Benchmarking Language-agnostic Intent Classification for Virtual Assistant Platforms

Gengyu Wang*, Cheng Qian*, Lin Pan, Haode Qi
Ladislav Kunc, Saloni Potdar
IBM Watson
{gengyu, cheng.qian, haode.qi, lada}@ibm.com
potdars@us.ibm.com

Abstract

Current virtual assistant (VA) platforms are beholden to the limited number of languages they support. Every component, such as the tokenizer and intent classifier, is engineered for specific languages in these intricate platforms. Thus, supporting a new language in such platforms is a resource-intensive operation requiring expensive re-training and re-designing. In this paper, we propose a benchmark for evaluating language-agnostic intent classification, the most critical component of VA platforms. To ensure the benchmarking is challenging and comprehensive, we include 29 public and internal datasets across 10 low-resource languages and evaluate various training and testing settings with consideration of both accuracy and training time. The benchmarking result shows that Watson Assistant, among 7 commercial VA platforms and pre-trained multilingual language models (LMs), demonstrates close-to-best accuracy with the best accuracy-training time trade-off.

1 Introduction

Virtual assistant (VA) platforms that enable customers to train and deploy their chatbots have seen growing demand in recent years. This has attracted significant interest from both industry and academia to develop new machine learning (ML) models and datasets for these task-oriented dialog systems. In a dialog system, intent classification as the core component identifies user intent of a user's utterance so that the system can respond appropriately by triggering dialog nodes in predefined dialog trees.

Although there has been a lot of exploration around implementing intent classification models for English, not much work has been extended to low-resource languages. Due to the vast number of world languages, it is not trivial for an enterprise VA platform to support its global customers.

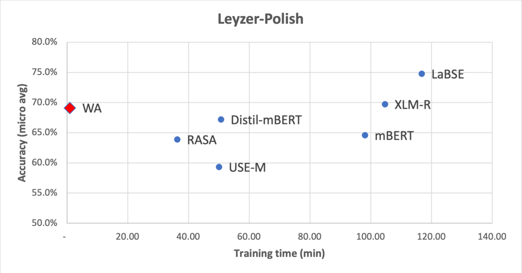

Figure 1: **Training time vs. accuracy on Leyzer (Polish) dataset for all models.** Full train set and test set are used. All methods, except WA and RASA, are trained using GPU. WA offers the best trade-off between training time and accuracy.

.

Currently, VA platforms usually take the following two methods to handle unsupported languages:

- Use without modification: VA platforms usually include language-specific components for each supported language, such as language models (LMs), tokenizers, part-of-speech taggers. Directly applying them to unsupported languages could dramatically hurt the performance. Several preprocessing steps, such as contraction handling, stemming, lemmatization, can produce unpredictable behavior when used with an unsupported language.
- Using translation: Translating unsupported language to the supported ones is an intuitive solution. However, low-quality translation can result in classification errors. Also, there is additional round-trip time and cost when including a translation component. In enterprise scenarios, this may lead to the deployed solution being more expensive.

While we see an increasing need to develop such a framework for non-English languages, developing a language-agnostic modeling paradigm that can serve a large number of languages carries important business applications as language-specific

[*] Equal contributions from the corresponding authors.

Proceedings of the Workshop on Multilingual Information Access (MIA), pages 69 - 76
July 15, 2022 ©2022 Association for Computational Linguistics

solutions are difficult and expensive to maintain.

In addition to the above challenge, there are two more considerations while developing such language-agnostic VAs. Firstly, due to the high cost of curating training data for multiple languages, real-world intent detection models usually must be able to train and perform well on few-shot training datasets. Secondly, the training time is also a critical factor to be considered. Given a commercial VA platform, authoring an assistant for a specific domain still takes dozens of hours, and the whole process involves hundreds to thousands of times of iteration. As model training is called in each iteration, keeping training time in the range of seconds is crucial.

