Monolingual and cross-lingual intent detection without training data in target languages

Jurgita Kapočiūtė-Dzikienė, Askars Salimbajevs, Raivis Skadiņš

electronics

MDPI

Article

Monolingual and Cross-Lingual Intent Detection without Training Data in Target Languages

Jurgita Kapočiūtė-Dzikienė [1,2,*], Askars Salimbajevs [3,4] and Raivis Skadiņš [3,4]

1 JSC Tilde Information Technology, Naugarduko Str. 100, LT-03160 Vilnius, Lithuania
2 Department of Applied Informatics, Vytautas Magnus University, Vileikos Str. 8, LT-44404 Kaunas, Lithuania
3 Tilde SIA, Vienības Str. 75A, LV-1004 Riga, Latvia; askars.salimbajevs@Tilde.lv (A.S.); raivis.skadins@tilde.lv (R.S.)
4 Faculty of Computing, University of Latvia, Raiņa Blvd. 19, LV-1586 Riga, Latvia
* Correspondence: jurgita.kapociute-dzikiene@vdu.lt

Abstract: Due to recent DNN advancements, many NLP problems can be effectively solved using transformer-based models and supervised data. Unfortunately, such data is not available in some languages. This research is based on assumptions that (1) training data can be obtained by the machine translating it from another language; (2) there are cross-lingual solutions that work without the training data in the target language. Consequently, in this research, we use the English dataset and solve the intent detection problem for five target languages (German, French, Lithuanian, Latvian, and Portuguese). When seeking the most accurate solutions, we investigate BERT-based word and sentence transformers together with eager learning classifiers (CNN, BERT fine-tuning, FFNN) and lazy learning approach (Cosine similarity as the memory-based method). We offer and evaluate several strategies to overcome the data scarcity problem with machine translation, cross-lingual models, and a combination of the previous two. The experimental investigation revealed the robustness of sentence transformers under various cross-lingual conditions. The accuracy equal to ~0.842 is achieved with the English dataset with completely monolingual models is considered our top-line. However, cross-lingual approaches demonstrate similar accuracy levels reaching ~0.831, ~0.829, ~0.853, ~0.831, and ~0.813 on German, French, Lithuanian, Latvian, and Portuguese languages.

Keywords: BERT; word and sentence transformers; monolingual and cross-lingual experiments; EN, DE, FR, LT, LV, PT languages

Citation: Kapočiūtė-Dzikienė, J.; Salimbajevs, A.; Skadiņš, R. Monolingual and Cross-Lingual Intent Detection without Training Data in Target Languages. *Electronics* **2021**, *10*, 1412. https://doi.org/10.3390/electronics10121412

Academic Editors: Pablo Gamallo, Patricia Martín-Rodilla and Daniel Gutiérrez Reina

Received: 13 April 2021
Accepted: 9 June 2021
Published: 11 June 2021

1. Introduction

Powered by recent significant Deep Neural Network (DNN) advancements in the field of Natural Language Processing (NLP), chatbots are becoming increasingly popular technology in real-time customer services [1]. The chatbot research area has already a long history dating back to 1966 [2]. The very first chatbot ELIZA, introduced by MIT Artificial Intelligence Laboratory, was adjusted to communicate with people suffering from psychological issues. ELIZA was examining keywords from the user's input and was prompting answers based on a pre-defined set of rules. Today chatbots are widely used in Marketing, Education, Healthcare, and other sectors. They even are created to entertain in interactive theater performances [3]. However, this pandemic especially revealed the necessity of chatbots in the news media [4], hospitals, or the healthcare system managing huge flows of incoming questions [5].

Chatbots provide support (usually adjusted to a single domain and able to answer FAQ-type questions), skills (do not need to be aware of the context, only comprehend a determined set of commands), or perform a virtual assistant (connecting both previous types) role. According to their primary goal, chatbots can be classified into informative (provide information), conversational (simulate human-like behavior in continuous conversations with a user), and task-based (provide a service or help a user in a pre-determined

task). Depending on their proximity to a user and provided services, chatbots can communicate inter-personally (pass user's information, but do not necessarily remember it), intra-personally (act like user's companions), or inter-agent-like (communicate with other chatbots). Based on an accessible or trained knowledge domain, chatbots can be grouped into closed (responding accurately to a limited set of questions on specific topics) and open (able to answer questions of any topics) domain types. Input processing manner and nature of response determine if a chatbot is generative or intent detection-based. Generative chatbots usually require huge amounts of training data and can learn how to generate responses from it. Intent detection-based chatbots are functioning as classifiers and therefore are limited to pre-defined responses. Despite advanced algorithms, any of these chatbot types cannot be prevented from failures in real-case user dialog scenarios. However, intend detection-based ones are more robust compared to generative and typically used in production chatbots, therefore chosen as our research direction.

Usually, chatbots are composed of four different components: Natural Language Understanding (NLU) (responsible for comprehension of user's requests meaning and structure), Dialog Management (controls a smooth flow of a conversation), Content (a template of how a chatbot must respond), and External Data (extracts data from external web services or databases). However, NLU has considered the most essential component: without understanding the user's request, all other components become secondary. Consequently, this research is focused on the improvement of the NLU component.

In general, the NLP area is completely dominated by research on the English language, which is resource-rich. In the machine learning (and especially deep learning) era, this fact explains why so much research has been conducted for English, and many accurate tools have been developed. However, no less important is paying attention to less popular and less-resourced languages that must constantly chase English. The gap is mostly due to less quality, quantity, or availability of resources: training datasets, corpora, monolingual refined embeddings, etc. A lack of resources or their proprietary usage often becomes an obstacle that hinders progress for such languages (especially complex ones, having a small number of speakers). Therefore, the goal of this research is to find measures how to address this sensitive multilingualism problem. One straightforward way is to choose available benchmark English datasets but translate (or machine translate, if possible) them into target languages. Hence, in this research, we rely on an assumption that machine translation does not distort data to such an extent that it degrades its quality significantly to become unsuitable for training NLU models, especially knowing that the quality of machine translation tools is significantly improved recently due to neural-based approaches. We even assume this problem is possible to tackle without machine translation by just employing multilingual transformers (as pre-trained vectorization models) able to capture sentence semantics. As an object of our research, we have chosen English, German, French, Lithuanian, Latvian, and Portuguese languages as our target languages. Such choice was performed on purpose: it includes different language groups (Germanic, Romance, and Baltic languages) and covers languages having different amounts of resources.

2. Related Work

Despite the fact that multilingual chatbots are in high demand usually, support of several languages is based on language identification first and application of an appropriate monolingual NLU model afterward. Consequently, there is very little related research on multilingualism in the scientific literature as well. Nevertheless, we will take a broader look at the methodologies used for creating NLU models.

According to the Scopus paper analysis by [6], the rapid growth of interest in chatbots is especially visible after 2016, with a special focus on innovative DNN technology in recent years. The essence of this related work analysis is to highlight methodologies that could be the most effective for the intent detection task. Additionally, it can be performed the best via comparative analysis under the same experimental conditions (same datasets, the same distribution for training/testing, etc.).

One of the most popular benchmark datasets is the ATIS (Airline Travel Information Systems) dataset introduced in [7] consisting of 17 intent categories, ~11 words per utterance with 4478, 500, and 893 utterances in train, development, and test subsets, respectively. Moreover, this dataset consists of speech transcripts and therefore represents a spoken language that is even more difficult to process compared to normative. With the best technique, the accuracy for this dataset reaches ~0.99 [8]. Indeed, besides intent detection, authors also tackle a slot-filling task (that searches for a specific piece of information as named entities or things) and prove that both tasks benefit the most from solving them jointly via their cross-impact. Their novel offered Bi-model based RNN semantic frame parsing approach (especially with a decoder) applied on jointly trained word embeddings was able to surpass other previously applied techniques. The second-best approach [9], using ATIS for intent detection, is based on the transformer-capsule model, especially suitable to model hierarchical relationships. GloVe embeddings [10] were fed into the transformer encoder (with 12-heads attention and feed-forward layer with 300 hidden dimensions), and then the produced vector was passed into a capsule network (composed of 100 capsules with 15 dimensions in each). Despite these authors demonstrated slightly lower performance compared to [8], it is in a range of ~0.99 of the accuracy. The competitively high accuracy of ~0.98 on ATIS was achieved with less refined technologies already in 2016 [11]. These authors tackled intent detection and slot-filling tasks independently and jointly, proving that their joint model gains over independent ones. They test an encoder-decoder model with aligned inputs in which the Bidirectional Long Short-Term Memory (BiLSTM) network is used on the encoder side and unidirectional LSTM on the decoder. Besides, the authors complemented their model with an attention mechanism, which improved the accuracy even further. The accuracy of ~0.98 on ATIS is achieved with the BERT-based model [12]. The architecture is composed of a BERT base model with fine-tuning as the encoder module and two decoders. The encoder represents an utterance grasping knowledge between the intent detection and slot-filling tasks, and then the first decoder performs intent detection. Then, the stack-propagation framework (enabling backpropagation down the stacked models) concatenates the output of the intent detection decoder and representations from the encoder as the input for the slot-filling decoder. Both intent detection and slot-filling sub-models are jointly learned by optimizing them simultaneously.

Another popular NLU benchmark dataset is SNIPS, introduced in [13]. This dataset contains 16 thousand crowdsourced queries distributed among seven intents with ~9 words per utterance. Similar to ATIS, this dataset also contains spoken language. The best accuracy reaching ~0.97 on this dataset is achieved with the BERT-based stack-propagation framework [12], giving promising results on ATIS as well. The method incorporating the contextual information at the representation and task-specific level allows achieving ~0.94 of the accuracy on SNIPS [14]. The context of each word is obtained via max-pooling over the outputs of BiLSTM for all sentence words except the target one. Thus, this first level aims to use the context of each word to predict the label of that word. The second level uses global context information to predict sentence-level labels. The range of ~0.92 of the accuracy can be achieved with another bidirectional interrelated slot-filling and intent detection model [15]. This method is based on the BiLSTM architecture as in [14]. It uses separate computed context vectors and separate attention mechanisms for slot and intent tasks.

When the dataset is stable (as in the experiments with the benchmark collections), the most often choice increasing the accuracy is the right choice of the methodology. Some authors address this issue by adding new training instances that expand the dataset but do not fundamentally change it. The innovative adversarial training approach jointly solving intent detection and slot-filling tasks with ATIS and SNIPS datasets injects perturbed inputs (adversarial examples) into the training data [16]. The perturbed word/character embeddings add a little noise to utterances that do not fool the model, but on the contrary, make it more robust. The classifier authors use a combination of LSTM encoder-decoder

with a stacked CRF applied on top of the BERT-large embedding model. The authors claim, their joint adverbial training model that applies a balance factor as a regularization term to the final loss function reaches state-of-the-art performance on the ATIS and SNIPS datasets.

The authors in [17] explore three datasets, i.e., HWU64 (containing ~25.7 thousand instances, 64 intents, ~7 words per instance), CLINC150 (~23.7 thousand instances, 150 intents, ~8 words per instance), and BANKING77 (~13 thousand customer service queries, 77 intents, ~12 words per instance). Unlike previous benchmark datasets (having 17 and 7 intents in ATIS and SNIPS, respectively), HWU64, CLINIC150, BANKING77 contain many more intents making this task even more complex. The authors imply dual sentence encoders (learned from interactions between input/context and relevel responses and therefore encapsulating conversational knowledge) such as USE (Universal Sentence Encoder) and ConverRT to support intent detection. The experimental investigation demonstrates the superiority of dual sentence embeddings over the fixed or fine-tuned BERT-large models, which is especially apparent with smaller intents (covered with ~10–30 cases).

The intent detection problem, which is the most relevant in chatbots, is tackled for other purposes as well. The authors [18] are solving the e-mail overload problem by classifying them into two intents: "to read" or "to do". The authors test context-free word embeddings (word2vec and GloVe), contextual word embeddings (ELMo and BERT), and sentence embeddings (DAN-based USE and Transformer-based USE), proving the superiority of ELMo, followed by Transformer-based USE and then DAN-based USE. This research compares a huge variety of word and sentence embedding types and once again proves that sentence embeddings are also a very powerful tool for intent detection problems.

