I trained a crf as vanilla bilstm-crf was not performing as good as crf with features. Training time was a constraint so I didnt try feature engineering on bilstm crf. I am mentioning the features used in CRF:

Feature for current word:

```
features = {
    'bias': 1.0,
    'word.lower()': word.lower(),
    'word[-3:]': word[-3:],
    'word[-2:]': word[-2:],
    'word[-4:]': word[-4:],
    'word[:2]': word[:2],
    'word.isupper()': word.isupper(),
    'word.istitle()': word.istitle(),
    'word.isdigit()': word.isdigit(),
    'postag': postag,
    'postag[:2]': postag[:2],
    'digit>=8': greater(word,8),
    'digit>=3': greater(word,3),
    'hassec': hassec(word),
    'hashttp': hashttp(word),
    'hasPfeat': hasPfeat(word),
    'hasbhk': hasgen(word,'bhk'),
    'hassq': hasgen(word,'sq'),
    'has/': hasgen(word,'/'),
    'has@': hasgen(word,'@'),
    'per': hasgen(word,'per'),
}
```

hasgen(a,b) func. Checks whether b is in a as substring

Feature from prev. and next word

```
        features.update({
    '-1:word.lower()': word1.lower(),
    '-1:word.istitle()': word1.istitle(),
    '-1:word.isupper()': word1.isupper(),
    '-1:postag': postag1,
    '-1:postag[:2]': postag1[:2],
    '-1digit>=8': greater(word1,8),
```

```
    '-1digit>=3': greater(word1,3),
    '-1hassec': hassec(word1),
    '-1hashttp': hashttp(word1),
    '-1hasPfeat': hasPfeat(word1),
    '-1hasbhk': hasgen(word1,'bhk'),
    '-1hassq': hasgen(word1,'sq'),
    '-1has/': hasgen(word1,'/'),
    '-1has@': hasgen(word1,'@'),
    '-1per': hasgen(word1,'per') or ((i>1)and hasgen(sent[i-2][0],'per')),
})
```