BotSharp Team
June 01, 2018

Building a Named Entity Recognition on BotSharp.

We talked about why CRF model is a good fit for (named entity recognization) NER system. This blog speaks about how we can build a Named Entity Recognizer using you own named entity data on BotSharp platform.

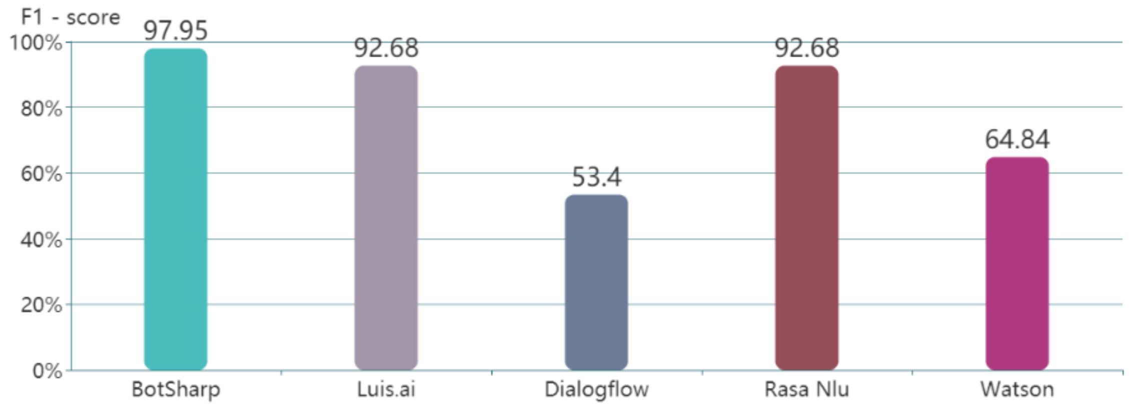
At the beginning, I will introduce about BotSharp, and what is the advantage of BotSharp NER system compared to other Platforms.

What is BotSharp platform?

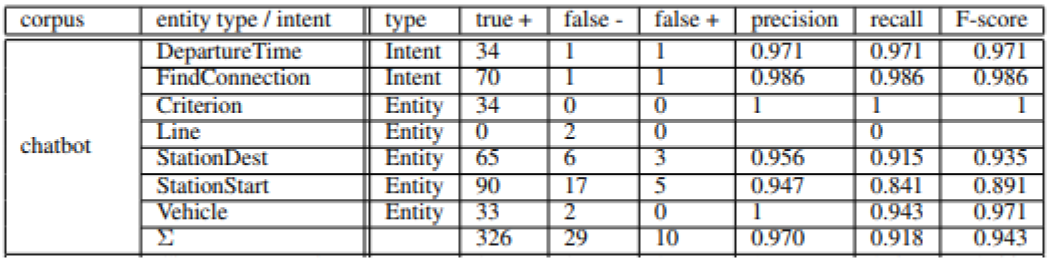
BotSharp CoreNLP(https://github.com/dotnetcore/BotSharp) is an open source NLP platform conducted by my mentor Haiping. It's witten in C# running on .Net Core that is full cross-platform framework. C# is a enterprise grade programming language which is widely used to code business logic in information management related system. BotSharp adopts machine learning algrithm in C/C++ interfaces directly which skips the python interfaces. That will facilitate the feature of the typed language C#, and be more easier when refactoring code in system scope.

How is the BotSharp NER system performance?

We reproduced an academic benchmark: “Evaluating Natura Language UnderStanding Services for Conversational Question Answer” which is published in 2017 summer. The Author of this paper assess the performance of Api.ai (Diagflow), Luis (Microsoft), Watson (IBM) and Rasa Nlu. **In regarding of named entity recognition performance based on chatbot corpus**, We re-calculate the f1-score of each engine. The results are shown below in this graph.

