Warsaw University of Technology

FACULTY OF
ELECTRONICS AND INFORMATION TECHNOLOGY

Institute of Telecommunications

Ph.D. Thesis

Multilingual Machine Translation System for Dialogue Agents
(Wielojęzyczny system tłumaczenia maszynowego
dla agentów dialogowych)

mgr inż. Marcin Sowański

thesis supervisor
dr hab. inż. Artur Janicki, prof. uczelni

WARSAW 2023

Multilingual Machine Translation System for Dialogue Agents

Abstract. This dissertation presents how machine translation (MT) can be used to translate training and evaluation resources for natural language understanding (NLU) models that are, among others, used in intelligent virtual assistants (IVA). The goal of this thesis is to prove that MT can be used as an efficient tool for language localization in the process of developing IVAs. Samsung's virtual assistant, Bixby, has been provided here as an example of the industrial implementation of this concept. The goal has been met and described in detail in this work. All models, datasets, and the source code described in this dissertation, excluding the resources used in industrial development, have been released to foster further research on this topic.

The idea of using MT models to translate the training set of dialog agents is well described in the literature, but there are no open-source MT models available that are adapted to the IVAs. The quality of not adapted MT models is insufficient and, most importantly, does not transfer semantic annotations used in NLU resources from source to target. State-of-the-art NLU models require various examples for each IVA domain, which causes another problem as MT tends to return the same translation for different source sentences. Solving all these problems would allow the development of dialogue agents for new languages to be cheaper and easier. Moreover, it would let more users use voice-based AI products that currently are mostly available in English.

The first part of this work discusses what resources are needed to build MT adapted to IVA. Available NLU and MT datasets are insufficient regarding domain coverage and the diversity of intents and slots. A new dataset called Leyzer is proposed that addresses that. The dataset is designed to be used as a benchmark for NLU and MT models. Leyzer covers 18 domains with 186 commands across English, Polish, and Spanish. One of the distinguishing features of the dataset is assigning naturalness level and verb patterns to each sentence. This novelty allows us to track the biases of MT and check the quality of translations.

In the second part of this work, the MT domain adaptation technique for the domain of IVA is presented. The performed experiments show how adapting the models with fine-tuning helps improve the results of MT. Created models can transfer semantic annotations used in NLU models, called slots, which solves the fundamental problem of this thesis. The adapted MT model outperformed the baseline models with +19.62 ± 1.6 BLEU points for the English-to-Polish model and +10.45 ± 1.92 BLEU points for the English-to-Spanish model, respectively. Furthermore, the Polish NLU model, trained on data translated by the fine-tuned English-to-Polish MT model, achieved an F_1-score of 87.54% for single-slot sentences and 65.47% for multi-slot sentences. This performance serves as an empirical validation of the MT model's effectiveness in facilitating multilingual NLU.

In the third part of this work, a solution for the absence of variability in MT outputs is presented. Following an in-depth analysis of eight different NLU corpora to identify the most frequently occurring verbs, a verb ontology is developed. This ontology, grounded in WordNet and VerbNet, when integrated with IVA-adapted Machine Translation models, enables the generation of multiple translation variants. This advancement not only captures the nuances of human language but also enriches the user experience in Intelligent Virtual Assistants. The presented model increased intent classification accuracy by 3.8% relative when compared to single-best translation.

Following the discussion of the three key components required for customizing MT, the study delves into its industrial implementation. Each element of this research has been applied commercially to address business challenges associated with localizing Natural Language Understanding (NLU) resources for Bixby, an IVA developed by Samsung Electronics.

This dissertation ends with a list of my academic achievements, including research articles, patents, and presentations I gave. All of these items contributed to this work or are thematically connected with it.

Keywords: machine translation, natural language understanding, intelligent virtual assistants

Wielojęzyczny system tłumaczenia maszynowego dla agentów dialogowych

Streszczenie. Niniejsza rozprawa przedstawia, w jaki sposób tłumaczenie maszynowe (MT) może być wykorzystane do tłumaczenia zasobów uczących i ewaluacyjnych dla modeli rozumienia języka naturalnego (NLU), które są używane między innymi w inteligentnych asystentach wirtualnych (IVA). Celem tej pracy jest udowodnienie, że MT może być efektywnym narzędziem do lokalizacji językowej w procesie rozwijania IVA dla nowych języków. Jako przykład przemysłowego wdrożenia tego konceptu w pracy użyto asystenta wirtualnego Bixby rozwijanego przez firmę Samsung Electronics. Cel rozprawy został osiągnięty i opisany szczegółowo w tej pracy. Wszystkie modele, zbiory danych i kod źródłowy opisane w tej rozprawie, z wyłączeniem zasobów użytych podczas prac wdrożeniowych, zostały udostępnione, aby wspierać dalsze badania w tej dziedzinie.

Pomysł wykorzystania modeli MT do tłumaczenia zbiorów treningowych agentów dialogowych jest dobrze opisany w literaturze, ale brakuje dostępnych modeli MT adaptowanych do domeny IVA. Jakość nieadaptowanych modeli MT jest niewystarczająca i, co najważniejsze, nie pozwala na przenoszenie semantycznych anotacji używanych w zasobach NLU z języka wyjściowego do języka docelowego. Współczesne modele NLU wymagają wielu, różnorodnych przykładów uczących dla każdej domeny IVA, którą obsługują, co prowadzi do kolejnego problemu, ponieważ MT zwykle zwraca to samo tłumaczenie dla różnych bliskoznacznych zdań źródłowych. Rozwiązanie tych problemów pozwoliłoby na tańszy i łatwiejszy rozwój agentów dialogowych dla nowych języków. Ponadto pozwoliłoby to na korzystanie z produktów opartych na AI sterowanych głosem przez większą liczbę użytkowników. Jest to istotne, ponieważ w chwili obecniej większość narzędzi AI dostępnych jest jedynie dla języka angielskiego.

W pierwszej części tej pracy omówiono, jakie zasoby są potrzebne do stworzenia MT zaadaptowanego do domeny IVA. Obecnie dostępne zbiory danych NLU i MT są niewystarczające pod względem ilości dostępnych domen a także różnorodności intencji i slotów. W rozprawie zaproponowano nowy zbiór danych o nazwie Leyzer, który rozwiązuje wymienione problemy. Zbiór ten został zaprojektowany do badania jakości modeli NLU i MT. Leyzer obejmuje 18 domen z 186 intencjami w językach angielskim, polskim i hiszpańskim. Jedną z wyróżniających cech tego zbioru danych jest przypisywanie każdemu zdaniu poziomu naturalności oraz wzorca czasownikowego, do którego należy. Ta cecha, niewystępująca w innych tego typu zasobach, pozwala nam śledzić inklinacje (ang. bias) modeli MT oraz lepiej określać jakość tłumaczeń.

W drugiej części tej pracy przedstawiona jest technika adaptacji domenowej MT dla domeny IVA. Przeprowadzone eksperymenty pokazują, jak adaptowanie modeli za pomocą fine-tuningu pozwala poprawić wyniki MT. Stworzone modele mogą przenosić semantyczne anotacje używane w modelach NLU, nazywane slotami, co rozwiązuje je-

den z trzech głównych problemów zdefiniowany w tej pracy. Zaadaptowany model MT uzyskał lepsze wyniki niż linia bazowa, uzyskując +19,62 ± 1,6 punktów BLEU dla modelu angielsko-polskiego i +10,45 ± 1,92 punktów BLEU dla modelu angielsko-hiszpańskiego. Dodatkowo polski model NLU, wytrenowany na danych przetłumaczonych przez adaptowany angielsko-polski model MT, osiągnął wynik F_1-score na poziomie 87,54% dla zdań zawierających jeden typ slotu oraz 65,47% dla zdań zawierających więcej niż jeden typ slot (encji nazwanej). Wyniki empirycznie potwierdzają skuteczności modelu MT w tworzeniu wielojęzycznego NLU.

W trzeciej części tej pracy przedstawione jest rozwiązanie problemu braku różnorodności w tłumaczeniach zwracanych przez modele MT. Po dogłębnej analizie ośmiu korpusów NLU w celu zidentyfikowania najczęściej występujących czasowników, opracowana została ontologia czasowników. Ontologia wykorzystuje bazy językowe WordNet i VerbNet, które w połączeniu ze zaadaptowanymi modelami MT umożliwiają generowanie wielu wariantów tłumaczenia. Rozwiązanie pozwala lepiej uchwycić niuanse języka naturalnego, a także umożliwia ulepszyć IVA. Zaprezentowany model zwiększył skuteczność klasyfikacji intencji o 3,8% w stosunku do modelu tłumaczącego na jeden wariant.

Po omówieniu trzech kluczowych komponentów niezbędnych do stworzenia adaptowanego MT, w niniejszej rozprawie omówiono wdrożenia przemysłowe. Każdy element wymienionych wcześniej badań został zastosowany komercyjnie. Pozwala to sprostać wyzwaniom biznesowym związanym z lokalizacją zasobów NLU dla asystenta Bixby, IVA opracowanego przez firmę Samsung Electronics.

Rozprawa kończy się listą moich osiągnięć naukowych, w tym artykułów naukowych, patentów i prezentacji, które wygłosiłem. Wszystkie wymienione elementy przyczyniły się do powstania tej pracy lub są z nią tematycznie związane.

Słowa kluczowe: tłumaczenie maszynowe, rozumienie języka naturalnego, inteligentni wirtualni asystenci

Acknowledgements

I would like to express my profound gratitude to Artur for his wisdom, patience, and constructive criticism. My appreciation also goes to my colleagues at Samsung R&D, who collaborated with me on this project. I would like to extend my sincere gratitude to Prof. Maciej Piasecki from the Wroclaw University of Technology, whose invaluable reviews and comments have significantly contributed to the improvement of this thesis. Special thanks to my mom for her constant optimism and support. Above all, I owe a heartfelt thank-you to my wife, Gosia, for her unwavering support and patience; this accomplishment would not have been possible without you.

I dedicate this dissertation to my late father, a wish left unspoken yet ever-present.

Contents

1. Introduction

The work described in this dissertation is the result of my participation in an Industrial Ph.D. program that was established between three institutions: Samsung R&D Institute Poland, the Warsaw University of Technology, and the Ministry of Education and Science. This unique program intersects academic and industry experiences. In this undertaking, I found myself in the vast technological ecosystem of Samsung, contributing to its widely recognized Bixby project. My primary task revolved around architecting and implementing a machine translation system specifically designed to translate NLU training resources into a range of European languages. This endeavor broadened Bixby's linguistic versatility and is a typical example of academic theories being put into practical, industrial use.

Bixby is a virtual assistant developed by Samsung Electronics. Virtual assistants use voice as a natural-language user interface to perform various actions for the user. As presented in Figure 1, Bixby, together with Google Assistant, Apple's Siri, and Amazon's Alexa, is one of the most frequently used dialogue agents. It is estimated that around 80 million U.S. adults use one of them each month [1].

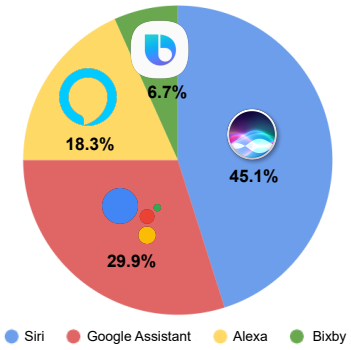

Figure 1. Market Share of Intelligent Virtual Assistants in the U.S. for the Year 2020, Measured in Number of Users (Source: Voicebot.ai [2]).

Bixby is designed to perform many tasks related to device control, to answer users' queries and serve as a chatbot. It is built on a modular framework, with each function encapsulated within what Samsung calls "capsules". These capsules, akin to skills in Amazon's Alexa or actions in Google Assistant, allow Bixby to perform specific tasks. Capsules are developed using Bixby Developer Studio, an integrated development environment (IDE), that provides tools for creating, testing, and publishing these capsules. Currently, Bixby supports eight languages: English, Korean, Chinese, Spanish, German, French, Italian, and Portuguese. These languages collectively account for nearly four billion speakers worldwide, as stated in the study by Joshi et al. [3]. This broad language support underscores Samsung's commitment to making Bixby accessible to a global audience.

To give further context for this dissertation, I will introduce a typical architecture of an intelligent virtual assistant (IVA) system. IVA comprises four components in a pipeline

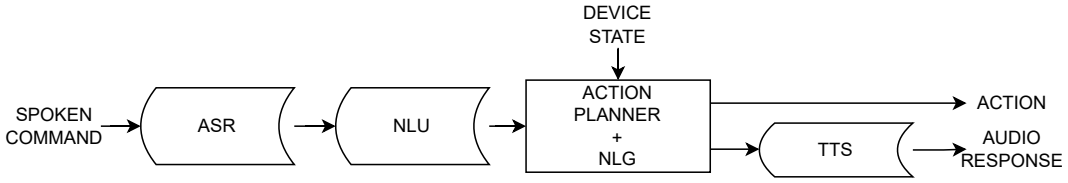

Figure 2. Component diagram of a typical intelligent virtual assistant system.

architecture, as presented in Figure 2. The first step is transcribing the user-spoken command (utterance) to text by the automatic speech recognition system (ASR). ASR output is then an input to the natural language understanding (NLU) model. The NLU component, using machine learning algorithms, interprets the user's command, extracting the underlying intent and relevant information and passing them to the next component. At this processing stage, IVA can comprehend the user's requests and the action planner executes the action. Subsequently, the natural language generation (NLG) module transforms the action planner's output into a coherent, contextually appropriate text response, ensuring that the system's responses are accurate and engaging. The text-to-speech (TTS) module then converts this generated text into spoken language, providing a clear and audibly pleasing voice output. This completes the interaction cycle, describing the performed action or prompting the next turn in the conversation.

The development of multilingual NLU is presently a primary focus in the field of natural language processing (NLP). These models facilitate the creation of IVAs capable of operating in multiple languages concurrently. However, the lack of multilingual learning data poses a significant challenge in developing such models, resulting in specific languages being underrepresented. Considering this, machine translation (MT) systems present a compelling solution for obtaining lots of multilingual training and evaluation data much more easily. Consequently, constructing multilingual NLU models by translating each training sentence into various languages using MT models appears feasible and promising.

To illustrate how MT can be used to translate NLU resources, I will present how NLU models are developed. Figure 3 shows the typical development process of an NLU system that is divided into four phases:

1. In the first phase (P1), the system requirements are outlined in the Voice User Interface (VUX). This sets the groundwork for the development process.
2. The second phase (P2) involves creating the NLU corpus for the initial language, which is usually English. This corpus forms the basis for the NLU model's understanding of language.
3. In the third phase (P3), the NLU model is trained and evaluated. Depending on the evaluation results, improvements can be made to the VUX, or a decision can be made to train a more effective NLU model.
4. The fourth phase (P4) presents a scenario where the source (SRC) corpus is translated

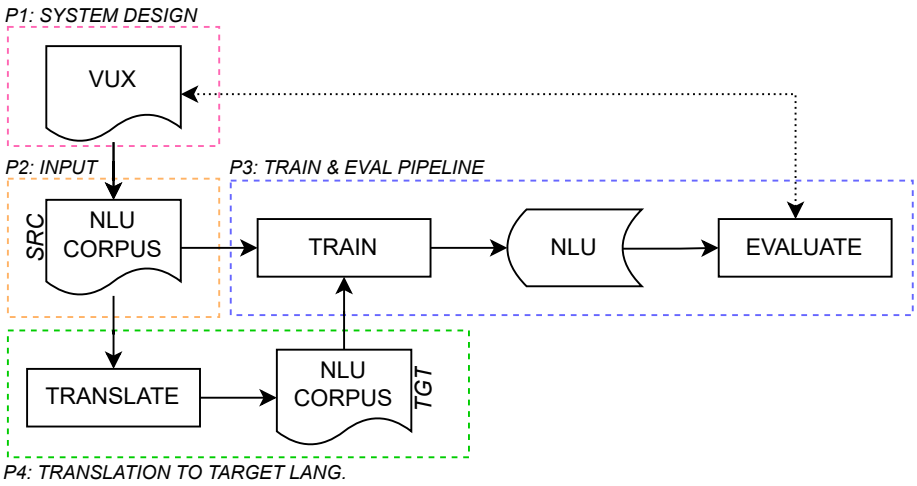

Figure 3. Typical process of NLU development.

into the target language (TGT) using MT. Once the translated NLU corpus is created, the same training pipeline used for the source data can be applied. This ensures consistency in the development process across different languages.

After a brief introduction to NLU and IVA, which are the two most important concepts needed to be able to understand my work, in the following chapters, I will present how MT can create NLU resources for new languages. I will describe how to create an MT model adapted to the domain of IVA and capable of transferring NLU annotations. This capability allows NLU model training without further processing. This work not only addresses the existing gap in multilingual training data but also paves the way for more linguistically versatile and globally accessible IVAs.

1.1. Background

This section introduces four concepts that are crucial to understanding this dissertation: NLU, MT, NLU localization techniques, and Large Language Models. A basic overview of each concept is, at the same time, an introduction to the following chapters.

1.1.1. Natural Language Understanding

NLU is a sub-field of Artificial Intelligence (AI) that focuses on interactions between computers and human languages. It involves computational techniques to interpret, recognize, and comprehend human language in a valuable and meaningful way. NLU enables machines to understand context, sentiment, and intent in human language. This understanding allows for more sophisticated and natural interactions between machines and humans, facilitating tasks such as language translation, sentiment analysis, and voice-activated command execution.

In this dissertation, I will refer to NLU that is used in IVA as a composition of the intent classification (IC) and slot filling (SF) tasks [4]. An example of this is presented in Figure 4.

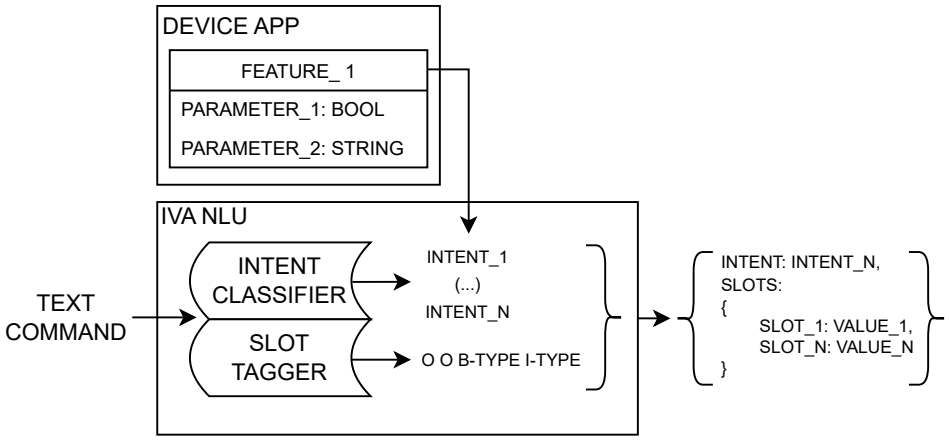

Figure 4. NLU model takes text on input and returns intent and slot annotations.

An intent represents the user's goal or intention when uttering a command to a dialogue system, and slots are the command's parameters. For example, in the utterance "play radiohead on spotify", "radiohead" and "spotify" are slots, and "play_music" is the intent. In NLU, it is typical to define another layer above intent and slots. Intents representing different features of the same application are typically grouped into domains. In our example, "play_music" intent would belong to the *Music* domain together with other similar features that allow the user to control the music player application. In this context, it is important to note that each NLU has a different set of intents and slot types because NLU developers define system features based on project requirements. Although some domains are common for multiple NLUs, they are usually named differently. The lack of a standard for naming is problematic for systems such as MT described in this dissertation. The following chapters will cover possible solutions to this issue.

TXT	play	**radiohead**	on	**spotify**
IOB	O	B-artist	O	B-app

Figure 5. Example of IOB annotation schema where each word in the sentence is annotated with an empty tag (O) or system pre-defined tag.

In this work, I will use the IOB (Inside, Outside, Beginning) tagging format to annotate slots in SF task. IOB is a simple format and a common annotation schema in various NLP tasks, for example, in a Named Entity Recognition (NER). Example from Figure 5 shows the IOB tagging scheme, where each token is a composition of prefix and slot type. Prefixes from which the IOB name comes are:

- I- (Inside): This prefix indicates that the token is inside an entity,
- B- (Beginning): This prefix indicates that the token is at the beginning of an entity,
- O (Outside): This prefix, after which there is no slot type, indicates that a token is outside an entity.