As can be seen, some researchers tackle problems having more or fewer intents, whereas others are focused on a few-shot intent detection scarcity problems of emerging classes. The authors in [19] offer the novel BiLSTM-based Semantic Matching and Aggregation Network approach. Their approach distills semantic components from utterances via multihead self-attention with additional dynamic regularization constraints. They experimentally compare their offered approach to 6 more methods (Matching Network, Prototypical Network, Relation Network, Hybrid Attention-based Prototypical Network, Hierarchical Prototypical Network, Multi-level Matching, and Aggregation Network) and prove their method achieves the best performance on two datasets. A very similar problem [20] is tackled with the novel two-fold pseudolabeling technique. The pseudolabeling process takes embedded user utterances and passes them to a hierarchical clustering method (in a bottom-up tree-manner), then the process goes top-down a tree and expands nodes having multiple labeled sentences with different labels. Once the pseudolabels are retrieved, the method performs BERT fine-tuning-based intent detection, which is a common solution for intent detection problems.

The other important intent detection direction covers multiple intents in the same utterance problems. The authors in [21] solve joint multiple intent detection and slot-filling problems with the Adaptive Graph-Interactive Framework method. Firstly, the self-attentive BiLSTM encoder represents some utterance which is then passed to the multilabel intent detection decoder, which computes context vectors using self-attention. Afterward, the adaptive intent-slot graph interaction layer leverages information about the multiple intents for slot prediction. Next to the offered method, authors also test more five state-of-the-art approaches (Attention BiRNN, Slot-Gated Atten, Bi-Model, SF-ID Network, Stack-Propagation), proving their offered method is superior on MixATIS and MixSNIPS datasets (appropriate ATIS and SNIPS versions but containing multiple intents). Either few-shot or multiple intent problems have additional mechanisms that go beyond the common intent detection problem-solving. However, parts responsible for the intent detection are tackled with the DNN-based techniques typically used for the common intent detection problems. Other intent detection monolingual research covers non-English languages; however, applied methods are in the same DNN-based trend.

Previously summarized approaches focus only on monolingual research, therefore, do not reveal their all potential. Recently some popular commercial virtual assistants (as

Google Home, Amazon, Apple Siri) were scaled to more regions and languages. Multilingual chatbots are gaining more and more attention from the scientific community as well. Thus, we direct our further method analysis towards multilingual intent detection problems (working well if applied separately on several languages) with a special focus on cross-lingual (working well if applied jointly on several languages) approaches. The paper [22] describes the offered joint model for intent detection and named entity recognition. The method firstly maps input tokens into share-space word embedding and then feeds them into the encoder to extract context information. Afterward, this content is propagated to downstream tasks. For the transfer learning experiments, authors train on high-resource languages and then: (1) transfer both encoder and decoder to a new multilingual model with fine-tuning; (2) transfer only encoder with fixed parameters to new multilingual model; (3) transfer only encoder with available learning rate by gradually freezing embeddings with training steps during fine-tuning. If precisely, authors use initial concatenated fastText embeddings trained on a three-filter Convolutional Neural Network (CNN); BiLSTM as the encoder; a multilayer perceptron for intent detection and CRF sequence labeler for NER with gelu activation function as the decoder. The authors applied their methods to English (~2.2 million utterances, 316 intents, and 282 slots), Spanish (~3 million utterances, 365 intents, 311 slots), Italian (~2.5 utterances, 379 intents, 324 slots), and Hindi (~0.4 million utterances, 302 intents, 267 slots) datasets. They observe performance improvements in all models with transfer learning, with the largest improvement with encoder transfer. The authors in [23] use the multilingual dataset containing annotated utterances in English (~43 thousand), Spanish (~8.6 thousand), and Thai (~5 thousand) and covering 3 domains, 12 intents, and 11 slots. They evaluate cross-lingual transfer methods based on (1) translated training data; (2) cross-lingual pre-trained embeddings; (3) multilingual machine translation encoder as contextual word representations. The joint intent detection and slot-filling model at first use a sentence classification model to identify the domain and then a domain-specific model to jointly predict intent and slots. The method architecture has self-attention BiLSTM and Conditional Random Fields (CRF) layers. The method is tested with several types of word embeddings (zero, XLU, encoder, CoVe, multilingual CoVE, and multilingual CoVE + autoencoder) trained by authors and available pre-trained ELMo encoders for Spanish. The authors found that languages with limited data benefit from cross-lingual learning. Despite it, multilingual contextual word representations outperform cross-lingual static embeddings. Due to these findings, the authors have to highlight a need for more refined cross-lingual methods. Another interesting cross-lingual research [24] uses a dataset containing ~6.9 thousand utterances across 16 COVID-19 specific intents in English, Spanish, French, and German languages. The authors explore: (1) monolingual and multilingual model baselines; (2) cross-lingual transfer from English to other languages; (3) zero-shot (in which only English data is used for training and model selection) code-switching for Spanglish (combining words and idioms from Spanish and English). These authors tested fastText, XLM-R, and ELMo embeddings. Authors prove that lower results are obtained under a zero-shot setting, and XLM-R cross-lingual sentence embeddings significantly outperform their other cross-lingual solutions. Another cross-lingual research [25], for the first time, presents multilingual modeling without degrading per-language performance. It demonstrates the robustness of pre-trained multilingual language models leading to significant performance gains for cross-lingual transfer tasks as natural language inference (15 languages), NER (English, Dutch, Spanish, and German), question answering (English, Spanish, German, Arabic, Hindi, Vietnamese and Chinese). Their XLM-Rbase (L = 12, H = 768, A = 12,270 million params) and XLM-R (L = 24, H = 1024, A = 16,550 million params) models outperform mBERT (compared to BERT for English, mBERT is trained on 104 languages with a shared word piece vocabulary, which allows the model to share embeddings across languages). The authors demonstrate their models significantly outperform mBERT on cross-lingual tasks, perform especially well on low-resourced languages.

Despite there is no consensus on which method is the best for intent detection problems, it effectively narrows the set of choices giving us guidance on which techniques are the most promising. However, as it can be seen from the related work analysis, very little has been carried out in the cross-lingual direction when transferring models (trained in downstream tasks) across different languages. The plentifulness of data resources for the English language and relatively little research carried out on some languages inspire us (1) to rely on machine translation tools when preparing datasets for target languages and (2) to seek cross-lingual-based solutions where less-resourced languages could benefit from others. The contribution of our research is due to the following reasons, we: (1) perform our experiments under monolingual and several cross-lingual settings; (2) tackle intent detection problem when training on English alone and testing on other target languages; (3) compare different approaches and embedding types over six languages (English, German, French, Latvian, Lithuanian, and Portuguese); (4) use a very small dataset (in which each intent is covered by a relatively small number of instances).

3. Methodology

The research question of our paper is how to create the multilingual intent detection method (by offering the vectorization technique, classifier, model, and training data usage strategy) without having annotated training data necessary prepared in the target language. To answer this research question, we choose several languages differing by their characteristics. The creation of such a multilingual method would open opportunities for other researchers solving intent detection problems to rely more on machine-translated data and cross-lingual approaches. If our hypotheses would be valid for all tested languages (taken from different language groups and differing in various characteristics), we anticipate that the offered multilingual method could also be applied for the broader group of languages (at least for Germanic, Romance, Balto-Slavic groups) having pre-trained multilingual BERT vectorization models. Moreover, the obtained know1edge about the offered methodology possibilities and boundaries could also be used in other supervised machine learning tasks. Thus, our offered approach (based on the machine-translated data or cross-lingual models) could be a superior alternative to previous approaches, typically demanding training data necessary created only for the target language.

Our research result is the different approach to the classification type problem (i.e., intent detection) solving and the type of result is the offered new technique able to tackle multilingual intent detection problems. Different approaches (combining vectorization techniques, classifiers, models) are tested on real datasets (for English, German, French, Lithuanian, Latvian, and Portuguese languages) under several training data usage strategies (relying on the machine-translated data and/or cross-lingual models) in the carefully designed controlled experiment with statistically significant results. The type of the performed research validation is the analysis. Thus, the research question (how to create multilingual intent detection method without annotated training data necessary prepared in the target language), the expected result (a technique/method able to solve multilingual intent detection problems without annotated training data necessary prepared in the target language) with the analysis research validation (as the controlled experiment) are combined into our research strategy. This strategy was applied and evaluated with *accuracy, precision, recall, f-score* metrics (typically used in the evaluation of intent detection problems); the obtained results with different approaches were compared to see if differences are statistically significant. Our research question would be confirmed if applying multilingual methods on target languages (not having training data but relying on the machine-translated English data and/or cross-lingual models) would achieve similar accuracy levels as with the monolingual methods on the English language with the original dataset.

3.1. Formal Description of the Task

The intent detection problem is a typical example of a supervised text classification task. Formally, such a task is determined as follows:

Let $D = \{d_1, d_2, \ldots, d_n\}$ be a set of documents (questions/statements an input from a user). Let $C = \{c_1, c_2, \ldots, c_m\}$ be a set of intents (classes). We have a closed-set classification problem where m is limited, and each c_j is defined in advance. Besides, we solve a single-label classification problem because each $d_i \in D$ can be attached to only one $c_j \in C$.

Let function η be a classification function that maps d_i into their correct classes: $D \rightarrow C$. Let $D^L \subset D$ be a training set of labeled instances (pairs of documents and their correct intents (d_i, c_j)) used to train a model.

Let Γ be a classification method that, from labeled instances, can learn a model (which is the approximation of η).

Our solving intent detection task aims to offer a classification method Γ that can find the best approximation of η, achieving as high an intent detection accuracy as possible on unseen instances $(D - D^L)$ also.

3.2. Datasets

The intent detection problem (described in Section 3.1) can be tackled with the appropriate dataset. For this reason, we have used the manually prepared English dataset that contains fluent questions and related answers about the app *Tildės Biuras* (more about it in https://www.tilde.lt/tildes-biuras accessed on 13 November 2020) prices, licenses, supported languages, and used technologies. Instances in the dataset were shuffled and randomly split into training and testing subsets by keeping the proportion for training and testing equal to 80% and 20% instances per intent, respectively (Table 1). The dataset covers only ~8.9 instances per intent on average in the training dataset, which means the solving intent detection task is challenging. Despite it, our case is by no means exceptional. There are many benchmark datasets with even fewer instances per intent on average (e.g., in [26]). Moreover, such datasets reflect the expectations of real customers that want to achieve the best possible chatbot's accuracy with minimum effort.

Table 1. Statistics about the used English dataset.

	Training	**Testing**
Number of intents	41	41
Number of instances	365	144
Instances per intent	8.9	3.5

Despite our available dataset is only in the English language, we plan to use it in a way that could prove that English resources could perfectly serve in solving intent detection problems for other languages as well. As the object of research, next to English (EN), we have chosen one more Germanic language (i.e., German (DE)), two Romance languages (French (FR) and Portuguese (PT)), and two Baltic languages (Lithuanian (LT) and Latvian (LV)), differing from each other by such characteristics as morphology, derivational systems, sentence structures, etc.

The EN training dataset (in Table 1) was Google machine-translated, whereas the testing dataset was manually translated into DE, FR, LT, LV, and PT languages. Such preparation was carried out on purpose. We simulate the common condition when training data is not available for some languages but can be easily prepared with the help of machine translation. The review of machine-translated data revealed that despite some not very precise translations, the gist in texts is retained, and therefore, automatic machine translation is a reliable way to translate the training data. The testing dataset is manually prepared because the intent detection model is usually tested by real users writing questions in their language. The sizes of datasets in different languages are in Table 2.

Table 2. Statistics (numbers of words) about datasets in different languages.

