A Appendix

A.1 Datasets

The experiments in the paper were performed on four datasets: the *Chatbot Corpus* (Chatbot), the *Ask Ubuntu Corpus* (AskUbuntu), the *Web Applications Corpus* (WebApplication), and the *20 News Groups Corpus* (20NewsGroups).

A chatbot was created on Telegram, where questions of the public transport of Munich were posted. The chatbot replied to the questions and thus data was collected for the Chatbot corpus. A detailed test and train split are provided in Table 4.

The AskUbuntu and WebApplication datasets are questions and answers from the StackExchange platform. A detailed breakdown is provided in Table 5 and 7 respectively.

The 20NewsGroups dataset comprises news posts labelled into several categories and a detailed breakdown is provided in Table 8.

A.2 Experimentation Details

The *Text-LeNet* architecture used in the experiments is defined as follows:

Text-LeNet : $[128, M, BN, 256, M, BN, 512, M, F, 128_D, DO, C]$

where, numbers 128, 256 and 512 represents the filters of *Convolution layer* which is followed by an activation function. *M* represents the *Max Pooling layer* and *BN* represents the *Batch Normalization layer*. *F* refers to a *Flatten layer*. 128_D represents the *Dense layer* of size 128 followed by a *Dropout layer* denoted by *DO*. Finally, *C* represents the *Linear classification layer* of dimension (128, number of classes).
The hyper-parameters settings are listed in Table 6.

A.3 Experimental Settings

On all datasets, six tokenization methods were used, namely: Word-based, SemHash, BPE, Char level BPE, Sentence Piece and BERT-based. For Word-based tokenizer, the datasets were pre-processed using the Spacy library to remove stop words from data. Spacy pre-trained model "en_core_web_lg" was used to parse the datasets. All text samples were also pre-processed by removing control tokens, in particular, the ones in the set *[Cc]*, which includes Unicode tokens from *U+0000* to *U+009F*. The small datasets (Chatbot, WebApplication and AskUbuntu) data samples within each class were made equal to the largest class sample size by augmenting the data. For SemHash, n value was kept at 3, and all trigrams were considered. For BPE, a dictionary size of 1000 was used for Chatbot, WebApplication and AskUbuntu, and 2000 for 20NewsCorpus dataset because of its larger size. Similar values were used for Char-based BPE and Sentence Piece tokenizers. The difference between Sentence Piece and BPE is that Sentence Piece uses stop words removal before tokenization while BPE and Char-based BPE do not. BERT-based tokenizer was pre-trained on a large corpus "bert-large-cased-vocab.txt"[1] that contains 29213 unique words and has been provided by the Huggingface[2] library.

For the purpose of benchmarking, nine sklearn classifiers were applied to the intent classification datasets: MLP, Random Forest, Linear SVC, Passive Aggressive, SGD Classifier[3], Ridge Classifier, Nearest Centroid, Bernoulli NB, KNN Classifier with an HD vector size of $d = \{512, 1024, 4098, 8192, 16384\}$. CNN-based architecture used $d = \{512, 1024, 4098, 8192\}$ as embedding sizes. 20NewsGroups dataset was run with MLP, Random Forest, Linear SVC and SGD classifiers for embedding sizes $d = \{512, 1024, 4096, 16384\}$ and with CNN for sizes $d = \{512, 1024, 4096\}$.

We noted in the experiments that Batch Normalization (Ioffe and Szegedy, 2015) is very helpful for

Intent	Train	Test
Departure Time	43	35
Find Connection	57	71

Table 4: Data sample distribution for the Chatbot dataset

Intent	Train	Test
Make Update	10	37
Setup Printer	10	13
Shutdown Computer	13	14
Software Recommendation	17	40
None	3	5

Table 5: Data sample distribution for the AskUbuntu dataset

[1] https://s3.amazonaws.com/models.huggingface.co/bert/bert-large-cased-vocab.txt

[2] https://github.com/huggingface/

[3] The SGD classifier here refers to the SVM classifier trained using SGD optimization as per the sklearn library and this notation is used henceforth.

Hyper-parameter	Value
Convolution Kernel Size	3
Convolution layer Padding	valid
Max-Pooling Kernel Size	3,2,3 for the three M layers respectively
Optimizer	RMSprop
Loss	Categorical cross entropy
Activation Function	Rectified Linear Unit (ReLU)
Batch Size	4 (small datasets) and 64 (20NewsGroups)
Learning Rate	0.001
Number of Epochs	15-40 (Depending upon the dataset)
Initializer	Xavier initialization

Table 6: Hyper-parameters for the experiments

Intent	Train	Test
Change Password	2	6
Delete Account	7	10
Download Video	1	0
Export Data	2	3
Filter Spam	6	14
Find Alternative	7	16
Sync Accounts	3	6
None	2	4

Table 7: Data sample distribution for the WebApplication dataset

Categories	Train	Test
alt.atheism	11314	7532
comp.graphics	11314	7532
comp.os.ms-windows.misc	11314	7532
comp.sys.ibm.pc.hardware	11314	7532
comp.sys.mac.hardware	11314	7532
comp.windows.x	11314	7532
misc.forsale	11314	7532
rec.autos	11314	7532
rec.motorcycles	11314	7532
rec.sport.baseball	11314	7532
rec.sport.hockey	11314	7532
sci.crypt	11314	7532
sci.electronics	11314	7532
sci.electronics	11314	7532
sci.space	11314	7532
soc.religion.christian	11314	7532
talk.politics.guns	11314	7532
talk.politics.mideast	11314	7532
talk.politics.misc	11314	7532
talk.religion.misc	11314	7532

Table 8: Data sample distribution for the 20NewsGroups dataset

faster convergence of the network and, therefore, it was added after every convolutional and dense layer. The clip value for the gradients was set to 1 during the backward pass. RMSProp (Hinton et al., 2012) was used as the optimizer with a learning rate of 1e-3 and categorical cross-entropy as the loss function. All the checkpoints were saved in the h5 format. Larq Compute Engine (Geiger and Team, 2020), a highly optimised engine for quantization of networks, was used to convert the h5 files to tflite format for BNNs.