If two entities of the same type immediately follow each other, the first token of the second entity will be tagged with B, not I. This helps to differentiate between the two.

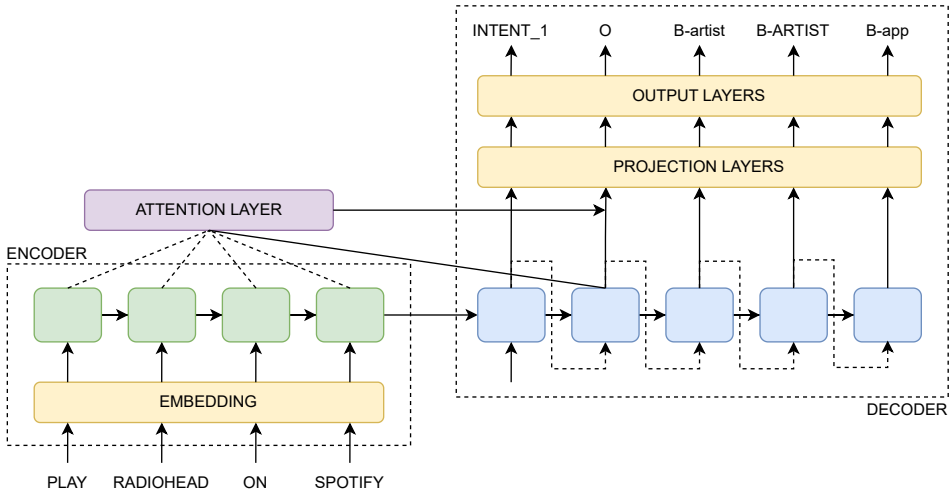

Figure 6. Architecture of sequence to sequence model with attention layer. Adapted from [5].

In IVAs, NLU is either a composition of two artificial neural network models for IC and SF, or it can be a single model that is capable of joint IC and SF. Input to IC and SF are sequences of tokens extracted from sentence, and output is also a sequence of tokens. For that reason, sequence-to-sequence (Seq2Seq) is the most popular architecture in NLU. Seq2Seq models have become a powerful tool in NLU and other sequence prediction tasks such as MT. The concept of Seq2Seq models was introduced by Sutskever et al. [6], where they proposed an end-to-end approach for training recurrent neural networks to map input sequences to output sequences. Seq2Seq models have two main components: an encoder and a decoder, and both are typically implemented as recurrent neural networks. The encoder processes the input sequence and compresses it into a fixed-length context vector, which the decoder then uses to generate the output sequence.

While several metrics can be used to evaluate NLU, I will use the two most common metrics: accuracy and F-score. Accuracy is defined as the ratio of correctly predicted instances to the total number of instances in the dataset. Mathematically, accuracy is calculated as the sum of true positive and true negative predictions divided by the total number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). In the context of intent classification, accuracy provides a straightforward measure of how well a model can correctly identify the intended action or purpose behind a given input, such as a user's query or command.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

The F-score is a harmonic mean of precision and recall. It ranges from 0 to 1, where 1

indicates perfect precision and recall, and 0 means neither precision nor recall. F-score is particularly useful in situations with imbalanced datasets, providing a balance between false positives (precision) and false negatives (recall). The F-score in its most common case, known as F_1-score, is calculated as twice the product of precision and recall divided by the sum of precision and recall, which can be written as:

$$F_1 = 2 * \frac{\text{Precission} * \text{Recall}}{\text{Precission} + \text{Recall}} \tag{2}$$

1.1.2. Neural Machine Translation

MT is the task of automatically translating text from one language to another. The evolution of MT technologies can be traced back to the development of the first dictionary-based systems in the 1950s and 1960s. Although the quality of early MT systems was limited, they attracted the public's and scientists' attention, helping to establish MT as one of the most prominent sub-fields in NLP. Over the years, the growth of MT has been driven by several factors: the rising demand for multilingual communication, the increase in digital content, and the expansion of the global tourism industry. Additionally, its adoption has increased across diverse industries, including automotive, healthcare, and military. As of 2021, the MT market size was estimated at 812.6 million USD, and it is projected to reach 4,069.5 million USD by 2030[1].

Following Chan [7], the history or MT development can be divided into four periods: germination (1967-1983), steady growth (1984-1993), rapid growth (1993-2003), and global development (2004-now).

The "germination period" began with the ALPAC report in 1966, which recommended reducing funding for MT-related research due to insufficient quality and lack of basic research. As a result, commercial MT systems did not emerge until later, and translation technology had a limited impact on translation practice and the industry. During this period, MT systems were primarily rule-based and dictionary-based, relying on linguistic rules and bilingual dictionaries for translation. One notable system was the Automated Language Processing Systems (ALPS), which introduced the concept of translation memory to reuse existing translations. However, these early systems had limitations, such as low reusability of translation memory. Factors like limited computer hardware and immature algorithms for bilingual data alignment constrained the development of MT technology.

From 1984 to 1992, the computer-aided translation (CAT) industry saw significant growth with the founding of companies like Trados in Germany and Star Group in Switzerland. Trados developed the TED plug-in and later the first commercial CAT system, while Star Group introduced the Transit 1.0 system with features like a translation editor and translation memory engine. During this period, CAT systems were primarily rule-based

[1] Analysis from Acumen Research and Consulting: `https://www.acumenresearchandconsulting.com/machine-translation-market`

and dictionary-based, utilizing linguistic rules and bilingual dictionaries. They incorporated translation memory for reusing existing translations. Still, they faced limitations such as low reusability and programming difficulties, partly due to constraints like limited computer hardware and immature bilingual data alignment algorithms.

Between 1993 and 2003, the field of MT experienced a shift towards statistical machine translation (SMT) systems [8]. These systems departed from the rule-based and dictionary-based approaches that had previously dominated the field. SMT systems utilized statistical methods to learn translation models from bilingual text corpora, relying on the frequency of words and phrases in these corpora to predict translations. This approach allowed SMT systems to generate more accurate translations for languages with large amounts of available bilingual data. In this period, phrase-based systems emerged as an evolution of SMT systems. These systems translated text in chunks or phrases rather than word-by-word, offering more contextually accurate translations. Phrase-based systems were particularly effective at capturing idiomatic expressions and collocations, which made them a valuable addition to the MT landscape. The growth of SMT and phrase-based systems was driven by factors such as the emergence of more commercial systems and the development of more built-in functions. The support of more document formats and languages for translation further facilitated the adoption of these systems.

The advent of neural networks led to the development of neural machine translation (NMT), a new approach that leverages deep learning techniques to improve the quality of MT. NMT systems model the entire translation process as a single neural network, eliminating the need for pre-defined linguistic rules and heuristics common in SMT systems.

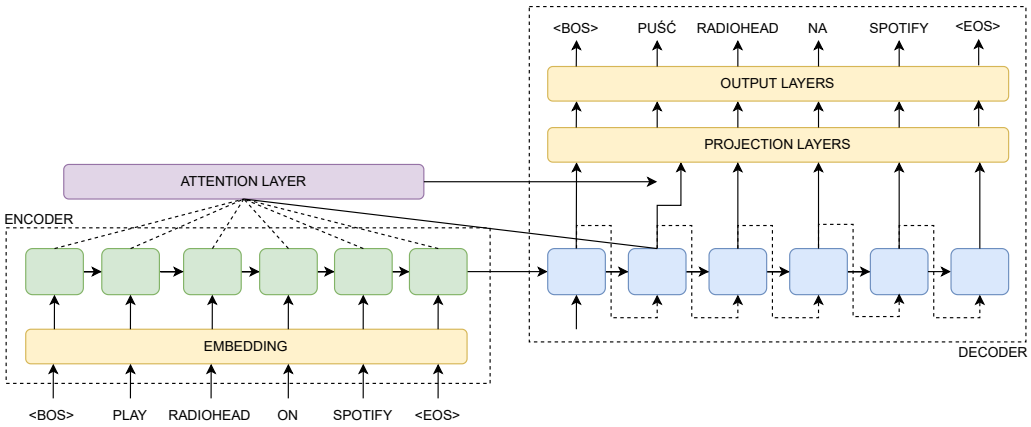

Figure 7. Encoder-decoder architecture of MT. Adapted from [5].

A significant milestone in the evolution of NMT was the introduction of the encoder-decoder architecture by Bahdanau et al. [9], [10]. As shown in Figure 7, this architecture consists of two main components: an encoder and a decoder. The encoder processes the input sentence, tokenizes it into words or subwords, then converts these tokens into

numerical representations (e.g., word embeddings) and compresses the information into a fixed-size context vector, often referred to as the "hidden state". This context vector is then passed to the decoder, which generates the output sentence in the target language. The encoder and decoder are typically implemented using recurrent neural networks (RNNs) or other Seq2Seq models. The encoder-decoder architecture enabled NMT systems to handle long sentences more effectively by capturing the semantic relationships between words and their context in the sentence. The encoder-decoder architecture was further enhanced with the introduction of attention-based methods. These methods allow the model to focus on different parts of the input sentence at each step of the output generation, thereby improving the quality of the translation, especially for long sentences with complex structures.

MT evaluation is crucial for assessing the quality of translated content. In this dissertation, I will use two evaluation metrics that are widely used and recognized. Bilingual evaluation understudy (BLEU) [11] is a widely-used metric that compares machine-generated translations to a set of reference translations, focusing on precision. BLEU was one of the first metrics to claim a high correlation with human quality judgments and remains one of the most popular. The metric computes the geometric mean of modified n-gram precisions, further adjusted by a brevity penalty to account for sentence length. More recently, bilingual evaluation understudy with representations from transformers (BLEURT) [12] has been introduced, which learns robust metrics for text generation by leveraging pre-trained models and fine-tuning on human judgments, aiming to capture more nuanced translation quality aspects.

1.1.3. NLU Localization Techniques

Localization is a process of adapting the NLU system to a new language. It involves translating language resources using MT to fulfill locale-specific requirements. In the IVA context, NLU localization demands careful handling of named entities such as contact names, locations, queries, event names, message content, subjects, and more, replacing them with locale-specific equivalents. For example, "navigate me to London" should be localized to refer to cities within the user's country, as destinations there are more commonly set. Some entities, particularly in the music domain, pose more complex localization challenges, requiring the management of both English and locale-specific names, such as artists, songs, and albums.

NLU localization strategies enable IVAs to understand and respond to user inputs across different languages and cultures. As the need for multilingual IVAs increases, the importance of these localization strategies will become even more evident. The two most common strategies for NLU localization that involve MT are to either translate the NLU training corpus (train-on-target) or to translate utterances from the target language to English and then pass them to an English NLU model (test-on-source) [13].

MT has been a key tool for automatic localization since the development of the first lin-

guistic corpora for NLU [14]–[16]. Early efforts primarily used SMT and reported promising results. For example, Jabain et al.[17] explored various localization strategies and employed an SMT model capable of transferring XML tags between languages. Their study showed that the performance of Italian NLU trained from French to Italian translation was only slightly lower than that of French NLU. This finding was later supported by Servan et al.[18]. Additionally, Stepanov et al. [19] discussed how to adapt SMT for NLU tasks and reported significant improvements in both translation quality and NLU performance for both close and distant language pairs, such as Spanish-Italian and Turkish-Italian.

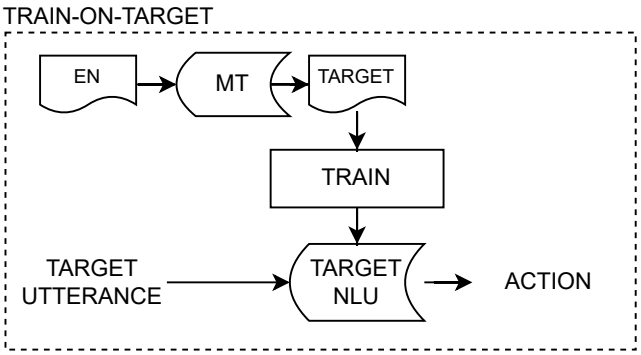

Figure 8. The train-on-target NLU localization technique utilizes a MT model to translate NLU training resources, subsequently replacing the English NLU model with the target NLU model.

The train-on-target strategy, presented in Figure 8, involves translating the source corpora into the target language to construct an NLU model tailored for that specific language. A typical system implementing this strategy consists of a multi-lingual NLU module. This module can either be a singular multi-lingual model or a composition of multiple single-language models. Additionally, an offline MT system is dedicated to translating the model training corpora.

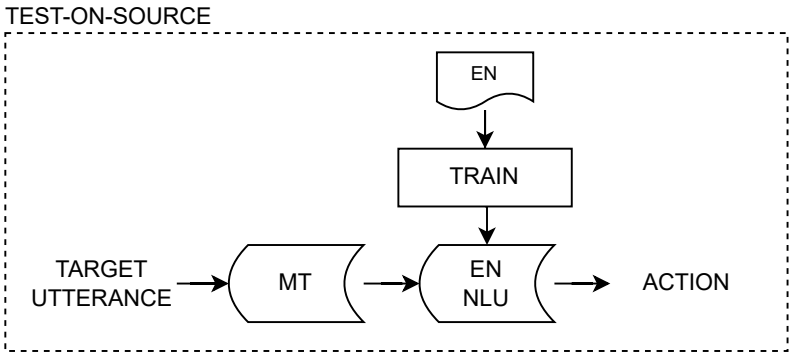

Figure 9. The test-on-target NLU localization technique employs a MT model to translate user utterances to English, which are then sent to the English NLU.

The test-on-source strategy, presented in Figure 9, involves on-the-fly translation of input (test) sentences to the source language, ensuring that the NLU model, trained in the source language, can understand and process the input. Systems adopting this strategy typically consist of a single-language NLU module and an MT system. The MT system can translate from multiple target languages to one source language, the same as the NLU module.

Recently, Abujabal et al. explored the localization of a German IVA system, which includes five domains and over 200 intents [20]. They employed a test-on-source strategy and found that 56% of the automatically translated and labeled utterances perfectly matched the ground-truth labels. Moreover, they reported that their MT-based approach resulted in a 90% reduction in the need for manually labeled data, while still achieving improved performance.

Hench et al. organized a workshop to accelerate the progress of multilingual NLU in IVA [21]. The workshop featured a zero-shot task, where a model trained only on the English portion of a dataset was tested on all other languages. Participants employed various strategies to enhance multilingual recognition. These strategies included using MT for data augmentation [22], train-on-target [23], and utilizing MT as a baseline for IC and SF fine-tuning [24].

While this work primarily discusses MT as a central component of localization, it is worth noting that MT often serves as a supporting model in data augmentation. For example, Rentschler et al. [25] used MT to augment IC in a conversational agent focused on the finance domain in German. They employed Google's MT API in a back-translation scheme and reported a significant improvement over the baseline system. Although this does not directly relate to localization strategies, it demonstrates MT's potential to enhance IC. Similarly, Quan et al. used MT for data augmentation on the CamRest676 and KVRET datasets [26]. They found that incorporating MT in data augmentation helps prevent the dialogue system from omitting key information in user utterances, leading to a significant improvement in the F_1-score.

1.1.4. Large Language Models

As mentioned in the previous section, MT has been a central area of research in the NLP community for several decades. Traditional approaches, such as rule-based and SMT, have paved the way for the current state-of-the-art NMT models. In this section, I will provide an overview of the evolution of MT and delve into the recent advancements brought about by Large Language Models (LLMs) in this domain.

Language modeling is the task of estimating the probability of a sequence of tokens in a text, represented as:

$$p(x) = p(x_1, \ldots, x_T) = \prod_{t=1}^{T} p(x_t | x_{<t}) \tag{3}$$

Here, x denotes a sequence of tokens, x_t is the token at position t, and $x_{<t}$ is the sequence

of tokens that come before x_t. This process is often called autoregressive sequence modeling, which involves predicting the next token based on the previous context at each time step.

LLMs have become increasingly proficient in various tasks, primarily due to incorporating two key techniques: transfer learning and instruction learning. Transfer learning involves training models on extensive unlabeled data and applying their acquired knowledge to various downstream tasks through fine-tuning. Instruction learning frames various NLP tasks as question-answering exercises over a given context, leveraging the existing knowledge of LLMs.

Although LLMs are usually not trained on the MT task, they generate translations of good quality. However, without including MT as one of the training tasks, they still perform worse than specialized MT systems. In light of this research, it becomes evident that continued investment in dataset creation is essential, both for accurately defining the scope of IVA challenges and for enhancing the testing methodologies for IVAs. Zhu et al. [27] showed that best-performing LLMs are still behind the supervised baseline MT in 83.33% of 102 languages and 202 English-centric translation directions. Wei et al. [28] showed that LLMs trained on the MT task can generally perform better than LLMs not trained on this task. Rosenbaum et al. [29] have used fine-tuned LLM [30] to synthesize IC and SL training data. Their results indicate that LLM used as MT can outperform MT baseline with Slot Alignment by +4.14 points absolute on ST F_1-score across six languages while matching performance on IC.

While LLMs exhibit promising results in MT, they are not without challenges. The computational cost of training and deploying such models remains a concern. Recent research indicates that state-of-the-art LLMs underperform in languages other than English [31], [32]. While multilingualism is a significant focus in LLM research, with some successful efforts in training multilingual models [33], most existing LLMs are predominantly trained on English data, with minimal inclusion of non-English languages. Furthermore, the black-box nature of these models makes it difficult to interpret their decisions, leading to potential issues in trustworthiness and reliability [34].

1.2. Problem Definition

IVAs have been available since the 1960s, but the release of their recent generation on smartphones and embedded devices has opened them to a broader audience. The most popular development approach for such systems is to release an initial set of languages, usually English as the first, and then the additional languages, usually starting from the biggest markets (Chinese, Spanish, German, etc.). Although there might be various reasons for choosing such an approach, it is clear that adding support for new languages is a time- and cost-consuming process.

Multilingual models of NLU are currently one of the main targets in the field of NLP, as

they allow the construction of IVA able to handle multiple languages simultaneously. One of the challenges in creating such multilingual models is the lack of multilingual learning data, leading to the under-representation of some languages. The most straightforward way to solve this problem is to translate training sentences manually. However, this method has several disadvantages. First, it is time-consuming and expensive because it requires language experts to translate the sentences. Second, manual translation of training sentences can lead to translation errors and ambiguities that can negatively affect the quality of NLU models. Third, manual translation can be difficult to maintain when languages change or new languages are added to the IVA.

In this context, using MT systems as a source of translations seems to be an attractive alternative for acquiring multilingual training and evaluation data. Therefore, creating multilingual NLU models by translating each training sentence into multiple languages using MT models seems possible and promising.

One of the key challenges in the translation of training sentences for NLU is that they consist of slots and annotations on the level of words that carry information for the NLU system. For example, in the sentence "play radiohead" typically, *radiohead* will be annotated as slot *music_artist*. MT systems employed for training NLU models should accurately preserve and localize specific slots within translated sentences.

MT systems, used to generate sentences for training NLU models, should also produce multiple correct translation variants. This is crucial as languages often have numerous grammatical forms and ways of conveying information. For instance, English has various verb forms, such as regular, irregular, and modal verbs, with potentially different translations in other languages. If an MT system generates only one translation variant, the NLU model might not learn to recognize others, compromising the model's quality. Hence, MT systems should create multiple accurate translation variants to cover all possible patterns, enhancing the performance of NLU models.

1.3. Aim and Motivation

There are over 6900 living languages worldwide, from which more than 91 have over 10 million users [3]. Despite this linguistic diversity, AI voice-based products are predominantly available in English. While the importance of multilingualism is well-recognized in NLP research, a significant gap remains between the top five languages and all others. If we want to build an unfragmented e-society, we must develop methods that will allow us to create multilingual NLU. MT plays a pivotal role in achieving this goal.

While the Conference On Machine Translation (WMT), a leading forum in the MT field, has shifted its focus from sentence-level to paragraph and even document-level translation, there remains a critical need for nuanced sentence-level translation, especially in the context of voice-activated virtual assistants. Contrary to the prevailing opinion that sentence-based translation is a 'solved problem,' the sentences used in commands

for virtual assistants present unique challenges. These include the need for semantic annotations that add contextual meaning to sentences, as well as the requirement for multivariant translations that offer more than a single 'best' translation. As mainstream MT research moves toward document coherence, my work aims to address these overlooked but essential aspects of sentence-level translation.

