

NTCIR13 MedWeb Task: Multi-label Classification of Tweets using an Ensemble of Neural Networks.

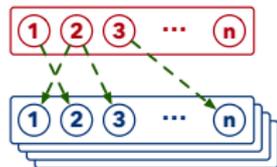
Hayate Iso, Camille Ruiz, Taichi Murayama, Katsuya Taguchi, Ryo Takeuchi, Hideya Yamamoto, Shoko Wakamiya and Eiji Aramaki
Social Computing Lab, Nara Institute of Science and Technology



NAIST®

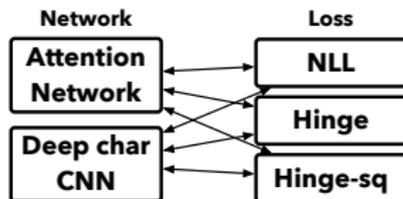
Overview

Resampling



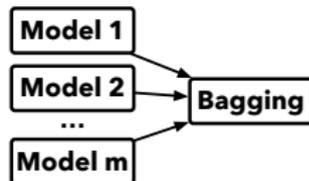
1. Make bootstrap samples

Model



2. Build 6 models for every bootstrap sample

Ensemble



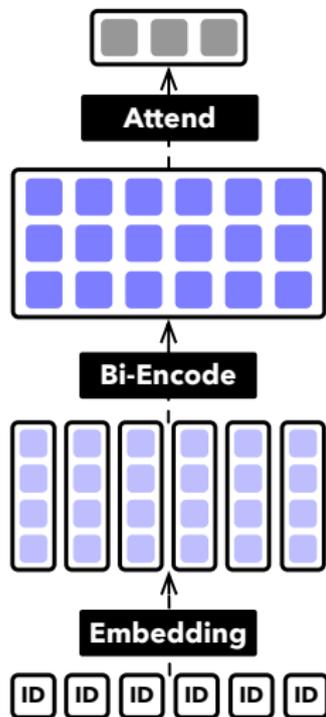
3. Average over all model outputs

- Our team tackled the MedWeb using neural networks that produced the best results with **88.0%** accuracy.
- Our high-level modeling procedure is:
 1. Resampling: Create Bootstrap samples.
 2. Model: Learn Neural Network with 6 settings.
 3. Ensemble: Average over the model outputs.

Features representation

- In this paper, we utilized two neural network models based on both Hierarchical Attention Network (HAN) and Character-level Convolutional Networks (CharCNN).
- The goal is to encode the tweet sentence into a fixed size sentence vector s , which will eventually undergo multi-label classification.

Hierarchical Attention Network



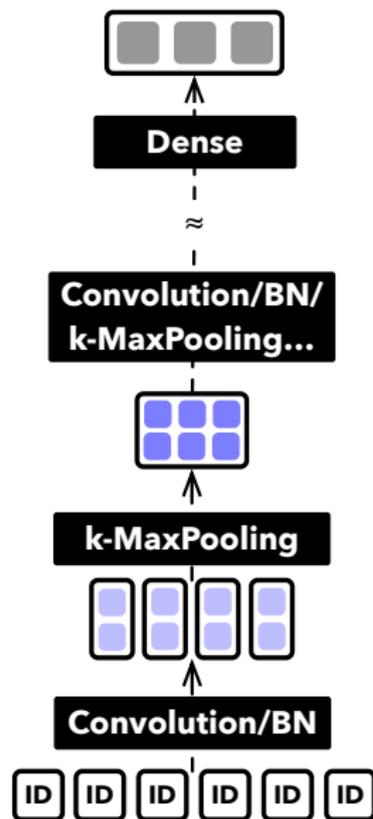
- Given a sentence with words w_t where T is the total number of words in the sentence and embed these words through the embedding matrix $W_e, x_t = W_e w_t$.
- Given the encode bidirectional GRU to encode the tweet sequence $h_t = \text{BiGRU}(x_t)$.
- Compose the tweet vector s with attention mechanism:

$$u_t = \tanh(W_w h_t + b_w),$$

$$\alpha_t = \frac{\exp(u_t^\top u_w)}{\sum_t \exp(u_t^\top u_w)},$$

$$s = \sum_t \alpha_t h_t$$

Character-level Convolutional Network



- In contrast to the HAN, the CharCNN is the deep learning method to compose sentence vector from character sequences.
- To accelerate learning procedure, we adapt Batch Normalization.
- We define the above procedure as CNN and iterate CNN three times:

$$v_{1,1:T_{v,1}} = CNN(c_{1:T_c})$$

$$v_{2,1:T_{v,2}} = CNN(v_{1,1:T_{v,1}})$$

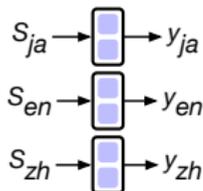
$$v_{3,1:T_{v,3}} = CNN(v_{2,1:T_{v,2}})$$

- Compose the sentence vector s the linear transformation for hidden features v_3 to compose the sentence vector:

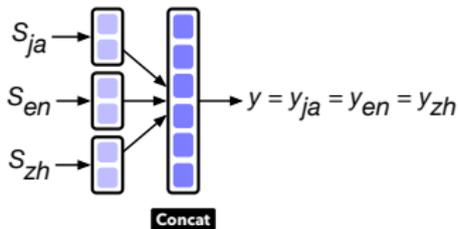
$$s = W_v v_{3,1:T_{v,3}} + b_v.$$

Integrating all three tasks

Language-Independent



Multi-Language



- Although we generally need to learn the neural network model for each task, the MedWeb task consists of the same label set for the different language datasets.

