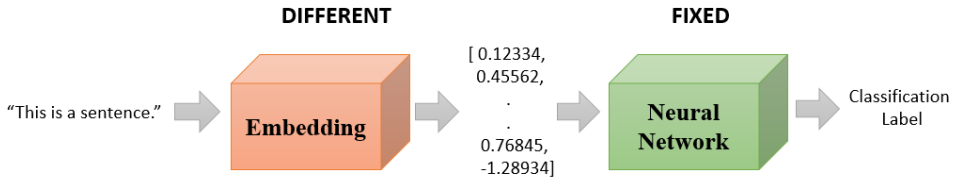
EMBEDDING BENCHMARK

1. Overview

The aim of this embedding benchmark test is to identify the accuracy and computational time for different pre-trained embedding models on different types of datasets in terms of classes, sample sizes and sequence lengths using a fixed neural network classifier.



2. Data Sets

Four different data sets were tested ranging from short and long sentences for intent classification, medium-length sentences for sentiment analysis, and long sentences for document classification. The summary of the data sets as follow:

Data Set	Type	Data Size	Classes	Train / Test	Average Length	Max Length
Conversational Agents[1]	Intent	25580	68	20571 / 5144	6	61
JIRA Issue Tracking[2]	Sentiment	5692	6	4553 / 1139	23	1536
BBC News[3]	Category	2225	5	1780 / 445	368	2427
Banking Agent[4]	Intent	412	13	329 / 83	4	13

Note: Refer to Appendix for distribution of data

1- https://github.com/xliuhw/NLU-Evaluation-Data
2- http://ansymore.uantwerpen.be/system/files/uploads/artefacts/alessandro/MSR16/archive3.zip
3- https://github.com/suraj-deshmukh/BBC-Dataset-News-Classification
4- Self-generated

3. Embedding

Nine different word embedding models were chosen and tested using various pooling methods to derive sentence embedding. The summary of embedding architectures as follow:

Architecture	Layers	Hidden	Heads	Parameters	Pooling	Output
USE (Large)	-	-	-	-	-	512
BERT (Large)	24	1024	16	340M	Mean (Layer -2)	1024
ELMo (Large)	2	4096	-	96.3M	Concat (Layer 1:3)	3072
XLNet (Large)	24	1024	16	340M	Mean (Layer 1)	1024
RoBERTa (Large)	24	1024	16	355M	Mean (Layer -2)	1024
XLM (en-2048)	12	2048	16	-	Mean (Layer 1)	2048
GPT-2 (Medium)	24	1024	16	345M	First+Last (Layer 1)	2048
Transformer-XL	18	1024	16	257M	(Layer 1:3)	3072
Glove + Flair	-	-	-	-	Mean	4196

Note: Refer to Appendix for more details on the architecture libraries

4. Results

The following result is based on (1) 16-cores, 1x 16GB P5000 GPU workstation, (2) standardized train-test-split across all embedding, (3) standardized neural network classifier with 2-layers, crossentropy loss, adam optimizer and batch size of 64.

Note: colour: best performance and colour: worse performance. '-' indicates that the embedding library cannot be used for the corresponding data set due to sequence length exceeding limit. F1 accuracy is based on the best score among three to five training/encoding (accuracy fluctuates).

Dataset	Embedding	Total Training/Encoding Time (Secs)	Total Inference Time (Secs)	F1 Accuracy (%)
Conversational Agents	**USE**	400	7	88.24
	BERT	408	85	87.03
	ELMo	309	78	86.99
	XLNet	573	146	86.08
	RoBERTa	405	100	87.29
	XLM	372	92	85.80
	GPT-2	423	106	84.64
	Transformer-XL	1762	291	83.94
	Glove + Flair	553	139	87.64
JIRA Issue Tracking	**USE**	500	8	87.36
	BERT	2200	662	86.65
	ELMo	180	46	87.00
	XLNet	131	35	85.33
	RoBERTa	-	-	-
	XLM	-	-	-
	GPT-2	-	-	-
	Transformer-XL	290	72	86.30
	Glove + Flair	406	108	87.27
BBC News	**USE**	340	6	97.98
	BERT	1045	277	98.42
	ELMo	940	230	97.75
	XLNet	404	96	97.30
	RoBERTa	-	-	-
	XLM	-	-	-
	GPT-2	-	-	-
	Transformer-XL	440	106	96.63
	Glove + Flair	2394	585	96.85
Banking Agent	**USE**	6	2	85.54
	BERT	5	1	91.56
	ELMo	3	< 1	92.77
	XLNet	7	< 2	90.36
	RoBERTa	6	1	87.95
	XLM	5	< 2	92.77
	GPT-2	6	1	90.36
	Transformer-XL	17	4	86.74
	Glove + Flair	6	1	89.15