Language	Training	Testing
EN	2826	1090
DE	2369	877
FR	2743	1222
LT	1929	751
LV	1991	855
PT	2812	1133

3.3. Used Approaches

The goal of this section is to offer the best Γ (presented in Section 3.1) for our solving supervised intent detection tasks. Therefore, we need to find the best combination of text representation and classification techniques.

For the text representation (vectorization) we have investigated the following approaches:

- **Word embeddings.** BERT (Bidirectional Encoder Representations from Transformers) [27] is a transformers model pre-trained on a large raw corpus in a self-supervised manner (by automatically generating inputs and labels from texts). Its learning is based on masked language modeling and next sentence prediction phases. The masked language modeling process takes a sentence, randomly masks some words, and then learns how to predict them. This way, the model learns bidirectional sentence representations. Thus, BERT is robust to word disambiguation problems: words written equally but with different meanings are represented with different vectors based on their context. Afterward, the next sentence prediction process learns to determine if two sentences follow each other in a sequence. This training manner allows learning inner language representations that later can be used to extract features for downstream classification tasks. In our experiments, we have investigated 4 monolingual English BERT models, i.e., *bert-base-cased*, *bert-base-uncased*, *bert-large-cased*, and *bert-large-uncased*. The difference between *base* and *large* models is in the number of stacked encoder layers (12 vs. 24 for base and large, respectively), attention heads (12 vs. 16), parameters (110 million vs. 340 million), and hidden layers (768 vs. 1024). Cased models are sensitive to the letter-casing and, vice versa, uncased models are not. We also investigated multilingual BERT models *bert-base-multilingual-cased* and *bert-base-multilingual-uncased* (a detailed description of used BERT transformer models can be found in https://huggingface.co/transformers/pretrained_models.html accessed on 13 November 2020), both trained on Wikipedia texts of 104 languages, including all languages that we use in this research.
- **Sentence embeddings**. Besides, BERT we have tested several models tuned to be used for text/sentence embedding generation [28]. The output of such transformer models is pooled to generate a fixed-size representation. In our experiments next to sentence BERT, we have RoBERTa [29], DistilBERT [30], DistilUSE, and XLNet [31] transformer models. RoBERTa is an optimized BERT approach. It does not have the next sentence prediction phase and, instead of masked language modeling, performs dynamic masking by changing masked tokens during training epochs. Besides, RoBERTa is trained on much larger amounts of data. DistilBERT is the smaller approximation of the BERT transformer model, retaining only half of its layers (with ~66 million parameters). Besides, DistilBERT even does not have token-type embeddings and the pooler. The DistilUSE transformer model is similar to DistilBERT, but it uses an additional down-projection layer on top of DistilBERT. The XLNet transformer, instead of masked language modeling, uses permutation language modeling in which all tokens are predicted but in random order. Besides, XLNet is trained on much larger amounts of data compared to BERT. We have experimented with 4 monolingual English sentence embedding models: *roberta-base-nli-stsb-mean-tokens*, *roberta-large-nli-stsb-mean-tokens*, *bert-large-nli-stsb-mean-tokens*, *distilbert-base-nli-stsb-mean-tokens* and 4 multilingual sentence embedding models: *distiluse-*

base-multilingual-cased-v2, *xlm-r-distilroberta-base-paraphrase-v1*, *xlm-r-bert-base-nli-stsb-mean-tokens*, *distilbert-multilingual-nli-stsb-quora-ranking* (more about these models can be found in https://www.sbert.net/docs/pretrained_models.html accessed on 13 November 2020). The *nli* and *stsb* notations stand for training on the Natural Language Inference data and testing on Semantic Textual Similarity Benchmark dataset, respectively. The *mean-tokens* notation represents the mean pooling with taking an attention mask into account. *Paraphrase* means that training is performed on millions of paraphrased sentences. *Quora-ranking* determines that the model is expanded by training it with contrastive loss and multiple negative ranking loss functions on the Quora Duplicate questions dataset.

For the intent detection, we have investigated the following approaches:

- **BERT-w + CNN**. The Convolutional Neural Network (CNN) classifier introduced in [32] was applied on top of concatenated BERT word embeddings. In our experiments, we have used the 1D CNN method adjusted for text [33] with the optimized architecture and hyper-parameter values in various language processing tasks, including intent detection for English, Estonian, Latvian, Lithuanian, and Russian(see Figure 3 in [34]). We reuse the architecture and hyper-parameter set of CNN in our experiments without any further optimization. The advantage of CNN is that it learns how to recognize patterns independently of their position in the text. Thus, the CNN method gets the vectorized texts (i.e., determining the length sequences of the corresponding word embeddings) on the input and learns to detect relevant patterns (consisting of 2, 3, or more adjacent tokens, so-called n-grams) (regardless of their position in the text) having the major impact on prediction of the right class.
- **BERT-w + BERT**. The BERT transformer model can be used in various classification tasks, including intent detection. If precisely, the pre-trained BERT model is fine-tuned with just one additional output layer of neurons corresponding to classes. Despite the parameters of such a model still have to be modified to adjust to the downstream intent detection task, the advantage of such an approach is that the pre-trained BERT model weights already encode a lot of information about the language. Since bottom layers are already well learned, the tuning process only slightly adjusts them in the way their output could serve as features in text classification.
- **BERT-s + FFNN**. BERT sentence embeddings as features are fed into the Feed Forward Neural Network (FFNN) as the classifier.
- **BERT-s + COS**. This approach, unlike previously described classification-based approaches, does not learn any generalized model. It simply stores all training data and computes the similarity between the testing instance and all training instances. The testing instance is assigned with the label of the training instance with which the similarity is the largest. The similarity between sentence embeddings is calculated using the cosine similarity measure [35].

These four approaches were implemented using a python 3.8.5 programming language with Tensorflow 2.3.1, Keras 2.4.3, and PyTorch 1.6.0 libraries. The word and sentence transformer models were taken from the huggingface repository.

Datasets (Section 3.2) and previously described machine learning methods were evaluated under the following training data usage strategies for tackling the data scarcity problem:

- **Monolingual machine-translated** (we call this strategy **MT-based** due to conciseness). Both training and testing are conducted in the same target language. These experiments will demonstrate the performance of monolingual models trained on machine-translated data. Results with the manually prepared EN dataset are particularly important: it will reveal what level of accuracy should be pursued with other languages. Results with other languages will reveal how far the results for other languages with translated texts lag.
- **Cross-lingual**. Under this condition, training is conducted on the EN training dataset alone, but testing is conducted on the testing dataset of some other target language

(e.g., DE, FR, LT, LV, and PT). These experiments do not use machine-translated training data at all but rely on multilingual BERT models. This will test the ability of BERT-based models to capture semantic similarities between the same texts written in different languages.

- **Combined**. These experiments combine the previous two approaches: the training is conducted on two datasets of two languages, i.e., original EN plus the machine-translated target language. Such experiments will reveal if both training data preparation methods are complementary. This will also help answer the question of whether it is sufficient to rely on BERT-based models alone or whether the role of the machine translator (or training data in the target language) is nevertheless crucial.
- **Cross-lingual without any target language data** (we abbreviate it to **train all** due to conciseness). Under this condition, training is conducted on all training datasets of all languages (both manually for EN and machine translated for other, but necessary excluding the target language). This represents the scenario when no target language data can be obtained (even machine-translated). We propose that by training on data machine-translated to multiple other languages, we can facilitate semantic interfaces between languages in BERT-based models. In case of success, these experiments can be especially beneficial for languages for which machine-translated data cannot be obtained or are of very poor quality.

4. Experiments and Results

The experimental investigation is based on the hypothesis that it is possible to find a good multilingual intent detection method that does not require original training data specifically prepared for the target language (in our case: German, French, Lithuanian, Latvian, and Portuguese) to achieve similar accuracy levels as with the monolingual method applied on the original dataset (in our case the original dataset is in English).

We have performed experiments under the controlled conditions in which some parameters were kept stable to see the impact of varied ones. We have controlled: different training and testing language pairs, vectorization types, classification approaches, and models. The randomness in our experiments was introduced (1) by selecting language representatives from several groups of languages and (2) by shuffling instances in our datasets (presented in Section 3.2) and randomly splitting them into training (80%) and testing (20%) subsets. Moreover, in each run, the training dataset part was once again shuffled and randomly split into training (80%) and validation (20%) subsets. This randomness guarantees that the training does not bind to the specific training instances, but at the same time, similar experimental conditions for the results to remain comparable are maintained.

The performance of each trained model was evaluated with the *accuracy*, *precision*, *recall*, and *f-score* metrics presented in Equations (1)–(4), respectively. The evaluation of *accuracy*, *precision*, *recall*, and *f-score* metrics was performed using *sklearn.metrics* in python.

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}, \quad (1)$$

where tp and tn represent correctly predicted c_i and c_j instances, respectively; $fp - c_j$ incorrectly predicted as c_i, and $fn - c_i$ incorrectly predicted as c_j.

$$precision = \frac{tp}{tp + fp} \quad (2)$$

$$recall = \frac{tp}{tp + fn} \quad (3)$$

$$f\_score = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

The *accuracy*, *precision*, *recall*, and *f-score* values were averaged in five runs, and the confidence intervals were calculated for all approaches (in Section 3.3) except BERT-s + COS.

The BERT-s + COS method is a memory-based approach that simply stores all training data and computes the similarity between each testing instance and all training instances. Since vectors representing training and testing instances are stable, each run results in absolutely the same predicted labels for the training dataset instances. There is no deviation in results; therefore, upper/lower bounds values of confidence intervals are always equal to 0.

A model is considered reasonable if the calculated accuracy is above random (Equation (5)) and majority (Equation (6)) baselines.

$$random\_baseline = \sum P^2(c_j), \tag{5}$$

where $P(c_j)$ is a probability of a class.

$$majority\_baseline = max(P(c_j)), \tag{6}$$

In our experiments, *random* and *majority* baselines are equal to ~0.04 and ~0.09, respectively. The low random baseline value demonstrates the difficulty of the task in which a "random guess" could not be "a solution"; the low majority baseline value shows that the dataset is not biased towards any class.

When comparing different evaluation results is important to determine if differences between them are statistically significant. For this purpose, the McNemar test [36] with 95% of confidence ($\alpha = 0.05$) has been used. Differences are considered statistically significant if the calculated p-value is below $\alpha = 0.05$. The evaluation of statistical significance was performed using *statsmodels.stats.contingency_tables* module in python.

During experiments under the *MT-based* strategy, we have tested all four approaches (described in Section 3.2). Evaluation results with BERT-w + CNN, BERT-w + BERT, BERT-s + FFNN, and BERT-s + COS are presented in Table A1, Table A2, Table 3, and Table A4, respectively. EN results are obtained on original data and represent the top-line. To see clearly which approach is the best for each target language, we have summarized accuracies in Figure 1. Methods based on sentence embeddings outperform methods based on word embeddings. The winner is BERT-s + FFNN followed by BERT-s + COS, despite differences between their accuracies for most languages are not statistically significant. The experimental investigation revealed that all four metrics (*accuracy*, *precision*, *recall*, and *f-score*) demonstrate similar trends. For comparison purposes, we have selected *accuracy* as the main metric in this and further experiments. It is the most common metric, besides, suitable for our dataset not biased towards major classes.

Next, experiments were performed under the *cross-lingual* strategy: i.e., when training multilingual models on the original EN training dataset alone and testing on some other target language.

The results of BERT-w + CNN, BERT-w + BERT, BERT-s + FFNN, and BERT-s + COS approaches under the *cross-lingual* strategy are summarized in Table A5, Table A6, Table A7, and Table A8, respectively. The summary of the highest accuracies for each target language is presented in Figure 2.

A combination of the first two data preparation approaches under the *combined* strategy also was evaluated. The training was performed on two datasets of two languages (i.e., EN + the target language), whereas testing was conducted on the testing dataset of the target language. We have shrunk the set of testing approaches to BERT-s + FFNN and BERT-s + COS because only they demonstrated good performance under the cross-lingual condition. The accuracies for BERT-s + FFNN and BERT-s + COS are summarized in Tables A9 and A10, respectively. The best accuracies for each target language are presented in Figure 3.