Language Independent learning

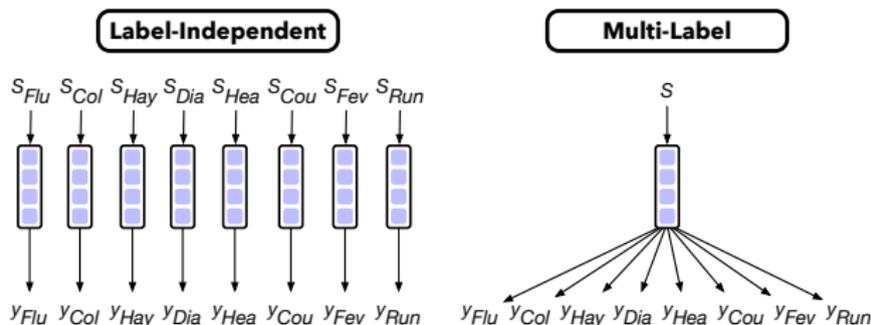
- For each task, we build one neural network model.

Multi-language learning

- Represent the three tweets of each language in a single vector for multi-language learning:

$$s^{\text{Multi}} = [s^{\text{ja}}; s^{\text{en}}; s^{\text{zh}}]$$

Multi-label learning



- Since the task is to perform a multi-label classification of 8 diseases or symptoms per tweet, there are two ways to approach this:

Label-Independent learning

- Build the classifier for each label, respectively:

$$\hat{y}_c = w_c^T s + b'_c \in \mathbb{R}$$

Multi-label learning

- Build one classifier for the 8 labels, simultaneously:

$$\hat{y} = W_c s + b_c \in \mathbb{R}^8$$

Loss functions

- To optimize the models, we experimented following three loss functions:

Negative Log-Likelihood

$$\mathcal{L}_{\text{NLL}} = \sum_i^N \sum_{c=1}^8 \ln(1 + \exp(-y_{c,i} \hat{y}_{c,i}))$$

Hinge

$$\mathcal{L}_{\text{Hinge}} = \sum_i^N \sum_{c=1}^8 \max(0, 1 - y_{c,i} \hat{y}_{c,i})$$

Hinge-Square

$$\mathcal{L}_{\text{Hinge-sq}} = \sum_i^N \sum_{c=1}^8 \max(0, 1 - y_{c,i} \hat{y}_{c,i})^2$$

Bagging ensemble

- Bagging is the ensemble strategy that averages over the outputs learned by resampled dataset.
- We made 20 resampled datasets for this purpose and use each dataset for training the HAN and CharCNN against the 3 loss functions, resulting in 6 methods.

Experiments: Label-independent v.s. Multi-label

Table: Comparison between label-independent or multi-label

Target	Exact match accuracy	
	Label-Independent	Multi-Label
Influenza	0.977	0.988
Diarrhea	0.973	0.979
Hay Fever	0.971	0.975
Cough	0.988	0.991
Headache	0.979	0.981
Fever	0.931	0.929
Runny nose	0.948	0.952
Cold	0.944	0.965
Exact match	0.767	0.823

Experiments: Multi-language and Model config

Table: Language Independent Learning vs. Multi-language Learning - This table shows that multi-language learning is more accurate than language independent learning in any of the languages and classifiers for this dataset. We also append the other team's results for each language, AKBL-ja-3, UE-en-1, TUA1-zh-3 for benchmark, respectively.

Setting		Exact match accuracy				
Encode	Loss	Language-Independent			Multi-Language	
		ja	en	zh	Single	Ensemble
Attention	NLL	0.823	0.791	0.789	0.823	0.841
	Hinge	0.823	0.795	0.809	0.844	0.841
	Hinge-sq	0.825	0.786	0.794	0.822	0.844
CharCNN	NLL	0.800	0.718	0.808	0.831	0.848
	Hinge	0.797	0.686	0.806	0.811	0.869
	Hinge-sq	0.772	0.670	0.784	0.811	0.866
Benchmark		0.805	0.789	0.786	-	-

Experiments: Ensemble results

Table: This table shows the results of our ensembles. Among the 9 ensembles we created, we submitted the last 3—particularly the ensembles using both HAN and CharCNN. Of the three, the ensemble with loss functions NLL and Hinge produced the highest accuracy: 88.0%.

Encode	Ensemble strategy		Exact match
		Loss	
Attention	NLL × Hinge × Hinge-sq	0.842	
	NLL × Hinge	0.836	
	NLL × Hinge-sq	0.844	
CNN	NLL × Hinge × Hinge-sq	0.861	
	NLL × Hinge	0.861	
	NLL × Hinge-sq	0.859	
Attention × CNN	NLL × Hinge × Hinge-sq	0.877	
	NLL × Hinge	0.880	
	NLL × Hinge-sq	0.878	

Summary

- Integrate all tasks into a single neural network.
- Two neural networks–HAN and CharCNN–with multi-language learning are combined.
- Ensemble all models with Bagging.
- The ensemble using the NLL and hinge loss produced the best results with **88.0%** accuracy.