In this paper, we conduct a comprehensive and robust evaluation of several modeling approaches across multiple low-resource languages in real-world settings and focus on their accuracy, training time, and computation requirements. We benchmark two commercial VA platforms, including IBM Watson Assistant (WA)[1], RASA[2], [3] and five representative multilingual LMs with different model sizes and architectures.

To benchmark the models on as many low-resource languages as possible, we include 9 public datasets from the research community across 5 languages and curate 20 real-world datasets from a commercial VA platform across 7 languages and 9 domains in the evaluation. We also create the few-shot version of these datasets to evaluate the models' performance on small datasets. Additionally, after observing the close accuracy results among the models, we follow Arora et al. (2020) and Qi et al. (2021) to create the TF*IDF and jaccard based difficult testing set to differentiate them better. [4]

Overall, our benchmark generates about 1000 data points, including accuracy and training time in default, few-shot training, and difficult testing settings. While LaBSE (Feng et al., 2020) produces the highest accuracy in almost all settings, along with all other LMs, their training time is too long to be used in commercial production. On the contrary, Watson Assistant achieves the best accuracy-training trade-off by achieving the competitive accuracy and consistent short training time of less than one minute. Figure1 demonstrates this comparison on one of the benchmarking datasets.

2 Related Work

Multilingual Intent Classification A line of work has studied commercial conversational AI services (Braun et al., 2017; Arora et al., 2020; Liu et al., 2019) and pretrained LMs (Casanueva et al., 2020; Larson et al., 2019; Arora et al., 2020; Bunk et al., 2020; Qi et al., 2021) on intent classification task in English. Li et al. (2020) built a benchmark on their proposed multilingual dataset, but only evaluated two multilingual pretrained LMs. Comparing to previous work, we conduct a comprehensive benchmarking study by evaluating seven conversational AI services or LMs on 9 public datasets and 20 internal datasets covering 10 languages.

Resource Efficiency When applying a VA system in a production environment, the training cost of the model is an important consideration. Most of the prior work only focuses on the accuracy of models but does not evaluate the training time they require given the same training resources. Casanueva et al. (2020) only compare three models. In our work, we compared the training time of the 7 models in addition to their accuracy.

Few-shot Training Li et al. (2020) and Casanueva et al. (2020) conducted zero-shot or few-shot training to resemble the training process of a commercial VA system, but did not conduct a comprehensive evaluation.

3 Benchmarking

In this section, we firstly introduce the three benchmarking settings in our experiments, and then describe the VA platforms and models we evaluate. Lastly, we present and analyze the results.

3.1 Experimental Settings

Standard Training This corresponds to the standard benchmark setting where we train on the full train set and evaluate on the full test set.

Few-shot Training In a real production environment, the dialog system is usually fine-tuned for specific topics with scarce labeled data. Therefore, we propose a few-shot setting where we create five few-shot subsets by sampling 5, 15, and 30 examples per intent class from each of the datasets.

Testing with difficult examples In experiments with the standard train/test splits in the data, we

[1] https://www.ibm.com/products/watson-assistant

[2] https://rasa.com

[3] We do not include other commercial VA providers due to the benchmarking prohibition in their terms of use.

[4] The difficult test sets inherit the same licenses and terms of original datasets. https://github.com/posuer/benchmark-multilingual-intent-classification

observe that most models can achieve high accuracy. One of the possible reasons could be that the semantic and lexical distribution of test and train set are very similar. To better evaluate and compare the performance of the models, we create difficult test subsets with selected examples from the original test set.

We use a similar setup as described in Arora et al. (2020) and Qi et al. (2021) to create two difficult test subsets, ***TF*IDF*** and ***jaccard***, for each of the datasets. Specifically, we firstly concatenate all tokenized training examples and transform them into a vector space of TF*IDF scores (Salton and McGill, 1986) (count scores for jaccard), then use the initialized TF*IDF (or jaccard) vectorizer to transform each testing example and calculate the cosines distance (or jaccard score). For each intent class, 5 farthest testing examples are selected to build the difficult subset.