Finally, experiments under the *train all* strategy were performed. During these experiments, training was conducted on all training datasets for all languages (excluding the target one) and testing was performed on the testing dataset of the target language. The accuracies for BERT-s + FFNN and BERT-s + COS approaches are summarized in

Tables A11 and A12, respectively. The best accuracies for each target language are presented in Figure 4.

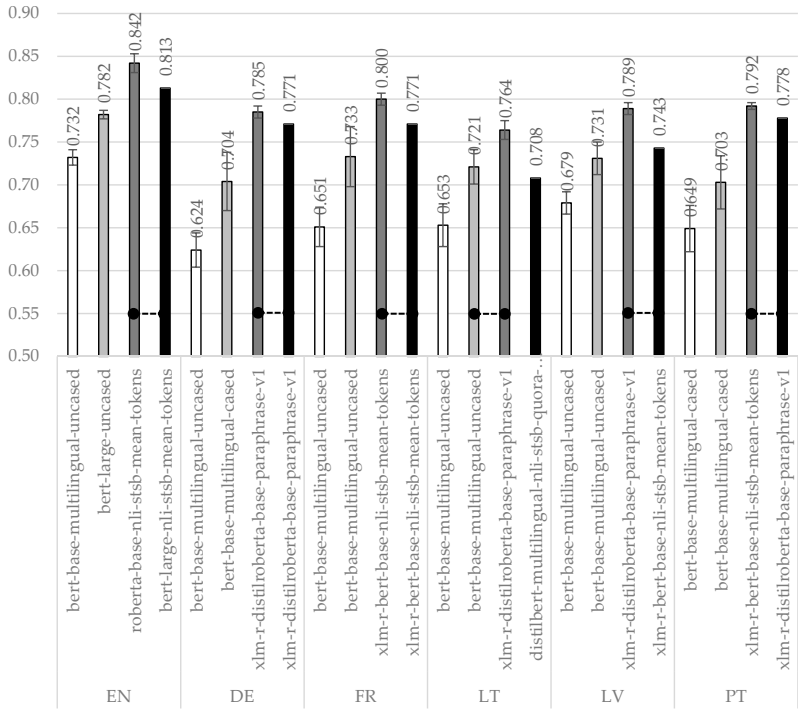

Figure 1. The best accuracies + confidence intervals with BERT-w + CNN, BERT-w + BERT, BERT-s + FFNN, and BERT-s + COS approaches under the *MT-based* strategy. Dashed lines connect the best-achieved accuracy (within the same language) with those accuracies to which differences are not statistically significant. EN results are obtained on original data and represent the top-line.

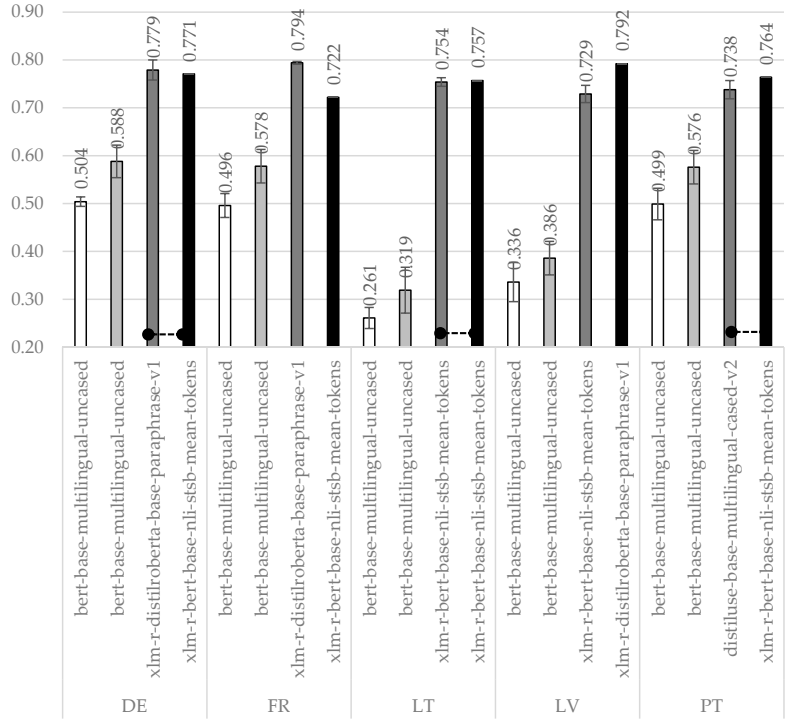

Figure 2. The best accuracies + confidence intervals with BERT-w + CNN, BERT-w + BERT, BERT-s + FFNN, and BERT-s + COS approaches under the *cross-lingual* strategy. For the notation, see Figure 1.

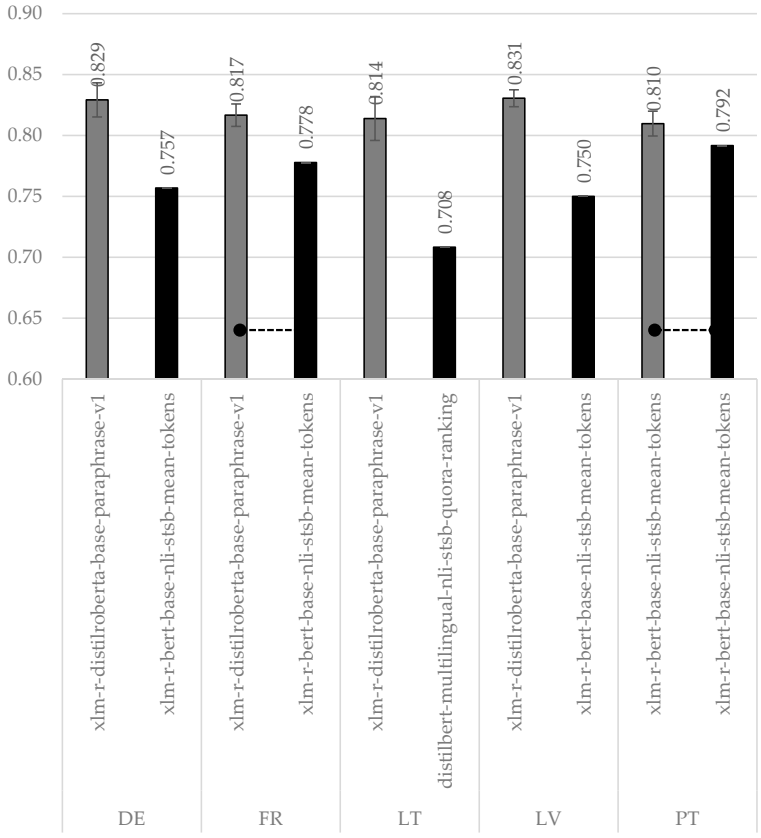

Figure 3. The best accuracies + confidence intervals of BERT-s + FFNN and BERT-s + COS models trained under the *combined* strategy. For the notation, see Figure 1.

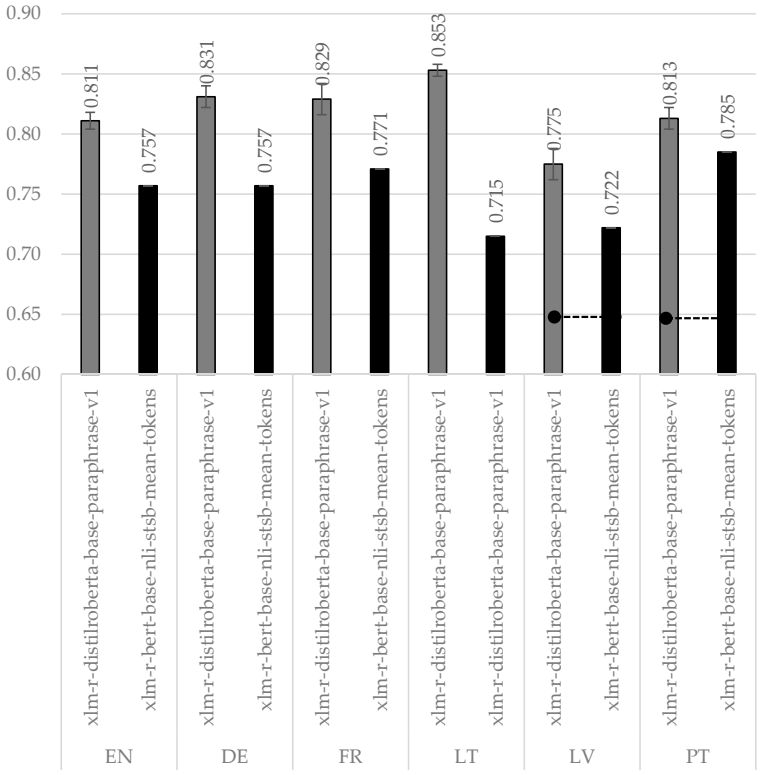

Figure 4. The best accuracies + confidence intervals with BERT-s + FFNN and BERT-s + COS approaches under the *train all* strategy. For the notation see Figure 1.

5. Discussion

Zooming into tables in Appendix A and figures allows us to make the statement that all results can be considered reasonable for solving intent detection tasks because they exceed random and majority baselines.

The best overall accuracy (equal to ~0.842) under the monolingual *MT-based* condition is achieved on the original English language dataset. This result represents our top-line, which will be used for comparison purposes to other approaches and languages. The most accurate approach (i.e., BERT-s + FFNN) uses the pre-trained *roberta-base-nli-stsb-mean-tokens* BERT model that is adjusted for the English language alone. It also explains why this particular model outperforms multilingual pre-trained sentence embeddings (i.e., *xlm-r-distilroberta-base-paraphrase-v1* and *xlm-r-bert-base-nli-stsb-mean-tokens*).

Experiments with the machine-translated training data (i.e. under the *MT-based* strategy) clearly show that this approach is successful. The best performer again is BERT-s + FFNN, with the machine-translated training data for all target languages allows achieving the best accuracies in the range (0.764–0.800) that is still rather close to our top-line.

In the cross-lingual experiments (under the *cross-lingual* strategy), the best-achieved accuracies are in the accuracy range (0.757–0.794). As we can see, they exceed the threshold of 75%, which is a surprisingly good result without having any training instances in target languages. For German and French, BERT-s + FFNN is a better option, whereas for Lithuanian, Latvian, and Portuguese, on the contrary, BERT-s + COS outperforms BERT-s + FFNN. However, it is still difficult to make hard conclusions on which of these approaches is the better option because differences are not statistically significant for German, Lithuanian, and Portuguese languages. Of the four multilingual sentence embedding models, the *xlm-r-bert-base-nli-stsb-mean-tokens* pre-trained sentence transformer model seems to be a slightly better option for Lithuanian and Portuguese, whereas *xlm-r-distilroberta-base-paraphrase-v1* for German, French, and Latvian.

Both experiments under *MT-based* or *cross-lingual* strategies reveal the superiority of sentence transformers over word transformers. Word embedding-based methods use sequences of concatenated word vectors to represent input texts of the pre-determined length. Due to it, even specific functional words (as articles, modal verbs, etc.) or sentence word-order greatly influence vectors representing those texts. In this respect, languages with relatively free word-order in a sentence (e.g., Lithuanian or Latvian) seem to more suffering: corresponding vectors are more diverse, and therefore it is more difficult to tune the model to be better generalize for some downstream tasks. This phenomenon is especially visible in Figure 2: the training is conducted with the English language; therefore, a model cannot adjust to different word orders. However, a sentence is not a sequence of words but their cumulative semantical meaning. Despite different syntactic and grammatical rules in different languages, the meanings of sentences in different languages remain the same. Sentence embeddings accumulate the meaning of the vectorized text as a whole and therefore seem a more natural and more appropriate way to represent texts for any language.

