Preposition Sense Disambiguation

Name: Ansh Khurana, Drumil Trivedi, Riya K Baviskar
Roll Number 170050035, 170020016, 170050011

Assumptions

1. Model is trained on training files present in the 'Train' folder. There would be one <head> tag per preposition in a given sentence.
2. Details about how to run the code and a high-level description of the files has been provided in the README.md file.

Approach

We followed the Preposition Sense Disambiguation (PSD) algorithm described in section 3 and 4 of the provided paper.

The problem of disambiguating the preposition sense can be seen as a classification task where the model needs to predict the correct sense_id based on the left and right contexts.

The method uses a vector left context representation $\mathbf{v}_\mathbf{l}$, for right context $\mathbf{v}_\mathbf{r}$ and a vector for context-interplay feature $\mathbf{v}_{\mathbf{inter}}$.

For $\mathbf{v}_\mathbf{l}$ we used the average word embeddings of $\mathbf{k}_\mathbf{l}$ (=1..4) left context words.

For $\mathbf{v}_\mathbf{r}$ we used the average word embeddings of $\mathbf{k}_\mathbf{r}$ (=1..4) right context words.

$\mathbf{v}_{\mathbf{inter}}$ is calculated using the method described in section 3. It is it to be the vector closest to both the subspace spanned by the left context word vectors and that spanned by the right context word vectors.

Thus [$\mathbf{v}_\mathbf{l}$, $\mathbf{v}_\mathbf{r}$, $\mathbf{v}_{\mathbf{inter}}$] are the input features for our classification models. These input features can be fed directly to the model, or we can use a linear combination of these features.

The outputs of the model are canonicalized version of the sense_ids provided in the XML files.

We treat independent models for every preposition since the classification label space for every preposition is unrelated.

sense_id

The sense ids are used as provided in the XML files without any alterations.

Implementation Details

For the word embeddings, we are using GloVe embeddings downloaded from https://nlp.stanford.edu/projects/glove/ (Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB download): glove.6B.zip)

Data processing steps have been set-up in data.py.

Implementation of the algorithms is available in main.py.

We have implemented the models using the sklearn library.

After running the main script, outputs are stored in the folder specified by --save_test_outs in the format described on moodle.

Methods

We implemented 3 supervised classification models:

1. SVM
2. MLP
3. KNN

Results

Chosen hyperparameters:

k_l = k_r = 3

MLP - 1 hidden layer with 20 neurons

KNN - 8 nearest neighbours

SVM - regularization parameter = 1

The outputs for the test set are stored in the provided <preposition.out> format. For each proposition, the model returns the predictions for each of the sentences in the test file. The predicted sense_id is saved the sentences in the following format.

<setence> | <predicted_sense_id>

For testing out our implementation, we had split the training set into an 80-20 train/validation set.

We obtained the following weighted (across different prepositions) accuracies on the train and validation set:

Method	Train Accuracy	Validation Accuracy	Train F1 score	Validation F1 score
MLP	**0.992**	**0.663**	**0.989**	**0.663**
KNN	0.770	0.618	0.740	0.590
SVM	0.734	0.604	0.670	0.543

The **MLP** model performs the best among all methods. Thus, we have saved our test outputs for the MLP algorithm in test_out/.