3.2 Models

In this work, we benchmark 7 different intent classification models or services. Among them, 5 are multilingual pre-trained LMs, and the remaining 2 are commercial VA platforms, IBM Watson Assistant and RASA.

Watson Assistant provides language-specific models for 13 popular languages, and a language-agnostic model that responds to all other languages. We focus on the latter for the experiments in the paper. We use public API to train and evaluate the model. For training time, we measure the round-trip latency from sending the training request until we receive the status that the model is trained and available for serving.

RASA is an automated dialogue framework that allows incorporating various text processing tools and pre-trained LMs. In our experiment, we follow the default setting that feeds count-based features to an intent classifier, DIET (Bunk et al., 2020). We fine-tune the model with each of the dataset for 100 epochs.

We also evaluate following multilingual pretrained LMs: multilingual BERT (**mBERT$_{\textbf{base-cased}}$**) (Devlin et al., 2018) , **Distil-mBERT$_{\textbf{base-cased}}$**[5] (Sanh et al., 2019), **XLM-R$_{\textbf{base}}$** (Conneau et al., 2019), USE-Multilingual (**USE-M$_{\textbf{large}}$**)[6] (Yang et al., 2019), and **LaBSE**[7] (Feng et al., 2020).

For mBERT, Distil-mBERT, XLM-R, and LaBSE model, we add a softmax classifier on top of the [CLS] token and fine-tune all layers. We use AdamW (Loshchilov and Hutter, 2018) with 0.01 weight decay and a linear learning rate scheduler. We choose a batch size of 32, epochs of 30 [8], max sequence length 128 and learning rate warmup for the first 50 iterations, peaking at 0.00005.

For USE-M, we train a softmax layer on top of the sentence representation and fine-tune all layers for 100 epochs. A learning rate of 0.01 and batch size of 32 are used for all train set variants. All models are trained or fine-tuned with a single CPU core or a single K80 GPU.

4 Benchmarking Datasets

Based on the availability and quality of public intent classification datasets, we propose our benchmark consisting of 9 public datasets across 5 languages, including *Hindi, Polish, Russian, Thai & Turkish*, and 20 internal datasets across 7 languages and 9 domains. A summary of dataset statistics and preprocessing details are provided in Table 1.

MTOP (Li et al., 2020) is an almost parallel multilingual dataset covering 6 languages and 11 domains (e.g., weather, calling, alarm, etc.). English utterances and annotations are generated by crowdsourced workers and annotators and then human translated to other languages. We use the Hindi and Thai subset of MTOP in our experiments.

Multilingual ATIS (MultiATIS) (Upadhyay et al., 2018) contains airline travel inquiries in Hindi and Turkish, which are manually translated from the original English ATIS dataset. In our experiments, utterances with more than one intent label (concatenated by white space) are expanded into multiple records, one for each intent label.

Leyzer (Sowański and Janicki, 2020) is a multilingual chatbot dataset which contains a large number of intents and covers 20 domains such as email, contacts, etc. This corpus is generated with a grammar-based approach. We use the Polish subset of Leyzer in our experiments.

[5] https://huggingface.co/distilbert-base-multilingual-cased

[6] https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/1

[7] https://huggingface.co/sentence-transformers/LaBSE

[8] We experimented with both 30 and 40 epochs settings and present the results of 30 epochs as it produced compatible results with shorter training time.

Public Datasets				
Language	Dataset	Train	Test	Intent Types
Hindi	MTOP	11,251	2,789	113
	MultiATIS	1,565	909	16
Polish	Leyzer	6,366	991	168
Russian	Chatbot-ru	5,517	1,380	79
	PSTU	1,082	271	7
Thai	MultiTOD	1,928	1,692	10
	MTOP	10,622	2,765	110
Turkish	Chatbot-tr	761	191	24
	MultiATIS	628	725	15