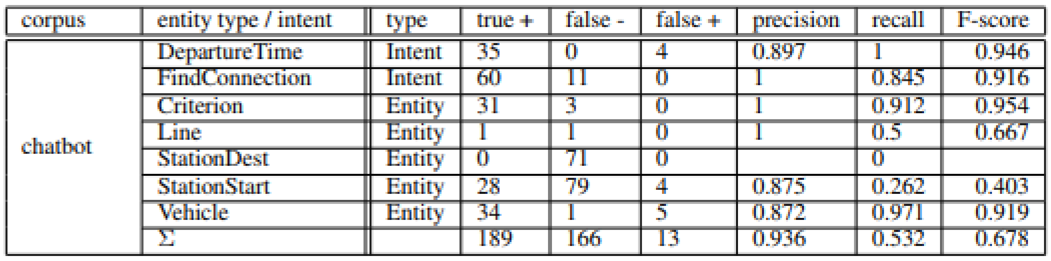


To show more detail about BotSharp performance in named entity recognition,I stole more detail data from <u>"Evaluating Natura Language UnderStanding Services for Conversational Question Answer"</u> for comparation.



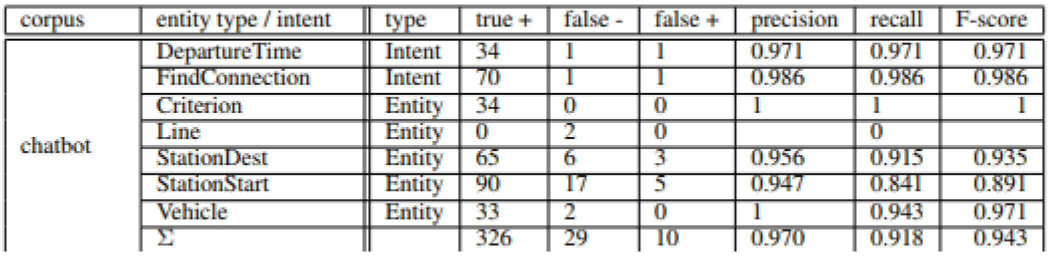
corpus	entity type / intent	type	true +	false -	false +	precision	recall	F-score
chatbot	DepartureTime	Intent	34	1	1	0.971	0.971	0.971
	FindConnection	Intent	70	1	1	0.986	0.986	0.986
	Criterion	Entity	34	0	0	1	1	1
	Line	Entity	0	2	0		0	
	StationDest	Entity	65	6	3	0.956	0.915	0.935
	StationStart	Entity	90	17	5	0.947	0.841	0.891
	Vehicle	Entity	33	2	0	1	0.943	0.971
	Σ		326	29	10	0.970	0.918	0.943

Results Luis.ai (Microsoft)



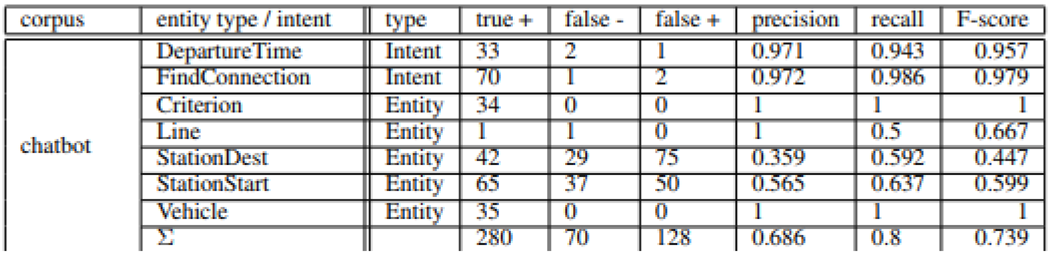
corpus	entity type / intent	type	true +	false -	false +	precision	recall	F-score
chatbot	DepartureTime	Intent	35	0	4	0.897	1	0.946
	FindConnection	Intent	60	11	0	1	0.845	0.916
	Criterion	Entity	31	3	0	1	0.912	0.954
	Line	Entity	1	1	0	1	0.5	0.667
	StationDest	Entity	0	71	0		0	
	StationStart	Entity	28	79	4	0.875	0.262	0.403
	Vehicle	Entity	34	1	5	0.872	0.971	0.919
	Σ		189	166	13	0.936	0.532	0.678