Results under the *combined* strategy show that *MT-based* and *cross-lingual* strategies are complementary. The best-achieved accuracies exceed the threshold of 80%, are in the interval [0.810–0.831], and are very close to our top-line equal to ~0.842. It seems that having machine-translated training instances in the target language (besides the training instances in English) boosts the accuracy level by ~5%, and this increase is considered statistically significant. In this setting BERT-s + FFNN approach is superior to BERT-s + COS, except for French and Portuguese languages, for which differences between these two approaches are insignificant. The best performing sentence embedding model is *xlm-r-distilroberta-base-paraphrase-v1*, except for Portuguese. The *xlm-r-bert-base-nli-stsb-mean-tokens* is the best for Portuguese, however the difference from *xlm-r-distilroberta-base-paraphrase-v1* is less than 0.3% and insignificant.

The interval of the best-achieved accuracies under the *train all* strategy is much wider (0.775–0.853) compared to all previously discussed, which means that larger diversity in the training data does not always lead to better performance. At the same time, it demonstrates how strongly the accuracy depends on the target language. If the Latvian language benefits the least, Lithuanian, on the contrary, benefits the most. Additionally, it is very difficult to explain why two similar Baltic languages (Lithuanian and Latvian) obtain such contradictory results. Surprisingly, the accuracy (~0.853) for the Lithuanian language even slightly exceeds our top-line. It allows us to conclude cautiously that very good results can be achieved even without any data in the target language, only with the correctly chosen technique. The best approach under the *train all* strategy is also BERT-s + FFNN, whereas for Latvian and Portuguese, BERT-s + COS from BERT-s + FFNN differ insignificantly.

To be able to compare the performance of all four training data usage strategies, we have summarized the best-achieved accuracies in Figure 5. The winner is the *train all* with the BERT-s + FFNN method, followed by *combined* (except for Latvian, where it is vice versa).

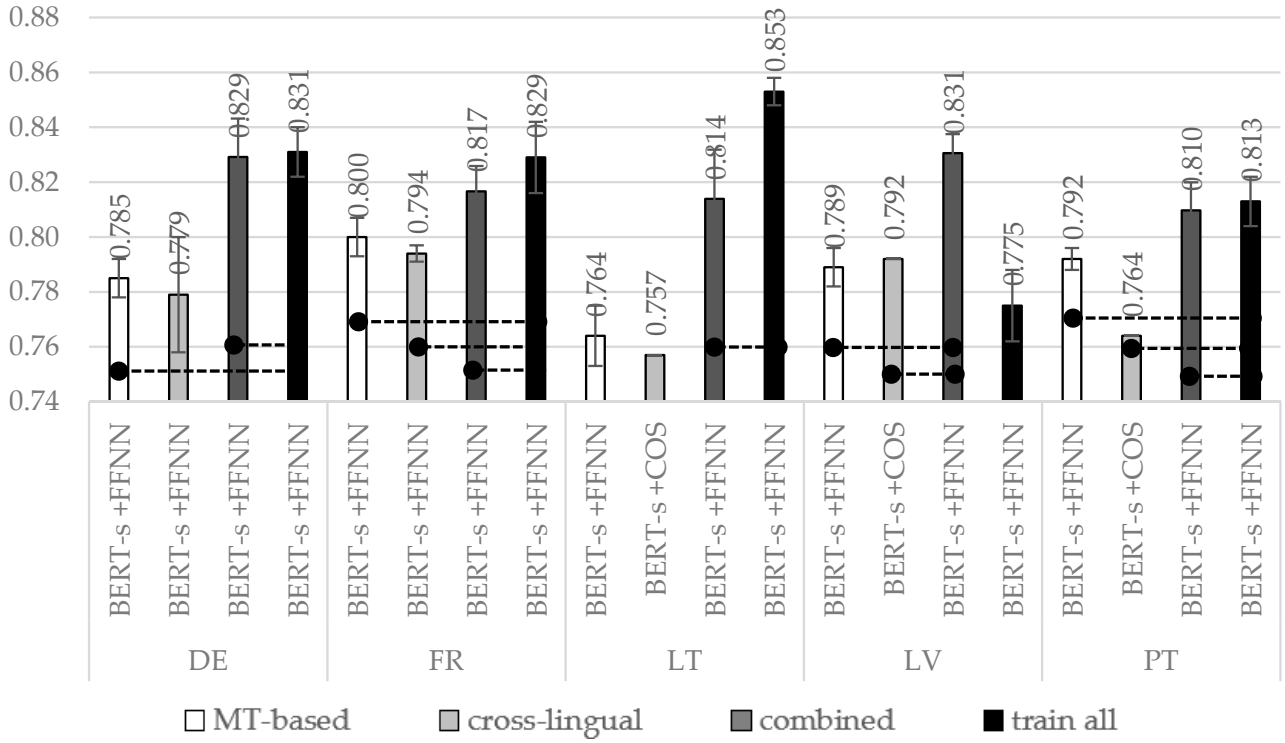

Figure 5. The best accuracies + confidence intervals for different languages under different conditions. For the notation, see Figure 1.

Overall, the results seem promising, especially having in mind that experiments for these target languages (i.e., German, French, Lithuanian, Latvian, and Portuguese) were performed without any original training data. Intent detection experiments strongly rely on the quality of machine translations (under *MT-based*, *combined*, and *train all* strategies). The review of German, French, Latvian, Lithuanian, and Portuguese machine translations confirmed that even though the translations are not always very precise, the gist is still retained. Despite this, Google is not the only machine translation tool, and it is always recommended to choose the best one. Moreover, the training data (even in the machine-translated form) for these languages is not mandatory. Rather good results can be achieved only with cross-lingual models transferred from training on English alone. All of it encourages us to continue experiments in the future by testing more approaches, more models on more datasets for more languages.

6. Conclusions and Future Work

In this research, we attempt to solve a two-fold problem: (1) a supervised intent detection problem for several languages (English, German, French, Lithuanian, Latvian, and Portuguese); (2) the annotated data scarcity problem, because such training data for some languages does not exist. For this reason, the English training dataset (containing 41 intent) was Google machine-translated into five target languages.

The intent detection problem was solved by using two BERT-based vectorization types (i.e., word and sentence embeddings) together with three eager learning classifiers (CNN, BERT fine-tuning, FFNN) and one lazy learning approach (Cosine similarity as the memory-based method). The annotated data scarcity problem was tackled by testing the following training data usage strategies: *MT-based* (when relying on the machine-translated training data), *cross-lingual* (when training on either English alone), *combined* (cross-lingual complemented with the machine-translated instances of the target language), and *train all* (cross-lingual complemented with the machine-translated instances in multiple languages excluding the target one). The experiments revealed the superiority of the *combined* and *train all* strategies on all five target languages. The experiments revealed the superiority of sentence transformers over word embeddings; in particular, FFNN applied on top of BERT-based sentence embeddings over the rest.

The best accuracy of ~0.842 (which is also our top-line) on the English language dataset is achieved with completely monolingual models (monolingual vectorization and monolingual classification method). However, without the original training dataset, similar accuracy levels equal to ~0.831, ~0.829, ~0.853, ~0.831, and ~0.813 were achieved for other languages like German, French, Lithuanian, Latvian, and Portuguese, respectively.

Thus, our research investigation claims the hypothesis that regardless of the tested language, the multilingual intent detection problem can be solved effectively (reaching similar accuracy levels >0.8 as in the monolingual experiments with the original English dataset) even without training data originally prepared for the target language. It allows us to assume that this hypothesis holds for the other languages (at least similar to the tested ones: i.e., from Germanic, Romanic, and Balto-Slavic branches). Moreover, since the intent detection problem is a typical text classification problem, the findings of this research allow us to assume that multilingual text classification problems can also be solved with similar approaches. In future research, it would be interesting to investigate both assumptions by including more languages, more domains, and solving other text classification problems.

Author Contributions: Conceptualization, J.K.-D., A.S., and R.S.; methodology, J.K.-D. and A.S.; software, J.K.-D.; validation, J.K.-D.; formal analysis, J.K.-D.; investigation, J.K.-D. and A.S.; resources, J.K.-D. and A.S.; data curation, J.K.-D. and R.S.; writing—original draft preparation, J.K.-D. and A.S.; writing—review and editing, R.S.; visualization, J.K.-D.; supervision, R.S.; project administration, R.S.; funding acquisition, R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by the European Regional Development Fund within the joint project of SIA TILDE and the University of Latvia "Multilingual Artificial Intelligence Based Human Computer Interaction" No. 1.1.1.1/18/A/148. The research presented in this paper has also received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 825081 and under the name COMPRISE (Cost-effective, Multilingual, Privacy-driven voice-enabled Services).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Experiments with the BERT-w + CNN method under the *MT-based* condition. The table contains averaged *accuracy, precision, recall, and f-score values* followed by confidence intervals. The best results for each target language are emphasized in bold.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
EN	bert-base-cased	0.697 ± 0.016	0.778 ± 0.017	0.674 ± 0.027	0.722 ± 0.020
	bert-base-uncased	0.714 ± 0.014	0.793 ± 0.009	0.689 ± 0.016	0.737 ± 0.011
	bert-large-cased	0.714 ± 0.019	0.794 ± 0.024	0.685 ± 0.016	0.735 ± 0.019
	bert-large-uncased	0.653 ± 0.022	0.739 ± 0.023	0.628 ± 0.022	0.679 ± 0.021
	bert-base-multilingual-cased	0.703 ± 0.019	0.782 ± 0.017	0.672 ± 0.019	0.723 ± 0.018
	bert-base-multilingual-uncased	**0.732** ± 0.009	**0.801** ± 0.014	**0.696** ± 0.012	**0.745** ± 0.012
DE	bert-base-multilingual-cased	0.614 ± 0.026	0.745 ± 0.034	0.560 ± 0.028	0.639 ± 0.027
	bert-base-multilingual-uncased	**0.624** ± 0.020	**0.748** ± 0.019	**0.578** ± 0.025	**0.652** ± 0.022
FR	bert-base-multilingual-cased	0.640 ± 0.014	0.743 ± 0.020	0.601 ± 0.014	0.665 ± 0.009
	bert-base-multilingual-uncased	**0.651** ± 0.023	**0.781** ± 0.029	**0.628** ± 0.027	**0.696** ± 0.027
LT	bert-base-multilingual-cased	0.651 ± 0.015	**0.776** ± 0.018	**0.572** ± 0.017	**0.659** ± 0.017
	bert-base-multilingual-uncased	**0.653** ± 0.025	0.774 ± 0.025	0.569 ± 0.025	0.656 ± 0.024
LV	bert-base-multilingual-cased	0.651 ± 0.018	0.783 ± 0.027	0.612 ± 0.021	0.687 ± 0.020
	bert-base-multilingual-uncased	**0.679** ± 0.013	**0.783** ± 0.012	**0.643** ± 0.012	**0.706** ± 0.010
PT	bert-base-multilingual-cased	**0.649** ± 0.027	**0.762** ± 0.022	**0.600** ± 0.037	**0.670** ± 0.019
	bert-base-multilingual-uncased	0.632 ± 0.015	0.725 ± 0.022	0.597 ± 0.026	0.655 ± 0.024