Internal Datasets				
Language	Domain	Train	Test	Intent Types
Finnish	COVID-19	1045	262	60
Greek	COVID-19	198	50	15
	Insurance	281	71	28
Norwegian Bokmål	banking	223	56	13
	customer service	304	76	18
	telco	317	80	19
	utilities	176	44	10
Norwegian Nynorsk	banking	224	57	13
	customer service	300	76	18
	teleco	350	88	21
	utilities	176	45	10
Polish	general	795	199	43
Russian	banking	1364	342	92
	COVID-19	1392	349	122
	general	623	158	46
Swedish	banking	211	54	13
	customer care	294	74	18
	teleco	345	87	21
	utilities	172	43	10
Turkish	customer care	184	46	9

Table 1: **Dataset Statistics.** Prepossessing has been done on all datasets (details in Datasets Section). Numbers reflect the actual size used in our experiment.

Multilingual Task Oriented Data (MultiTOD) (Schuster et al., 2018) contains annotated utterances in English, Spanish, and Thai across the topics like weather, alarm, etc. English utterances are first produced by native English speakers and labelled by annotators, then translated into Spanish and Thai by native speakers of the target languages.

Chatbot-ru (Russian)[9], **PSTU** (Russian)[10], and **Chatbot-tr** (Turkish)[11] are three intent classification datasets publicly released on Github. For each of the three datasets, we split them into train and test set in a stratified fashion, using intent type as the class labels. Intents with only one utterance are discarded.

Internal Datasets To enable benchmarking with real-world data and evaluate the models in more languages, we curate 20 internal datasets in 8 languages across 9 domains from users of a virtual assistant platform. Different from the public datasets, these internal datasets are used in enterprise production environment to train real-world virtual assistants and serve customers in domains including banking, COVID-19 and telecommunication. The detailed size, domain and language information of these datasets are listed in Table 1.

Dataset Preprocessing We conducted following preprocessing for above datasets. We firstly transform all utterances in the train sets into lower case and perform deduplication. After this process, we use the original data without duplication for experiments. Test sets of Leyzer and MultiATIS contain utterances with intents unseen in the training data. We keep such utterances in the test sets to ensure a fair comparison with others' work on these datasets.

5 Results and Analysis

Standard Training Setting Table 2 shows results of WA, RASA, and 5 pretrained LMs on 9 public datasets across 5 languages. We train on the full train sets and report results on the full test sets, measured by accuracy. In Table 3, we present the results for internal datasets in the same setting. Overall, LaBSE performs best among the 7 models on both public and internal datasets. However, considering that fine-tuning large LMs, such as LaBSE, requires significantly more computational resources, WA makes a great trade-off by achieving 84.8% average accuracy that is only 4.5% lower than LaBSE.

Few-shot Training Setting In table 4, we present the accuracy of models trained on the full set and three subsets consisting of 5/15/30 examples per intent type and evaluated on the full test set. We obtain the accuracy per language by averaging the accuracy of all datasets in that language.

Among the models, WA shows an advantage over RASA and mBERT in the few-shot setting of 5 examples per intent based on the average accuracy across the 5 languages in Table 4. For all models, we observe significant drop in accuracy in *5 examples per intent* train set compared to the *full* train set, decreasing from about 80% to about 60% on average. This shows that the limitation in

[9] https://github.com/Koziev/chatbot/blob/master/data/intents.txt

[10] https://github.com/Perevalov/pstu_assistant/blob/master/data/data.txt

[11] https://github.com/zerocodenlu/chatbot-tr/blob/master/data/nlu/intent_data.csv

Models	Hindi		Polish	Russian		Thai		Turkish		Average
	MTOP	MultiATIS	Leyzer	Chatbot-ru	PSTU	MultiTOD	MTOP	Chatbot-tr	MultiATIS	
WA	90.7	87.6	69.1	81.5	79.7	96.6	89.8	80.6	87.2	84.8
RASA	88.5	88.3	64.0	66.7	75.3	96.6	89.5	81.7	88.3	82.1
mBERT	92.9	90.0	64.6	81.9	79.7	97.1	92.5	77.5	85.7	84.6
XLM-R	94.3	89.9	69.7	86.1	81.5	96.9	94.2	84.8	89.1	87.4
USE-M	75.4	81.6	59.3	84.5	80.8	97.4	93.5	83.2	84.8	82.3
LaBSE	94.4	91.6	74.8	87.2	83.8	97.4	94.5	87.4	92.6	89.3
Distil-mBERT	92.5	89.1	67.2	79.4	80.1	97.2	92.0	78.5	87.2	84.8

Table 2: **Accuracy on 9 public datasets for WA, RASA, and 5 pretrained LMs.** Each model is trained on full train set and evaluated on full test set.

