

Embeddings in Natural Language Processing

Quantitative Comparison on Downstream Tasks

Part B: Empirical Results

Ester Hlav

ECBM E6040 Neural Networks and Deep Learning
Columbia University

August 2019

Project Summary

Goal: Fair quantitative comparison of embeddings on standard NLP tasks and datasets

Motivation: Different embeddings advantageous in different applications -- need for an objective evaluation performance metrics (in terms of accuracy, training time, *etc.*)

Tools:

- **PyTorch** (data pre-processing, batching of embeddings, MLP, Logistic Regression)
- **SentEval** (dataset library for specific NLP tasks, *e.g.* sentiment analysis, question answering, language modeling, *etc.*)
- **Flair** (library with pre-trained embeddings, *e.g.* Word2Vec, GloVe, BERT, XLM, *etc.*)
- **Hyperopt** (optimization library for hyperparameter tuning, in particular used for Sequential Bayesian Optimization) and implementation of Hyperopt extension

Conclusion: Transformer-based contextualized embeddings architectures (*i.e.* *BERT*, *RoBERTa*) achieve state-of-the-art results and outperform static and other contextualized embeddings

Project Organization

1. Built a pipeline to connect **SentEval** (tasks, models) and **Flair** (embeddings library)
2. Bayesian Optimization for Sequential Hyperparameter Search using **Hyperopt** library
3. Extension of Hyperopt: **log based U/log based N** priors
4. Setup: training and **optimization** for all tasks/embeddings
5. **Results** reported in terms of accuracy, pre-processing time, training time, and architecture

1. SentEval, Flair and Hyperopt

SentEval - An Evaluation Toolkit for Universal Sentence Representations by Conneau *et al.* 2018

- a toolkit for evaluating the quality of universal sentence representations, but limited in generalization for different embeddings and new tasks with
 - standard *datasets* emerged from consensus in NLP community
 - standard *architectures* to evaluate the embeddings, yet with flexibility for fine-tuning

Flair - Contextual String Embeddings for Sequence Labeling by Akbik *et al.* 2018

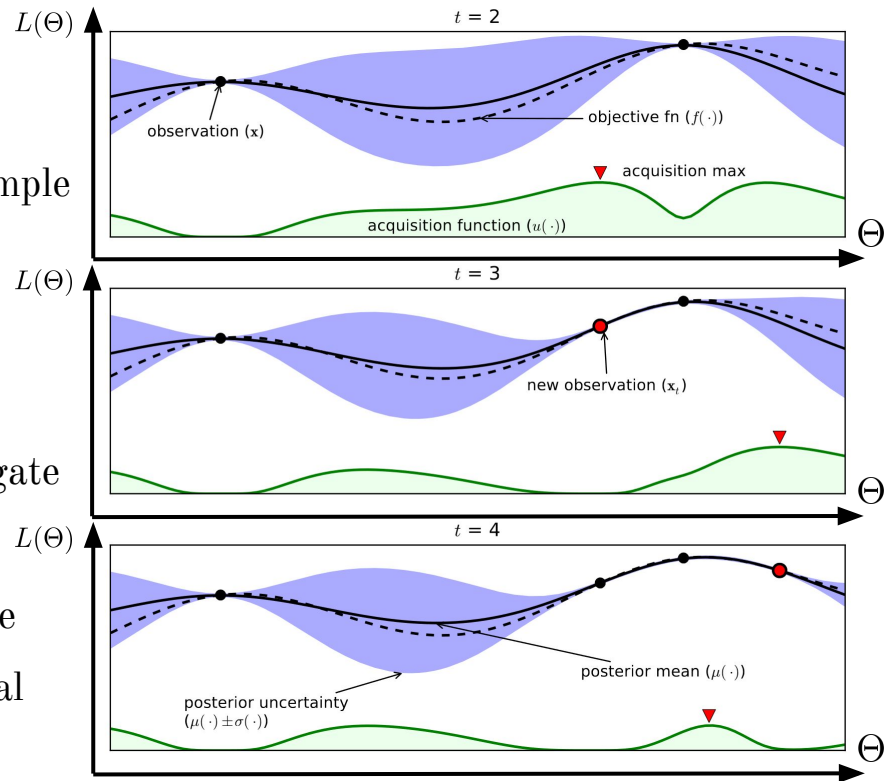
- a library for pre-trained embeddings with a uniform API for easy access and use

