I. Architecture

Architecture used in the project is closely followed by this paper: http://www.aclweb.org/anthology/W16-5307

A. Bi LSTM

Each sentence is split into two halves according to the position of the target word: forward pass and backward pass. Each pass will then be feed into a LSTM. The outcome of this phase is a output vector h at the end of the pass. We then have two output vectors for each pass and concatenate these vectors and forward them to the next layer.

B. Hidden layer

The hidden layer comes to shrink down the dimension of the LSTM outputs and forward them to softmax layer.

C. Softmax layer

In softmax layer, each sample will be be multiplied with its own softmax parameters. The model will try to minimize the cross entropy loss using the result of softmax operation.

II. Preprocessing

A. Word preprocessing

No complicated preprocessing was performed during training. Sentences are stripped newline tag and split into tokens by space character. All the tokens are converted to lowercase.

B. Vocabulary

In order to make sure all words in the dataset has its own embeddings, I first read all the sentences and make a list of unique words in the dataset. As a result, I have a list of about 45k words. After this, I will try to find a pre trained vector for each word, if it is not possible then I will initialize a random vector for that word.

C. Lemma-to-senses

In order to build a dictionary to map a specific lemma to a set of possible senses, I have to read all the data from semcor and ALL and then update them. So after this, I have a biggest lemma-to-senses as possible. During training, I assumed no new senses other than the ones in provided dataset will appear.

D. Wrapper

All the data point is wrapped around a custom class which has 6 properties to be extracted.

1. fw: contains the forward pass of the sentence until the target word.
2. bw: contains the backward pass of the sentence until the target word.
3. word_id: instance ID
4. sense_id: sense ID
5. word_int: lemma int (used to find softmax params for this target word)
6. possible_senses: a set of possible sense IDs for this lemma

III. Experimental result

- Figure 1 summarizes the training process by plotting the loss against number of epochs
- Figure 2 compares train accuracy and val accuracy. We can see that the model is suffering from overfitting.
- Figure 3 gives us a clearer look at how well the model is doing on the validating set and suggests that this model did not converge and can do better if it was trained more.
- Table 1 sums up the F1 score of the model on a variety of datasets.

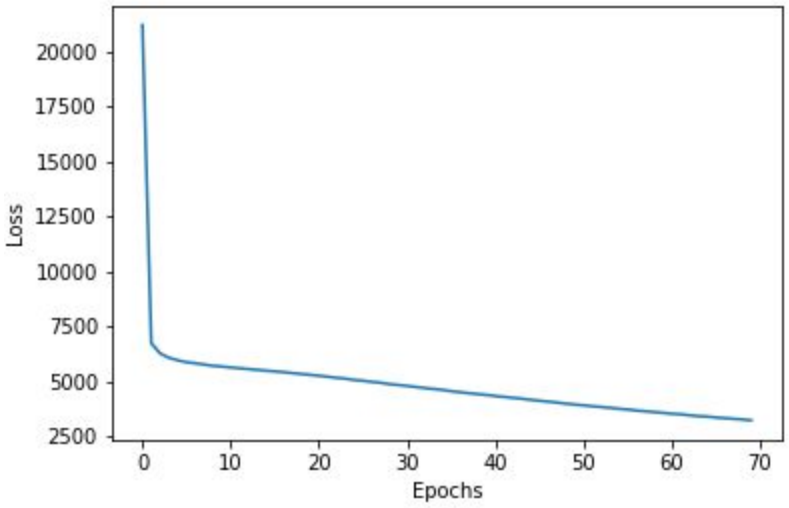

Figure 1

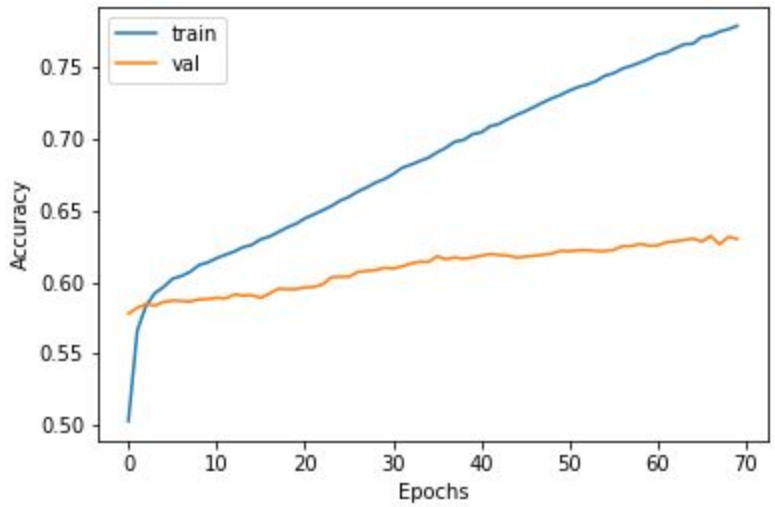

Figure 2

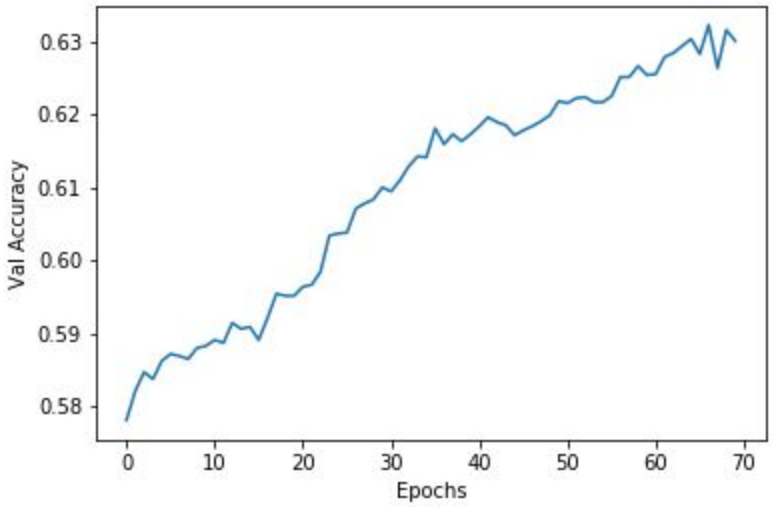

Figure 3

	semeval2015	senseval3	semeval2013	senseval2	semeval2007	all
F1 score	0.59	0.654	0.602	0.657	0.589	0.63

Table 1