Models	Finnish	Greek	Norwegian Bokmål	Norwegian Nynorsk	Polish	Russian	Swedish	Turkish	Average
WA	66.9	70.2	74.8	73.9	68.3	77.6	75.0	80.6	73.4
RASA	64.6	66.1	75.7	73.8	62.3	70.0	68.5	77.8	69.9
mBERT	71.9	65.1	74.6	73.1	84.4	78.3	78.6	75.0	75.1
XLM-R	75.8	84.2	86.6	82.0	79.4	79.4	85.9	75.0	81.0
USE-M	65.8	56.1	66.6	64.6	78.9	78.3	70.2	72.2	69.1
LaBSE	78.1	85.9	89.9	86.6	87.4	81.9	88.8	86.1	85.6
Distil-mBERT	69.6	71.2	73.8	67.4	80.9	76.1	73.4	72.2	73.1

Table 3: **Macro accuracy over internal datasets for each language.** Models are trained on the full train set of each dataset and evaluated on the full test set. Averaged accuracy at the last row is the simple averaging.

	Hindi				**Polish**				**Russian**				**Thai**				**Turkish**				**Average**			
Models	5	15	30	full	5	15	30	full	5	15	30	full	5	15	30	full	5	15	30	full	5	15	30	full
WA	50.4	60.7	76.0	89.1	60.1	67.2	69.6	69.1	51.3	64.6	71.8	80.6	63.3	77.4	82.9	93.2	61.9	72.5	76.5	83.9	57.4	68.5	75.4	83.2
RASA	29.6	46.3	61.2	88.4	47.1	58.9	61.0	63.9	32.7	44.6	51.3	71.0	43.6	62.3	73.2	93.0	43.4	64.0	69.0	85.0	39.3	55.2	63.1	80.2
mBERT	62.3	75.6	80.4	91.5	61.4	66.9	68.2	64.6	48.3	58.4	67.8	80.8	56.7	81.0	85.8	94.8	48.6	69.3	75.6	81.6	55.4	70.2	75.6	82.6
XLM-R	64.8	78.8	82.4	92.1	66.7	73.6	73.8	69.7	52.8	67.2	73.9	83.8	72.0	46.4	47.3	95.6	55.7	73.3	82.0	87.0	62.4	67.8	71.9	85.6
USE-M	24.6	37.3	48.3	78.5	60.0	62.2	61.0	59.3	63.5	67.6	72.8	82.7	81.2	87.5	89.8	95.4	75.2	80.8	84.6	84.0	60.9	67.1	71.3	80.0
LaBSE	74.8	85.0	89.5	93.0	69.9	75.3	74.9	74.8	60.2	69.5	74.7	85.5	73.9	88.1	91.3	96.0	66.7	81.1	84.6	90.0	69.1	79.8	83.0	87.8
Distil-mBERT	54.9	69.4	78.4	90.8	57.6	65.1	65.5	67.2	32.9	57.4	68.8	79.7	54.3	78.8	84.7	94.6	46.3	63.8	74.8	82.9	49.2	66.9	74.4	83.0

Table 4: **Few-shot setting on public datasets with full test set.** Accuracy for each language is averaged over all datasets for that language. Second row corresponds to 5, 15, 30 & all examples per intent in the train set.

