)**	**0.766 (0.001)**	**0.752 (0.001)**	**0.778 (0.002)**
MaxGRU500-MCEMJ	0.685 (0.003)	0.737 (0.003)	0.757 (0.001)	0.745 (0.001)	0.737 (0.001)	0.768 (0.001)
MaxGRU200-MCEMJ	0.706 (0.002)	**0.741 (0.001)**	0.761 (0.001)	0.755 (0.001)	0.743 (0.002)	0.768 (0.001)

Table 7: Evaluation on prediction of future diseases.