This dissertation aims to develop an MT system capable of reliably and predictably translating training sentences for dialogue agents. The objective is to match, if not surpass, the quality of expert translations.

2. Theses of this work

The general objective of this work is to show that specialized machine translation models can significantly automate the process of developing dialogue assistants for new languages. This objective was further specified using the below three theses:

1. **[T1] Machine translation, when adapted to the language of Intelligent Virtual Assistants, serves as an efficient tool for localizing natural language understanding models,**
2. **[T2] To translate natural language understanding training resources, which comprise semantic annotations, machine translation must preserve and appropriately translate named entity locations,**
3. **[T3] Generating multiple variants when translating training data for Intelligent Virtual Assistants improves the natural language understanding accuracy.**

T1 is elaborated in Chapter 3. It underscores the necessity of multilingual datasets for MT models and introduces a dataset and its creation method that offers a more accurate evaluation of dialogue agents [35]. Next, a domain-adapted MT model for IVA is introduced and an analysis of the impact of adaptation on model quality is presented.

T2, detailed in Chapter 4 and based on [36], comprises a description of an MT model designed to transfer NLU slots between input and target languages through a flexible XML-like annotation format. Experiments are conducted to evaluate an NLU model trained on translated data, and the impact of translation on NLU quality is discussed.

T3, explored in Chapter 5 and rooted in [37], introduces MT models tailored for IVA applications, enabling multiple valid translations. The translation process, guided by a derived verb-frame ontology, showcases the superiority of multi-verb translation over traditional methods. I perform experiments to check if multi-verb translation improves intent classification accuracy compared to single-best translation. Several MT models and an IVA verb ontology are also presented.

This dissertation is divided into two main parts. The first part, covering Chapters 1 to 5, provides a theoretical background on the effectiveness of MT in translating NLU resources, along with the most up-to-date solutions in this area. Conversely, Chapter 6 showcases a practical implementation of this concept with slightly different MT model architectures, influenced by the project schedule. This system, commercialized at Samsung R&D Institute Poland at the end of 2019, was operational until 2022. The earlier chapters also suggest how the next generation of this system might be structured. The theses have been implemented within systems at Samsung R&D, demonstrating their effectiveness in commercial environments. Chapter 6 offers an overview of the system architecture and presents the experimental results, affirming the practical and commercial viability of the discussed concepts.

3. Machine Translation Adapted to Intelligent Virtual Assistants

3.1. Language Resources for Virtual Assistants and Machine Translation

The development of multilingual IVAs relies heavily on the use of open-source corpora that cover a wide range of languages and domains. Ideally, these corpora should include a variety of intents and slots to match the diverse queries that users might pose. A key feature of a high-quality corpus is its diversity in sentence structures, which is crucial for enhancing an IVAs ability to understand and respond to different conversational patterns. This diversity in sentence formulation directly impacts the IVAs effectiveness in real-world interactions across multiple languages and domains. Therefore, enriching existing corpora becomes essential for the development and evaluation of IVAs that are both multilingual and domain-specific.

In Table 1, the summarization of the existing corpora is presented. All listed corpora are used to test IVAs. Existing corpora are divided into two groups and later compared to the corpora described in this work.

Table 1. Statistics of existing corpora compared to Leyzer, proposed in this work. The first group consists of resources designed to train and test IVAs without focusing on multilingual setups. The second group concerns multilingual IVAs. Language abbreviations according to ISO 639-1:2002.

Dataset	Languages	# Utterances	# Domains	# Intents	# Slots
ATIS [38]	en	5,871	1	26	83
NLU++ [39]	en	3,080	2	62	17
SLURP [40]	en	16,496	18	46	55
CLINC150 [41]	en	23,700	10	150	0
Liu et al. [42]	en	25,716	19	64	54
TOP [43]	en	44,279	-	25	36
SNIPS [44]	en, fr	2,943/1,136	-	7	72
MTOD [45]	en, es, th	5,083-43,323	3	12	11
MTOP [46]	6 langs.	15,193-22,286	11	117	78
PRESTO [47]	6 langs.	72,107-109,528	8	34	285-303
MultiATIS++ [48]	9 langs.	1,353-5,871	1	17-18	71-84
MASSIVE [49]	52 langs.	16,434	18	60	55
Leyzer (this work)	en, es, pl	22,325-27,119	20	181-193	91-97

The first group contains corpora with only one language created to evaluate NLU models. The most popular corpus among them is The Air Travel Information System (ATIS) [38], which consists of spoken queries from the flight domain in the English language. ATIS has a small number of intents and is heavily unbalanced, with most utterances belonging to three intents. Still, it owes its popularity to the fact that it was the first corpus of its

kind widely available to the research community. Larson et al. [41] created a dataset to study out-of-scope queries that do not fall into any of the system's supported intents. The presented corpus consists of 23,700 queries equally distributed among 150 intents, which can be grouped into ten general domains. Liu et al. [42] created a dataset that is a use case of a home robot that can be used to train and compare multiple NLU platforms (Rasa, Dialogflow, LUIS, and Watson). The dataset consists of 25716 English sentences from 21 domains that can be divided into 64 intents and 54 slot types.

The corpora that belong to the second category of IVAs datasets were designed to train and evaluate multilingual IVAs. The SNIPS [44] dataset has a small number of intents; each intent, however, has a large number of sentences. MTOD is a multilingual dataset for English, Spanish, and Thai to study various cross-lingual transfer scenarios. The dataset consists of 3 domains: Alarm, Reminder, and Weather, with a small number of intents and slots (11 intents and 12 slots total). Different languages have different numbers of sentences, with English having 43,323, Spanish having 8,643, and Thai having 5,083. It follows that there is a large number of sentences per intent and slot type.

The MASSIVE, MultiATIS++, and PRESTO datasets represent recent advancements in multilingual resources for IVA. MASSIVE and MultiATIS++ build upon the foundations of their predecessors, SLURP and ATIS, respectively. Both these corpora were initially crafted using MT systems and subsequently refined by linguistic experts, showcasing a blend of automated and human expertise. On the other hand, PRESTO stands out with its extensive collection of 552,924 sentences, offering around 2,000 test cases for each intent and approximately 95,000 utterances per language. Beyond its sheer volume, PRESTO introduces a unique dimension to the latest NLU corpora by identifying a range of linguistic phenomena. This includes 21% revisions (e.g., "send this screenshot to Mike and cc John, I mean Josh"), 20% disfluency cases, 14% code mixing instances, and various other linguistic nuances. Compared with Leyzer, PRESTO shares domains like fitness, food, social, and communications, underscoring the overlapping interests in the field. However, PRESTO further broadens the horizon by exclusively delving into finance, health, transportation, and shopping domains. Collectively, these datasets - MASSIVE, MultiATIS++, and PRESTO - provide a broad overview of current trends and focus areas in multilingual NLU research. Each dataset brings its own unique contributions and insights, thereby enriching the field of IVA research.

3.2. Leyzer: A Dataset for Multilingual Virtual Assistants

In this section [2], I will present Leyzer[3], a dataset containing a large number of utterances created for the purpose of investigation of cross-lingual transfer learning in NLU systems. While creating my dataset, I focused on testing localization strategies that use MT and multilingual word embeddings.

When I began my work on Leyzer in 2020, there were only a few NLU corpora that could be used to test and develop NLU models for IVA. NLU corpora for IVA have evolved significantly since 2019. Initially, the available corpora such as ATIS, TOP, MTOD, and SNIPS were relatively limited in their scope, primarily focusing on self-management tasks. They did not cover domains typically associated with commercial assistants like Siri or Bixby. Although many new resources have been developed since its creation, Leyzer still is among the biggest in terms of the number of domains, intents (where *intent* is understood as an utterance-level concept representing system functionality available for the user) and slots (where *slot* is defined as a word-level concept representing the parameters of a given intent) in the area of multilingual datasets focused on problems of the localization of IVA datasets. Leyzer has been publicly released, with the code to allow reproduction of the experiments, and is available at `https://github.com/cartesinus/leyzer`.

When designing Leyzer, I focused on problems that commercial IVA systems often face:

1. Number of languages and their linguistic phenomena, which represents a challenge of building a multilingual system and handling phenomena such as flexion, which has an impact on slot recognition,
2. Number of domains and their distribution, that introduce two major challenges:
   a) how to train a model to equally represent each domain, even if the trainset is not balanced in terms of the number of sentences per domain,
   b) how to treat sentences that are similar or identical in more than one domain,
3. Number of intents and how they differ. This introduces the problem of having multiple intents that differ only by one parameter or word,
4. Number of slots and their values, that introduces a challenge of how to train a model that will recognize slots not by their values but rather by their syntactic function in the sentence.

Leyzer is task-oriented NLU corpus, and together with other corpora of this type, such as SNIPS, PRESTO, and MASSIVE (all listed in Table 1), it aims at modeling voice-controlled interaction between user and device.

[2] This section is partially based on my article [35] that was presented at the International Conference on Text, Speech and Dialogue (TSD) conference in 2020.

[3] Named after Ludwik Lejzer Zamenhof, a Polish-Jewish linguist and the inventor of the international language Esperanto, the most widely used constructed international auxiliary language worldwide. `https://en.wikipedia.org/wiki/L._L._Zamenhof`

Leyzer consists of single-turn utterances in contrast to multi-turn corpora such as MultiWoz [50]. Multi-turn corpora feature multiple rounds of questions and answers (a dialogue) between the user and the system. These corpora typically cover domains that necessitate more complicated interactions or would be too complex to express in a single sentence. For instance, making a restaurant reservation usually involves more than just saying "make a reservation at a ramen shop for two people at 7 pm". It is not only difficult to say this without pausing, but the restaurant might only have availability at 7:30, which could still work for the user. In contrast, single-turn utterances often function as voice interface "buttons". Users employ them to control devices for which they have a mental model. For example, setting an alarm with the command "set the alarm at 8 am" is a straightforward interaction that doesn't require additional steps. Voice interactions can be concise or require several turns to complete an action. Therefore, single-turn and multi-turn NLU corpora are complementary and should ideally be integrated into a single corpus that includes both types of interactions.

3.2.1. Domain Selection in Leyzer dataset

Following [51], I have created 20 domains representing popular applications that can be used on mobile devices, computers, or embedded devices.

The Leyzer corpus includes a diverse set of domains, some unique to Leyzer, while others are shared with other corpora. Common domains found in multiple corpora include Alarm, Weather, Calendar/Reminder, Communication, Transport, Booking, Music, and News. For example, Weather is present in more than half of the corpora, and Calendar/Reminder is found in nearly half of the corpora, as shown in Table 1.

The Leyzer corpus differentiates from other NLU corpora through its unique domains, such as Console, G Drive, Translate, and YouTube, which are not found in other corpora. ATIS is the only corpus that focuses on Flight Booking. NLU++ includes Banking and Hotel domains, which are not present in other corpora. CLINC150 covers a wide range of domains, including Work, Auto and Commute, Travel, Home, and Kitchen, the latter of which can be interpreted as similar to the Cooking domain in the MASSIVE corpus. Liu's corpus is unique in its exclusive focus on the Alarm domain. Snips include Restaurant Booking, Taxi, and Maps, which are not found in other corpora. MTOD and MTOP have exclusive domains like Timer and Reminder. PRESTO stands out with its Finance, Health, Transportation, and Shopping domains. MASSIVE includes IoT, Lists, QA, Takeaway, and Transport, which are unique to this corpus.

Leyzer domains can be categorized into groups with similar functions:

- **Communication** with Email, Facebook, Phone, Slack, and Twitter domains in that group. All these domains contain a kind of command to send a message.
- **Internet** with Web Search and Wikipedia. The aim of these domains is to search for information on the web and, therefore, these domains will have a lot of open-title queries.

- **Media and Entertainment** with Spotify and YouTube domains in that group. The root function of these applications is to find content with name entities connected with artists or titles.
- **Devices** with Air Conditioner and Speaker domains. These domains represent simple physical devices that can be controlled by voice.
- **Self-management** with Calendar and Contacts domains. These domains consist of actions that involve time planning and people.
- **Other** non-categorized domains represent functions and language not common to the other categories. In that sense, the remaining domains can be represented as intentionally not matching other domains.

Table 2. Statistics of sentences, intents, and slots across domains and languages in the Leyzer dataset.

Domain	**# Intents**	**# Slots**	**# English Utt.**	**# Spanish Utt.**	**# Polish Utt.**
Airconditioner	10	4	578	1018	304
Calendar	10	4	1039	749	1106
Console	6	5	1370	-	1030
Contacts	11	5	1180	1769	1530
Email	11	8	1418	7483	1341
Facebook	7	5	696	1307	1193
Fitbit	5	3	227	139	927
Google Drive	13	6	401	376	1098
Instagram	10	7	1042	959	1506
News	4	3	283	1351	961
Phone	6	4	488	386	463
Slack	14	9	487	99	642
Speaker	7	3	368	98	159
Spotify	18	9	1935	3386	1877
Translate	9	7	4575	812	3321
Twitter	6	4	250	218	324
Weather	10	3	490	94	371
Websearch	7	3	3184	6315	3096
Wikipedia	8	2	956	156	352
Yelp	12	6	657	146	994
Youtube	9	4	488	259	3165
Total	187	99	22325	27394	25938

In Table 2, a list of domains, intents, and utterances available in Leyzer for each language is presented. Each domain has a different number of intents and slots, reflecting the complexity and specificity of each domain. For example, the Slack domain has a notably high number of slots (9), indicating that it requires more detailed information to fulfill user requests. As mentioned above, several domains differ in size to better reflect proportions from the real-world problems where some applications will only have a few

possible ways to express commands, while the other ones will have an almost infinite number of valid expressions. The Leyzer dataset covers a wide range of domains, from common ones like Calendar and Weather to more specialized ones like Console and Fitbit. This diversity makes the dataset suitable for training and evaluating NLU models across various applications.

3.2.2. Intent and Slot Selection in Leyzer dataset

There is a close relationship between intents and slots in Leyzer, as the intents represent functions or actions that users want to perform, while the slots are the parameters of these intents. In many cases, intents represent the same action, but they have been distinguished based on the number of parameters. During the creation of intents, my principle was that intents must differ from each other either by the language (different important keywords) or by the number of slots they have. The reason for that is purely pragmatic, as there cannot be two identical sentences with different intents to avoid the system's instability. The model input is a sentence, and its output is the intent, so if, in the training corpus, we had two identical sentences pointing to different intents, then the model would not be able to learn to which intent this sentence should be assigned.

Table 3. Representative patterns from selected domains of the corpus.

Domain	**Intent**	**Example Sentence Pattern**
Calendar	AddEventWithName	add an event called **$EVENT_NAME**
Email	ShowEmailWithLabel	show me my emails with label **$LABEL**
Facebook	ShowAlbumWithName	show photos in my album **$ALBUM**
Slack	SendMessageToChannel	send **$MESSAGE** to **$CHANNEL** on slack
Spotify	PlaySongByArtist	play **$SONG** by **$ARTIST**
Translate	TranslateTextToLanguage	translate **$TEXT** to **$TRG_LANG**
Weather	OpenWeather	what's the weather like
Websearch	SearchTextOnEngine	google **$TXT_QUERY**

Table 3 provides representative patterns from selected domains in the Leyzer corpus, showing the relationship between intents and slots. Each row in the table represents a domain, an associated intent, and an example sentence pattern that illustrates how slots are used within the intent. For example, in the Email domain, the intent *ShowEmailWithLabel* enables users to view emails with a specific label. The slot $LABEL represents the label of the emails. This intent highlights the importance of categorization in email management and shows how users can filter their emails based on labels. Similarly, in the Slack domain, the intent *SendMessageToChannel* enables users to send a message to a specific Slack channel. The slots $MESSAGE and $CHANNEL represent the message's content and the channel's name, respectively. This intent demonstrates the need for targeted communication within team collaboration platforms. These examples illustrate how intents represent actions or functions that users want to perform, while slots serve as

parameters for these intents. The diversity of intents and slots across different domains highlights the versatility of the Leyzer corpus in capturing various user interactions.

The slots in Leyzer can be categorized into two groups:

- **Open-titled** – where the number of slot values is practically infinite and, therefore, cannot be listed. Open-title slots are challenging for NLU systems because they force them to generalize the unseen data.
- **Close-titled** – where the values of the slots can be listed.

3.2.3. Naturalness Level and Verb Patterns

The quality of IC models is tied to how many and how diverse the training examples are. Most NLU resources have a few to several hundred examples per intent. For example, the MASSIVE dataset has, on average, 275 examples per intent, while Leyzer has around 190. The average number of examples per intent varies depending on the domain and how the intents are set up. Some very specific domains might only have as few as five examples per intent, while others could have more than 1,000 examples.

In contrast to Leyzer, other NLU corpora do not differentiate between examples. It is important to note, however, that some utterances are more natural and commonly used by users. In its latest release, two more sub-intent modalities were added to Leyzer: naturalness level and verb patterns. These unique features help researchers gain extra insights into system performance and improve certain areas. It allows for checking whether the intent classification model is working right in its most important cases (*L0* and *L1*), where the accuracy should be nearly 100%. Accuracy for *L2* and *REPHRASE* can be lower because users are less likely to use them.

Commercial NLU systems, which often include hundreds or thousands of intents, also use this rule-of-thumb strategy and focus on the most natural utterances to improve accuracy. This heuristic is important because the current state-of-the-art models still cannot correctly recognize all user commands.

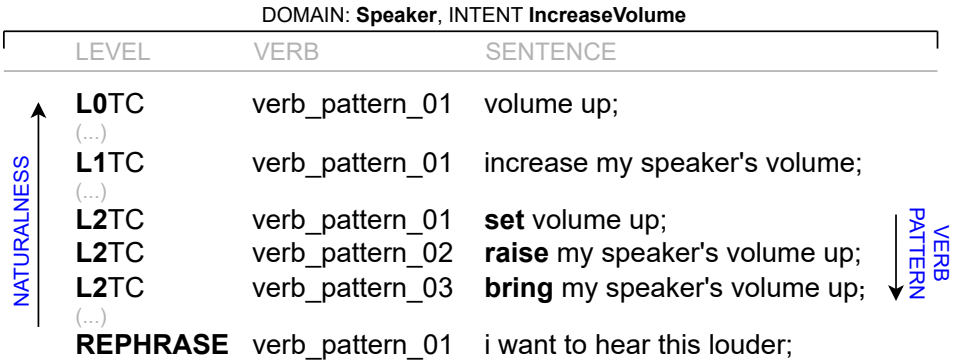

Figure 10. Example of various levels of naturalness and verb patterns in IncreaseVolume intent of Speaker domain.

In Figure 10, the structure of intents available in the Leyzer dataset is presented. The naturalness is divided into four levels:

- **L0**: Typical and most natural way to utter a command to the NLU system,
- **L1**: Straightforward yet not most obvious ways to request for command,
- **L2**: Correct utterances yet somewhat unlikely to be uttered by the user,
- **REPHRASE**: rephrased in the lexical or semantic way (as in Wu et al. [52]).

To illustrate the concept of naturalness level, let's revisit the example from Figure 10. The *IncreaseVolume* intent allows users to raise the volume of their speakers. In this case, users typically use voice commands as if pressing a physical button. The most straightforward way to express this intent vocally would be "increase the volume of my speaker". However, according to Zipf's Law of Least Effort, the shorter "volume up" is more natural because it conveys the same meaning with less effort from the user. *Level 2* includes alternative ways of expressing the same intent, though they may not be as commonly used. The purpose of *level 2* is not to capture popular expressions but to cover a wide range of possible utterances. These utterances may not be frequently used, but they are still plausible. We can use *level 2* utterances to test whether an NLU model can handle less common expressions and whether it was designed to cover a broad range of utterances (high recall). Rephrase is intended as a sanity check for the system to understand utterances that express the same goal but are crafted to challenge the system. In our example, "I want to hear this louder" is an ambiguous way of expressing that the music (or other sounds) should be played at a higher volume. Another *rephrase* example that does not explicitly reference sound could be "increase the decibels, please" when referring to music.