	Hindi			**Polish**			**Russian**			**Thai**			**Turkish**			**Average**		
Models	full	jaccard	tf*idf	full	jaccard	tf*idf	full	jaccard	tf*idf	full	jaccard	tf*idf	full	jaccard	tf*idf	full	jaccard	tf*idf
WA	89.1	55.6	49.5	69.1	68.0	68.8	80.6	67.5	64.0	93.2	70.4	60.9	83.9	61.9	57.9	83.2	64.7	60.2
RASA	88.4	56.7	50.0	63.9	59.5	60.2	71.0	64.5	58.2	93.0	70.2	67.5	85.0	62.3	60.9	80.2	62.7	59.4
mBERT	91.5	67.8	62.4	64.6	65.7	66.5	80.8	76.5	73.1	94.8	77.8	71.3	81.6	59.4	52.8	82.6	69.4	65.2
XLM-R	92.1	68.3	62.6	69.7	71.1	70.6	83.8	78.8	76.4	95.6	80.7	76.2	87.0	72.0	67.1	85.6	74.2	70.6
USE-M	78.5	40.1	34.2	59.3	59.2	58.5	82.7	76.0	76.0	95.4	78.9	73.1	84.0	61.2	58.4	80.0	63.1	60.0
LaBSE	93.0	72.2	68.7	74.8	76.8	76.8	85.5	79.0	78.3	96.0	82.2	75.2	90.0	79.1	75.6	87.8	77.8	74.9
Distil-mBERT	90.8	63.1	57.0	67.2	66.2	65.5	79.7	69.7	70.4	94.6	78.3	72.1	82.9	59.2	55.2	83.0	67.3	64.0

Table 5: **Difficult test accuracy comparison on public datasets.** Accuracy for each language is averaged over all datasets in the corresponding language. *full*, *jaccard*, and *tf*idf* refer to full, jaccard and tf*idf test sets accordingly.

		Hindi		Polish	Russian		Thai		Turkish	
Models	Resource	MTOP	MultiATIS	Leyzer	Chatbot-ru	PSTU	MultiTOD	MTOP	Chatbot-tr	MultiATIS
WA	CPU	**0.64**	**0.34**	**0.92**	**0.79**	**0.45**	**0.40**	**1.03**	**0.38**	**0.45**
RASA	CPU	66.49	8.32	36.34	35.99	15.48	7.11	73.61	2.12	3.08
mBERT	GPU	175.89	24.70	98.11	83.17	16.44	29.12	160.35	11.45	9.50
XLM-R	GPU	185.41	25.85	104.68	90.82	17.92	31.69	174.50	12.63	10.46
USE-M	GPU	103.46	19.94	50.06	40.44	14.70	14.73	72.84	6.95	7.39
LaBSE	GPU	207.02	28.77	116.80	101.58	19.98	35.48	195.59	14.01	11.62
Distil-mBERT	GPU	90.02	12.55	50.75	44.11	8.73	15.46	85.34	6.09	5.04

Table 6: **Training time.** Macro averaged training time in minutes and resource types while training on full train set and evaluating on full test set for each public dataset.