Table A2. Experiments with the BERT-w + BERT method under the *MT-based* strategy. For the notation, see Table A1.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
EN	bert-base-cased	0.749 ± 0.019	0.827 ± 0.021	0.739 ± 0.023	0.780 ± 0.020
	bert-base-uncased	0.765 ± 0.033	**0.836** ± 0.016	0.751 ± 0.027	0.791 ± 0.020
	bert-large-cased	0.688 ± 0.150	0.813 ± 0.009	0.678 ± 0.201	0.719 ± 0.156
	bert-large-uncased	**0.782** ± 0.005	0.828 ± 0.014	**0.776** ± 0.010	**0.801** ± 0.010
	bert-base-multilingual-cased	0.732 ± 0.011	0.804 ± 0.009	0.712 ± 0.011	0.755 ± 0.009
	bert-base-multilingual-uncased	0.768 ± 0.027	0.829 ± 0.022	0.768 ± 0.027	0.798 ± 0.022
DE	bert-base-multilingual-cased	**0.704** ± 0.034	**0.770** ± 0.035	**0.681** ± 0.036	**0.723** ± 0.035
	bert-base-multilingual-uncased	0.703 ± 0.029	0.760 ± 0.026	0.672 ± 0.031	0.713 ± 0.027
FR	bert-base-multilingual-cased	0.715 ± 0.017	0.785 ± 0.012	0.702 ± 0.023	0.741 ± 0.014
	bert-base-multilingual-uncased	**0.733** ± 0.035	**0.794** ± 0.022	**0.718** ± 0.031	**0.754** ± 0.024
LT	bert-base-multilingual-cased	0.692 ± 0.022	0.741 ± 0.023	0.669 ± 0.029	0.703 ± 0.021
	bert-base-multilingual-uncased	**0.721** ± 0.020	**0.767** ± 0.024	**0.688** ± 0.024	**0.725** ± 0.024
LV	bert-base-multilingual-cased	0.710 ± 0.016	0.785 ± 0.014	0.700 ± 0.021	0.740 ± 0.017
	bert-base-multilingual-uncased	**0.731** ± 0.019	**0.798** ± 0.023	**0.729** ± 0.028	**0.762** ± 0.025
PT	bert-base-multilingual-cased	**0.703** ± 0.031	**0.778** ± 0.028	**0.679** ± 0.031	**0.725** ± 0.027
	bert-base-multilingual-uncased	0.699 ± 0.018	0.773 ± 0.019	0.673 ± 0.017	0.719 ± 0.012

Table A3. Experiments with the BERT-s + FFNN method under the *MT-based* strategy. For the notation, see Table A1.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
EN	roberta-base-nli-stsb-mean-tokens	**0.842** ± 0.011	0.774 ± 0.018	**0.806** ± 0.009	0.762 ± 0.006
	roberta-large-nli-stsb-mean-tokens	0.808 ± 0.014	0.871 ± 0.014	0.795 ± 0.017	0.831 ± 0.015
	bert-large-nli-stsb-mean-tokens	0.817 ± 0.009	0.863 ± 0.019	0.806 ± 0.012	0.833 ± 0.015
	distilbert-base-nli-stsb-mean-tokens	0.799 ± 0.009	0.857 ± 0.014	0.785 ± 0.011	0.819 ± 0.010
	distiluse-base-multilingual-cased-v2	0.760 ± 0.020	0.843 ± 0.019	0.728 ± 0.024	0.781 ± 0.021
	xlm-r-distilroberta-base-paraphrase-v1	0.806 ± 0.011	**0.872** ± 0.011	0.793 ± 0.015	**0.831** ± 0.011
	xlm-r-bert-base-nli-stsb-mean-tokens	0.806 ± 0.006	0.857 ± 0.013	0.789 ± 0.011	0.821 ± 0.008
	distilbert-multilingual-nli-stsb-quora-ranking	0.790 ± 0.008	0.835 ± 0.009	0.770 ± 0.015	0.801 ± 0.011

Table 3. *Cont.*

Language	BERT Model	Accuracy	Precision	Recall	F-Score
DE	distiluse-base-multilingual-cased-v2	0.735 ± 0.010	0.833 ± 0.011	0.703 ± 0.014	0.762 ± 0.006
	xlm-r-distilroberta-base-paraphrase-v1	**0.785** ± 0.007	0.836 ± 0.012	**0.770** ± 0.008	0.802 ± 0.009
	xlm-r-bert-base-nli-stsb-mean-tokens	0.774 ± 0.005	**0.849** ± 0.016	0.769 ± 0.010	**0.807** ± 0.011
	distilbert-multilingual-nli-stsb-quora-ranking	0.692 ± 0.022	0.785 ± 0.018	0.678 ± 0.024	0.727 ± 0.021
FR	distiluse-base-multilingual-cased-v2	0.731 ± 0.012	0.824 ± 0.011	0.696 ± 0.018	0.754 ± 0.015
	xlm-r-distilroberta-base-paraphrase-v1	0.782 ± 0.011	0.840 ± 0.014	0.760 ± 0.016	0.798 ± 0.013
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.800** ± 0.007	**0.843** ± 0.003	**0.791** ± 0.010	**0.816** ± 0.005
	distilbert-multilingual-nli-stsb-quora-ranking	0.754 ± 0.016	0.784 ± 0.017	0.724 ± 0.020	0.753 ± 0.017
LT	distiluse-base-multilingual-cased-v2	0.640 ± 0.005	0.776 ± 0.011	0.571 ± 0.010	0.658 ± 0.006
	xlm-r-distilroberta-base-paraphrase-v1	**0.764** ± 0.011	**0.843** ± 0.011	0.706 ± 0.019	0.768 ± 0.015
	xlm-r-bert-base-nli-stsb-mean-tokens	0.732 ± 0.021	0.803 ± 0.018	0.692 ± 0.024	0.744 ± 0.022
	distilbert-multilingual-nli-stsb-quora-ranking	0.751 ± 0.010	0.825 ± 0.021	**0.718** ± 0.008	0.768 ± 0.013
LV	distiluse-base-multilingual-cased-v2	0.685 ± 0.013	0.814 ± 0.011	0.660 ± 0.019	0.729 ± 0.015
	xlm-r-distilroberta-base-paraphrase-v1	**0.789** ± 0.007	**0.869** ± 0.004	0.756 ± 0.013	**0.809** ± 0.007
	xlm-r-bert-base-nli-stsb-mean-tokens	0.786 ± 0.011	0.844 ± 0.019	0.744 ± 0.015	0.791 ± 0.015
	distilbert-multilingual-nli-stsb-quora-ranking	0.756 ± 0.007	0.818 ± 0.011	**0.761** ± 0.009	0.789 ± 0.010
PT	distiluse-base-multilingual-cased-v2	0.700 ± 0.021	0.802 ± 0.022	0.669 ± 0.023	0.730 ± 0.022
	xlm-r-distilroberta-base-paraphrase-v1	0.779 ± 0.017	**0.885** ± 0.008	0.777 ± 0.019	0.827 ± 0.013
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.792** ± 0.004	0.856 ± 0.011	**0.789** ± 0.007	**0.821** ± 0.007
	distilbert-multilingual-nli-stsb-quora-ranking	0.750 ± 0.016	0.809 ± 0.016	0.733 ± 0.026	0.770 ± 0.020

Table A4. Experiments with the BERT-s + COS method under the *MT-based* strategy. The best results for each language are presented in bold.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
EN	roberta-base-nli-stsb-mean-tokens	0.764	0.833	0.748	0.788
	roberta-large-nli-stsb-mean-tokens	0.757	0.837	0.752	0.793
	bert-large-nli-stsb-mean-tokens	**0.813**	0.862	**0.800**	**0.830**
	distilbert-base-nli-stsb-mean-tokens	0.757	0.841	0.744	0.790
	distiluse-base-multilingual-cased-v2	0.694	0.824	0.679	0.745
	xlm-r-distilroberta-base-paraphrase-v1	0.708	0.801	0.680	0.735
	xlm-r-bert-base-nli-stsb-mean-tokens	0.778	**0.864**	0.769	0.814
	distilbert-multilingual-nli-stsb-quora-ranking	0.715	0.822	0.732	0.775
DE	distiluse-base-multilingual-cased-v2	0.736	0.811	0.749	0.779
	xlm-r-distilroberta-base-paraphrase-v1	**0.771**	**0.857**	**0.768**	**0.810**
	xlm-r-bert-base-nli-stsb-mean-tokens	0.757	0.852	0.739	0.792
	distilbert-multilingual-nli-stsb-quora-ranking	0.674	0.797	0.693	0.741
FR	distiluse-base-multilingual-cased-v2	0.688	0.777	0.683	0.727
	xlm-r-distilroberta-base-paraphrase-v1	0.743	**0.864**	0.737	0.795
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.771**	0.863	**0.761**	**0.808**
	distilbert-multilingual-nli-stsb-quora-ranking	0.701	0.830	0.717	0.769
LT	distiluse-base-multilingual-cased-v2	0.667	0.791	0.648	0.713
	xlm-r-distilroberta-base-paraphrase-v1	0.674	0.798	0.626	0.702
	xlm-r-bert-base-nli-stsb-mean-tokens	0.694	**0.808**	0.658	**0.725**
	distilbert-multilingual-nli-stsb-quora-ranking	**0.708**	0.753	**0.688**	0.719
LV	distiluse-base-multilingual-cased-v2	0.653	0.798	0.681	0.735
	xlm-r-distilroberta-base-paraphrase-v1	0.694	0.816	0.690	0.747
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.743**	**0.827**	**0.723**	**0.772**
	distilbert-multilingual-nli-stsb-quora-ranking	0.667	0.749	0.719	0.734
PT	distiluse-base-multilingual-cased-v2	0.639	0.749	0.651	0.697
	xlm-r-distilroberta-base-paraphrase-v1	**0.778**	**0.870**	**0.776**	**0.820**
	xlm-r-bert-base-nli-stsb-mean-tokens	0.771	0.868	0.742	0.800
	distilbert-multilingual-nli-stsb-quora-ranking	0.708	0.796	0.701	0.746

Table A5. Experiments with the BERT-w + CNN method under the *cross-lingual* strategy. For the notation, see Table A1.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
DE	bert-base-multilingual-cased	0.369 ± 0.008	0.710 ± 0.018	0.318 ± 0.015	0.439 ± 0.015
	bert-base-multilingual-uncased	**0.504** ± 0.010	**0.738** ± 0.034	**0.436** ± 0.019	**0.547** ± 0.020
FR	bert-base-multilingual-cased	0.435 ± 0.031	0.711 ± 0.029	0.359 ± 0.030	0.476 ± 0.022
	bert-base-multilingual-uncased	**0.496** ± 0.025	**0.748** ± 0.030	**0.445** ± 0.031	**0.557** ± 0.026
LT	bert-base-multilingual-cased	0.219 ± 0.016	0.702 ± 0.020	0.197 ± 0.025	0.307 ± 0.031
	bert-base-multilingual-uncased	**0.261** ± 0.022	**0.669** ± 0.049	**0.246** ± 0.033	**0.359** ± 0.039
LV	bert-base-multilingual-cased	0.222 ± 0.026	0.686 ± 0.048	0.191 ± 0.023	0.298 ± 0.027
	bert-base-multilingual-uncased	**0.336** ± 0.041	**0.687** ± 0.029	**0.271** ± 0.035	**0.387** ± 0.035
PT	bert-base-multilingual-cased	0.410 ± 0.071	0.757 ± 0.054	0.324 ± 0.061	0.449 ± 0.063
	bert-base-multilingual-uncased	**0.499** ± 0.033	**0.769** ± 0.016	**0.399** ± 0.053	**0.524** ± 0.047

Table A6. Experiments with the BERT-w + BERT method under the *cross-lingual* strategy. For the notation, see Table A1.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
DE	bert-base-multilingual-cased	0.525 ± 0.034	0.694 ± 0.039	0.512 ± 0.023	0.589 ± 0.024
	bert-base-multilingual-uncased	**0.588** ± 0.034	**0.748** ± 0.023	**0.573** ± 0.034	**0.648** ± 0.029
FR	bert-base-multilingual-cased	0.568 ± 0.037	0.724 ± 0.067	0.544 ± 0.060	0.621 ± 0.060
	bert-base-multilingual-uncased	**0.578** ± 0.035	**0.746** ± 0.072	**0.570** ± 0.054	**0.646** ± 0.060
LT	bert-base-multilingual-cased	0.215 ± 0.026	**0.697** ± 0.050	0.259 ± 0.014	0.377 ± 0.009
	bert-base-multilingual-uncased	**0.319** ± 0.048	0.644 ± 0.024	**0.342** ± 0.041	**0.446** ± 0.040
LV	bert-base-multilingual-cased	0.303 ± 0.054	**0.697** ± 0.064	0.326 ± 0.027	0.444 ± 0.037
	bert-base-multilingual-uncased	**0.386** ± 0.035	0.694 ± 0.029	**0.383** ± 0.032	**0.493** ± 0.029
PT	bert-base-multilingual-cased	0.536 ± 0.035	0.676 ± 0.020	0.498 ± 0.044	0.572 ± 0.034
	bert-base-multilingual-uncased	**0.576** ± 0.035	**0.724** ± 0.017	**0.551** ± 0.040	**0.625** ± 0.030