Hyperopt - A Python Library for Optimizing the Hyperparameters of ML Algorithms by Bergstra

- sequential model-based optimization library for function minimization

2. Sequential Bayesian Optimization: an overview

- Hyperparameter tuning - *Grid Search / Random Search* are naive and computationally expensive
- Bayesian Optimization approach - *sequentially* sample hyperparameters based on:
 - **surrogate** of the loss function (typically a Gaussian Process)
 - **acquisition function** derived from the surrogate (typically expected improvement)
- Acquisition function *argmax* gives the next sample
- Enables a more efficient search in high dimensional hyperparameter space



2. Sequential Bayesian Optimization: TPE and Hyperopt

- **Tree-structured Parzen Estimator (TPE) method** [3] creates a tree-like structure of Gaussian Processes (*surrogate*) and is driven by the expected improvement (*acquisition*)

$$\text{EI}_{y^*}(x) := \int_{-\infty}^{\infty} \max(y^* - y, 0) p_M(y|x) dy = \frac{\gamma y^* \ell(x) - \ell(x) \int_{-\infty}^{y^*} p(y) dy}{\gamma \ell(x) + (1 - \gamma) g(x)} \propto \left(\gamma + \frac{g(x)}{\ell(x)} (1 - \gamma) \right)^{-1}$$

- Tree-structure is fundamental for model comparison: SVM vs Decision Tree; LR vs MLP
- **Hyperopt** [4] : Python library for TPE

```
trials = Trials()
best = fmin(train_models,
            space=space,
            algo=tpe.suggest,
            max_evals=100,
            trials=trials)
```

```
from hyperopt import hp
space = hp.choice('classifier_type', [
    {
        'type': 'svm',
        'C': hp.lognormal('svm_C', 0, 1),
        'kernel': hp.choice('svm_kernel', [
            {'ktype': 'linear'},
            {'ktype': 'RBF', 'width': hp.lognormal('svm_rbf_width', 0, 1)},
        ]),
    },
    {
        'type': 'dtree',
        'criterion': hp.choice('dtree_criterion', ['gini', 'entropy']),
        'max_depth': hp.qlognormal('dtree_max_depth_int', 3, 1, 1),
        'min_samples_split': hp.qlognormal('dtree_min_samples_split', 2, 1, 1),
    },
])
```

3. Extension of Hyperopt: \log_b priors vs \log_e priors

- **Learning rate** and **batch size** are defined uniformly on power scales of 2 and 10, respectively, i.e. $10^{-1}, 10^{-2}, \dots, 10^{-5}$ and $2^5, 2^6, \dots, 2^{10}$

$$\log_n(n^i) = \frac{\ln(n^i)}{\ln(n)} = i \frac{\ln(n)}{\ln(n)} = i \implies \begin{cases} i \sim U(a, b) \implies e^{i \ln(n)} \sim \log_n U(a, b) \\ X \sim \log_n U(a, b) \implies \log_n(X) \sim U(a, b) \end{cases}$$

And thus: $n^i \sim \log_n U(a, b) \implies \log_n(n^i) = i \sim U(a, b)$

- Hyperopt only supports $\log_e U$ and $\log_e N$ distributions - not suitable for learning rate or batch size
- Extension of Hyperopt: **implemented $\log_b U$ and $\log_b N$ distributions**

$$\begin{aligned} X \sim \log N(\mu, \sigma^2) &\implies f_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln(x)-\mu}{\sigma}\right)^2} & F_X(x) &= \frac{1}{2} \operatorname{erfc}\left(-\frac{\ln(x)-\mu}{\sigma\sqrt{2}}\right) \\ X \sim \log_b N(\mu, \sigma^2) &\implies f_X(x) = \frac{1}{x\ln(b)\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\log_b(x)-\mu}{\sigma}\right)^2} & F_X(x) &= \frac{1}{2} \operatorname{erfc}\left(-\frac{\log_b(x)-\mu}{\sigma}\right) \end{aligned}$$

4. Setup: Embeddings

Embedding	Type	Trained on Task	Embedding size in Flair	Paper - date
Word2vec	Static	Language Model	300	Mikolov et al. - 2013
GloVe	Static	Co-Occurrence Model	100	Pennington et al. - 2014
ELMo	Contextualized	Language Model	3072	Peters et al. - 2018
BERT	Contextualized	Masked Language Model	3072	Devlin et al. - 2018
Flair	Contextualized	Language Model	2048	Akbik et al. - 2018
XLM	Contextualized	Cross-lingual Masked Language Model	4096	Devlin et al. - 2019
T-XL	Contextualized	Masked Language Model	3072	Dai et al. - 2019
XLNet	Contextualized	Autoregressive Language Model	2048	Yang et al. - 2019
RoBERTa	Contextualized	Masked Language Model	1024	Liu et al. - 2019