5. Conclusion

The following recommendation and observations can be drawn from the above experiment:

5.1. Recommendation

Use the ELMo embedding interface provided by Flair over the currently used TensorFlow Hub module. The Flair implemented ELMo package produces better performance mainly in computational time (refer to table below).

The TFHub default output returns a fixed mean-pooling of all contextualized word representation of size 1024. The Flair implementation returns a concatenation of all word representation of size 3072 which might result in a slightly higher accuracy. The Flair implementation also automatically uses GPU if available but the TFHub module doesn't, which explains why the encoding and inference time is significantly faster.

Data Set	TensorFlow Hub			Flair		
	Encoding Time	Inference Time	Accuracy	Encoding Time	Inference Time	Accuracy
Conversation	3000	50s	87.09	309	78	86.99
JIRA	8500	132	76.65	180	46	87.00
BBC	15800	225	98.20	940	230	97.75
Banking	40	2	90.36	3	< 1	92.77

5.2. Observations

5.2.1. Accuracy

(i) USE is one of the best performing in terms of accuracy. It achieved the highest accuracy for conversational agent and JIRA issue tracking, and the second highest for BBC news. However, it has the worst accuracy for banking agent data set.

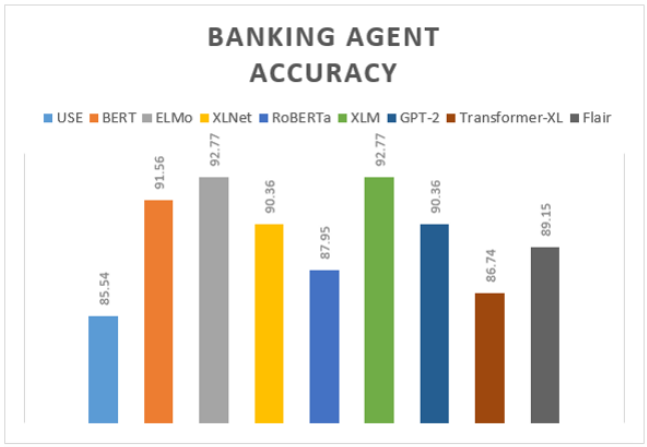
(ii) BERT achieved the best accuracy for BBC news category classification and is consistently one of the highest for the other data sets.

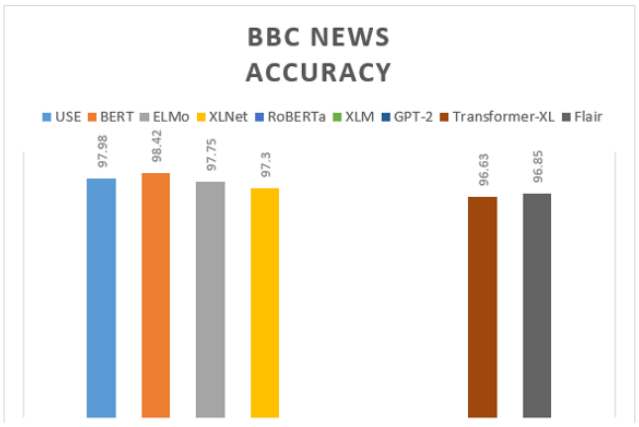
(iii) ELMo achieved the best accuracy for banking agent intent classification and is also consistently one of the highest for the other data sets.

