1 Error propagation

1.1

For example:

1. China is a country. 2. He likes china.

We can see the first sentence China is a country name and it is a named entity. But the second china is a noun. Thus this can bring errors.

2

2.1

We used result from MBT to generate POS-tag from NLTK. From result, we see that there are 19465 difference of POS-tagging between MBT Tagger and NLTK. And we count the number of main difference between MBT Tagger and NLTK. We found 1836 differences of CD in MBT but JJ in NLTK. And 1253 difference of NN in MBT but NNP NLTK. And ('JJ', 'NNP'): 962, ('JJ', 'NN'): 673, ('NNP', 'JJ'): 731 and so on.

2.2

Then we further investigate how the POS-tag influence the named entity. We generate the words which cause different prediction between MBT and NLTK. We find that most words are not have big influence on named entity recognition such as *that(85 different POS-tag)*. But some words do have influence on named entity recognition. For example, *Europen (27 different POS-tag)*. In NLTK, many *Europen* are MBT but in NLTK it is NNP. In ins way the prediction of MBT will be not accurate. Because the way of prediction of model highly depends on NNP part-of-speech.

2.3

Using code from assignment 2 and part-of-speech from NLTK we predict the named entity. Then we compare the result from NLTK and MBT. We see for both model the POS-tag from NLTK is slightly better than MBT, which corresponds to our result from 2.2. Because NLTK predicts words like *European, British...* as NNP, which increases the possibility to predict as named entity.

Table 1: Best feature selection

Number of feature	Best features	Best precision	Best recall	Best f1
5-NLTK	Token, Prevtoken, Cap, Pos, Chunklabel	0.778	0.737	0.750
5-MBT	Token, Prevtoken, Cap, Pos, Chunklabel	0.771	0.723	0.740

Table 2: Outcome of word embeddings

Best features	Best precision	Best recall	Best f1
A mixed system-NLTK	0.785	0.787	0.786
A mixed system-MBT	0.784	0.784	0.785

3 Evaluation Metrics

Predicted	B-LOC	B-MISC	B-ORG	B-PER	I-LOC	I-MISC	I-ORG	I-PER	O
Gold									
B-LOC	1348	18	134	70	1	1	7	46	43
B-MISC	28	458	54	47	0	1	8	24	82
B-ORG	184	43	1038	175	1	1	26	57	136
B-PER	68	20	45	1258	0	0	1	141	84
I-LOC	5	0	0	0	155	4	40	30	23
I-MISC	1	8	0	3	3	128	12	23	38
I-ORG	21	8	5	11	45	8	521	111	105
I-PER	4	1	0	13	1	0	15	1093	29
O	29	41	70	101	3	37	36	105	38132

P: 0.775359110733929 R: 0.7352997074459117 F1: 0.7489500540371716

3.1

The accuracy is the correct prediction percentage, which are the entries on diagonal divided by total number. In this case: $44131/46666 = 0.946$.

3.2

Result:

Table 3:

	0	1
0	True Positives	False Positives
1	False Negatives	True Negatives

Table 4:

Attributes	Macro	Micro	Weighted
Precision	0.775	0.946	0.946
Recall	0.735	0.946	0.946
F1	0.749	0.946	0.945

Precision = True Positives / (True Positives + False Positives)
Recall = True Positives / (True Positives + False Negatives)
F1 Score = 2 * (Precision*Recall)/(Precision+Recall)

These three all have corresponding micro-average or macro-average. Macro-average refers to the arithmetic mean of each statistical index value of all categories. And macro-average is each example in the data set does not divide the category, statistically establishes a global confusion matrix, and then calculates the corresponding index.

According to the figure 3, the macro score is more informative for named entity recognition. Because the weighted of micro score is based on the number of each sample. But for named entity recognition sample O has larger number and it is useless for named entity recognition. Thus, macro score is more suitable for named entity recognition.

3.3

Table 5:

Attributes	O	Other
O	38132	422
Other	540	7572

Accuracy: 97.9%, F1(macro): 96.4%, F1(micro): 97.9%

4 Input representation

I think POS and token are highly correlated Because the method of tokening will determine the part-of-speech. POS and Capitalization are not correlated. Because they are irrelevant .

5 Input representation

Sparse matrix's the main entries are zeros and contains less information. While dense matrix is relatively small vectors and contains more information. And sparse matrix is easier to calculate for computer.

6 End-to-end models

Predicted	B-LOC	B-MISC	B-ORG	B-PER	I-LOC	I-MISC	I-ORG	I-PER	O
Gold									
B-LOC	1279	62	122	36	3	1	5	0	155
B-MISC	31	453	51	12	3	3	2	1	142
B-ORG	142	79	1008	54	1	4	27	1	344
B-PER	70	44	148	921	3	5	16	20	388
I-LOC	6	1	0	3	149	3	50	6	39
I-MISC	0	14	6	3	3	125	13	5	44
I-ORG	13	12	36	8	32	25	571	17	120
I-PER	5	6	16	50	10	7	66	666	330
O	183	211	396	138	25	82	152	56	37225

P: 0.685203034754767 R: 0.6656313290813023 F1: 0.6690473776414725

According to figure 3 and 6, it is clear to see that the LSTM output has worse performance than previous result. Only accuracy of I-ORG is better than previous. And the worst prediction is on I-PER. And precision, recall and F1 are also worse than previous result. I would choose the first system because the first system has better accuracy, precision, recall, and F1 scores.