	5 ex/intent		30 ex/intent		full	
Models	Time	Acc.	Time	Acc.	Time	Acc.
HINDI						
MTOP						
WA	0.44	45.7%	0.56	75.1%	0.64	90.7%
RASA	2.15	21.0%	10.90	54.6%	66.49	88.5%
mBERT	8.02	51.0%	33.44	82.1%	175.89	92.9%
XLM-R	8.45	68.4%	35.21	88.4%	185.41	94.3%
USE-M	6.80	18.4%	21.32	44.1%	103.46	75.4%
LaBSE	9.40	73.6%	39.22	88.8%	207.02	94.4%
Distil-mBERT	4.10	46.2%	17.08	78.5%	90.02	92.5%
MultiATIS						
WA	0.37	55.1%	0.34	76.9%	0.34	87.6%
RASA	0.38	38.1%	1.35	67.4%	8.32	88.3%
mBERT	0.98	73.6%	3.85	78.8%	24.70	90.0%
XLM-R	1.03	61.1%	4.10	76.5%	25.85	89.9%
USE-M	2.91	30.8%	4.93	52.5%	19.94	81.6%
LaBSE	1.15	76.0%	4.55	90.2%	28.77	91.6%
Distil-mBERT	0.50	63.6%	1.98	78.4%	12.55	89.1%
POLISH						
Leyzer						
WA	0.44	60.1%	0.70	69.6%	0.92	69.1%
RASA	2.62	47.2%	12.83	60.8%	36.34	64.0%
mBERT	11.91	61.4%	43.71	68.2%	98.11	64.6%
XLM-R	13.08	66.7%	48.18	73.8%	104.68	69.7%
USE-M	7.62	60.0%	23.43	61.0%	50.06	59.3%
LaBSE	14.55	69.9%	53.32	74.9%	116.80	74.8%
Distil-mBERT	6.36	57.6%	23.18	65.5%	50.75	67.2%
RUSSIAN						
Chatbot-ru						
WA	0.34	52.4%	0.48	73.2%	0.79	81.5%
RASA	2.06	22.6%	10.57	42.8%	35.99	66.7%
mBERT	6.04	50.1%	28.53	70.2%	83.17	81.9%
XLM-R	6.64	52.9%	31.27	76.2%	90.82	86.1%
USE-M	4.82	67.6%	14.89	75.6%	40.44	84.5%
LaBSE	7.39	63.2%	34.91	79.6%	101.58	87.2%
Distil-mBERT	3.23	39.2%	15.16	68.9%	44.11	79.4%
PSTU						
WA	0.29	50.2%	0.35	70.5%	0.45	79.7%
RASA	0.56	42.8%	3.27	59.8%	15.48	75.3%
mBERT	0.62	46.5%	3.16	65.3%	16.44	79.7%
XLM-R	0.70	52.8%	3.47	71.6%	17.92	81.5%
USE-M	2.64	59.4%	4.48	70.1%	14.70	80.8%
LaBSE	0.81	57.2%	3.89	69.7%	19.98	83.8%
Distil-mBERT	0.34	26.6%	1.69	68.6%	8.73	80.1%

	5 ex/intent		30 ex/intent		full	
Models	Time	Acc.	Time	Acc.	Time	Acc.
THAI						
MultiTOD						
WA	0.40	77.3%	0.38	90.9%	0.40	96.6%
RASA	0.32	65.7%	1.18	90.1%	7.11	96.6%
mBERT	0.81	62.7%	4.18	92.1%	29.12	97.1%
XLM-R	0.90	82.2%	4.58	93.9%	31.69	96.9%
USE-M	2.66	90.0%	4.02	94.0%	14.73	97.4%
LaBSE	1.02	77.1%	5.12	94.3%	35.48	97.4%
Distil-mBERT	0.44	69.9%	2.23	91.2%	15.46	97.2%
MTOP						
WA	0.41	49.3%	0.47	75.0%	1.03	89.8%
RASA	2.43	21.5%	11.61	56.2%	73.61	89.5%
mBERT	7.57	50.6%	31.67	79.6%	160.35	92.5%
XLM-R	8.30	61.8%	34.76	0.8%	174.50	94.2%
USE-M	5.55	72.5%	15.99	85.5%	72.84	93.5%
LaBSE	9.26	70.6%	38.66	88.3%	195.59	94.5%
Distil-mBERT	4.04	38.8%	16.90	78.1%	85.34	92.0%
TURKISH						
Chatbot-tr						
WA	0.42	56.0%	0.36	74.9%	0.38	80.6%
RASA	0.44	39.3%	1.41	67.5%	2.12	81.7%
mBERT	1.85	51.3%	7.45	73.3%	11.45	77.5%
XLM-R	2.03	60.7%	8.19	83.2%	12.63	84.8%
USE-M	2.92	72.8%	5.37	83.2%	6.95	83.2%
LaBSE	2.26	68.1%	9.12	83.8%	14.01	87.4%
Distil-mBERT	0.98	48.7%	3.96	72.8%	6.09	78.5%
MultiATIS						
WA	0.35	67.7%	0.40	78.1%	0.45	87.2%
RASA	0.33	47.6%	0.78	70.5%	3.08	88.3%
mBERT	0.84	45.9%	2.77	77.9%	9.50	85.7%
XLM-R	0.93	50.8%	3.06	80.7%	10.46	89.1%
USE-M	2.68	77.7%	3.61	85.9%	7.39	84.8%
LaBSE	1.05	65.4%	3.40	85.5%	11.62	92.6%
Distil-mBERT	0.45	44.0%	1.47	76.8%	5.04	87.2%
MACRO AVERAGE						
WA	**0.38**	57.1%	**0.45**	76.0%	**0.60**	84.8%
RASA	1.26	38.4%	5.99	63.3%	27.62	82.1%
mBERT	4.29	54.8%	17.64	76.4%	67.64	84.6%
XLM-R	4.67	61.9%	19.20	71.7%	72.66	87.4%
USE-M	4.29	61.0%	10.89	72.5%	36.72	82.3%
LaBSE	5.21	**69.0%**	21.35	**83.9%**	81.21	**89.3%**
Distil-mBERT	2.27	48.3%	9.30	75.4%	35.34	84.8%

