

SDTM Mapping using Machine Learning (Natural Language Processing and Artificial Neural Network)

Sam Tomioka¹, Kevin Lee²

1 Sunovion Pharmaceuticals

2 Clindata Insight



sdtm-mapper

Summary

CDISC SDTM mapping faces challenges because raw data are not well standardized due to lack of CDISC CDASH compliance, lack of company database standards, or study specific requirements.

Combining natural language processing (NLP), natural language model with traditional machine learning (ML) algorithms have enabled possibilities to assist CDISC SDTM mapping.

Recent natural language research has opened up new possibilities for SDTM mapping. Bidirectional LSTM + ELMo + FFNN achieved human level performance.

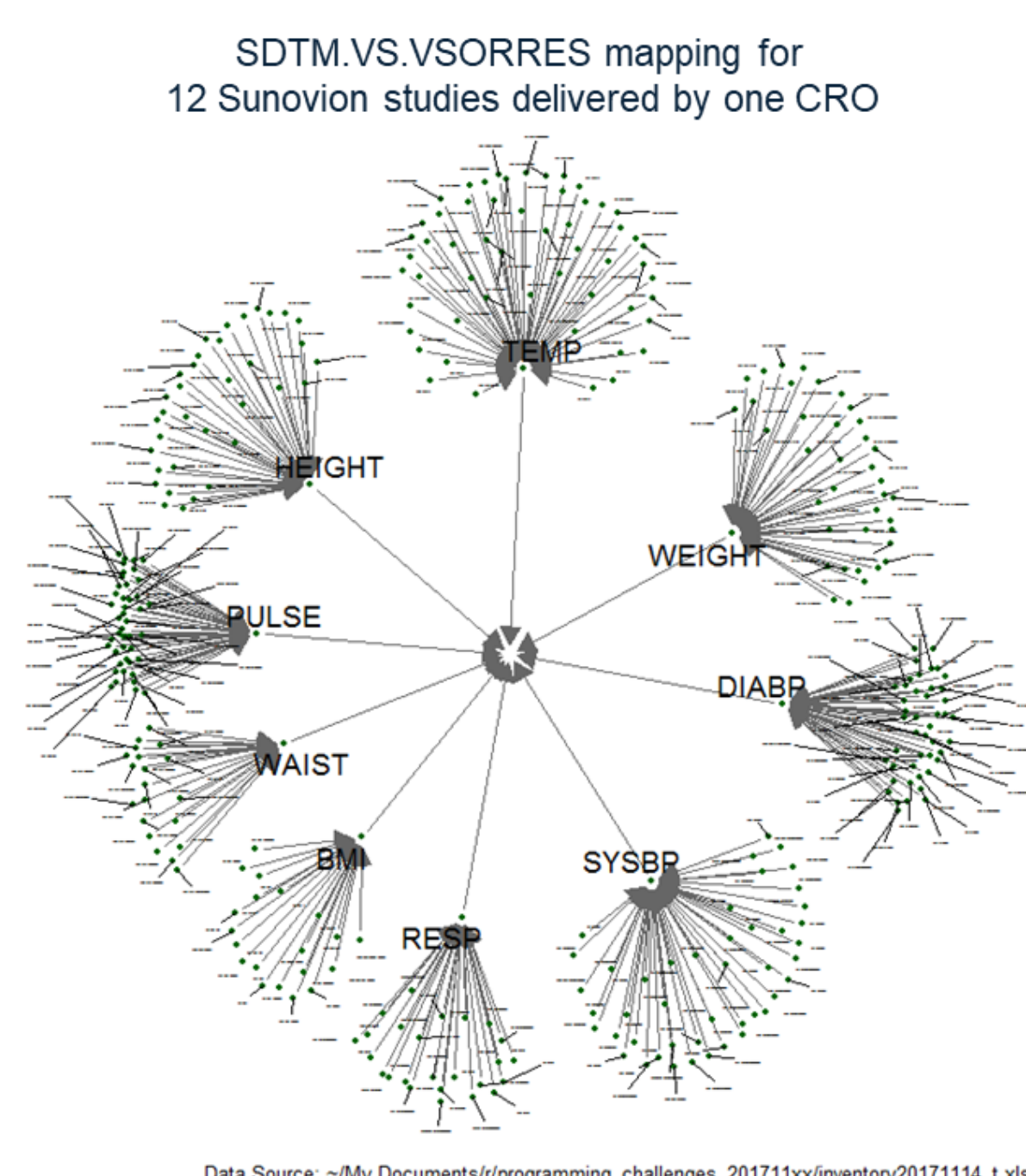
sdtm-mapper is a Python library which is a framework to extract metadata from SAS datasets and train on pre-built neural network models.