For classifiers, that were a part of the grid-based search, the paper reports the results for the best hyperparameters. For all other classifiers the default hyperparameter settings provided by the sklearn library were used. A 5-fold cross-validation was used in the experiments. A total of 5 simulations were performed and the average results are reported in the paper. All sklearn-based classifier experiments were performed on the CPU and CNN-based experiments were performed on NVIDIA Tesla GPUs.

A.4 Experiments

Tables 9-11 compares the F_1 scores of nine classifiers: MLP, Random Forest, Linear SVC, Passive Aggressive, SGD Classifier, Ridge Classifier, Nearest Centroid, Bernoulli NB, KNN Classifier on all three small datasets with HD versions of six tokenization methods and Non HD versions of SemHash and BPE for small datasets and Non HD SemHash for 20NewsGroups Corpus. SP is an acronym for Sentence Piece. Table 12 compares the results of all the tokenizers using four classifiers: MLP, SGD Classifier, Linear SVC and Random Forest for the 20NewsGroups dataset. SemHash tokenizer, in general, achieved better results compared to other tokenizers followed by BERT tokenizer on all four datasets.

Classifier	HD Word	HD SemHash	HD BPE	HD Char BPE	HD SP	HD BERT	Non HD SemHash	Non HD BPE
MLP	0.92	**0.93**	0.89	0.90	0.88	0.87	0.92	0.91
Passive Aggr.	0.92	**0.93**	0.90	0.88	0.91	0.90	0.92	**0.93**
SGD Classifier	0.86	**0.89**	0.85	0.88	0.86	**0.89**	**0.89**	**0.89**
Ridge Classifier	**0.92**	**0.92**	0.88	0.87	0.91	**0.92**	0.90	0.91
KNN Classifier	0.81	**0.84**	0.78	0.60	0.80	0.79	0.79	0.72
Nearest Centroid	**0.91**	**0.91**	0.88	0.86	0.86	0.85	0.90	0.89
Linear SVC	0.91	**0.92**	0.90	0.90	0.89	0.90	0.90	**0.92**
Random Forest	0.90	**0.92**	0.82	0.83	0.83	0.82	0.88	0.90
Bernoulli NB	0.76	0.87	0.74	0.84	0.65	0.79	0.91	**0.92**

Table 9: F_1 scores of all sklearn classifiers for AskUbuntu dataset.

Classifier	HD Word	HD SemHash	HD BPE	HD Char BPE	HD SP	HD BERT	Non HD SemHash	Non HD BPE
MLP	0.93	**0.98**	0.88	0.93	0.95	**0.98**	0.96	0.94
Passive Aggr.	**0.96**	**0.96**	0.92	0.94	0.89	0.94	0.95	0.91
SGD Classifier	0.95	**0.97**	0.93	0.89	0.89	0.92	0.93	0.93
Ridge Classifier	0.94	**0.95**	0.77	0.91	0.84	0.88	0.94	0.94
KNN Classifier	0.86	0.86	0.80	0.83	0.80	**0.91**	0.75	0.71
Nearest Centroid	0.79	0.89	0.90	0.86	0.88	0.88	0.89	**0.94**
Linear SVC	0.93	**0.95**	0.89	0.92	0.88	0.94	0.94	0.93
Random Forest	0.88	0.91	0.77	0.83	0.85	0.92	**0.95**	**0.95**
Bernoulli NB	0.81	0.88	0.81	0.89	0.84	0.91	**0.93**	**0.93**

Table 10: F_1 scores of all sklearn classifiers for Chatbot dataset.

Classifier	HD Word	HD SemHash	HD BPE	HD Char BPE	HD SP	HD BERT	Non HD SemHash	Non HD BPE
MLP	**0.82**	**0.82**	0.77	0.78	0.75	0.80	0.77	0.77
Passive Aggr.	**0.83**	**0.83**	0.76	0.80	0.80	0.80	0.82	0.80
SGD Classifier	0.76	0.79	0.66	0.64	0.64	**0.84**	0.75	0.74
Ridge Classifier	0.79	**0.84**	0.79	0.78	0.76	0.81	0.79	0.80
KNN Classifier	0.80	0.81	0.78	0.67	0.80	**0.84**	0.72	0.75
Nearest Centroid	0.77	**0.80**	0.72	0.76	0.75	0.75	0.74	0.73
Linear SVC	0.83	**0.85**	0.79	0.77	0.80	0.82	0.82	0.80
Random Forest	0.77	0.80	0.61	0.68	0.67	0.78	**0.87**	0.85
Bernoulli NB	0.61	0.70	0.57	0.62	0.57	0.61	0.74	**0.75**

Table 11: F_1 scores of all sklearn classifiers for WebApplication dataset.

Classifier	HD Word	HD SemHash	HD BPE	HD Char BPE	HD SP	HD BERT	Non HD SemHash
MLP	0.44	0.61	0.34	0.47	0.40	0.51	**0.72**
SGD Classifier	0.43	0.59	0.41	0.26	0.45	0.49	**0.70**
Linear SVC	0.41	0.64	0.28	0.48	0.37	0.43	**0.75**
Random Forest	0.17	0.33	0.13	0.20	0.13	0.21	**0.58**

Table 12: F_1 scores of sklearn classifiers for 20NewsGroups dataset.