Results Api.ai (Google Diagflow)



corpus	entity type / intent	type	true +	false -	false +	precision	recall	F-score
chatbot	DepartureTime	Intent	34	1	1	0.971	0.971	0.971
	FindConnection	Intent	70	1	1	0.986	0.986	0.986
	Criterion	Entity	34	0	0	1	1	1
	Line	Entity	0	2	0		0	
	StationDest	Entity	65	6	3	0.956	0.915	0.935
	StationStart	Entity	90	17	5	0.947	0.841	0.891
	Vehicle	Entity	33	2	0	1	0.943	0.971
	Σ		326	29	10	0.970	0.918	0.943

Results Rasa Nlu



corpus	entity type / intent	type	true +	false -	false +	precision	recall	F-score
chatbot	DepartureTime	Intent	33	2	1	0.971	0.943	0.957
	FindConnection	Intent	70	1	2	0.972	0.986	0.979
	Criterion	Entity	34	0	0	1	1	1
	Line	Entity	1	1	0	1	0.5	0.667
	StationDest	Entity	42	29	75	0.359	0.592	0.447
	StationStart	Entity	65	37	50	0.565	0.637	0.599
	Vehicle	Entity	35	0	0	1	1	1
	Σ		280	70	128	0.686	0.8	0.739

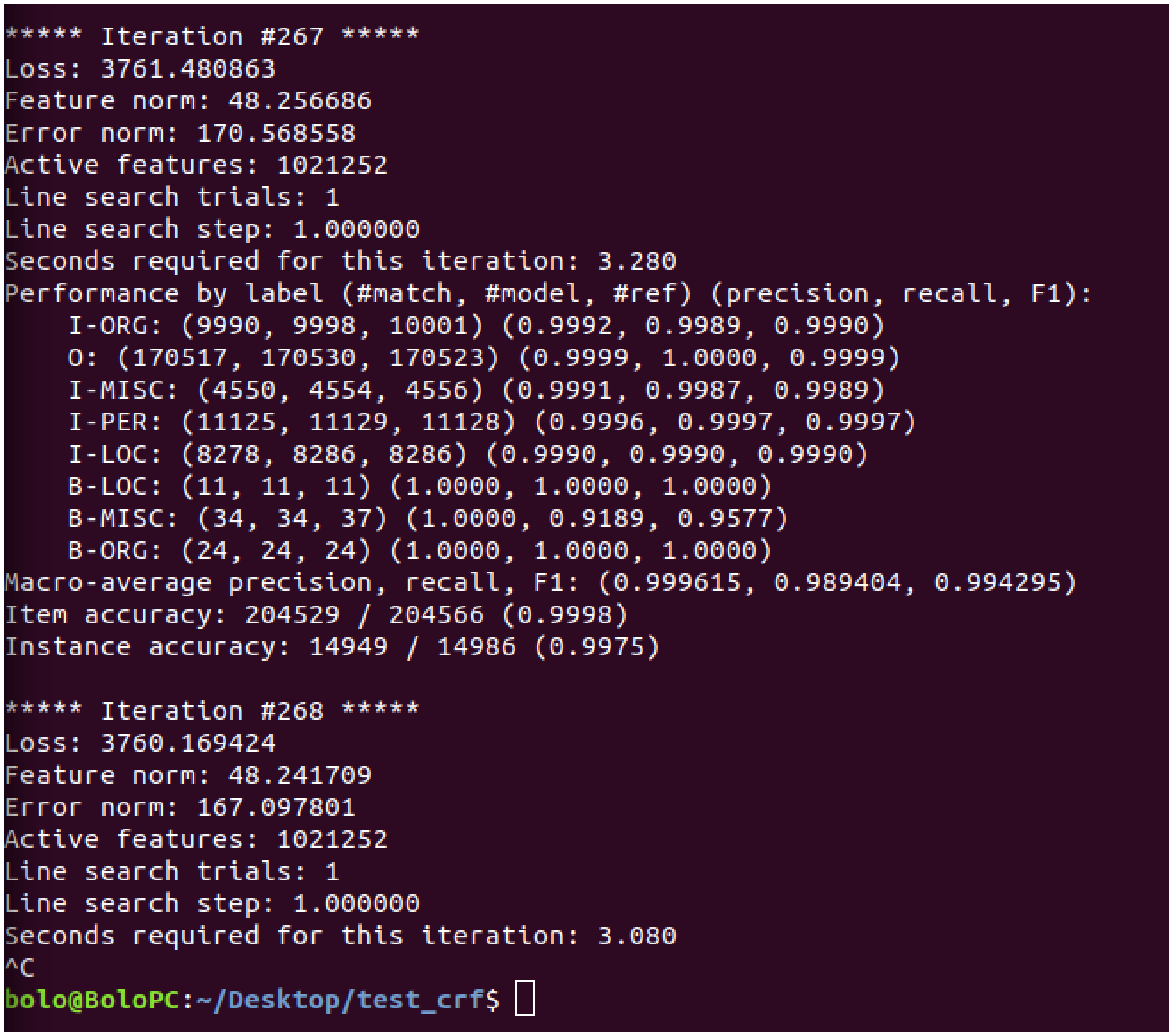
Results Watson (IBM)