Table A7. Experiments with the BERT-s + FFNN method under the *cross-lingual* strategy. For the notation, see Table A1.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
DE	distiluse-base-multilingual-cased-v2	0.760 ± 0.012	0.832 ± 0.012	0.716 ± 0.013	0.769 ± 0.010
	xlm-r-distilroberta-base-paraphrase-v1	**0.779** ± 0.021	**0.870** ± 0.014	**0.750** ± 0.025	**0.806** ± 0.020
	xlm-r-bert-base-nli-stsb-mean-tokens	0.749 ± 0.010	0.838 ± 0.013	0.730 ± 0.007	0.780 ± 0.009
	distilbert-multilingual-nli-stsb-quora-ranking	0.700 ± 0.010	0.818 ± 0.025	0.677 ± 0.011	0.741 ± 0.013
FR	distiluse-base-multilingual-cased-v2	0.718 ± 0.016	0.823 ± 0.024	0.690 ± 0.020	0.751 ± 0.022
	xlm-r-distilroberta-base-paraphrase-v1	**0.794** ± 0.003	**0.878** ± 0.012	**0.774** ± 0.004	**0.823** ± 0.004
	xlm-r-bert-base-nli-stsb-mean-tokens	0.767 ± 0.014	0.851 ± 0.022	0.741 ± 0.014	0.792 ± 0.017
	distilbert-multilingual-nli-stsb-quora-ranking	0.707 ± 0.009	0.821 ± 0.018	0.676 ± 0.018	0.741 ± 0.010
LT	distiluse-base-multilingual-cased-v2	0.647 ± 0.024	0.754 ± 0.034	0.614 ± 0.034	0.677 ± 0.034
	xlm-r-distilroberta-base-paraphrase-v1	0.657 ± 0.023	0.811 ± 0.016	0.610 ± 0.014	0.696 ± 0.010
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.754** ± 0.009	**0.841** ± 0.013	**0.717** ± 0.012	**0.774** ± 0.012
	distilbert-multilingual-nli-stsb-quora-ranking	0.625 ± 0.018	0.759 ± 0.025	0.585 ± 0.012	0.660 ± 0.016
LV	distiluse-base-multilingual-cased-v2	0.613 ± 0.015	0.768 ± 0.029	0.575 ± 0.018	0.657 ± 0.020
	xlm-r-distilroberta-base-paraphrase-v1	0.726 ± 0.016	**0.852** ± 0.015	**0.695** ± 0.025	**0.765** ± 0.015
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.729** ± 0.018	0.825 ± 0.014	0.682 ± 0.025	0.747 ± 0.020
	distilbert-multilingual-nli-stsb-quora-ranking	0.618 ± 0.010	0.778 ± 0.021	0.580 ± 0.017	0.664 ± 0.017
PT	distiluse-base-multilingual-cased-v2	**0.738** ± 0.019	0.853 ± 0.014	0.714 ± 0.026	0.777 ± 0.020
	xlm-r-distilroberta-base-paraphrase-v1	0.771 ± 0.011	0.864 ± 0.013	**0.743** ± 0.016	0.799 ± 0.013
	xlm-r-bert-base-nli-stsb-mean-tokens	0.771 ± 0.011	**0.875** ± 0.006	0.742 ± 0.009	**0.803** ± 0.007
	distilbert-multilingual-nli-stsb-quora-ranking	0.699 ± 0.007	0.800 ± 0.017	0.675 ± 0.012	0.732 ± 0.012

Table A8. Experiments with the BERT-s + COS method under the *cross-lingual* strategy. The best results for each language are presented in bold.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
DE	distiluse-base-multilingual-cased-v2	0.715	0.809	0.687	0.743
	xlm-r-distilroberta-base-paraphrase-v1	0.764	0.841	0.736	0.785
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.771**	**0.859**	**0.767**	**0.811**
	distilbert-multilingual-nli-stsb-quora-ranking	0.667	0.724	0.657	0.689
FR	distiluse-base-multilingual-cased-v2	0.653	0.792	0.669	0.725
	xlm-r-distilroberta-base-paraphrase-v1	0.701	**0.851**	0.691	**0.763**
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.722**	0.826	**0.692**	0.753
	distilbert-multilingual-nli-stsb-quora-ranking	0.660	0.740	0.646	0.690
LT	distiluse-base-multilingual-cased-v2	0.618	0.749	0.617	0.677
	xlm-r-distilroberta-base-paraphrase-v1	0.736	**0.897**	0.702	0.788
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.757**	0.830	**0.761**	**0.794**
	distilbert-multilingual-nli-stsb-quora-ranking	0.611	0.721	0.614	0.663
LV	distiluse-base-multilingual-cased-v2	0.576	0.727	0.599	0.657
	xlm-r-distilroberta-base-paraphrase-v1	**0.792**	**0.882**	**0.756**	**0.814**
	xlm-r-bert-base-nli-stsb-mean-tokens	0.778	0.842	0.748	0.792
	distilbert-multilingual-nli-stsb-quora-ranking	0.597	0.760	0.623	0.685
PT	distiluse-base-multilingual-cased-v2	0.674	0.815	0.663	0.731
	xlm-r-distilroberta-base-paraphrase-v1	0.743	0.859	0.723	0.785
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.764**	**0.860**	**0.739**	**0.795**
	distilbert-multilingual-nli-stsb-quora-ranking	0.681	0.779	0.692	0.733

Table A9. Experiments with the BERT-s + FFNN method under the *combined* strategy. For the other notation, see Table A1.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
DE	distiluse-base-multilingual-cased-v2	0.781 ± 0.009	0.843 ± 0.012	0.762 ± 0.018	0.800 ± 0.014
	xlm-r-distilroberta-base-paraphrase-v1	**0.829 ± 0.014**	**0.880 ± 0.009**	**0.818 ± 0.015**	**0.848 ± 0.012**
	xlm-r-bert-base-nli-stsb-mean-tokens	0.779 ± 0.010	0.863 ± 0.009	0.766 ± 0.015	0.811 ± 0.010
	distilbert-multilingual-nli-stsb-quora-ranking	0.733 ± 0.007	0.811 ± 0.017	0.716 ± 0.010	0.761 ± 0.006
FR	distiluse-base-multilingual-cased-v2	0.787 ± 0.013	0.846 ± 0.007	0.765 ± 0.020	0.803 ± 0.014
	xlm-r-distilroberta-base-paraphrase-v1	**0.817 ± 0.009**	**0.870 ± 0.009**	**0.807 ± 0.012**	**0.837 ± 0.011**
	xlm-r-bert-base-nli-stsb-mean-tokens	0.800 ± 0.015	0.844 ± 0.008	0.791 ± 0.014	0.817 ± 0.011
	distilbert-multilingual-nli-stsb-quora-ranking	0.785 ± 0.009	0.820 ± 0.012	0.781 ± 0.010	0.800 ± 0.009
LT	distiluse-base-multilingual-cased-v2	0.729 ± 0.004	0.800 ± 0.010	0.708 ± 0.011	0.751 ± 0.007
	xlm-r-distilroberta-base-paraphrase-v1	**0.814 ± 0.018**	**0.875 ± 0.012**	**0.779 ± 0.026**	**0.824 ± 0.019**
	xlm-r-bert-base-nli-stsb-mean-tokens	0.767 ± 0.018	0.826 ± 0.010	0.731 ± 0.018	0.776 ± 0.014
	distilbert-multilingual-nli-stsb-quora-ranking	0.765 ± 0.008	0.823 ± 0.010	0.752 ± 0.011	0.786 ± 0.008
LV	distiluse-base-multilingual-cased-v2	0.739 ± 0.020	0.824 ± 0.031	0.755 ± 0.020	0.787 ± 0.022
	xlm-r-distilroberta-base-paraphrase-v1	**0.831 ± 0.007**	**0.892 ± 0.010**	**0.804 ± 0.007**	**0.846 ± 0.006**
	xlm-r-bert-base-nli-stsb-mean-tokens	0.800 ± 0.019	0.864 ± 0.016	0.756 ± 0.022	0.807 ± 0.019
	distilbert-multilingual-nli-stsb-quora-ranking	0.740 ± 0.011	0.827 ± 0.014	0.738 ± 0.013	0.780 ± 0.012
PT	distiluse-base-multilingual-cased-v2	0.761 ± 0.018	0.845 ± 0.013	0.749 ± 0.018	0.794 ± 0.013
	xlm-r-distilroberta-base-paraphrase-v1	0.807 ± 0.005	**0.874 ± 0.007**	0.793 ± 0.008	**0.831 ± 0.007**
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.810 ± 0.010**	0.867 ± 0.008	**0.796 ± 0.013**	0.830 ± 0.010
	distilbert-multilingual-nli-stsb-quora-ranking	0.781 ± 0.011	0.856 ± 0.008	0.765 ± 0.019	0.808 ± 0.012

Table A10. Experiments with the BERT-s + COS method under the *combined* strategy. The best results for each language are presented in bold.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
DE	distiluse-base-multilingual-cased-v2	0.722	0.802	0.715	0.756
	xlm-r-distilroberta-base-paraphrase-v1	**0.757**	0.832	0.738	0.782
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.757**	*0.861*	**0.739**	**0.796**
	distilbert-multilingual-nli-stsb-quora-ranking	0.681	0.802	0.699	0.747
FR	distiluse-base-multilingual-cased-v2	0.688	0.764	0.681	0.720
	xlm-r-distilroberta-base-paraphrase-v1	0.743	0.859	0.730	0.789
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.778**	**0.871**	**0.773**	**0.819**
	distilbert-multilingual-nli-stsb-quora-ranking	0.715	0.836	0.725	0.776
LT	distiluse-base-multilingual-cased-v2	0.674	0.811	0.681	**0.741**
	xlm-r-distilroberta-base-paraphrase-v1	0.674	0.798	0.626	0.702
	xlm-r-bert-base-nli-stsb-mean-tokens	0.694	**0.816**	0.658	0.728
	distilbert-multilingual-nli-stsb-quora-ranking	**0.708**	0.753	**0.688**	0.719
LV	distiluse-base-multilingual-cased-v2	0.653	0.757	0.681	0.717
	xlm-r-distilroberta-base-paraphrase-v1	0.694	0.816	0.690	0.747
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.750**	**0.831**	**0.735**	**0.780**
	distilbert-multilingual-nli-stsb-quora-ranking	0.667	0.749	0.719	0.734
PT	distiluse-base-multilingual-cased-v2	0.646	0.753	0.655	0.701
	xlm-r-distilroberta-base-paraphrase-v1	0.771	0.854	**0.767**	0.808
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.792**	**0.866**	**0.767**	**0.814**
	distilbert-multilingual-nli-stsb-quora-ranking	0.701	0.795	0.699	0.744