Verb patterns, as the name implies, categorize sentence patterns based on the verb used. All sentences with a specific verb will share the same verb pattern, regardless of the sentence structure. I have focused on verbs because they are crucial in single-turn commands. If we consider voice commands as linguistic equivalents of physical actions, then verbs represent the core components of these actions. Verb patterns enable us to test whether the NLU system can understand various ways of expressing an action, even if they are not the most direct. For example, in Figure 10, the verbs *set*, *raise*, and *bring* were grouped under the same naturalness level. While these verbs represent different actions, they should all be interpreted as increasing the speaker's volume in this specific context.

3.2.4. Corpus Generation

In contrast to approaches used in most NLU corpora presented in Table 1, where utterances are gathered using crowd-sourcing annotators, I decided to use finite-state grammars. I believe that all concerns about grammar-based generated text, namely on their lack of naturalness, can be eliminated if the quality control procedure is implemented. Grammar-based corpora have two noteworthy advantages: they are cheap in

generation and remodeling. They can cover all possible ways to express a given intent, which crowd-sourced approaches can easily miss.

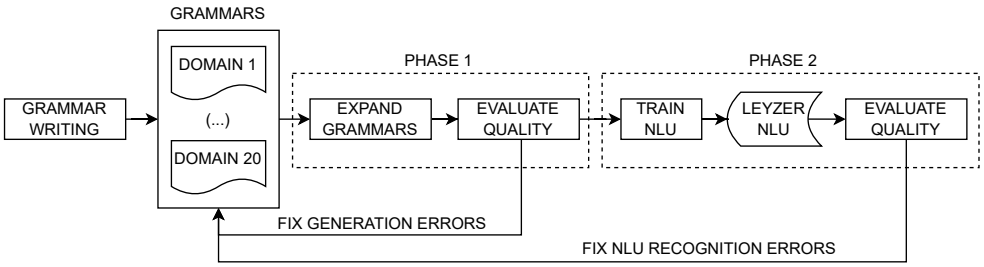

Figure 11. Leyzer grammar creation process. English grammar was created first and used as a seed for Polish and Spanish grammars.

The Leyzer creation process, presented in Figure 11, consisted of four steps: creating English grammar, creating grammars for Spanish and Polish, slot expansion and splitting data into train-, dev- and test sets. Starting with English, I have created 20 grammars with sentence patterns in the JSpeech Grammar Format (JSGF). Each domain's initial set of intents was inspired by example commands available in Almond Virtual Assistant [51]. In the second step, I extended a list of intents for each domain with more challenging features for NLU models. For example, I have added several intents where slot values are from the open list (i.e., the content of the message). NLU model trained to recognize such intents cannot pay attention to words in such slot yet must correctly recognize boundaries of such slot.

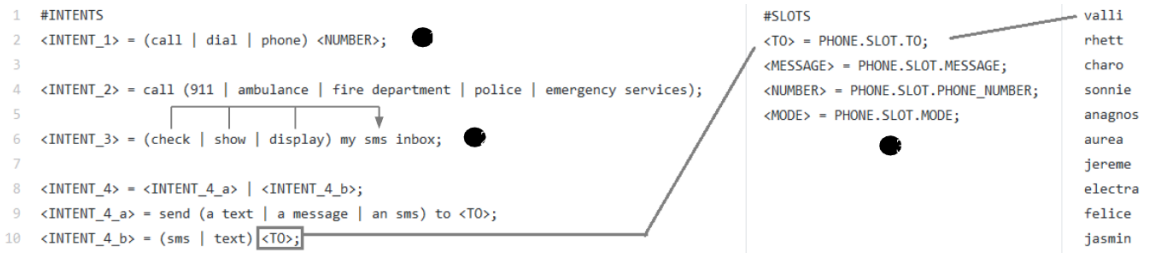

Figure 12. Simplified JSGF grammar for Call domain.

In Figure 12, a fragment of a simplified version of *Call* domain grammar is presented. JSGF is a simple but powerful format for grammar generation. In each rule presented in our example, one of two basic operators can be used. In an alternative that is represented with round brackets (i.e., *(a|b)*), only one element from the list is chosen to be generated. In an optional that is represented with box brackets (i.e., *[a|b]*), either one element or none is selected for generation. Non-terminals defined in angle brackets (i.e., <*name*>) are used to annotate slots in grammars. Each slot definition is linked during the generation with an external file where slot values are stored. Slot values were crawled from the Internet or

created manually. Depending on slot type, from a few to a few hundred values for each slot were gathered.

I have first written grammars for English. English plays a central role in Leyzer. All new intents, domains, and corpora features are first implemented in English and then localized to Spanish and Polish. While creating English grammar, I have first created definitions of all intents in each domain without filling them. Each intent was divided into four levels, representing how natural a given sentence is.

All sentence patterns in the corpus were generated from grammars. Each of such patterns represents a possible way to utter a sentence without explicitly giving the content of the slots. Later on, grammars were filled with the slot values. Since sentences generated in such a fashion might contain some unnatural expressions or grammatical errors, I requested verification by language experts. Wherever it was possible, incorrect sentences were fixed, and sentences were removed if that was impossible.

3.3. Comparative Analysis of Grammar-Based and Crowd-Sourced NLU Corpora

The grammar-based NLU corpora are considered less natural than crowd-sourced or expert-made corpora. This argument, however, may be valid only if there is no verification in the development process. If grammar is inspected during creation, then all problems, such as repetitions and incorrect sentence structure, can be easily eliminated, making grammar-based corpora undistinguishable from crowd-sourced corpora. In this section, I will compare Leyzer's intents with their equivalents in crowd-sourced corpora.

Table 4. Comparison of patterns in MASSIVE and Leyzer AddEvent*

MASSIVE calendar_set (1147 examples)		Leyzer AddEvent* (224 examples)	
%	Pattern	%	Pattern
12.99	set (event | reminder) *event*	27.67	create *event*
10.98	add *event* to calendar	18.75	save (meeting | reminder) *event*
8.63	remind (me) [about] *event*	18.75	schedule *event*
2.79	i need *event*	17.41	add *event*
2.70	schedule *event*	3.57	make *event*
2.00	make *event*	2.68	remember *event*
1.83	create *event*	1.78	remind (me) [about] *event*
1.83	i have *event*	1.78	put *event* on the calendar
8.63	please + *basic pattern*		
3.22	can you + *basic pattern*		
43.4	*other*	7.61	*other*

The prevalent assumption that grammar-based NLU corpora are less natural than crowd-sourced or expert-made corpora may not hold true under scrutiny. As Tables 4, 5, and 6 demonstrate, a careful comparison between the intents in the Leyzer corpora

Table 5. Comparison of patterns in MASSIVE audio_volume_up and Leyzer IncreaseVolume*

MASSIVE audio_volume_up (134 examples)		Leyzer IncreaseVolume* (96 examples)	
%	Pattern	%	Pattern
14.92	turn volume up	20.83	make speaker louder
11.19	increase volume	14.58	bring volume up
6.71	raise volume	14.58	increase volume
3.73	make speaker louder	14.58	raise volume
11.19	please + *basic pattern*	12.50	turn volume up
5.22	can you + *basic pattern*	6.25	set volume
5.22	(i need | i want) + *basic pattern*	6.25	volume up
38.79	*other*	10.43	*other*

Table 6. Comparison of patterns in MTOD checkSunrise and Leyzer Sunrise*

MTOD checkSunrise (102 examples)		Leyzer Sunrise* (146 examples)	
%	Pattern	%	Pattern
25.49	what time *is* sunrise	10.95	get sunrise
21.56	what time *does* sunrise	10.95	show me sunrise
12.74	when *does* sunrise	10.95	when [does] sunrise
11.76	when *is* sunrise	8.21	check sunrise
10.78	what time *will* sunrise	8.21	check *what* time is sunrise
4.9	when *will* sunrise	8.21	check *when* time is sunrise
4.9	*at* what time is sunrise	8.21	find out when sunrise
7.87	*other*	9.68	*other*

and their equivalents in crowd-sourced corpora such as MASSIVE and MTOD reveals noteworthy similarities.

Both types of corpora contain a diverse set of patterns for the same intent. For example, in Table 4, both MASSIVE and Leyzer use multiple phrasings for adding an event—ranging from "set (event | reminder)" in MASSIVE to "create *event*" in Leyzer. Similar observations can be made for audio volume adjustment (Table 5) and checking sunrise times (Table 6).

Both the grammar-based and crowd-sourced corpora contain basic patterns, often preceded by polite phrases like "please" or query initiators like "can you" (see Tables 4 and 5). This suggests that grammar-based corpora can capture the nuances of natural language to some extent.

Another crucial aspect is the percentage of patterns classified as "other". In Table 4, the percentage is significantly higher for the crowd-sourced MASSIVE corpora (43.4%) compared to Leyzer (7.61%). This could indicate that grammar-based corpora like Leyzer are more focused and possibly less noisy.

Although the percentage distribution of specific patterns varies, this could be at-

tributed to the sample size or other external factors and does not necessarily indicate a qualitative difference between the two types of corpora.

In summary, the grammar-based NLU corpora, when carefully crafted, can exhibit properties very similar to those of crowd-sourced corpora. This observation challenges the prevailing notion that grammar-based corpora are inherently less natural, suggesting instead that they can be a robust alternative for NLU tasks.

3.4. Machine Translation Adapted to IVA

Domain adaptation in NMT involves fine-tuning a pre-trained neural network to specialize in a particular domain. This becomes relevant when there is a noticeable data distribution discrepancy between the model's original training data and the target domain dataset. Typically, NMT models are trained on data sourced from web crawling, which predominantly includes news articles and similar content. This often leads to a mismatch in data distribution that can be mitigated through domain adaptation.

Luong and Manning [53] pioneered the concept of adaptation in NMT. They explored this adaptation through a process of continued training, where an NMT model initially trained using large corpora in one domain could later be used to initialize a new NMT model for another domain. Their findings suggested a significant enhancement in performance through the fine-tuning of the NMT model. Specifically, they showed that training the MT model on out-of-domain data and then fine-tuning it using a small in-domain parallel corpus led to a boost in performance.

Through various methods that impact model parameters, domain adaptation enables the model to specialize in translating texts within a particular domain, aligning its performance and relevance to the targeted context. In this section, different aspects of domain adaptation will be examined, beginning with adjustments at the corpus level and followed by model-level adaptations that both have been used in this dissertation.

Adaptation on the corpus level is a data selection technique. It involves choosing training and evaluation corpora to focus on in-domain distribution while also including some out-of-domain examples for general quality. This method is easy to implement, but it requires knowing the domain data distribution upfront. Deviations from this distribution during inference can lead to quality decreases. In its most simple implementation, domain data are mixed with out-of-domain, and both are fed to the model.

Duh et al. [54] introduced a method that aims to select general-domain sentences that are similar to in-domain text while being dissimilar to the average general-domain text. The approach involves creating Language Models (LMs) for both source and target languages, trained on general and domain-specific data. Their method showed translation improvements ranging from 0.1 to 1.7 BLEU. Wang et al. [55] proposed a method that leverages sentence embeddings to score out-of-domain data. The core idea is to train models using in- and out-of-domain data and then score the out-of-domain data. Training

data is selected from the out-of-domain data based on a cut-off threshold applied to these scores.

In scenarios where the exact test data or domain is unknown during training, online optimization techniques can be applied. Lü et al. [56] proposed an online model optimization method where similarity between the input sentence and predefined models is calculated in real-time to determine the weights of each model.

Another type of domain adaptation technique is applied at the model level. Fine-tuning [57] is a technique that involves training a model on a general domain and then fine-tuning it on a specific domain. This method is commonly used due to its simplicity and effectiveness. While fine-tuning is an effective method to improve in-domain quality it does negatively impact general domain performance, which is known as catastrophic forgetting [58]. Several techniques have been proposed to overcome this problem. Thompson et al. [59] used Elastic Weight Consolidation [60] as a regularization technique during fine-tuning. Bapna and Firat [61] proposed task-specific adapters for each domain. Additionally, knowledge distillation [62] has been proposed as another solution to this challenge [63].

3.4.1. Domain Adaptation with LoRA Adapters

In this study, the domain adaptation technique Low-Rank Adaptation (LoRA) was employed to enhance the performance of the multilingual machine translation model M2M100. LoRA is a method designed to tailor LLMs for specific tasks without adding extensive additional parameters. As described in Hu et al. [64], LoRA operates by introducing low-rank modifications to the weight matrices of the targeted layers or modules within the neural network. Specifically, it decomposes the original weight matrix into a product of two low-rank matrices. This decomposition enables the model to capture task-specific information with fewer parameters, thus facilitating adaptation while maintaining computational efficiency.

In the configuration used for adapting the M2M100 model, LoRA was directed towards the “q_proj” and “k_proj” modules with a specified rank (“r”) of 8, a LoRA alpha value of 32, and a dropout rate of 0.1. The low-rank factorization facilitated by LoRA permits the generation of compact adapter models, roughly 5 MB in size, which are capable of real-time integration with the baseline M2M100 model. This allows the system to dynamically switch adapters to accommodate the input data better, thereby enhancing the translation quality. Through this mechanism, LoRA serves as an effective domain adaptation technique, bridging the gap between the general-purpose multilingual capabilities of M2M100 and the specific needs of the English-to-Spanish and English-to-Polish translation tasks undertaken in this study.

The training logs show a consistent improvement in the BLEU score over epochs, reflecting a positive impact of LoRA on the model’s translation accuracy. Particularly, for English-to-Polish translation, as presented in Table 8, the BLEU score improved from 24.91 to 38.11, and for English-to-Spanish translation presented in Table 7, it ascended from

Table 7. Training and validation metrics across epochs for LoRA-adapted English-to-Spanish translation model

Epoch	Training Loss	Validation Loss	BLEU
1	8.2793	7.8473	26.92
2	7.8807	7.5783	28.54
3	7.7234	7.4743	32.67
4	7.6478	7.4225	35.56
5	7.5943	7.3928	36.95
6	7.5515	7.3752	37.84
7	7.5255	7.3621	38.53
8	7.5248	7.3521	37.84
9	7.5123	7.3387	38.19
10	7.4964	7.3428	38.16
11	7.4982	7.3328	41.03
12	7.4839	7.3287	41.42
13	7.4776	7.3265	41.48
14	7.4830	7.3285	40.22
15	7.4671	7.3264	40.93

Table 8. Training and validation metrics across epochs for LoRA-adapted English-to-Polish translation model

Epoch	Training Loss	Validation Loss	BLEU
1	7.8621	7.6870	24.91
2	7.6340	7.5312	29.79
3	7.5582	7.4595	34.82
4	7.5047	7.4264	36.19
5	7.4888	7.4167	36.23
6	7.4560	7.4013	36.63
7	7.4477	7.3907	37.05
8	7.4422	7.3743	37.75
9	7.4311	7.3748	37.57
10	7.4294	7.3679	37.53
11	7.4114	7.3697	38.19
12	7.4224	7.3620	38.17
13	7.4334	7.3608	38.09
14	7.4133	7.3621	38.24
15	7.4158	7.3599	38.10

26.92 to 40.93 over 15 epochs. However, as will be described in detail in the next section, upon evaluating the model on the test set, it can be seen that the LoRA adaptation did not yield satisfactory results. The average BLEURT and BLEU scores of the LoRA Adaptation are significantly lower compared to the Base model and the Fine-Tuning approach across

both in-domain (IVA_MT) and out-of-domain (WMT) datasets. This discrepancy in performance could be ascribed to potential issues with the implementation or the training regimen of the LoRA adapters. Despite the promising outcomes observed during training, the LoRA adaptation did not prove to be effective in enhancing the translation quality in these test scenarios. The stark contrast between the training and testing performance underscores the necessity for further investigation into the applicability and optimization of LoRA for domain adaptation in multilingual machine translation tasks, particularly within the framework of the M2M100 model.

3.4.2. Domain Adaptation via Fine-Tuning

Fine-tuning is a widely employed adaptation technique in the domain of artificial neural networks. It involves taking a pre-trained model and further training it on a new dataset to tailor its performance to specific tasks or domains. During this process, all the parameters of the original model are typically updated, although the extent of these updates can vary. Depending on the complexity of the model architecture and the new dataset, fine-tuning generally requires fewer epochs compared to training a model from scratch. This makes fine-tuning a time-efficient strategy for model adaptation.

In my experiments, the fine-tuning process was conducted over 10 epochs for English-to-Polish model and 7 epochs for English-to-Spanish model. The number of epochs was chosen after the initial set of experiments to allow the models to adequately adapt to the data without risking overfitting. I have also experimented with extending the number of training epochs but observed no significant improvements. For optimization, the Adam algorithm [65] was used with an initial learning rate set at $2e-5$. The batch size was fixed at 4.

Table 9. Training and validation metrics across epochs for English-to-Spanish adaptation via fine-tuning

Epoch	Training Loss	Validation Loss	BLEU
1	0.0135	0.0122	66.83
2	0.0090	0.0112	68.12
3	0.0067	0.0110	68.26
4	0.0051	0.0110	68.70
5	0.0037	0.0112	68.70
6	0.0027	0.0113	68.99
7	0.0023	0.0115	69.28

Compared to other adaptation techniques like LoRA Adapters, fine-tuning showed superior performance, particularly in BLEU scores. This suggests that fine-tuning is more effective for the specific demands of translating for IVAs. However, one limitation was the computational cost, as fine-tuning required more resources than some other techniques,

Table 10. Training and validation metrics across epochs for English-to-Polish adaptation via fine-tuning

Epoch	Training Loss	Validation Loss	BLEU
1	0.0178	0.0171	57.44
2	0.0130	0.0159	58.89
3	0.0091	0.0157	60.16
4	0.0073	0.0159	60.59
5	0.0054	0.0161	60.65
6	0.0040	0.0166	61.53
7	0.0031	0.0169	61.04
8	0.0024	0.0172	61.94
9	0.0018	0.0175	61.73
10	0.0014	0.0176	61.62

albeit for fewer epochs. In my experiments, the adaptation of the M2M100 model was feasible only when using an A100 GPU card with 40GB of RAM, which presents a significant limitation in terms of computational resources.

For future work, exploring different sets of hyperparameters or employing regularization techniques could potentially yield even better results. This fine-tuning approach could extend to other NLU tasks or even different neural architectures.

3.4.3. Results of Domain Adaptation

As described in more detail in the previous section, domain adaptation is the process of tuning a pre-trained model on a specific domain or dataset to enhance its performance for that particular context. In this work, I compare two domain adaptation techniques that are commonly used. LoRA adaptation is compared with data selection with fine-tuning. The reason to select these particular methods is that they are the simplest. One of the goals of this dissertation is to provide engineers and researchers working on multilingual NLU with tools that they can use. Selected methods are easy to reproduce, replicate, and finally to understand how they work.

I have developed two MT models: one for English-to-Polish and another for English-to-Spanish. These languages were selected for two primary reasons:

1. Linguistic Diversity: English, Polish, and Spanish belong to different language families—Germanic, Slavic, and Romance, respectively—allowing for a more comprehensive examination of translation challenges across diverse linguistic structures.
2. Resource Availability: Spanish is one of the most widely spoken languages globally and has a wealth of natural language understanding (NLU) resources. In contrast, Polish is less-resourced, making it an ideal candidate for exploring machine translation capabilities in low-resource settings. The variation in resource availability between

these languages also enables a more nuanced positioning of our results, offering insights into the performance scalability of our models.

This selection aligns with the overarching goal of my thesis, which aims to develop machine translation solutions for languages with limited resources.

As illustrated in Figure 13, the baseline model (M2M100) was adapted using a parallel corpus that encompassed both in-domain (IVA_MT) and out-of-domain (general MT) data. This dataset will be further detailed in the subsequent chapter, as it was initially crafted for slot-transfer applications but also serves the purpose of IVA domain adaptation. The dataset is publicly available and can be accessed online[4]. For the English-to-Polish adaptation, the dataset comprised 20.4k training, 3.68k validation, and 5.49k testing utterances, along with 1k test cases from the WMT20 dataset. On the other hand, the English-to-Spanish dataset included 8.42k training, 1.53k validation, and 2.21k testing utterances, supplemented by 3k test cases from the WMT13 dataset. It should be noted that both WMT datasets were used exclusively for testing purposes.

Upon completing the adaptation, I have evaluated both the baseline and adapted models using two different test sets: one derived from the adaptation corpus and another from WMT. The former helps us understand the effectiveness of the adaptation process, while the latter confirms that the model has not overly adapted to the training data.