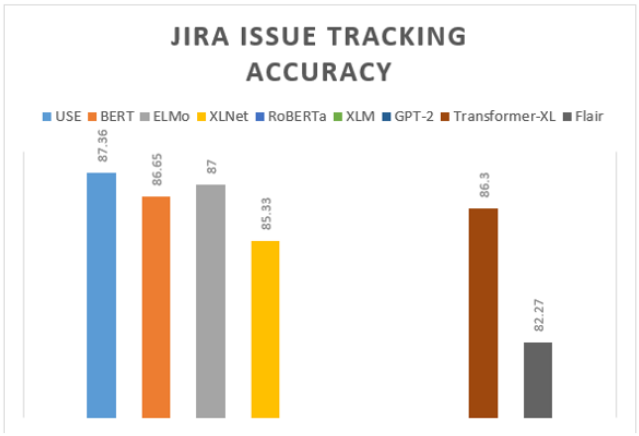
(iv) XLNet has the worst accuracy for sentiment analysis and has above average performance for other data sets.

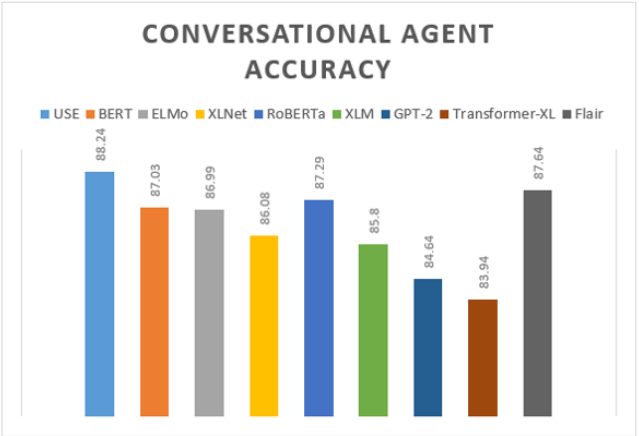
(v) XLM has the best accuracy for banking agent intent classification together with ELMo. However, for the conversational agent intent classification (larger data set) it has one of the worse performance.

(vi) Transformer-XL is the worst performing with two out of four data sets producing the lowest accuracy.


BANKING AGENT ACCURACY
USE BERT ELMo XLNet RoBERTa XLM GPT-2 Transformer-XL Flair
85.54
91.56
92.77
90.36
87.95
92.77
90.36
86.74
89.15


BBC NEWS ACCURACY
USE BERT ELMo XLNet RoBERTa XLM GPT-2 Transformer-XL Flair
97.98
98.42
97.75
97.3
96.63
96.85


JIRA ISSUE TRACKING ACCURACY
USE BERT ELMo XLNet RoBERTa XLM GPT-2 Transformer-XL Flair
87.36
86.65
87
85.33
86.3
82.27


CONVERSATIONAL AGENT ACCURACY
USE BERT ELMo XLNet RoBERTa XLM GPT-2 Transformer-XL Flair
88.24
87.03
86.99
86.08
87.29
85.8
84.64
83.94
87.64

5.2.2. Computational Time

(i) Transformer-XL takes the longest to compute consistently for both data set on intent classification. It is relatively fast for sentiment and category classification.

(ii) USE is the best performing for computational time. It achieved the fastest training time for the category data set and the fastest inference time for every data set except the banking agent; which inference time is still fast.

(iii) ELMo is the best performing for training/encoding time achieving significantly faster computational time for both intent classification data sets.

(iv) BERT performed on average for most of the data sets except for the JIRA issue tracking on sentiment analysis.

(v) Flair performed on average for most of the data sets except for the BBC news on category classification.

5.2.3. Long Sequence Length

Sentences with a longer length may not be able to use all the pre-trained embedding libraries easily.