7 Error Analysis

7.1

Figure 7.4 shows the part of wrong prediction of models. It is clear to see that the wrong prediction is mainly about Country and people's name. And we noticed that each sentence is divided by ————- and each sentence has different length. So our first hypothesis is the length of sentence can influence the accuracy of prediction. And then we look at the wrong prediction on specific named entity. We notice that there are many wrong prediction of person's name. For example, *67 B-PER was predicted as O* and *82 I-PER was predicted as O.* Thus, our second hypothesis is this model is bad at predict PER attributes.

	Word	Golden	Prediction
2	JAPAN	B-LOC	O
13	Nadim	B-PER	O
14	Ladki	I-PER	B-PER
16	AL-AIN	B-LOC	B-ORG
72	Uzbekistan	B-LOC	B-PER
91	Uzbek	B-MISC	O
93	Igor	B-PER	B-ORG
94	Shkvyrin	I-PER	I-ORG
99	misdirected	O	B-MISC
118	Oleg	B-PER	O
119	Shatskiku	I-PER	O
144	Soviet	B-MISC	B-LOC
163	Asian	B-MISC	B-ORG

Figure 1: Wrong prediction in sample.

7.2

According to the paper, the hypothesis need to be precise and clear. Thus, we need to investigate the 10% dataset carefully.

For the first hypothesis, we carefully investigate the accuracy of different sentence length. We found out when sentences have more than 50 tokens the accuracy drops significantly. Thus, our first hypothesis is **model is bad at predicting the sentences which have more than 50 tokens.**

For second hypothesis, We separate the first name and last name, for example, we count Qinghe Gao are two names. And we found out when name have 4 letters there are 27 out of 70 errors. But when name's letters are more than 4, the errors become large. Thus, our second hypothesis is **model is bad at predicting the name which have more than 5 letters.**

7.3

We further investigate the correct prediction and also training and validation dataset. We found for the first hypothesis. Many sentences still have more than 50 tokens. Thus, the model will perform consistently on both experiment step and prediction step. And we also find that for second hypothesis there are many names which have more than 5 letters.

7.4

Finally, we can verify our hypothesis on full .

For the first hypothesis, we calculate the accuracy for each sentence which is divided by ———-. The accuracy is right prediction over total prediction. And then we average the accuracy of sentences which have same tokens. Figure 7.4 shows our result of first hypothesis. We can see when sentences have more than 55 tokens the accuracy drop quickly. Thus, the model is bad at sentences which have more than 55 tokens.

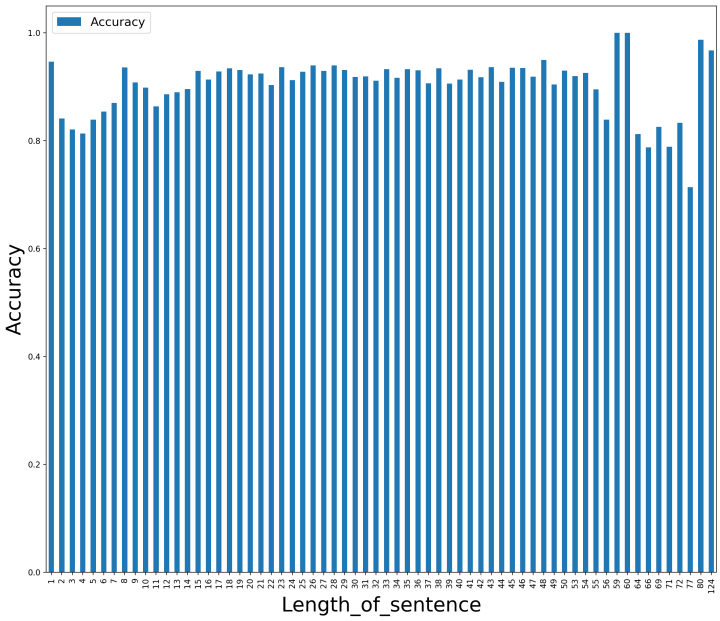

Figure 2: First hypothesis.

And for second hypothesis, first we count the whole wrong prediction. We found there are 4167 wrong prediction. Then we further count the number of wrong predictions on name(for both B-PER and I-PER), which is 1184. Thus, 28.4% wrong prediction on PER. Then, we further count the wrong prediction names which have more than 5 letters, which is 693. And 58.5% wrong prediction is the name which have more than 5 letters. Thus, the model is bad at predicting the name which have more than 5 letters.

8 Error Analysis

In this part, we do adversarial examples in this part. And still we want to test hypothesis: the model is bad at predicting the name which have more than 5 letters. And as for adversarial examples, we need to manipulate the input without changing golden labels. Thus, our steps show as follow:

- Generate name list. In this step, we need to generate a name list which contains the names that have less than 5 letters(include 5 letters). This name list could extract the name from train, validation or just test dataset.
- Replace name. In this step, we randomly pick up a name in name list and substitute it with 5-more-letter name in input dataset. We also can do another way around. Like substituting all the 5-less-letter name with the name which have more than 5 letters.
- Then analyze the prediction

The adversarial examples show as data-adversarial dataset. Here are some example in the dataset. The adversarial examples show as data-adversarial dataset. Here are some example in the dataset.

- Original. when Uzbek striker Igor **Shkvyrin** took advantage of a misdirected . . .
- Adversarial. when Uzbek striker Igor **Dion** took advantage of a misdirected . . .

The whole data set is in zip file.

References