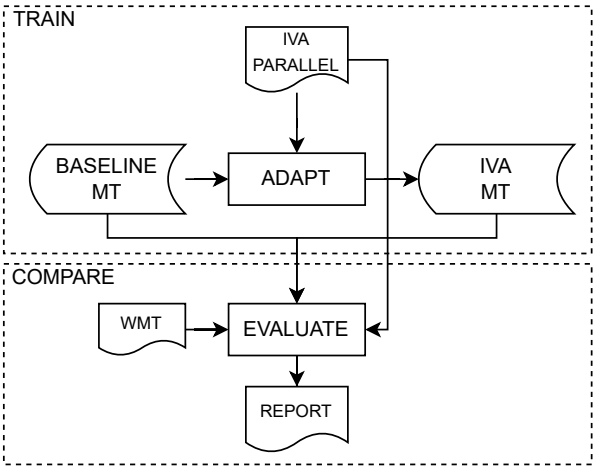

Figure 13. Domain adaptation and evaluation process.

The results of the adapted models are summarized in Table 11. As expected, the models adapted for the IVA domain showed significant improvements, although performance on out-of-domain datasets dropped. The BLEURT scores reported are the arithmetic means of individual scores.

The confidence intervals for BLEU and BLEURT were computed using bootstrap resampling, a non-parametric statistical method. Specifically, multiple bootstrap samples were generated from the original dataset by randomly drawing sentences with replacement.

[4] Hugging Face Dataset: `https://huggingface.co/datasets/cartesinus/iva_mt_wslot`

Table 11. Comparison of domain adaptation techniques: LoRA and Fine-Tuning applied to Base M2M100 model and compared with GPT-3 model across IVA_MT in-domain and WMT out-of-domain datasets.

Dir.	Dataset	Metric	GPT-3.5	Base M2M100	LoRA Adapt.	Fine-Tuning
en-es	IVA_MT	BLEU	40.72 ± 1.32	32.04 ± 1.29	9.51 ± 0.91	**51.17** ± 1.39
		BLEURT	0.73 ± 0.01	0.64 ± 0.01	0.32 ± 0.01	**0.76** ± 0.01
	WMT13	BLEU	26.09 ± 0.68	**31.29** ± 0.71	19.63 ± 0.57	20.87 ± 0.58
		BLEURT	**0.66** ± 0.01	**0.66** ± 0.01	0.47 ± 0.01	0.54 ± 0.01
en-pl	IVA_MT	BLEU	25.41 ± 0.96	22.79 ± 1.09	9.33 ± 0.96	**45.03** ± 1.28
		BLEURT	0.69 ± 0.01	0.63 ± 0.02	0.27 ± 0.01	**0.74** ± 0.02
	WMT20	BLEU	16.08 ± 1.14	**22.36** ± 1.22	8.50 ± 0.91	16.67 ± 1.03
		BLEURT	0.65 ± 0.01	**0.72** ± 0.01	0.33 ± 0.01	0.63 ± 0.01

The score is then calculated for each bootstrap sample, forming a distribution of BLEU and BLEURT scores. The 95% confidence interval is derived from this distribution by selecting the 2.5th and 97.5th percentiles, providing a range in which we are 95% confident that the "true" BLEU and BLEURT score of the system resides. This interval serves as an indicator of the score's stability and the system's performance variability.

Based on this robust statistical framework, it can be confidently stated that fine-tuning significantly outperforms both the LoRA adaptation and the baseline M2M100 model. Despite LoRA's training and validation metrics trailing by approximately 20 points, it failed to achieve comparable performance to fine-tuning. For example, in the English-to-Polish pair, LoRA started at a BLEU score of 24.91 and ended at 38.24, whereas fine-tuning started at 57.44 and ended at 61.62. A similar trend was observed for English-to-Spanish, suggesting that LoRA's learning paradigm may not be as efficient as full-network retraining in this specific context.

In both the English-to-Spanish and English-to-Polish pairs, the baseline M2M100 model outperformed GPT-3 on the WMT datasets. This is in line with M2M100's design as a dedicated translation model. However, GPT-3 showed competitive performance, especially for the well-resourced Spanish language, confirming the known efficacy of large language models in languages with abundant data.

GPT-3 outperformed the baseline M2M100 model in the IVA domain but was still surpassed by the fine-tuned models, further highlighting the effectiveness of fine-tuning for domain-specific tasks.

In the following section, I will delve into the details of the impact of fine-tuning on model performance.

3.4.4. Analyzing Impact of Fine-Tuning

BLEURT correlates with human judgments better than BLEU [66]–[68]. Therefore, I have decided to use it to analyze the impact of domain adaptation. I have started my

analysis by computing the Kernel Density Estimation (KDE) distribution of BLEURT scores for MT outputs from both a baseline model and an adapted model across two distinct datasets: IVA MT (in-domain) and WMT20 (out-of-domain). KDE is a non-parametric method for estimating the probability density function of a variable.

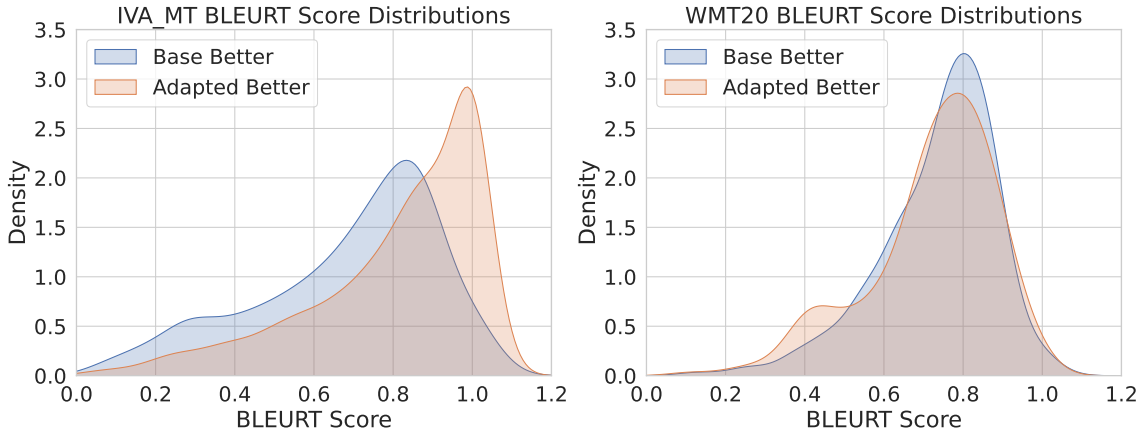

Figure 14. Kernel Density Estimation of BLUERT score for MT outputs from baseline and adapted English-to-Polish MT model. The left panel shows in-domain (IVA_MT) scores, and the right panel shows out-of-domain (WMT20) scores.

Figure 14, which focuses on the English-to-Polish translation model, shows how the model performs across different domains. The left panel of the figure shows the in-domain dataset. In this case, the adapted model performs better, as indicated by the "Adapted Better" curve having more scores clustered around 1.0. This higher density at higher BLEURT scores suggests that the adapted model generally produces better translations for in-domain tasks. However, it's worth noting that the adapted model also shows a small peak at lower BLEURT scores, indicating a cluster of poorly translated samples that could affect the overall score. Conversely, in the out-of-domain dataset presented on the right panel, the baseline model performs slightly better. This is evidenced by a higher peak in higher BLEURT values for the baseline model. The adapted model expresses a lower main peak due to the existence of a smaller side-peak to the left, resulting from a cluster of poorly translated samples. This subtlety suggests that while the adapted model is specialized for in-domain tasks, it does not generalize as effectively to out-of-domain scenarios. The statistical significance of the BLEURT score differences between the baseline and adapted models across both datasets further validates the distinct behavior of each model.

In Table 12, two examples of the best and worst translations from in-domain (IVA_MT) and out-of-domain (WMT) are measured by the difference between BLEURT. In the first example (ID=1), the positive impact of adaptation can be observed. This is a typical example of vocabulary mismatch. The base model could not translate (localize) the name of the vacuum cleaning robot to the correct Polish "rumba" and left this word untranslated. Also, the missing context verb was not translated correctly. In the second

Table 12. Best and worse translation examples comparing MT model before and after domain adaptation.

ID	Column	Value
1	Testset	IVA_MT (in-domain)
	Input	switch on the roomba
	Reference	włącz rumbę
	Base MT	przejście na roomba
	Adapted MT	włącz rumbę
	BLEURT Δ	+0.9540
2	Testset	IVA_MT (in-domain)
	Input	push repeat on this song
	Reference	powtórz tę piosenkę
	Base MT	powtórz tę piosenkę
	Adapted MT	powtórz ten utwór
	BLEURT Δ	−0.2334
3	Testset	WMT20 (out-of-domain)
	Input	any sentence of his was actually accepted
	Reference	właściwie każde jego zdanie było akceptowane
	Base MT	każdy wyrok jego został faktycznie przyjęty
	Adapted MT	wszystkie jego wypowiedzi zostały faktycznie zaakceptowane
	BLEURT Δ	+0.3102
4	Testset	WMT20 (out-of-domain)
	Input	how to dress a baby for a baptism?
	Reference	jak ubrać dziecko na chrzest?
	Base MT	jak ubrać dziecko na chrzest?
	Adapted MT	jak się ubrać dziecko na baptism?
	BLEURT Δ	−0.4152

example (ID=2), we see a sentence where the adapted model performed worse than the base model (negative BLEURT delta). Although the error is subtle because both “utwór” and “piosenka” in this context are correct expressions for a song, after careful analysis, this can be interpreted as model over-fitting because the training corpus consisted of more training examples with “utwór”. Over 65% of 5393 test sentences received a better BLEURT score for the adapted model, with an additional 9.5% sentences that received a score making the base and adapted model translations equal. In the third and fourth examples (ID=3, ID=4), the result of adaptation on out-of-domain sentences is presented. Quality of translation in out-of-domain test cases dropped. Over 72% of 1001 test sentences received a worse BLEURT score for the adapted model, with an additional 8.8% sentences that received a score making the base and adapted model translations equal.

In the IVA MT (in-domain) dataset, the adapted model generally outperforms the baseline model, particularly for shorter sentences. Specifically, the adapted model achieves an average BLEURT score of approximately 0.89, compared to 0.77 for the baseline, with an average sentence length of around 34.6 words and high lexical diversity (1.00). Conversely,

in the WMT20 (out-of-domain) dataset, the baseline model tends to excel, most notably in sentences of medium to long lengths. Here, the baseline model shows a more consistent performance across a range of BLEURT scores, highlighting its adaptability to diverse text types.

3.5. Conclusions

Chapter 3 addressed the challenges and solutions for adapting MT for the IVA domain. One of the most significant limitations in the current landscape is the scarcity of comprehensive NLU corpora. To mitigate this, I introduced Leyzer - a multilingual, multi-domain dataset uniquely designed to evaluate NLU systems. Leyzer stands out for its breadth, covering three languages, 20 domains, and 187 intents, and its depth, assigning naturalness levels and verb patterns to each utterance.

The dual role of Leyzer was crucial to this research. It served not only as a robust benchmark for evaluating the adapted MT models but also contributed to the adaptation dataset itself. This enabled a more nuanced understanding of how MT models perform when adapted to specific domains and languages.

The exploration of multiple adaptation techniques, LoRA Adapters and Fine-tuning allowed for a comparative analysis that highlighted the strengths and limitations of each method, thereby providing insights into their suitability for different scenarios. Fine-tuning emerged as the more effective technique, particularly in the IVA domain, with improvements of +19.62 ± 1.6 and +10.45 ± 1.92 BLEU points for English-to-Polish and English-to-Spanish models, respectively.

The findings of this chapter strongly support the central thesis that MT, when aptly adapted, can serve as a pivotal tool for localizing NLU models. Furthermore, the comparative analysis of adaptation techniques contributes to a more comprehensive understanding of the landscape, highlighting the importance of methodological choice in achieving optimal performance. By demonstrating significant performance gains through Fine-Tuning, this chapter not only robustly defends Thesis T1 but also opens avenues for future work focused on the methodological nuances of MT adaptation for multilingual NLU.

4. Entity Translation and Transfer

This chapter is partially based on article “Slot Lost in Translation? Not Anymore: A Machine Translation Model for Virtual Assistants with Type-Independent Slot Transfer”, presented by the author of this thesis at the 30th International Conference on Systems, Signals and Image Processing (IWSSIP) conference on 28 June 2023 in Ohrid, North Macedonia.

4.1. Slot Transfer Task

One of the key challenges in the translation of training sentences for NLU is that they consist of slots and annotations on the level of words that carry information for the NLU system. For example, in the sentence “play radiohead” typically, *radiohead* will be annotated as slot *music_artist*. MT systems used to translate sentences for NLU training must be able to annotate and transfer slots.

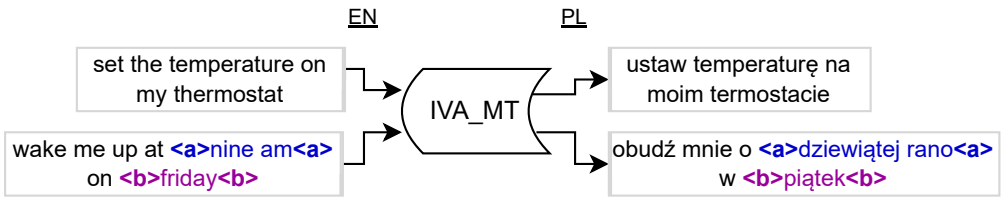

Figure 15. Example of plain text input and annotated input translation.

Slot transfer is a process of re-annotating the entity in the same part of the sentence in the target (translation) sentence as in the source. The slot type must remain the same as in the source, while the value of that slot must either be translated, localized, or transferred intact, depending on the slot type. For example, in the source sentence, “play ⟨artist⟩radiohead⟨artist⟩” slot type ⟨artist⟩ needs to be re-annotated in the target sentence. However, in this particular example, slot value “radiohead” should not be translated or localized.

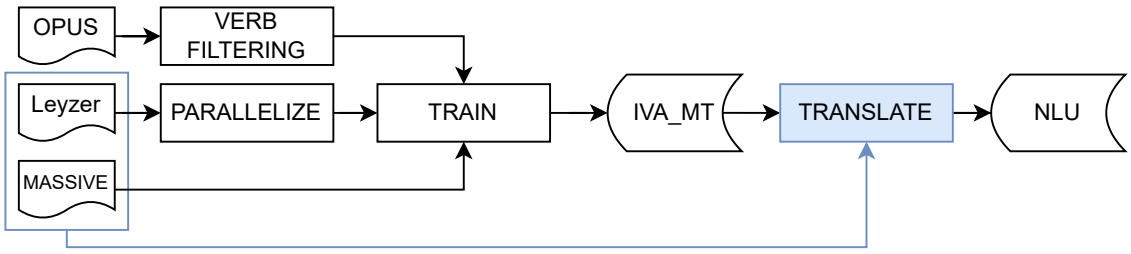

Figure 16. The proposed method for parallel data collection.

In this work, I aim to build an MT that can generate high-quality translations and transfer slots between source and target sentences that will be used to prepare data for

IVA model training. To create such a model, I propose a language-independent method consisting of three stages, as shown in Figure 16:

1. Preparation of a parallel dataset with slot annotations for transfer task. Leyzer needs to be paralyzed, and from OPUS, sentences similar to IVA were selected,
2. Creation of MT models from parallel corpora that can transfer slots between languages,
3. Training of NLU models from translated resources. Evaluation and analysis of the impact of MT on NLU quality were performed on testset derived from MASSIVE corpus.

4.2. Parallel Dataset with Slot Annotations for Slot Transfer Task

To adapt an MT model to the IVA domain, we need sentences with the following attributes: short (below 160 characters), simple (one independent clause), imperatives, or interrogatives with a pragmatical goal of performing some action by an IVA. As presented in Table 13, seven data sources were used to create the parallel dataset. I selected the two biggest IVA corpora: MASSIVE and Leyzer [35] and added OPUS [69] as a counterbalance that provides generalization and protects the model from overfitting.

Table 13. Composition of the parallel dataset used to train and evaluate IVA_MT model. All sizes are given in terms of sentences.

Corpus	Train Size	Validation Size	Test Size
MASSIVE 1.1	11514	2033	2974
Leyzer 0.2.0	3974	701	1380
OpenSubtitles (OPUS)	2329	411	500
KDE (OPUS)	1154	241	241
CCMatrix (OPUS)	1096	232	237
Ubuntu (OPUS)	281	60	59
GNOME (OPUS)	14	3	3
total	*20362*	*3681*	*5394*

MASSIVE is a multilingual dataset created to evaluate IVAs. It comprises 18 domains, 60 intents, and 55 slots across 51 languages. The corpus consists of interrogative and imperative utterances directed at a device. MASSIVE is parallel and can be used to train MT models without additional processing. I took all sentences from MASSIVE and replaced slot types with consecutive alphabet letters as shown in Fig 15.

Leyzer is another dataset created to evaluate IVAs. It comprises 21 domains and 193 intents across three languages (English, Polish, and Spanish). The corpus consists of imperative utterances uttered to a device. Leyzer is not a parallel corpus, therefore, I used the multilingual Universal Sentence Encoder model [70] to generate an English-to-Polish parallel subset. As a result, 6055 parallel sentences were extracted covering 124 out of 192

intents available in Leyzer. Slot types specific to Leyzer were also replaced with consecutive alphabet letters.

Finally, five sub-corpora available in the OPUS project were used. KDE, Ubuntu, and GNOME contain similar sentences to typical IVA sentences. In the case of all OPUS-derived sentences that do not contain slots, I used polyglot[5] to detect name entities that were later replaced with consecutive letters of the alphabet. Polyglot has an error rate of about 20% in annotating entities; it either incorrectly marks portions of the sentence as entities when they are not, or fails to annotate actual entities. However, this is not a significant issue for our task. We are training the model to transfer any annotated words, regardless of whether they are correctly identified as entities or not. The focus is on the ability to transfer these annotations accurately, not on the semantic correctness of what is being annotated. Therefore, the errors in Polyglot's entity recognition do not adversely affect the task at hand. CCMatrix and OpenSubtitles were selected to counterbalance the dataset and help models gain generalization power. From OpenSubtitles, I selected only sentences that consisted of verbs extracted earlier from MASSIVE and Leyzer (in total, 234 unique verbs).

As a result, I created a dataset consisting of 20,362 training, 3,681 validation, and 5,293 test sentences. The presented dataset covers 184 different intents available in MASSIVE and Leyzer datasets and after manual clustering based on the type of verb, an additional 72 intents in KDE, Ubuntu, and GNOME subsets were assigned. Slot annotations in my dataset have been taken from 55 unique types of slots from MASSIVE, 37 unique slots from Leyzer, and three additional types extracted from OPUS corpus using the name entity recognition model.

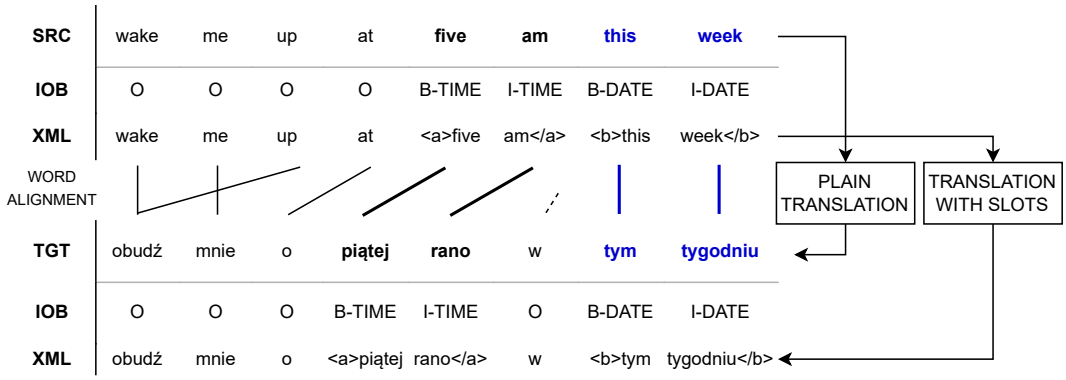

Figure 17. Slot Annotation and Word Alignment in Multilingual NLU.