Table A11. Experiments with the BERT-s + FFNN method under the *train all* strategy. For the other notation, see Table A1.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
EN	distiluse-base-multilingual-cased-v2	0.790 ± 0.008	0.836 ± 0.015	0.786 ± 0.012	0.810 ± 0.013
	xlm-r-distilroberta-base-paraphrase-v1	**0.811** ± 0.007	**0.854** ± 0.007	**0.799** ± 0.011	**0.826** ± 0.007
	xlm-r-bert-base-nli-stsb-mean-tokens	0.811 ± 0.023	0.851 ± 0.024	0.796 ± 0.024	0.823 ± 0.022
	distilbert-multilingual-nli-stsb-quora-ranking	0.781 ± 0.009	0.848 ± 0.009	0.750 ± 0.011	0.796 ± 0.008
DE	distiluse-base-multilingual-cased-v2	0.785 ± 0.004	0.835 ± 0.006	0.772 ± 0.010	0.802 ± 0.007
	xlm-r-distilroberta-base-paraphrase-v1	**0.831** ± 0.009	**0.872** ± 0.013	**0.803** ± 0.010	**0.836** ± 0.011
	xlm-r-bert-base-nli-stsb-mean-tokens	0.761 ± 0.010	0.825 ± 0.004	0.746 ± 0.015	0.784 ± 0.008
	distilbert-multilingual-nli-stsb-quora-ranking	0.742 ± 0.012	0.800 ± 0.018	0.718 ± 0.017	0.756 ± 0.010
FR	distiluse-base-multilingual-cased-v2	0.771 ± 0.007	0.844 ± 0.009	0.762 ± 0.010	0.801 ± 0.008
	xlm-r-distilroberta-base-paraphrase-v1	**0.829** ± 0.013	**0.880** ± 0.013	0.820 ± 0.019	0.849 ± 0.016
	xlm-r-bert-base-nli-stsb-mean-tokens	0.818 ± 0.007	0.876 ± 0.004	0.803 ± 0.012	0.838 ± 0.005
	distilbert-multilingual-nli-stsb-quora-ranking	0.728 ± 0.005	0.807 ± 0.009	0.727 ± 0.010	0.765 ± 0.009
LT	distiluse-base-multilingual-cased-v2	0.733 ± 0.013	0.818 ± 0.008	0.744 ± 0.014	0.780 ± 0.011
	xlm-r-distilroberta-base-paraphrase-v1	**0.853** ± 0.005	**0.891** ± 0.005	**0.846** ± 0.008	**0.868** ± 0.005
	xlm-r-bert-base-nli-stsb-mean-tokens	0.793 ± 0.016	0.841 ± 0.016	0.757 ± 0.024	0.797 ± 0.020
	distilbert-multilingual-nli-stsb-quora-ranking	0.754 ± 0.005	0.836 ± 0.008	0.727 ± 0.015	0.778 ± 0.008
LV	distiluse-base-multilingual-cased-v2	0.729 ± 0.004	0.825 ± 0.008	0.739 ± 0.004	0.780 ± 0.004
	xlm-r-distilroberta-base-paraphrase-v1	**0.775** ± 0.013	**0.858** ± 0.009	**0.746** ± 0.010	**0.798** ± 0.010
	xlm-r-bert-base-nli-stsb-mean-tokens	0.775 ± 0.009	0.835 ± 0.021	0.746 ± 0.014	0.788 ± 0.015
	distilbert-multilingual-nli-stsb-quora-ranking	0.656 ± 0.013	0.745 ± 0.021	0.658 ± 0.016	0.699 ± 0.016
PT	distiluse-base-multilingual-cased-v2	0.758 ± 0.008	0.839 ± 0.009	0.756 ± 0.013	0.795 ± 0.011
	xlm-r-distilroberta-base-paraphrase-v1	**0.813** ± 0.009	**0.874** ± 0.005	**0.791** ± 0.015	**0.830** ± 0.006
	xlm-r-bert-base-nli-stsb-mean-tokens	0.792 ± 0.004	0.851 ± 0.019	0.776 ± 0.014	0.812 ± 0.010
	distilbert-multilingual-nli-stsb-quora-ranking	0.700 ± 0.015	0.819 ± 0.021	0.678 ± 0.018	0.742 ± 0.015

Table A12. Experiments with the BERT-s + COS method under the *train all* strategy. The best results for each language are presented in bold.

Language	BERT Model	Accuracy	Precision	Recall	F-Score
EN	distiluse-base-multilingual-cased-v2	0.694	0.802	0.714	0.755
	xlm-r-distilroberta-base-paraphrase-v1	0.708	0.837	0.674	0.747
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.757**	**0.838**	0.726	**0.778**
	distilbert-multilingual-nli-stsb-quora-ranking	**0.757**	0.808	**0.749**	**0.778**
DE	distiluse-base-multilingual-cased-v2	0.694	0.813	0.692	0.748
	xlm-r-distilroberta-base-paraphrase-v1	0.750	0.817	**0.730**	0.771
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.757**	**0.825**	**0.730**	**0.774**
	distilbert-multilingual-nli-stsb-quora-ranking	0.701	0.770	0.680	0.722
FR	distiluse-base-multilingual-cased-v2	0.681	0.790	0.686	0.734
	xlm-r-distilroberta-base-paraphrase-v1	0.722	0.820	0.717	0.765
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.771**	**0.862**	**0.755**	**0.805**
	distilbert-multilingual-nli-stsb-quora-ranking	0.688	0.767	0.683	0.722
LT	distiluse-base-multilingual-cased-v2	0.604	0.740	0.607	0.667
	xlm-r-distilroberta-base-paraphrase-v1	**0.715**	**0.821**	0.672	**0.739**
	xlm-r-bert-base-nli-stsb-mean-tokens	0.708	0.785	**0.683**	0.731
	distilbert-multilingual-nli-stsb-quora-ranking	0.646	0.749	0.650	0.696
LV	distiluse-base-multilingual-cased-v2	0.604	0.759	0.636	0.692
	xlm-r-distilroberta-base-paraphrase-v1	0.688	0.807	0.648	0.718
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.722**	**0.855**	**0.707**	**0.774**
	distilbert-multilingual-nli-stsb-quora-ranking	0.646	0.780	0.671	0.721
PT	distiluse-base-multilingual-cased-v2	0.688	0.808	0.682	0.740
	xlm-r-distilroberta-base-paraphrase-v1	0.750	**0.853**	0.745	0.796
	xlm-r-bert-base-nli-stsb-mean-tokens	**0.785**	0.852	**0.781**	**0.815**
	distilbert-multilingual-nli-stsb-quora-ranking	0.674	0.786	0.684	0.731

References

1. Nithuna, S.; Laseena, C.A. Review on Implementation Techniques of Chatbot. In Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July 2020; pp. 157–161. [CrossRef]
2. Weizenbaum, J. ELIZA–A computer program for the study of natural language communication between man and machine. *Commun. ACM* **1996**, *9*, 36–45. [CrossRef]
3. Bhushan, R.; Kulkarni, K.; Pandey, V.K.; Rawls, C.; Mechtley, B.; Jayasuriya, S.; Ziegler, C. ODO: Design of Multimodal Chatbot for an Experiential Media System. *Multimodal Technol. Interact.* **2020**, *4*, 68. [CrossRef]
4. Maniou, T.A.; Veglis, A. Employing a Chatbot for News Dissemination during Crisis: Design, Implementation and Evaluation. *Future Internet* **2020**, *12*, 109. [CrossRef]
5. Battineni, G.; Chintalapudi, N.; Amenta, F. AI Chatbot Design during an Epidemic like the Novel Coronavirus. *Healthcare* **2020**, *8*, 154. [CrossRef]
6. Adamopoulou, E.; Moussiades, L. An Overview of Chatbot Technology. In *Artificial Intelligence Applications and Innovations*; Maglogiannis, I., Iliadis, L., Pimenidis, E., Eds.; IFIP Advances in Information and Communication Technology; Springer: Cham, Switzerland, 2020; Volume 584, pp. 373–383. [CrossRef]
7. Hemphill, C.T.; Godfrey, J.J.; Doddington, G.R. The ATIS Spoken Language Systems Pilot Corpus. In Proceedings of the Speech and Natural Language, Hidden Valley, PA, USA, 24–27 June 1990.
8. Wang, Y.; Shen, Y.; Jin, H. A Bi-Model Based RNN Semantic Frame Parsing Model for Intent Detection and Slot Filling. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 309–314. [CrossRef]
9. Obuchowski, A.; Lew, M. Transformer-Capsule Model for Intent Detection (Student Abstract). In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13885–13886. [CrossRef]
10. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [CrossRef]
11. Liu, B.; Lane, I. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In Proceedings of the Interspeech, San Francisco, CA, USA, 8–12 September 2016; pp. 685–689. [CrossRef]
12. Qin, L.; Che, W.; Li, Y.; Wen, H.; Liu, T. A Stack-Propagation Framework with Token-Level Intent Detection for Spoken Language Understanding. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th

International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 9 July 2019; pp. 2078–2087. [CrossRef]
13. Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; et al. Snips Voice Platform: An embedded Spoken Language Understanding system for private-by-design voice interfaces. *arXiv* **2018**, arXiv:1805.10190.
14. Veyseh, A.P.B.; Dernoncourt, F.; Nguyen, T.H. Improving Slot Filling by Utilizing Contextual Information. *arXiv* **2019**, arXiv:1911.01680.
15. Haihong, E.; Niu, P.; Chen, Z.; Song, M. A Novel Bi-directional Interrelated Model for Joint Intent Detection and Slot Filling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy, 8 July–2 August 2019; pp. 5467–5471. [CrossRef]
16. Cao, X.; Xiong, D.; Shi, C.; Wang, C.; Meng, Y.; Hu, C. Balanced Joint Adversarial Training for Robust Intent Detection and Slot Filling. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 4926–4936. [CrossRef]
17. Casanueva, I.; Temčinas, T.; Gerz, D.; Henderson, M.; Vulič, I. Efficient Intent Detection with Dual Sentence Encoders. In Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, Seattle, WA, USA, 9 July 2020; pp. 38–45. [CrossRef]
18. Alibadi, Z.; Du, M.; Vidal, J.M. Using Pre-trained Embeddings to Detect the Intent of an Email. ACIT 2019. In Proceedings of the 7th ACIS International Conference on Applied Computing and Information Technology, Honolulu, HI, USA, 29–31 May 2019; Article No.: 2. pp. 1–7. [CrossRef]
19. Nguyen, H.; Zhang, C.; Xia, C.; Yu, P. Dynamic Semantic Matching and Aggregation Network for Few-shot Intent Detection. In Proceedings of the Association for Computational Linguistics: EMNLP 2020, Seattle, WA, USA, 26 May 2021; pp. 1209–1218. [CrossRef]
20. Dopierre, T.; Gravier, C.; Subercaze, J.; Logerais, W. Few-shot Pseudo-Labeling for Intent Detection. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 4993–5003. [CrossRef]
21. Qin, L.; Xu, X.; Che, W.; Liu, T. AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling. In Proceedings of the Association for Computational Linguistics: EMNLP 2020, Seattle, WA, USA, 5–10 July 2020; pp. 1807–1816. [CrossRef]
22. Tan, L.; Golovneva, O. Evaluating Cross-Lingual Transfer Learning Approaches in Multilingual Conversational Agent Models. In Proceedings of the 28th International Conference on Computational Linguistics: Industry Track, Barcelona, Spain, 8–11 December 2020; pp. 1–9. [CrossRef]
23. Schuster, S.; Gupta, S.; Shah, R.; Lewis, M. Cross-lingual Transfer Learning for Multilingual Task Oriented Dialog. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 3795–3805. [CrossRef]
24. Arora, A.; Shrivastava, A.; Mohit, M.; Lecanda, L.S.M.; Aly, A. Cross-Lingual Transfer Learning for Intent Detection of Covid-19 Utterances. In *EMNLP 2020 Workshop NLP-COVID Submission*; online; 20 November 2020.
25. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzman, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, online. 5–10 July 2020; pp. 8440–8451. [CrossRef]
26. Braun, D.; Hernandez, M.A.; Matthes, F.; Langen, M. Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, 15–17 August 2017; pp. 174–185. [CrossRef]
27. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [CrossRef]
28. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using SiameseBERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) 2019, Hong Kong, China, 3–7 November 2019; pp. 3982–3992. [CrossRef]
29. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
30. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
31. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv* **2019**, arXiv:1906.08237.
32. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
33. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751. [CrossRef]
34. Kapočiūtė-Dzikienė, J.; Balodis, K.; Skadiņš, R. Intent Detection Problem Solving via Automatic DNN Hyperparameter Optimization. *Appl. Sci.* **2020**, *10*, 7426. [CrossRef]

35. Gunawan, D.; Sembiring, C.A.; Budiman, M.A. The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents. *J. Phys. Conf. Ser.* **2018**, *978*, 012120. [CrossRef]
36. McNemar, Q.M. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. *Psychometrika* **1947**, *12*, 153–157. [CrossRef] [PubMed]