PhUSE SDTM-Mapper Project has started

Introduction

CDSIC Study Data Tabulation Model (SDTM) is the standard data model to represent clinical trial data. While the FDA selected SDTM as the standard data model for submission in 2004, no attempt has been made until recently to automate the SDTM creation using machine learning.

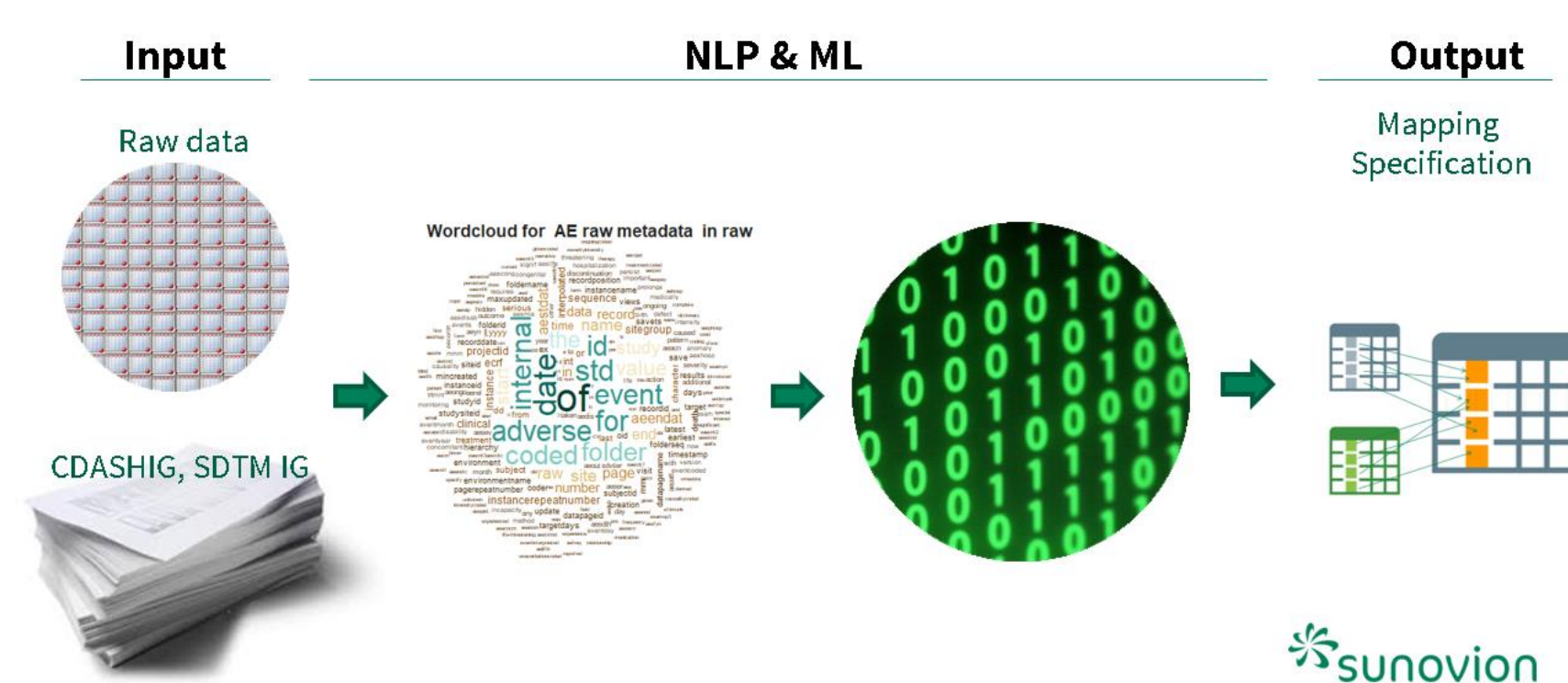
SDTM mapping continues to be challenging because the incoming raw data structures have high variability as represented in the network graph below.



The network graph shows the single SDTM variable VSORRES represented as a central node, and 9 child nodes represent VSTESTCD. The leaf nodes represent the variables in raw datasets. Edges represent the mapping.

Methods

Previous work was done with a combination of traditional natural language representation of the raw data metadata with traditional machine learning algorithms as represented in the figure below.



- Metadata are extracted from raw datasets (typically Proc Contents of SAS7bdat)
- Metadata are pre-processed and transformed into vector representations.