Figure 17 illustrates the slot annotation process in the dataset, which employs a simplified XML annotation schema. This schema is designed to be compatible with the IOB format, allowing for seamless conversion between the two. One of the challenges highlighted in the figure is the issue of word alignment between languages. For instance,

[5] `https://pypi.org/project/polyglot`

the English phrase “wake up” corresponds to a single word “obudź” in the target language. This presents a unique challenge for the slot projection algorithm, which must accurately align and annotate multi-word phrases in the source language with their single-word counterparts in the target language. In the system, I address this by annotating each component of the multi-word phrase individually. For example, if a verb like “wake up” needs to be annotated, it would appear as “<a>wake me <a>up” in the XML format.

4.3. Machine Translation with Slot Transfer

While Chapter 1 focused on the domain adaptation of the M2M100 model to IVAs, this chapter aims to evaluate the model’s slot transfer capabilities through experiments. The M2M100 [71] model was used as a base. It provides an excellent base for future expansion, especially when considering low-resource languages, as it was trained to translate 100 languages. Moreover, this architecture is considered state-of-the-art, and most systems participating in WMT-22 implemented similar Transformer architecture [72].

Figure 18. Training loss (per step) and BLEU score for each of 10 evaluation epochs.

I have adapted the base model for 10 epochs, as the foundation model was already pre-trained on the MT task. Adam [65] was used for optimization with an initial learning rate of $2e-5$. Training progress is presented in Figure 18. All data available in the training part of the corpus was used. Each epoch was evaluated on the validation subset. The batch size was 4.

The final results presented in Table 14 are from the epoch that reached the highest BLEU score [11] on the validation set and was later evaluated on the test set part of the iva_mt_wslot corpus. While in Chapter 1, I used the same dataset, the difference is that in

Table 14. Results of English-to-Polish MT model adapted to IVA domain capable of transferring slots.

Model	BLEU	BLEU w/ slots	Slot F1 (%)
(*reference*) GPT-3 few-shot adapt.	23.23 ± 0.91	45.59 ± 0.76	55.07 ± 2.03
Baseline M2M100-418M	22.79 ± 1.09	-	-
IVA_WSLOT M2M100-418M	42.57 ± 1.30	59.97 ± 1.07	65.41 ± 1.81

this experiment, I used annotated sentences to train, evaluate, and test model capabilities to transfer slots, which is represented with "BLEU w/slots". The confidence intervals for F1 and BLEU scores were determined using bootstrap resampling, a non-parametric statistical technique. For F1 scores, 1,000 bootstrap samples were generated from 50% of the original dataset of reference and hypothesis sequences. Each sample was randomly drawn with replacement, and the F1 score was calculated for each resample. This yielded a distribution of F1 scores, from which the 95% confidence interval was extracted, specifically between the 2.5th and 97.5th percentiles. The same bootstrap resampling method was employed for BLEU scores. These confidence intervals serve as indicators of score stability and offer a range within which we are 95% confident that the "true" score resides, thereby providing insight into the system's performance variability.

The results clearly indicate that the model adapted to the IVA domain is 19.78 BLEU points better than the baseline, with confidence intervals of ±1.30 and ±1.09 for the adapted and baseline models, respectively. In a manual analysis, only minor translation problems were observed in both models. No significant issues were identified in the translations or the slot transfers. While there were instances of slot misalignment, these did not follow any discernible recurring pattern, indicating that the errors were largely random and not systematic.

The "BLEU w/ slots" metric, which includes additional XML tags to represent slot values, inherently inflates the BLEU score. These extra tokens are accounted for in both the reference and hypothesis sentences, making the metric not directly comparable to the standard BLEU score. However, achieving a score near 60 in this specialized evaluation framework suggests effective handling and alignment of these additional tokens. It is a strong indicator of good performance when evaluated within this specific methodology.

Slot transfer was measured with a weighted-averaged F_1-score, the harmonic mean of the precision and recall. For sentences with only one slot type, results are much higher: they yield a weighted-averaged F_1-score of 87.52% ± 1.31. The baseline model has neither BLEU w/ slots nor Slot F_1-score results since this model cannot transfer slots.

In addition to the M2M100 model, the GPT-3 (specifically, the *text-davinci-003* version) was utilized as a state-of-the-art commercial reference system for comparison. GPT-3 was

prompted with a few-shot prompt consisting of five examples to perform the translation and slot annotation tasks as presented in Listing 1.

The comparative analysis revealed that while GPT-3 delivered commendable results, the adapted IVA M2M100-418M model outperformed it, especially regarding BLEU score and Slot F1 score as detailed in Table 14. It is, however, noteworthy to mention that the results from GPT-3, a general-purpose model, were impressively competitive, underscoring its potential as a reference system in the domain of machine translation and slot transfer.

```
Translate the following English sentences to Polish and annotate the slots as shown in the examples:
Example 1:
English: "wake me up at <a>five am<a> <b>this week<b>",
Polish: "obudź mnie o <a>piątej rano<a> <b>w tym tygodniu<b>"

Example 2:
English: "play <a>radiohead<a> <b>creep<b>",
Polish: "odtwórz <b>creep<a> od <a>radiohead<a>"

Example 3:
English: "hello i want to turn off my <a>wemo plug<a>",
Polish: "chcę wyłączyć <a>wtyczkę wemo<a>"

Example 4:
English: "do i have any <a>appointments<a>",
Polish: "czy mam jakieś <a>spotkania<a>"

Example 5:
English: "please insert a <a>data<a> medium",
Polish: "proszę włożyć płytę z <a>danymi<a>"
```

Listing 1. Examples of few-shot prompts used for GPT-3.

4.4. Conclusions

Chapter 4 explored the details of slot transfer in the context of MT for IVAs. The chapter underscored the crucial role of slots and annotations in training sentences for NLU, emphasizing the need for MT systems to annotate and transfer these slots adequately. The presented IVA_MT model displayed the ability to translate and transfer slots between source and target sentences effectively, which is crucial for the performance of slot-filling models used in NLU. Through rigorous evaluation, it was found that the IVA_MT model significantly improved the slot F_1-score, a direct metric for evaluating the preservation and appropriate translation of named entity locations, which are integral to semantic annotations in NLU training resources.

The chapter also introduced a language-independent method for creating such a model, detailing the preparation of a parallel dataset with slot annotations for the slot

transfer task. The adaptation of the M2M100 model, a state-of-the-art architecture, was discussed, highlighting its potential for future expansion, especially for low-resource languages. Furthermore, the chapter provided a comparative analysis with GPT-3, a commercial state-of-the-art baseline, demonstrating that the presented model surpassed GPT-3 in terms of both BLEU scores and slot F_1-score measurements, indicating a favorable advancement over the commercial baseline. Due to time constraints, the experiments in this chapter focus solely on the English-to-Polish model.

The experimental results and analyses presented in this chapter robustly defend Thesis T2, asserting that to translate NLU training resources, which comprise semantic annotations, MT must preserve and appropriately translate named entity locations. The significant improvement in slot F_1-score measurements affirms the model's ability to accurately handle named entity locations, thereby enhancing the translation of NLU training resources and affirming the feasibility and effectiveness of the proposed IVA_MT model in addressing the challenges outlined in Thesis T2.

5. Multiverb and Multivariant Machine Translation

This chapter is partialy based on the article “Optimizing Machine Translation for Virtual Assistants: Multi-Variant Generation with VerbNet and Conditional Beam Search” [37].

Multilingual NLU models are a major focus in NLP as they enable virtual assistants to manage multiple languages. However, the scarcity of multilingual training data often leads to the underrepresentation of some languages. While the manual translation of training sentences can address this problem, it is a time-consuming and costly process prone to errors and ambiguities that can compromise model quality. Moreover, manual translation struggles to adapt to language changes or the introduction of new languages to the virtual assistant.

In this context, using MT systems as a source of translations seems to be an attractive alternative for acquiring multilingual learning data. Creating multilingual NLU models by translating a learning sentence into multiple languages using MT models seems possible and promising.

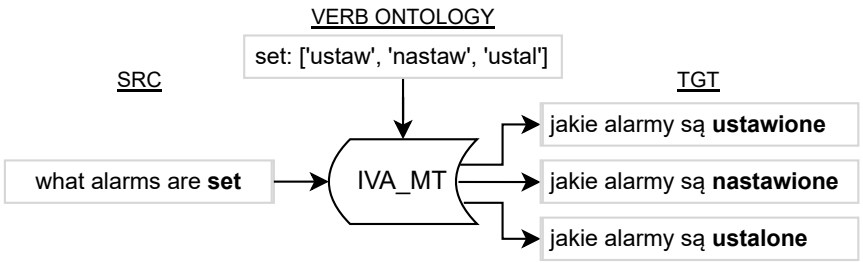

Figure 19. Example of multiple variants translations based on verb ontology and constrained beam search.

MT systems, used to generate sentences for training NLU models, should produce multiple correct translation variants. This is crucial as languages often have numerous grammatical forms and ways of conveying information. For instance, English has various verb forms, such as regular, irregular, and modal verbs, with potentially different translations in other languages. If an MT system generates only one translation variant, the NLU model might not learn to recognize others, compromising the model’s quality. Hence, MT systems should create multiple accurate translation variants to cover all possible patterns, enhancing the performance of NLU models.

This chapter outlines the development of a multi-variant MT model that leverages a verb ontology tailored for the IVA domain. A secondary objective is to create an ontology that is user-friendly and easily modifiable by NLU developers. To achieve these goals, verbs were extracted from various IVA corpora and lexically matched to classes in VerbNet 3.0 [73], [74]. Subsequently, using the linkage to Princeton WordNet 3.0 [75], translations for these verbs in the target language were extracted. It should be noted that while

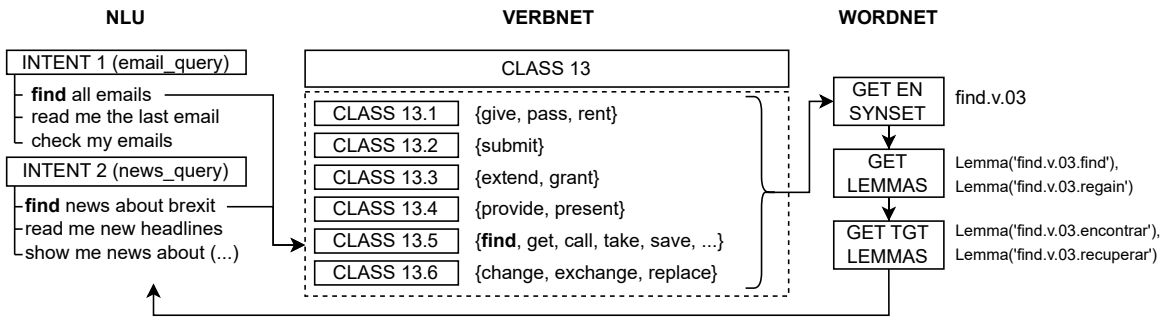

Figure 20. Overview of the presented method. NLU verbs are initially matched to VerbNet classes through a lexical matching process, where a verb is considered a match if it is identical to a word in the VerbNet class. Each VerbNet class contains multiple WordNet synsets, which are subsequently retrieved. These synsets are then matched back to the original NLU verbs, resulting in an extended list of NLU verbs.

verbs can have multiple meanings, our experimental results indicate that the MT model's performance is not adversely affected by this polysemy. Even verbs that are "incorrectly" linked due to their multiple meanings contribute to improved system accuracy. Moreover, since the ontology is text-based, any such inaccuracies can be easily identified and rectified by NLU developers. In Figure 20, I present processing steps used to find verb equivalent in the target language to increase the variance of training resources. The proposed method consists of the following stages:

1. Creation of multilingual dictionary with verb translation for the IVA domain,
2. Creation of MT model (based on M2M100 architecture) from parallel corpora and creation of tools that guide decoding (constrained beam search) to generate multiple hypotheses,
3. Translation of NLU training resources, training of NLU model, and evaluation and analysis of the impact of MT on NLU quality.

5.1. Verb Ontology for IVA NLU

I start my investigation by analyzing verbs in NLU corpora. Verbs carry crucial information about the event or action being described [76]. IVA commands semantics is composed of a verb and its parameters. In this work, I analyzed eight popular NLU corpora (listed in Table 15) and extracted 374 English verbs. I then created a ranking list [6] where the frequency of occurrences of verbs in all corpora is counted. The first verb on the list represents the most frequently used verb in all analyzed corpora.

In Table 15, the top five positions on verb occurrence ranking are presented. The highest ranked verbs are: *set*, *show*, *remind*, *play* and *give*. Most analyzed NLU corpora

[6] Available at: `https://github.com/cartesinus/multiverb_iva_mt/blob/main/data/nlu_corpora-common_verbs.tsv`

consisted of calendar, alarm, and music domains, which explain why given verbs are most popular.

While creating the ranking list, I noticed that each NLU corpus presents the same trend where the most frequent verbs can be found in around 20% of utterances. Figure 21 illustrates that the trend in IVA corpora closely resembles the Zipf distribution, albeit with some deviations. A similar trend can be found in other linguistic resources, for example, VerbNet [77].

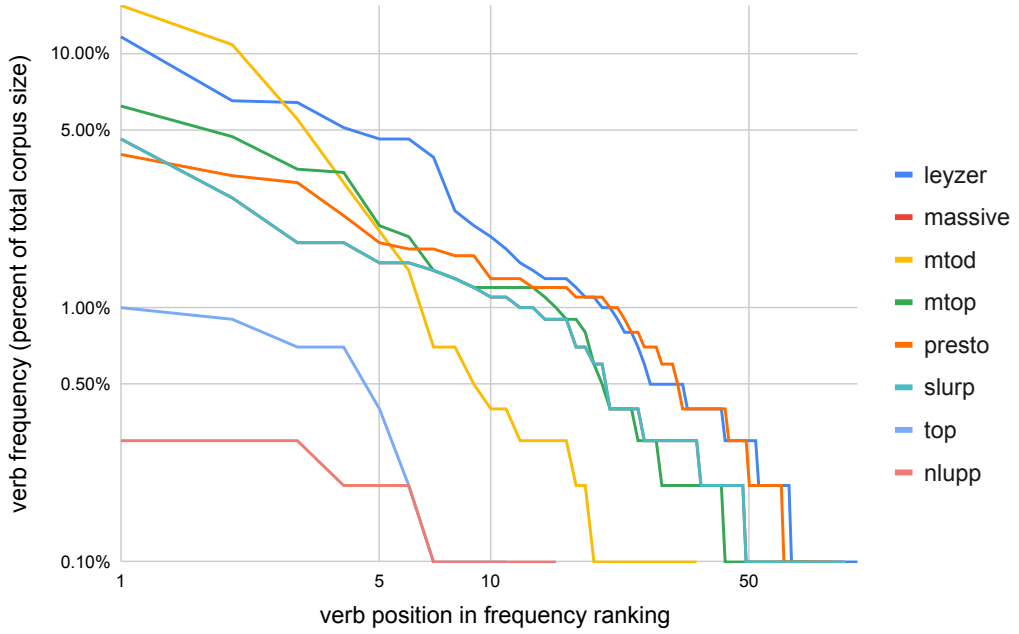

Figure 21. Verb frequency and verb position on the ranking list for selected IVA datasets presented in logarithmic scale.

5.1.1. Mapping IVA Verbs to Levin Classes and VerbNet

Verbs extracted from NLU corpora often span multiple domains. For instance, the verb *set* could be used to set an alarm or adjust screen brightness. To address this complexity, I utilized Levin's verb classification [78] to categorize verbs of similar semantic properties. Levin classified 3,024 verbs into 48 broad and 192 fine-grained classes based on patterns of syntactic alternations that correlate with semantic properties. These classes are employed in this article to identify IVA verb frames. Although Levin's classes were initially designed to understand syntactic and semantic alternations in verbs, they can be adapted to comprehend IVA capabilities. The key is to interpret these verbs in the context of virtual actions and outputs. While IVAs cannot perform all human tasks, they can simulate a wide array of actions in a virtual setting.

While automated verb classification methods have been explored [79], these approaches primarily focus on general language and rely on syntactic features. They have shown promising results in classifying verbs into Levin classes, but their applicability

to the specialized language of IVAs remains uncertain. Annotated corpora and theories like speech act theory [80] provide valuable insights into human-machine interactions. However, they often do not focus on the specific verbs employed in IVAs, nor are there resources readily available for the automatic or semi-automatic classification of such verbs. This creates a verification challenge, as existing methods cannot be definitively cross-referenced for accuracy in this specialized domain. Therefore, we developed our own classification method to better address the unique linguistic features of IVA interactions.

Out of 270 verbs, 14.88% could not be found in VerbNet or did not belong to WordNet synset, making it impossible to use in the algorithm. 7.04% verbs matched more than one VerbNet class. 7.27% verbs belong to a VerbNet class where no other verb from NLU corpora belongs.

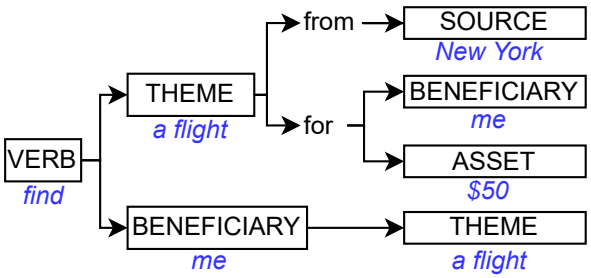

Figure 22. Example of frames available in VerbNet for verb belonging to Verbs of Change of Possession (Class 13).

VerbNet defines semantic frames in which a given verb can be found. The example presented in Figure 22 shows four semantic frames belonging to class 13 where verb *find* appears. Verbs that belong to that class reflect the change of possession. From the frames presented in the example, several utterances belonging to the different IVA domains can be constructed.

Below, I present verbs found in NLU corpora that have been successfully matched to VerbNet classes. Using those classes, other instances (verbs) of the same frame can be found. The ten most frequent classes found in NLU corpora are:

1. **Verbs of Change of Possesion** (Class 13) where 10.73% of the IVA verbs belong. Verbs in this class encompass a set of verbs that denote a transfer of possession or provision of an item or service from one entity to another. In the IVA context, users often employ these verbs to request specific actions or information from the assistant. For instance, when a user says, "Give me my latest photos from the gallery," they are essentially asking the virtual assistant to provide (or "give") them access to specific data (photos) from a particular location (gallery). The user (recipient) is indirectly instructing the assistant (provider) to fetch or display the desired content (item). The following sub-classes can be further distinguished:

   a) 13.1 with *give*, *pass*, and *rent*

b) 13.2 with *submit*,
c) 13.3 with verbs such as *extend* and *grant* that relate to the change of possession that will take place in the future,
d) 13.4 with *provide*, *present* that can be described as "X gives something to Y that Y needs or deserves",
e) 13.5 (Get and Obtain Verbs) with *find*, *get*, *call*, *take*, *save*, *order*, *keep*, *book*, *buy*, *select* and other,
f) 13.6 with *change*, *exchange*, *replace* that relate to exchanging one thing for another,

2. **Verbs of Communication** (Class 37) where 9.34% of the IVA verbs belong. In IVAs, users often use them to request information or actions from the assistant. For example, "Explain this concept to me" seeks clarity, while "read" suggests the assistant's better content access. Verbs like "email" indicate the assistant's role in facilitating communication. These verbs underscore the evolving role of virtual assistants, from simple tools to sophisticated communication entities. The following sub-classes can be further distinguished:
   a) 37.1 (Verbs of Transfer of Message) with *tell*, *read*, *write*, *ask*, *explain*, *dictate*, *summarize* that are verbs of type of communicated message,
   b) 37.2 with *remind*, *update*, *notify*, *inform*
   c) 37.3 with *call*, which gathers the verbs of a manner of speaking and verbs in this class are distinguished from each other by how the sound is expressed. This is not a perfect match for IVA, but members are also not very far from IVA context,
   d) 37.4 with *email*, *phone*, *broadcast*, *ring* that relate to communication via these instruments of communication and are zero-related to the same noun,
   e) 37.5 with *speak*, *talk* that do not take sentential complement,
   f) 37.6 with *chat*
   g) 37.7 with *repeat*, *say*, *report*, *note*, *suggest*
   h) 37.8 with *complain* that specify the speaker's attitude or feeling,
   i) 37.9 with *alert* and *brief*.
3. **Verbs of Creation and Transformation** (Class 26) where 6.92% of the IVA verbs belong. Members of that class are transitive verbs where one argument (agent) creates (brings something into existence) or transforms an entity (changes its state or form). While IVAs don't physically create or transform objects, they do "create" virtual outputs for users. For instance, when a user asks Bixby to "arrange my meetings for the day", the IVA organizes the user's schedule, effectively "creating" a structured day plan. Similarly, when a user asks Bixby to "convert USD to EUR", the IVA transforms the currency value, providing a new output,
4. **Aspectual Verbs** (Class 55) where 5.19% of the IVA verbs belong. These verbs describe the initiation, termination, or continuation of an activity. Users often employ these verbs to control the start, continuation, or cessation of tasks performed by the VA.