4. Setup: Tasks

Dataset	Topic	Training samples	Testing samples	NB Labels	Examples
CR	Product reviews	3.4K	0.6K	2	“We tried it out christmas night and it worked great .”
TREC	Question-type classification	6K	0.5K	6	“What are the twin cities ?” → LOC:city
SICK-E	Natural Language Inference	4.5K	4.9K	3	Premise + Hypothesis → Contradiction
SST-5	Sentiment analysis	8.5K	2.2K	5	“nothing about this movie works.”
MRPC	Paraphrase detection	4.1K	1.7K	2	Premise + Hypothesis → Paraphrase
SUBJ	Subjectivity status	8.5K	1.5K	2	“A movie that doesn’t aim too high , but doesn’t need to.”
MPQA	Opinion polarity	9.3K	1.7K	2	“don’t want” → neg ; “would like to tell” → pos
SST-2	Binary sentiment analysis	67K	1.8K	2	“Audrey Tautou has a knack for picking roles that magnify her [..]” → pos

5. Results: Accuracy (500 steps of TPE)

Emb \ Task	CR	TREC	SICK-E	SST-5	MRPC	SUBJ	MPQA	SST-2
Word2Vec ₃₀₀	80.95	83.00	78.49	45.88	72.87	92.20	87.68	81.99
GloVe ₁₀₀	69.14	76.00	76.07	40.14	72.29	89.80	86.36	75.56
ELMo ₃₀₇₂	84.66	93.40	80.84	47.56	74.03	93.80	88.87	85.39
BERT ₃₀₇₂	87.13	92.80	79.01	48.91	76.12	95.00	88.43	87.26
Flair-fast ₂₀₄₈	76.19	87.60	78.57	37.96	74.49	89.20	82.59	72.27
XLM ₄₀₉₆	80.25	88.2	56.69	--	73.39	91.27	87.62	--
T-XL ₃₀₇₂	79.01	87.8	56.69	--	70.43	92.60	90.07	--
XLNet ₂₀₄₈	82.01	90.40	82.16	--	70.15	90.80	89.0	--
RoBERTa ₁₀₂₄	90.65	93.60	75.12	--	74.20	97.00	89.06	--
SOTA **	86.30 *	96.10	84.50	52.40	80.40	95.50 *	93.30 *	89.50

* 10-Fold cross validation for SOTA, not consistent with 15%/15% val/test split for results

** task specific (state-of-the art results, cited from SentEval paper) vs generic trained embeddings

5. Results: Architectures - MLP vs. Logistic Regression

Emb \ Task	CR	TREC	SICK-E	SST-5	MRPC	SUBJ	MPQA	SST-2
Word2Vec ₃₀₀	MLP@1x120 sigmoid	MLP@2x100 ELU	MLP@3x30 ELU	MLP@2x20 tanh	MLP@2x80 ELU	MLP@2x140 ELU	MLP@2x60 ELU	MLP@2x20 tanh
GloVe ₁₀₀	MLP@2x20 ELU	MLP@2x120 ELU	MLP@3x10 tanh	MLP@2x40 tanh	MLP@2x120 ELU	MLP@1x120 sigmoid	LOGR	MLP@2x80 ELU
ELMo ₃₀₇₂	MLP@2x40 sigmoid	MLP@1x60 ELU	MLP@2x50 sigmoid	MLP@2x40 sigmoid	LOGR	LOGR	LOGR	MLP@3x80 ELU
BERT ₃₀₇₂	MLP@2x40 ELU	LOGR	MLP@2x40 tanh	MLP@3x40 ELU	LOGR	LOGR	LOGR	MLP@2x20 ELU
Flair-fast ₂₀₄₈	MLP@2x60 ELU	MLP@2x120 ELU	MLP@3x40 ELU	MLP@1x80 ELU	MLP@1x80 tanh	MLP@2x120 ELU	MLP@2x20 tanh	MLP@3x20 tanh
XLM ₄₀₉₆	MLP@1x100 ELU	LOGR	LOGR	--	LOGR	LOGR	MLP@2x40 sigmoid	--
T-XL ₃₀₇₂	MLP@2x140 ELU	MLP@2x20 tanh	LOGR	--	MLP@2x20 ELU	LOGR	MLP@2x80 ELU	--
XLNet ₂₀₄₈	MLP@1x40 sigmoid	LOGR	MLP@1x20 ELU	--	LOGR	MLP@2x140 tanh	MLP@2x20 sigmoid	--
RoBERTa ₁₀₂₄	MLP@1x100 sigmoid	MLP@2x120 ELU	MLP@2x20 tanh	--	LOGR	MLP@2x140 tanh	MLP@2x20 sigmoid	--