For sentences with length more than 512, the following pre-trained embedding libraries cannot be used:

(i) RoBERTa
(ii) XLM

For sentence with length more than 1024, the following pre-trained embedding library cannot be used:

(i) GPT-2

The Flair package does not allow easy modification to the number of positional embedding for transformer architectures.

Appendix

Intent	Total	Train	Test	Intent	Total	Train	Test
chitchat	118	93	25	affirm	16	11	5
change	69	52	17	goodbye	12	10	2
cancel	59	46	13	get_info	12	10	2
greeting	29	26	3	view	14	10	4
deny	21	20	1	thanks	11	9	2
receive	24	20	4	renew	8	7	1
pay	19	15	4	-	-	-	-

Table 1: Data Distribution for Intents in Banking Agent

Category	Total	Train	Test
sport	511	413	98
business	510	404	106
politics	417	334	83
tech	401	332	69
entertainment	389	297	89

Table 2: Data Distribution for Categories in BBC News

Sentiment	Total	Train	Test
Love/Joy (0)	2458	1973	485
Surprise (1)	27	24	3
Anger (2)	342	276	66
Sadness (3)	622	495	127
Fear (4)	7	5	2
Neutral (5)	2236	1780	456

Table 3: Data Distribution for Sentiments in JIRA Issue Tracking

Intent	Total	Train	Test	Intent	Total	Train	Test
alarm_query	203	162	41	iot_hue_lighton	39	31	8
alarm_remove	123	98	25	iot_hue_lightup	142	114	28
alarm_set	297	238	59	iot_wemo_off	100	80	20
audio_volume_down	80	64	16	iot_wemo_on	80	64	16
audio_volume_mute	163	130	33	lists_createoradd	294	235	59
audio_volume_other	24	19	5	lists_query	369	295	74
audio_volume_up	145	116	29	lists_remove	330	264	66
calendar_query	1002	802	200	music_dislikeness	25	20	5
calendar_remove	533	426	107	music_likeness	204	163	41
calendar_set	1451	1161	290	music_query	276	221	55
cooking_query	6	5	1	music_settings	80	64	16
cooking_recipe	415	332	83	news_query	877	702	175
datetime_convert	97	78	19	play_audiobook	241	193	48
datetime_query	626	501	125	play_game	237	190	47
email_addcontact	90	72	18	play_music	1205	964	241
email_query	759	607	152	play_podcasts	379	303	76
email_querycontact	221	177	44	play_radio	551	441	110
email_sendemail	694	555	139	qa_currency	378	302	76
general_affirm	554	443	111	qa_definition	504	403	101
general_commandstop	320	256	64	qa_factoid	1052	841	211
general_confirm	550	440	110	qa_maths	166	133	33
general_dontcare	450	360	90	qa_stock	270	216	54
general_explain	684	547	137	recommend_events	324	259	65
general_greet	25	20	5	recommend_locations	257	206	51
general_joke	122	98	24	recommend_movies	112	90	22
general_negate	939	751	188	social_posts	541	433	108
general_praise	785	628	157	social_query	186	149	37
general_quirky	1088	870	218	takeaway_order	199	159	40
general_repeat	585	468	117	takeaway_query	215	172	43
iot_cleaning	172	138	34	transport_query	399	319	80
iot_coffee	198	158	40	transport_taxi	185	148	37
iot_hue_lightchange	224	179	45	transport_ticket	239	191	48
iot_hue_lightdim	126	101	25	transport_traffic	200	160	40
iot_hue_lightoff	246	197	49	weather_query	1062	849	213

Table 4: Data Distribution for Intents in Conversational Agent

Architecture	Framework	Library
USE (Large)	TensorFlow	TensorFlow Hub
BERT (Large)	TensorFlow	Bert-as-a-service
ELMo (Large)	PyTorch	Flair
XLNet (Large)	PyTorch	Flair
RoBERTa (Large)	PyTorch	Flair
XLM (en-2048)	PyTorch	Flair
GPT-2 (Medium)	PyTorch	Flair
Transformer-XL	PyTorch	Flair
Glove + Flair	PyTorch	Flair

Table 5: Embedding Architecture Framework and Library References