The relationship between the user's utterance and the expected action is direct: the aspectual verb provides clear cues about the desired phase of the task, whether it is an initiation, continuation, or termination,

5. **Verbs of Change of State** (Class 45) where 4.50% of the IVA verbs belong. All of the verbs in this class relate to the change of state, with several sub-classes that define this state in more detail. When users employ these verbs in their utterances, they typically expect the IVA to either provide information related to the change or execute an action that results in the desired change. The relationship between the user's utterance and the expected action is direct: the verb provides clear cues about the nature and direction of the desired change,
6. **Verbs of Putting** (Class 9) where 4.15% of the IVA verbs belong. These verbs refer to putting an entity at some location. For instance, users might use Put Verbs to set reminders or arrange tasks. E.g., "Set a reminder for tomorrow.". With Verbs of Putting in Spatial Configuration, *suspend* is relevant in contexts like pausing tasks or suspending processes. Funnel Verbs could be used in contexts like adding items to lists or pushing tasks to a queue. Finally, Coil Verbs are connected with programming capabilities, *loop* might be used to indicate repetitive tasks.
7. **Verbs of Predicative Complements** (Class 29) where 4.15% of the IVA verbs belong. Verbs belonging to that class are foundational to human communication, especially when seeking information, validation, or expressing opinions. When users employ these verbs in their interactions with IVAs, they typically expect the assistant to provide relevant information, confirm their beliefs, or assist in categorizing or naming items. Appoint and Characterize Verbs are used when seeking specific information or categorization. For instance, "How would you rate this song?" or "Describe this image." Dub Verbs can be used in contexts like naming alarms or playlists. E.g., "Call this playlist 'Workout Tunes.". Declare Verbs might be used to express opinions or seek validation. E.g., "I believe it is going to rain today. What do you think?". Conjecture Verbs can be used when users are unsure about something and seek the assistant's input. For example, "I guess it is late. What's the time?",
8. **Verbs of Sending and Carrying** (Class 11) where 3.81% of the IVA verbs belong. Users employ these verbs to command the IVA to transfer, move, or retrieve information or perform specific tasks related to sending and carrying. Recognizing these verbs and their nuances is crucial for IVAs to ensure they respond appropriately to user commands, especially in contexts like messaging, reminders, and navigation. Send Verbs are frequently used in the context of message dispatching. For instance, users might say, "Send this message to John" or "Mail this document to my boss." The expected action is for the IVA to facilitate the dispatching of the message or document to the intended recipient. Bring and Take verbs can be employed in commands like "Bring up my last email" or "Take me to the home screen." The user expects the IVA to

Table 15. Top 5 English verbs from occurrence ranking and occurrence frequency in each of selected NLU corpora.

Dataset	*Set*	*Show*	*Remind*	*Play*	*Give*
Leyzer [35]	0.7%	11.6%	0.3%	1.1%	6.5%
MASSIVE [49]	1.8%	1.5%	1.3%	4.6%	1.1%
MTOD [81]	15.4%	3.1%	10.8%	0.0%	0.4%
MTOP [46]	6.2%	2.1%	4.7%	3.5%	1.2%
PRESTO [47]	0.4%	3.1%	0.2%	0.7%	0.3%
SLURP [40]	1.8%	1.5%	1.3%	4.6%	1.1%
TOP [43]	0.0%	0.7%	0.0%	0.0%	0.7%
NLU++ [39]	0.1%	0.2%	0.0%	0.0%	0.1%

retrieve specific information or navigate to a particular interface. Carry Verbs might be used metaphorically. For instance, “Carry this reminder over to tomorrow” would mean the user wants the IVA to reschedule a reminder,

9. **Verbs of Removing** (Class 10) where 3.11% of the IVA verbs belong. The relationship between users employing these verbs and the expected action is that users command the IVA to remove, eliminate, or refine something. Remove Verbs are commonly used in tasks like file management or editing. For instance, “Delete the third paragraph” or “Remove this contact from my list.”. Banish and Clear Verbs might be used in contexts like clearing notifications, “Clear all my notifications”, or managing tasks, “Recall the email I just sent”,
10. **Verbs of Assuming Position** (Class 51) where 2.77% of the IVA verbs belong. The relationship between users employing these verbs and the expected action is that users are commanding the IVA to navigate, guide, or move through digital spaces or tasks. Verbs of Inherently Directed Motion can be used in navigational tasks or browsing. For example, “Go to the next email” or “Exit the current application”. Leave Verbs in a digital context might be used as “Leave this group chat” or “Leave the current session”. Manner of Motion Verbs can be metaphorically used in digital tasks. For instance, “Slide to the next photo” or “Jump to the main menu”. Chase Verbs can be used in “Follow the latest news on this topic” or “Follow this artist on my music app”.
11. Remaining 30.45% consists of 38 verb classes.

5.1.2. Mapping VerbNet to WordNet

VerbNet maps each verb to the corresponding synset in WordNet. The algorithm used to find target language synsets used VerbNet version 3.2 and WordNet version 3.0 which are available in NLTK [82] library.

Table 16. Average number of target verbs generated in verb ontology.

Language	English Verbs	Avg. Num. of Target Verbs
es-ES	185	3.51
fr-FR	200	5.09
it-IT	187	4.24
pl-PL	89	2.63
pt-PT	188	3.76
sv-SE	116	2.46

As a result of mapping VerbNet to WordNet, I created verb ontology [7] that is represented by a dictionary where the key is an English verb, and values are verb translations in the target language as presented in the below examples. Each entry consists of between 1 and 10 possible translations that were extracted from the described mapping.

1. en-es: {*find*: [*encontrar, recuperar, conseguir*]}
2. en-fr: {*find*: [*retrouver, trouver, analyser*]}
3. en-it: {*find*: [*rinvenire, notare, osservare*]}
4. en-pl: {*find*: [*znajdź, poszukaj, odnajdź*]}
5. en-pt: {*find*: [*achar, encontrar, atingir*]}
6. en-sv: {*find*: [*upptäcka, hitta, finna*]

Table 16 displays the number of English verbs and the corresponding average number of target verbs extracted for each. For the Polish ontology, only 89 English verbs could be mapped. This constraint is due to the limited number of Polish synsets available in the WordNet 3.0 version provided by NLTK, compared to synsets for other languages.

5.2. Constrained Variant Generation Using Verb Ontology

Verb ontology guides MT to generate translation variants of the target verb. I use constrained decoding implemented in the Transformers library to create a translation consisting of a target verb (force word). The selected beam size is 5. The translations cannot consist of n-grams bigger than two more than once, and a single translation is generated for each constrained verb. All translations with more than two tokens bigger or smaller than the first-best are removed. If the input sentences contain slot annotations, we can expect constrained examples also to have slot annotations.

My translator (multiverb_iva_mt[8]) generate translations using following algorithm:

1. First translation is always a result of unconstrained translation (single-best),
2. For each target verb from verb ontology, the verb from the single-best translation with the target verb is replaced,

[7] `https://github.com/cartesinus/multiverb_iva_mt/tree/main/data/verb_translations`

[8] Code available at: `https://github.com/cartesinus/multiverb_iva_mt`

3. Finally, new variants generated by constrained beam search were added.

The final result is a list of translations that consist of at least one translation, but in the case when the input verb is found in verb ontology, typically, three variants are generated.

5.3. Comparative Analysis of Multi-Variant Translation Methods: Back-translation, Sampling, and GPT-3

In the domain of machine translation, generating multiple variants of a translation has been a focal point for enhancing the robustness and expressiveness of translated text. Two prevailing techniques for generating these variants are back-translation [83] and sampling [84], which have been widely adopted due to their proven effectiveness in generating diverse yet coherent translations. Back-translation involves translating a sentence to a target language and then back to the source language, while Sampling uses probabilistic models to choose different possible translations. These methods serve as strong baselines for evaluating innovative approaches to machine translation. In this section, we compare our machine translation library, which leverages a custom verb ontology for generating translation variants, against these well-established techniques. We aim to demonstrate the advantages of incorporating semantic understanding through verb ontology in generating multiple translation variants.

Another contemporary approach to generating multiple translation variants involves using large-scale language models like GPT-3, specifically its *text-davinci-003* version. By employing a sophisticated prompting mechanism, GPT-3 can generate many coherent and contextually relevant translation variants. Brown et al. [85] have demonstrated that GPT-3 performs at or near state-of-the-art levels across a wide range of natural language processing tasks, making it a compelling baseline for comparison. In this study, I utilize GPT-3 as an advanced control group, contrasting its performance with BackTranslation, Sampling, and our verb ontology-based method to provide a comprehensive evaluation landscape.

5.4. Multivariant Machine Translation

The proposed method to create verb ontology for IVAs can be used to generate multiple variants of translations. I tested my method on the NLU training set translation task, where English corpora were translated to Polish, and the NLU model was trained from them. In my experiments, I show that verb ontology can significantly improve IC while maintaining SF results intact compared to single-best translation.

My MT models extended with verb ontology presented in this work are the first open-source models adapted to the domain of IVA that can return multi-variant translation. I released verb ontology, verb ranking list, and source code of IC and SF training codes to the research community for all six languages described in this chapter. Additional data for the following language pairs presented in this chapter were published:

English-to-Spanish [86], English-to-French [87], English-to-Polish [88], English-to-Portuguese [89], and English-to-Swedish [90]. In the future, I plan to extend experiments to other languages.

5.4.1. Impact of Multi-verb Translation on NLU

To assess the efficacy of the proposed multivariant translation technique, a set of experiments was designed to compare it against established paraphrase generation algorithms. For contextual evaluation, two reference models are also introduced. These reference models are trained and tested solely on an untranslated subset of the dataset in question.

The experimental setup employs the English training corpus from the Leyzer dataset, comprising 17,290 utterances. Each method translates these utterances into Polish, generating multiple translation variants in the process. Subsequently, the translated output is partitioned into a new training and validation set, following an 80:20 ratio. The Inferential Consistency (IC) and Semantic Fidelity (SF) models, if applicable, are then trained on these sets. Evaluation is conducted using an independent Polish test set that has not undergone translation.

In the preceding section, the methodologies of Back-translation, Sampling, and ChatGPT prompting have been elaborated. For single-best translation, the method termed "Single-best IVA" is employed; this utilizes the M2M100 model adapted for the IVA domain and identifies the most accurate translation using a beam-search algorithm. Conversely, the multi-verb translation approach generates an array of translation alternatives. This is achieved through a constrained beam search, steered by the proposed verb ontology, to yield multiple semantically nuanced output variants.

Table 17, presents the impact of multiple variant generation on IC and SF model results. Reference models in English and Polish yield results above 95% for both IC and SF, affirming that high-quality translated training data can lead to strong performance metrics. As for the methods aimed at generating multiple translation variants, Back-translation and Sampling achieve lower performance, with intent accuracies of 77.07% and 79.00% respectively. These methods, although popular, demonstrate a noticeable gap in performance compared to the reference models. GPT-3 prompting, on the other hand, performs significantly better with an intent accuracy of 86.50%, though it still falls short of the reference models. Our proposed method, multi-verb translation, outperforms all other methods with an intent accuracy of 87.53%, closely approaching the high-performance benchmarks set by the reference models. These results underscore the effectiveness of generating translation variants based on verb ontology, especially when compared to Back-translation, Sampling, and GPT-3 prompting.

The proposed multi-verb improvement to the translation generation positively impacts IC model results. The accuracy of multi-verb translation is 3.8%, relatively better than single-best translation. However, it is 7.95% relatively lower than the baseline model. As presented in Table 18, each English sentence generates an average of 1.74 Polish transla-

Table 17. Comparison of NLU Intent Accuracy and Slot F_1-score between baseline, single-best translation, and multi-verb translation on Leyzer dataset.

Method	**Intent Accuracy [%]**	**Slot F_1-score [%]**
English reference (untranslated)	96.05	98.24
Polish reference (untranslated)	95.48	98.07
Back-translation	77.07	-
Sampling	79.00	-
Single-best IVA	83.73	88.21
GPT-3 prompting	84.58	-
Multi-verb IVA	87.53	88.15

Table 18. Average number of translations generated for a single English input per language.

Target Language	**Avg. Num. Translations**
es-ES	1.73
fr-FR	2.63
pl-PL	1.74
pt-PT	1.91
sv-SE	1.46

tions. This, in my opinion, is the main factor why multi-verb translation generates a better training dataset for the IC model. Leyzer test set evaluates multiple variants in which given intent can be uttered, including different levels of naturalness and verb patterns; therefore more variant training set improves results. Further improvements to IC could be made if more variants were created in verb ontology. Polish ontology (Table 16) consists of 89 verbs, which is the smallest of all presented languages.

Multi-verb translation does not improve the results of the SF model. My method does not generate different variants of slot values; therefore, during training, the SF model cannot generalize to new test cases. The difference in F_1-score between single-best and multi-variant is not statistically significant.

5.5. Conclusions

In this chapter, I explored the necessity for MT systems to produce multiple correct translation variants, ensuring comprehensive coverage of all linguistic patterns. The methodology introduced, which focuses on creating a verb ontology for IVAs, has demonstrated its potential to generate several translation variants. When applied to the task of translating NLU training sets, this method results in noticeable improvements in intent classification, particularly for English-to-Polish translations. This substantiates thesis T3, which posits that multi-variant translation enhances NLU accuracy.

A comparative evaluation was conducted to assess the efficacy of the proposed method against other prevalent techniques for generating multivariant translations, such as Back-translation, Sampling, Single-best IVA, and GPT-3 prompting. As illustrated in Table 17, the multi-verb IVA approach outperforms these methods in terms of both intent accuracy and slot F_1-score, thereby highlighting its superiority.

Furthermore, my MT models, enhanced with the verb ontology, stand out as pioneering open-source models tailored to the IVA domain capable of returning multi-variant translations. This capability is paramount, especially when considering human language's dynamic and multifaceted nature. The ability to recognize and understand multiple correct translations of a command or query can significantly enhance the efficacy and versatility of virtual assistants, especially in a multilingual setting.

As this chapter bridges to the subsequent sections of this dissertation, it is evident that integrating MT systems capable of generating multiple translation variants is not just an enhancement but a necessity. Such systems ensure that virtual assistants are equipped to understand and respond to users' commands in a manner that is accurate and contextually relevant, irrespective of linguistic nuances.

6. Industrial Implementations

This chapter presents the results of industrial implementations of MT for NLU used in the Bixby virtual assistant.

Bixby is a multi-lingual, multi-domain virtual assistant developed by Samsung Electronics that can perform actions based on voice commands. All the actions available in the system are grouped into the so-called capsules. A capsule serves as a voice interface for existing Android applications or web services, enabling them to understand and execute specific features, such as taking a photo in a camera app. These features, referred to as 'actions,' are inherently language-independent. However, to train Bixby to interpret natural language commands that trigger these actions, capsule developers must provide training utterances. These utterances are language-specific and serve as the bridge between the user's voice command and the action to be executed. Each action requires multiple training utterances for each language supported by the capsule.

Expanding into new markets, especially in the realm of virtual assistants like Bixby, necessitates support for multiple languages. For Bixby, this means localization of linguistic resources such as the training utterances, which is one of the most time-consuming parts of Bixby's development for new languages. Typically, utterances in English are created first and then localized into other languages.

In the project described in this chapter, I served as the project leader. I was responsible for setting project goals, including system quality metrics, for my team members and ensuring that they were met. I was also responsible for selecting the system architecture and feature management. Finally, I also implemented some of the system features myself.

6.1. Introduction

Bixby's NLU uses a Random Forest algorithm to make sense of user utterances based on annotated training examples. While Random Forest cannot fully grasp the nuances of a natural language, it does more than memorize features extracted from words. The key to its effectiveness lies in the quality of training examples. NLU developers carefully select examples that are broad enough to capture various user intents but specific enough to avoid redundancy. For instance, instead of using semantically close examples like "make a photo" and "I want a photo", a more generalized example like "selfie" would be more effective. This careful way of selecting training examples highlights the challenges MT must overcome to automate the work of NLU developers. To be effective in localization, MT needs to create translations that are as varied and detailed as the examples selected by humans, making the NLU system more robust.

To save the development time of localization from one language to another, we created a system for automatically translating Bixby capsules called Bixby Capsule Translator (BCT). Since the notion of correctness is unclear, we proposed an evaluation procedure

Figure 23. Example of Bixby capsule training sentence.

with a rich error category taxonomy that covers both syntactic and semantic problems that MT systems generate. The proposed evaluation procedure produces an easy-to-interpret numerical metric and a list of errors that can be used to improve system quality.

6.2. Machine Translation System for Cloud Bixby NLU

As presented in Figure 24, the system for Bixby capsule translation comprises three main components: MT model, slot alignment model, and rule-based post-editing module. The system takes an English capsule, an archive consisting of NLU, and extracts training examples from it. Each training example is translated into plain-text format (no slot or intent annotation) that is translated with the MT model. Slot annotations are transferred using an external alignment model. Finally, the post-editor fixes both translation and alignment errors based on rules created by language experts.

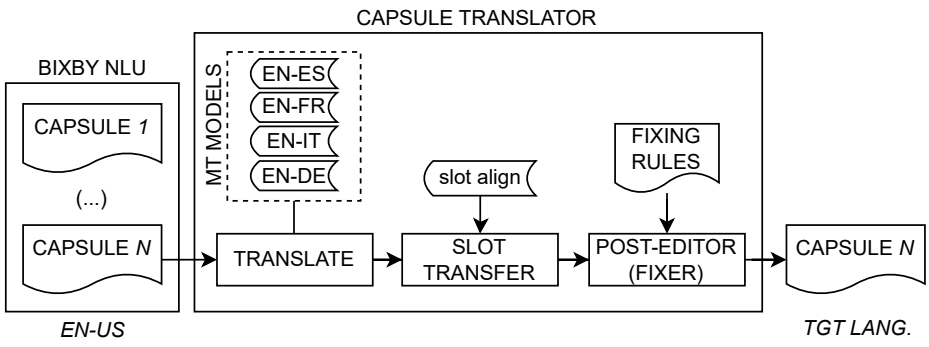

Figure 24. Architecture of the Bixby Capsule Translator, highlighting the integration of the MT model, slot alignment model, and rule-based post-editing module, with a workflow from English capsule extraction to expert post-editing.

We use GRU [91] sequence-to-sequence transducers [6], [10] with attention [9] as NMT model architecture. We extracted a joint byte-pair encoding (BPE) [92] vocabulary of size 60000 from the parallel corpus for each model. All models were implemented and trained in the Nematus toolkit [93].

Training corpora were divided into out-of-domain and in-domain segments for the purpose of training baseline and adapted models, respectively. For the out-of-domain corpus, we utilized all available data from the OPUS corpus [69], excluding translation pairs

identified as incorrect. To ascertain the quality of these translations, we first employed an automated filtering step that removed sentence pairs with a token count discrepancy exceeding five tokens. Subsequently, a manual evaluation was conducted by language experts who reviewed approximately 50% of the corpus. The focus of this manual check was to swiftly eliminate sentences that were flagrantly incorrect translations, rather than to identify minor translation errors. This two-step approach—automated filtering followed by expert review—ensured the high quality of our out-of-domain training corpus. For in-domain corpus, we use Bixby training sentences extracted from English capsules and manually translated into Spanish, Italian, French, and German. To avoid over-fitting to Bixby data, we have increased the size of the in-domain corpus using the method proposed by Axelrod [94]. Statistics of out-of-domain and in-domain corpora are presented in Table 19.

Table 19. Corpora size for Bixby Capsule Translator MT model training.