1. TF-IDF :

The weight of each token is computed with TF-IDF which is defined as

$$w_{t,d} = tf_{t,d} \times idf_t \text{ where}$$

$$tf_{t,d} = \begin{cases} 1 + \log_{10} count(t, d) & \text{if } count(t, d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{and } idf_t = \log_{10} \left(\frac{N}{df_t} \right)$$

2. 3-hidden-layer neural probabilistic language model:

The original model as shown below is modified and is trained on a 200 billion Google News corpus. The weighted sum divided by the square root of the

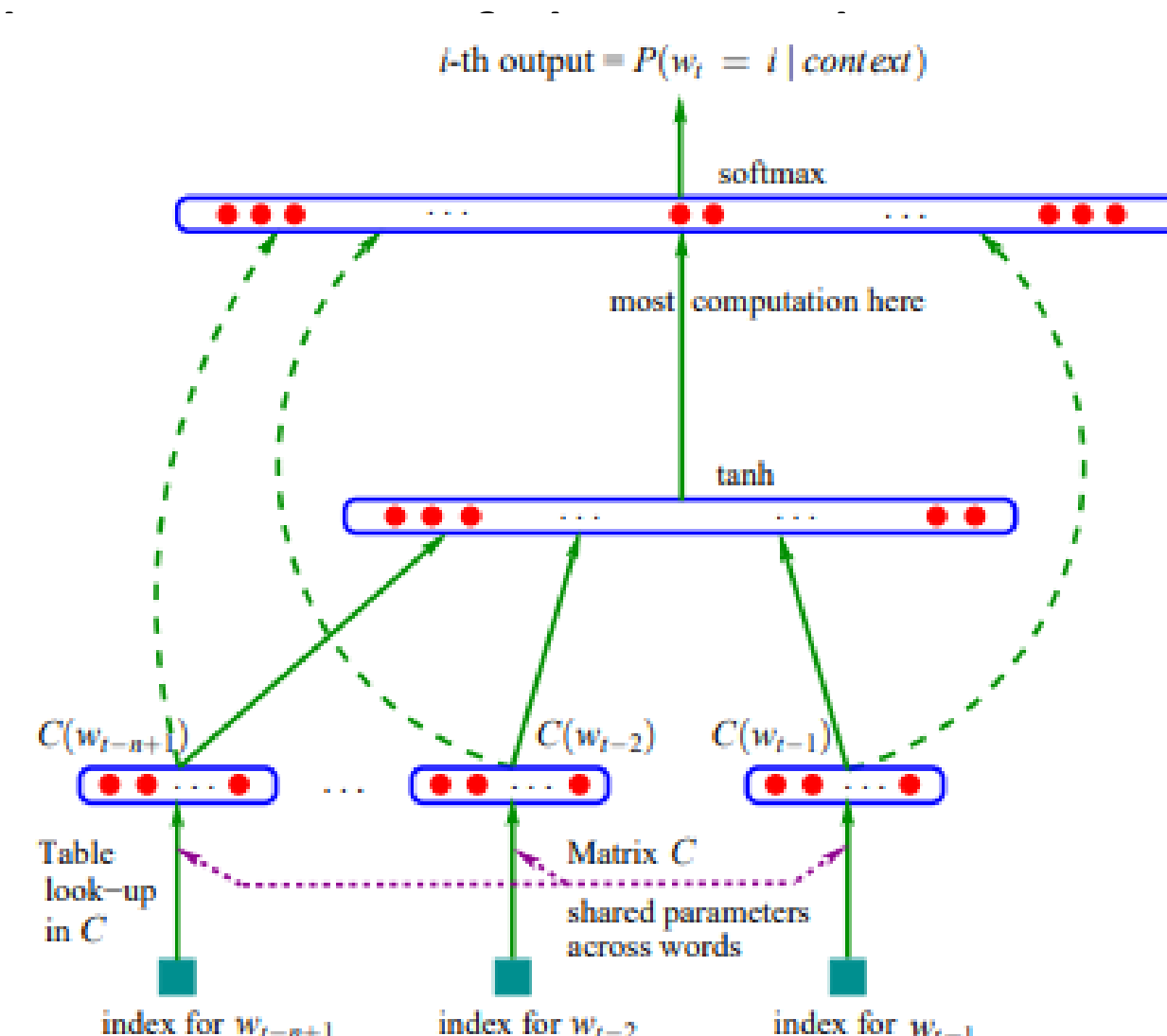
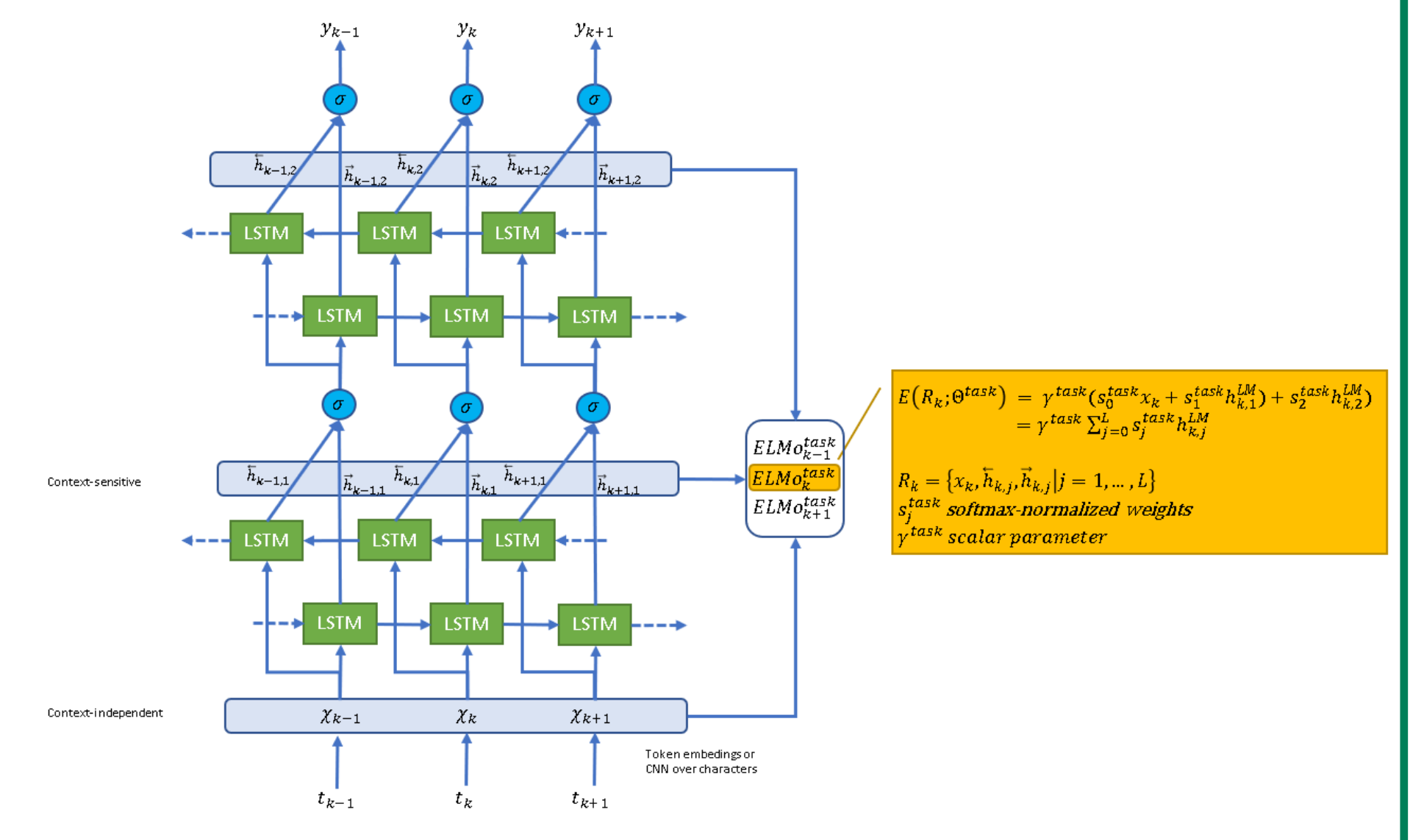


Figure 1: Neural architecture: $f(i, w_{t-1}, \dots, w_{t-n+1}) = g(i, C(w_{t-1}), \dots, C(w_{t-n+1}))$ where g is the neural network and $C(i)$ is the i -th word feature vector.

3. biLSTM+Elmo+FFNN

The third model which is packaged in sdtm-mapper is based on trainable Elmo from bidirectional LSTM which is trained further with FFNN.



Results

The traditional machine learning approach with TF-IDF yields satisfactory results, but most of the models were difficult to generalize. The top-performing model was the logit boosting model which achieved 0.965.



Figure 6. Results of SDTM mapping (Accuracy with 95% CI and Kappa)

The gradient boosting with neural probabilistic model achieved 0.88.

The trainable ELMo with FFNN achieved 0.96-0.98 accuracy and generalized well.

Conclusion

With the use of deep contextualized representations of raw data metadata, SDTM – mapping could be performed at the human level for new studies. Essentially the same method can be applied in other applications such as medical coding.

PhUSE working group has been initiated to further enrich the training data and to encourage industry to learn how to use this approach through sdtm-mapper.



More info



Question?