Table 7: **Training time (minutes) and accuracy on full test set for each public datasets.** *5 ex/intent*, *30 ex/intent*, and *full* refer to 5 examples per intent, 30 examples per intent, and full train set accordingly.

training data brings a huge challenge for all models, and the few-shot train sets provide a better testbed for the ability to handle such situations, which is crucial for real-world VA systems.

Difficult Test Setting In Table 2, we observe that most models can achieve about 90% accuracy. To better compare these models, we evaluate them on the difficult test sets, *jaccard* and *tf*idf*. Results are presented in Table 5. In this setting, we observe a significant gap between the original test set and difficult sets for all models. Among all the models, mBERT performs the best as it shows the least accuracy drop. However, WA still stands on top considering the trade-off between training time and accuracy, which will be further explained below.

5.1 Training Time vs. Accuracy Trade-off

We record the training time per dataset along with the resource requirement and accuracy in Table 6. Pretrained LMs require significantly longer training time compared to WA. The detailed result of each public dataset is in Table 7.

In Figure 1, we present a visualization of accuracy and training time for each model on the Leyzer dataset. WA achieves comparable performance to XLM-R but only requires less than 1 minute training time, compared to 104 minutes for XLM-R on the Leyzer dataset. WA offers the best trade-off in terms of accuracy vs. training time.

6 Conclusion

In this paper, we propose a robust evaluation framework to benchmark 7 intent classification models in multiple languages. On 9 public datasets and 20 internal datasets covering 10 languages. The benchmark results show that while LaBSE produces the highest accuracy in almost all evaluation settings, Watson Assistant achieves competitive performance with much less cost of training time and resource. The large LMs does not always outperform the models that only need CPUs. Through our work, we hope to encourage more research and development on language-agnostic chatbot solutions.

References

Gaurav Arora, Chirag Jain, Manas Chaturvedi, and Krupal Modi. 2020. Hint3: Raising the bar for intent detection in the wild. *arXiv preprint arXiv:2009.13833*.

Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, and Manfred Langen. 2017. Evaluating natural language understanding services for conversational question answering systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 174–185.

Tanja Bunk, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol. 2020. Diet: Lightweight language understanding for dialogue systems. *arXiv preprint arXiv:2004.09936*.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv preprint arXiv:2008.09335*.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Haode Qi, Lin Pan, Atin Sood, Abhishek Shah, Ladislav Kunc, Mo Yu, and Saloni Potdar. 2021. Benchmarking commercial intent detection services with practice-driven evaluations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 304–310.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2018. Cross-lingual transfer learning for multilingual task oriented dialog. *CoRR*, abs/1810.13327.

Marcin Sowański and Artur Janicki. 2020. Leyzer: A dataset for multilingual virtual assistants. In *International Conference on Text, Speech, and Dialogue*, pages 477–486. Springer.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*.