* MLP@1x120 sigmoid = architecture @ hidden layers x hidden units, activation function, ** MLP = Multi-Layer Perceptron, LR = Logistic Regression

5. Results: Pre-processing time

Emb \ Task	CR	TREC	SICK-E	SST-5	MRPC	SUBJ	MPQA	SST-2
Word2Vec ₃₀₀	2.0s	2.1s	6.5s	7.0s	7.4s	7.2s	1.7s	24.4s
GloVe ₁₀₀	2.2s	1.9s	6.4s	6.7s	7.4s	7.0s	1.7s	23.3s
ELMo ₃₀₇₂	4m 22s	3m 58s	13m 4s	13m 12s	14m 23s	13m 43s	3.46m	46m58s
BERT ₃₀₇₂	2m 17s	3m 7s	10m 26s	7m 13s	7m 27s	6m 27s	5min10s	37m42s
Flair-fast ₂₀₄₈	1m 59s	1m 43s	5m 30s	6m 34s	7m 20s	6m 51s	1.29s	22m27s
XLM ₄₀₉₆	38m 52s	30m 32s	11m 16s	--	1h 56m	1h 40m	13m 56s	--
T-XL ₃₀₇₂	2h 24m	2h 10m	49m 32s	--	19m 7s	18m 19s	14m 49s	--
XLNet ₂₀₄₈	2m 34s	3m 33s	12m 50s	--	11m 29s	6m 52s	6m 36s	--
RoBERTa ₁₀₂₄	1m 34s	3m 38s	9m 45s	--	10m 33s	5m 51s	4m 43s	--

5. Results: Training time

Emb \ Task	CR	TREC	SICK-E	SST-5	MRPC	SUBJ	MPQA	SST-2
Word2Vec ₃₀₀	10m 46s	31m 46s	24m 48s	43m 59s	18m 49s	42m 12s	32m 22s	7h44m
GloVe ₁₀₀	9m 56s	25m 21s	21m 31s	30m 0s	12m 41s	28m 18s	17m13s	6h10m
ELMo ₃₀₇₂	15m 6s	28m 53s	50m 31s	1h 35m	28m 5s	32m 21s	36m22s	10h31m
BERT ₃₀₇₂	14m 10s	27m 14s	36m 9s	1h 17m	25m 47s	31m 58s	28m42s	8h44m
Flair-fast ₂₀₄₈	24m 35s	1h 8m	44m 26s	1h 19m	36m 3s	1h 21m	53m28s	11h34m
XLM ₄₀₉₆	32m 32s	29m 20s	1h 21m	--	26m 35s	1h 20m	1h 17m	--
T-XL ₃₀₇₂	1h 38m	42m 6s	1h 16m	--	2h 29m	1h 17m	1h 50m	--
XLNet ₂₀₄₈	11m 5s	16m 38s	27m 56s	--	17m 34s	48m 12s	35m 10s	--
RoBERTa ₁₀₂₄	17m 7s	1h 26m	1h 0m	--	39m 13s	1h 26m	1h 2m	--

5. Results: Accuracy for Stacked Embeddings

Emb \ Task	CR	TREC	SICK-E	SST-5	MRPC	SUBJ
Word2vec + GloVe (400)	77.07	86.00	79.62	44.71	72.93	91.67
ELMo + GloVe (3172)	84.83	94.20	81.65	48.6	75.36	94.53
BERT + GloVe (3172)	87.65	94.40	80.72	48.19	75.88	94.87
ELMo + Word2vec (3172)	82.89	93.20	82.16	46.88	75.07	93.33
BERT + Word2vec (3172)	88.71	94.40	80.82	48.19	72.70	95.67
ELMo + BERT (6144)	86.60	94.60	82.18	48.24	75.71	95.80
ELMo + BERT+Word2vec (6544)	86.77	93.00	82.36	50.86	74.72	96.47
SOTA **	86.30 *	96.10	84.50	52.40	80.40	95.50 *