corpus	Entity type / intent	type	true +	false -	false +	precision	recall	F- score
chatbot	Departure Time	Intent	34	1	1	0.971	0.971	0.971
	FindConnection	Intent	70	1	1	0.986	0.986	0.986
	Criterion	Entity	29	0	0	1	1	1
	Line	Entity	0	0	0	-	-	-
	StationDest	Entity	45	3	0	1	0.9375	0.9677

	StatonStart	Entity	78	0	2	0.975	1	0.9873
	Vehicle	Entity	29	0	0	1	1	1
	Σ		181	3	2	0.989	0.984	0.9795

Results BotSharp NLP

Based on the same chatbot corpus benchmark , we did cross-validation spliting the data by 5 and the validation results are shown in Results BotSharp NLP. As comparing to others, we can see BotSharp has a better performance on Named Entity Recognition on both precision and recall . Besides, BotSharp still have a high accuracy in NER in large number corpus. We train the NER system using CoNLL 2003 benchmark.



```
***** Iteration #267 *****
Loss: 3761.480863
Feature norm: 48.256686
Error norm: 170.568558
Active features: 1021252
Line search trials: 1
Line search step: 1.000000
Seconds required for this iteration: 3.280
Performance by label (#match, #model, #ref) (precision, recall, F1):
    I-ORG: (9990, 9998, 10001) (0.9992, 0.9989, 0.9990)
    O: (170517, 170530, 170523) (0.9999, 1.0000, 0.9999)
    I-MISC: (4550, 4554, 4556) (0.9991, 0.9987, 0.9989)
    I-PER: (11125, 11129, 11128) (0.9996, 0.9997, 0.9997)
    I-LOC: (8278, 8286, 8286) (0.9990, 0.9990, 0.9990)
    B-LOC: (11, 11, 11) (1.0000, 1.0000, 1.0000)
    B-MISC: (34, 34, 37) (1.0000, 0.9189, 0.9577)
    B-ORG: (24, 24, 24) (1.0000, 1.0000, 1.0000)
Macro-average precision, recall, F1: (0.999615, 0.989404, 0.994295)
Item accuracy: 204529 / 204566 (0.9998)
Instance accuracy: 14949 / 14986 (0.9975)

***** Iteration #268 *****
Loss: 3760.169424
Feature norm: 48.241709
Error norm: 167.097801
Active features: 1021252
Line search trials: 1
Line search step: 1.000000
Seconds required for this iteration: 3.080
^C
bolo@BoloPC:~/Desktop/test_crf$
```