References

- [1] **SentEval: An Evaluation Toolkit for Universal Sentence Representations** by Alexis Conneau and Douwe Kiela (Facebook AI Research, 2018)
- [2] **Flair: Contextual String Embeddings for Sequence Labeling** by Alan Akbik, Duncan Blythe and Roland Vollgraf (Zalando Research 2018)
- [3] **Algorithms for Hyper-Parameter Optimization** by James Bergstra, Remi Bardenet, Yoshua Bengio, Balazs Kegl (NIPS 2011)
- [4] **Hyperopt** by James Bergstra , Dan Yamins, David D. Cox (Scipy conf. 2013)

Annex: Results and Tasks from SentEval Paper

Model	MR	CR	SUBJ	MPQA	SST-2	SST-5	TREC	MRPC	SICK-E
<i>Representation learning (transfer)</i>									
GloVe LogReg	77.4	78.7	91.2	87.7	80.3	44.7	83.0	72.7/81.0	78.5
GloVe MLP	77.7	79.9	92.2	88.7	82.3	45.4	85.2	73.0/80.9	79.0
fastText LogReg	78.2	80.2	91.8	88.0	82.3	45.1	83.4	74.4/82.4	78.9
fastText MLP	78.0	81.4	92.9	88.5	84.0	45.1	85.6	74.4/82.3	80.2
SkipThought	79.4	83.1	93.7	89.3	82.9	-	88.4	72.4/81.6	79.5
InferSent	81.1	86.3	92.4	90.2	84.6	46.3	88.2	76.2/83.1	86.3
<i>Supervised methods directly trained for each task (no transfer)</i>									
SOTA	83.1 ¹	86.3 ¹	95.5 ¹	93.3 ¹	89.5 ²	52.4 ²	96.1 ²	80.4/85.9 ³	84.5 ⁴

Table 3: Transfer test results for various baseline methods. We include supervised results trained directly on each task (no transfer). Results ¹ correspond to AdaSent (Zhao et al., 2015), ² to BLSTM-2DCNN (Zhou et al., 2016), ³ to TF-KLD (Ji and Eisenstein, 2013) and ⁴ to Illinois-LH system (Lai and Hockenmaier, 2014).

	name	N	task	C	examples	label(s)
→	MR	11k	sentiment (movies)	2	"Too slow for a younger crowd , too shallow for an older one."	neg
→	CR	4k	product reviews	2	"We tried it out christmas night and it worked great ."	pos
→	SUBJ	10k	subjectivity/objectivity	2	"A movie that doesn't aim too high , but doesn't need to."	subj
→	MPQA	11k	opinion polarity	2	"don't want"; "would like to tell";	neg, pos
→	TREC	6k	question-type	6	"What are the twin cities ?"	LOC:city
→	SST-2	70k	sentiment (movies)	2	"Audrey Tautou has a knack for picking roles that magnify her [..]"	pos
→	SST-5	12k	sentiment (movies)	5	"nothing about this movie works."	0

Table 1: **Classification tasks.** C is the number of classes and N is the number of samples.


	name	N	task	output	premise	hypothesis	label
	SNLI	560k	NLI	3	"A small girl wearing a pink jacket is riding on a carousel."	"The carousel is moving."	entailment
→	SICK-E	10k	NLI	3	"A man is sitting on a chair and rubbing his eyes"	"There is no man sitting on a chair and rubbing his eyes"	contradiction
	SICK-R	10k	STS	[0, 5]	"A man is singing a song and playing the guitar"	"A man is opening a package that contains headphones"	1.6
	STS14	4.5k	STS	[0, 5]	"Liquid ammonia leak kills 15 in Shanghai"	"Liquid ammonia leak kills at least 15 in Shanghai"	4.6
→	MRPC	5.7k	PD	2	"The procedure is generally performed in the second or third trimester."	"The technique is used during the second and, occasionally, third trimester of pregnancy."	paraphrase
	COCO	565k	ICR	sim		"A group of people on some horses riding through the beach."	rank

Table 2: **Natural Language Inference and Semantic Similarity tasks.** NLI labels are contradiction, neutral and entailment. STS labels are scores between 0 and 5. PD=paraphrase detection, ICR=image-caption retrieval.