Transl. Direction	Out-of-Domain Sentences	In-domain Sentences
en-es	62M	671K
en-fr	58M	595K
en-it	35M	714K
en-de	36M	694K

The model training procedure is divided into two stages. We train the baseline model on out-of-domain corpora in the first, longer stage. The baseline model is optimized to maximize BLEU and GLEU [95] scores on the WMT shared news translation test set and does not include any information about the Bixby domain. In the second stage of the training, we take the last epoch of the baseline model and resume training on in-domain data, as proposed by Chu et al. [96]. As a result, our translation model is well adapted to the Bixby domain, but at the same time, it can translate unseen data when new capsules are added to the Bixby ecosystem.

To translate the whole capsule, our system parses capsule content to extract language-dependent training sentences and translate them with NMT models. The parts of the capsule that are language-independent are stored in the system memory, and once all training sentences are translated, the translated capsule is returned.

6.3. Domain Adaptation

To evaluate the model adaptation method, we used a Bixby parallel test set. This set was manually created and translated by language experts from English to the target language. In Bixby, capsules are trained to interpret voice commands using a set of training sentences. These same sentences are later used to evaluate the accuracy of the capsule's learning through a Random Forest classifier. To prevent over-fitting and artificially in-

flating the performance metrics of our NMT model, we deliberately avoided using these capsule training sentences to construct the NMT test set. Instead, we employed a different approach to generate test cases. Language experts were provided with the context of the application and its features, and they were tasked with creating test cases that would be contextually appropriate. This method aims to mitigate the risk of over-fitting by not directly borrowing from the training data, offering a more genuine evaluation of the NMT model's performance. As a result, we created a test set consisting of 5220 test cases for each language.

We evaluated the model before and after adaptation to show the impact of domain adaptation on the translation model. Each model was scored using the BLEU metric and GLEU metrics. We used Moses [97] implementation of BLEU scorer and NLTK [82] implementation of GLEU scorer.

Table 20. Results of MT model domain adaptation.

Transl. Direction	Baseline		Adapted	
	BLEU	**GLEU**	**BLEU**	**GLEU**
en-es	51.90	54.16	67.35	68.06
en-fr	37.86	40.83	59.09	60.20
en-it	39.69	43.68	57.92	59.96
en-de	35.65	40.27	52.45	54.68

As presented in Table 20, model adaptation has significantly improved baseline results.

6.4. Manual Evaluation of Translation Quality

We extracted 1000 translated training sentences from 10 translated capsules to evaluate system quality and gave them to language experts for evaluation. Correctness of each sentence was judged based on an evaluation procedure, which was adjusted to the specificity of the task. Each translated sentence is either accepted when the translation is perfect or rejected in case of any error. Additionally, we assign an error category and subcategory for each rejected sentence. Error taxonomy was created based on the frequency of occurrence of specific structures or vocabulary characteristics for a given domain. As a result, the following general error categories have been distinguished: grammatical, lexical, adequacy, notation, and tagging errors. Grammatical errors include preposition errors, article errors, verb form errors, part of clause errors, word order errors, interrogation errors, and agreement errors. Among lexical errors, we identify word sense disambiguation errors, part-of-speech errors, multilingual expression errors, name entities errors, and localization errors. Adequacy errors, in turn, cover omission errors, addition errors, lack of translation, use of polite forms (which is considered incorrect within our domain scope),

and changes in general meaning. Notation errors refer to spelling and format errors. Finally, we identify the "other" category, primarily for occasionally occurring errors.

Table 21. Results of human evaluation of capsule translation measured with Translation Sentence Acceptance Rate.

Transl. Direction	Sentence Acceptance Rate
en-es	88.20%
en-fr	84.50%
en-it	81.80%
en-de	75.40%
average	*82.48%*

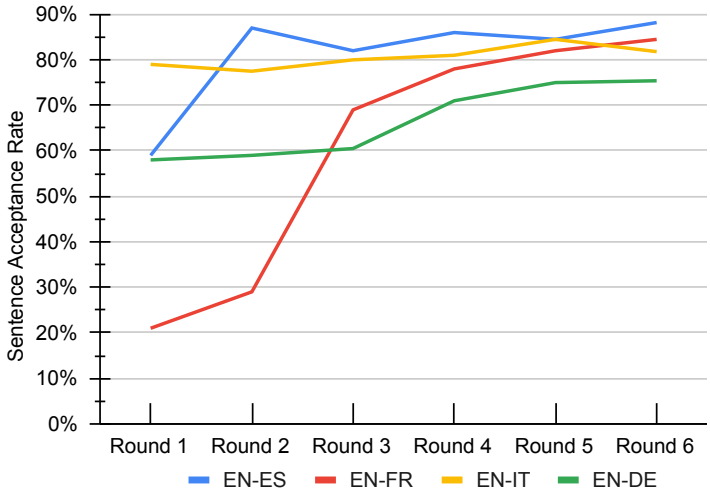

Figure 25. Quality improvements of Bixby Capsule Translator over six development rounds.

To evaluate system performance, we introduce an auxiliary metric, the Translation Sentence Acceptance Rate, which counts the number of accepted sentences divided by the number of all test cases. This metric is an easily interpretable numeric value that represents the number of perfectly translated sentences. The initial results of the system were not satisfactory; therefore, we designed an improvement process. Throughout six improvement rounds, as shown in Figure 25, we analyzed the results of the system and improved the MT alignment model and created post-editor rules. In Table 21, we present the system's final results for round 6, which were obtained using the procedure described earlier.

6.5. Automated Quality Checker

Automatic Post-Editing (APE) is the task of refining the output of an MT system using human-revised machine-translated content as training material. The primary objective

of APE is to rectify errors in machine-translated text, and therefore it is widely used in commercial settings. Commercial MT systems often employ APE to reach the desired quality standards, improve productivity [98], and minimize translation costs [99]. Typically, post-editing in commercial environments follows the MT model, utilizing a quality estimation model and a database containing both accurate and erroneous translations [100].

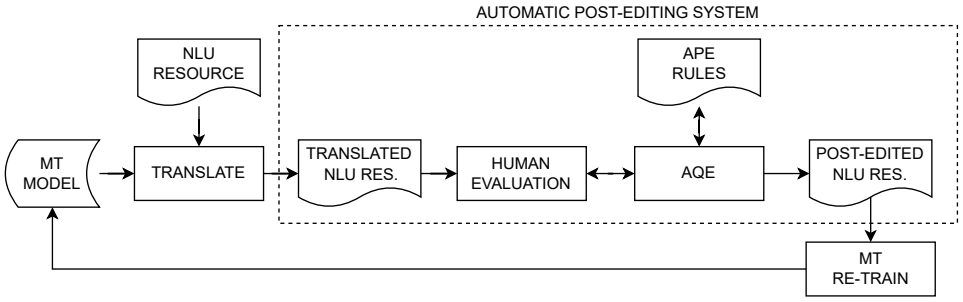

Figure 26. Schematic representation of the AQE system architecture, illustrating the integration of NLU resources, post-editing processes, and the subsequent refinement of the MT model.

We designed an APE system called Automated Quality Checker (AQC) to improve BCT quality further. AQC is a decision support system designed to help NLU developers and translators judge whether the translation returned by the MT model should be accepted or rejected (regarding translation correctness). As presented in Figure 26, AQE is implemented after NLU resources have been translated with the MT model. NLU developers evaluate translation quality using APE rules from prior improvement iterations, visual inspection, and translation analysis. As a result, post-edited NLU resources offer better quality than raw MT translations and aid in training a refined MT model.

To provide a deeper understanding, Figure 27 describes the architecture of AQC. The system comprises a machine-learning Automatic Quality Estimation (AQE) model and a rule-based system Error Pattern Tracker (EPT). Results of AQE and EPT are used to perform confirmation analysis (to show the user if a translation is correct based on text features). Additionally, the user is presented with a visual analysis tool to help assess translation quality.

In Figure 28, we present the user interface of AQC. The NLU developer uploads the NLU capsule to the system, and a list of all training resources is presented with estimated quality returned by the AQE. AQE returns three labels: *rejected* for sentences where the system was capable of detecting an error, *accepted* for sentences where no error was found and AQE model returns high classification probability, or *not_sure* for sentences without found error yet with low AQE classification probability. If the system can find a translation error in a sentence, it highlights part of a sentence using red color, and after hovering the mouse pointer over that part, a short description of the error is given (we call this "Confirmation Analysis")

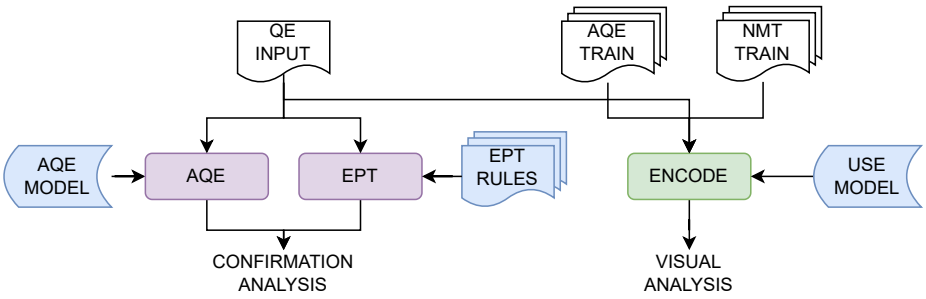

Figure 27. The system architecture of quality estimation system composed of AQE and EPT sub-systems.

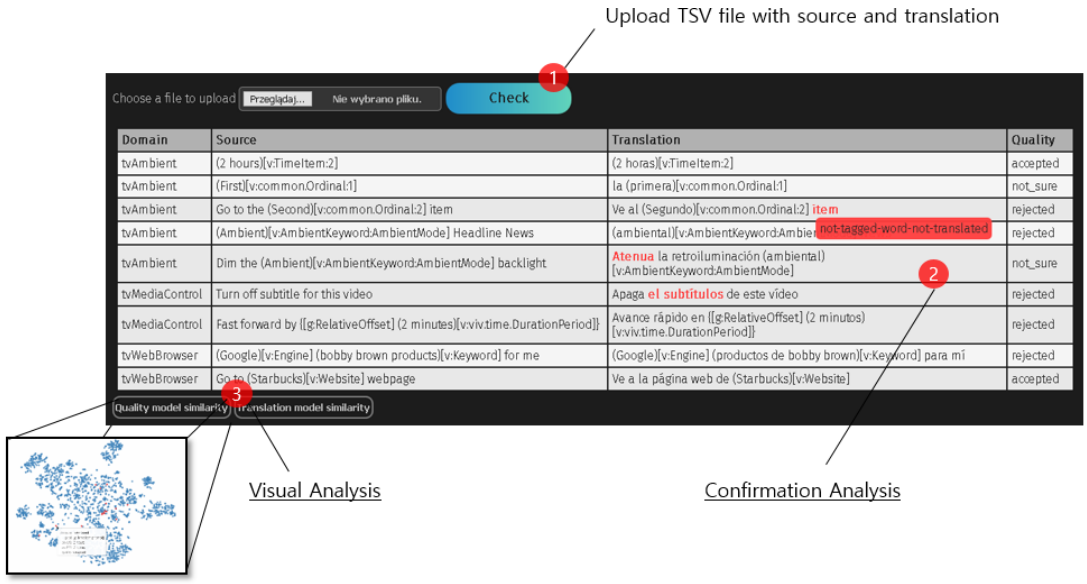

Figure 28. User interface of Automated Quality Checker system. Uploaded source-translation pairs can be analyzed in the system with visual analysis or text analysis (confirmation analysis).

6.6. Automatic Quality Estimation

The visual analysis tool was designed to help NLU developers support the credibility of model hypotheses. As presented in Figure 29, NLU developers can explore AQE model vector spaces in a two-dimensional graph created with t-SNE [101]. This tool analyzes if AQC input is similar to AQE model training data. Red dots represent AQC input, blue dots represent AQE model training, and each cluster represents a different domain. Input sentences should correspond to the AQE input domain, sentence structure, and expected quality (accepted or rejected). If not, it means that the model is trained on different data and, therefore, is not credible to assess translation quality.

The architecture of the AQE model, presented in Figure 30, consists of a lookup layer containing embeddings for target words and their source-aligned words. These embeddings are fed to a bidirectional LSTM to encode source and target tokens and segments

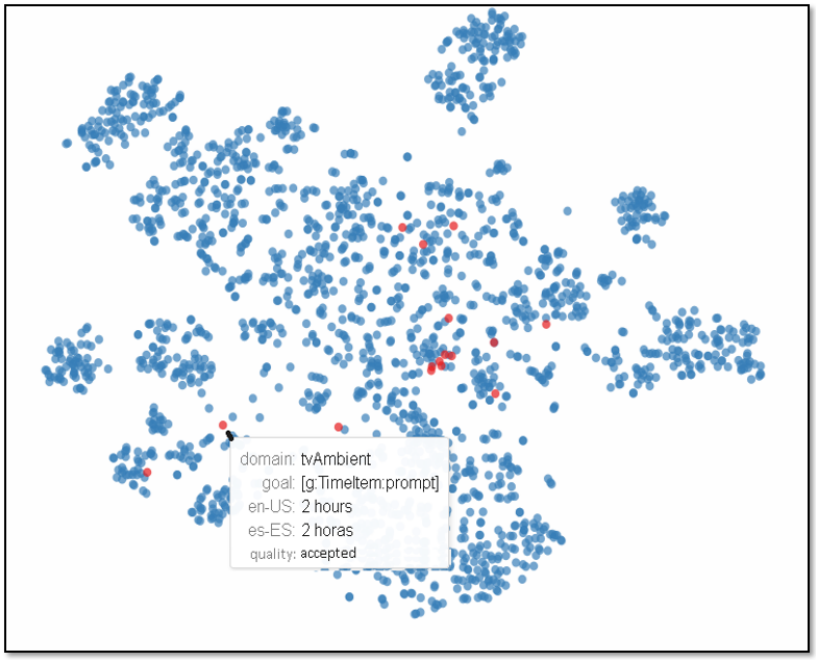

Figure 29. Visual analysis of source-translation pairs. Users can visually compare the position of input sentences (red dots) in the vector space of correct translations (blue dots).

that are concatenated and fed to a dense layer. The output contains a softmax layer that produces the final *accepted* or *rejected* decision. The model was trained for 40 epochs with an early stop parameter, batch size 128, and Adam as optimizer. We designed this model to predict sentence-level scores.

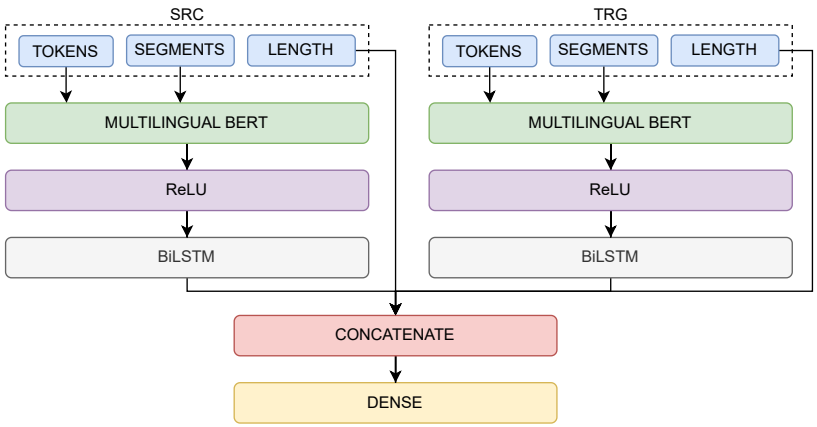

Figure 30. Architecture of AQE model. The model was trained on the source and translated sentences to predict if the translation was correct.

Estimation of the quality in DNN models tend to work best if a runtime distribution is close to the distribution of training corpora (catastrophic forgetting). To address this problem we have introduced the Semantic Textual Similarity (STS) metric that:

1. Inform the user how "far" runtime corpora distribution is from the model training distribution. The normalized score indicates whether the model will generalize to a given distribution.
2. Is used to change classification results. In cases where runtime examples are "far" from model training corpora, the model adjusts its confidence (increasing NOT_SURE predictions).

In Figure 31, we present a graphical interpretation of the STS measure. Assuming that the dataset consists of domains, in our context, Bixby-like language, News (e.g., from WMT), we can encode sentences with word embedding such as the BERT model and later visualize them using t-SNE (as in the example). In each domain similar sentences will gather into the same clusters because of their syntactic structure. We can then calculate the centroid of each cluster and measure the distance between centroids of these clusters, which is the STS measure. During the training of AQE, we know the distribution of in-domain and out-of-domain data. Therefore, when a user uploads input data we can measure STS between input and AQE training data, which then can be used to interpret confidence of AQE prediction.

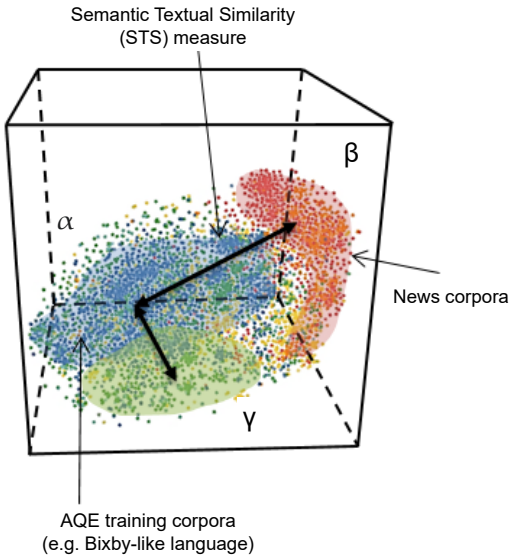

Figure 31. Graphical interpretation of the Semantic Textual Similarity (STS) measure. STS is the cosine distance from the centroid of the given data domain.

Results of the en-all AQE model trained on Samsung proprietary dataset and evaluated separately for all languages are presented in Table 22. In Figure 32, we present the receiver operating characteristic curve (ROC) that shows the trade-off between the model's True Positive Rate (Sensitivity) and False Positive Rate at various threshold settings.

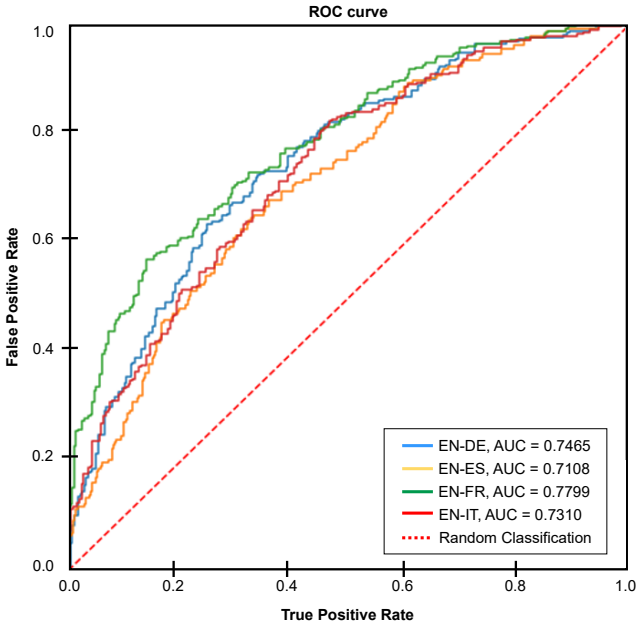

Figure 32. A receiver operating characteristic curve (ROC) plot of the correct-reject ratio (true negatives/ no) against correct-accept ratio (true positives / n1) for different thresholds. The ROC curve lies in the unit square, with random choice corresponding to the diagonal and perfect discrimination corresponding to the edges.

6.7. Error Pattern Tracker

Error Pattern Tracker (EPT) is a collection of processes, tools, and resources for evaluating and improving the MT quality of the BCT. EPT is built around the idea of an error pattern, which is a group of translation errors with a common description and standard detection method for which a common fix can be implemented.

The main purpose of EPT is to help to monitor and improve BCT translation quality. This can be done by identifying error patterns in translations and then tracking the number of occurrences of each pattern to measure translation quality and indicate which translation issues should be fixed first.

Some examples of error patterns:

1. Using the infinitive instead of the imperative form of a verb in the translation of a command
2. Translating the English verb “play” to Spanish “tocar” instead of “reproducir”, in the context of *SamsungMusic* application
3. Tagging the indefinite articles “un”, “une” or “de” together with the following noun

As can be seen from the above examples, error patterns may differ in generality (ex. 1 vs ex. 2), may require additional context besides just the source and the target utterances (ex. 2 requires the knowledge of the utterance domain), may refer to grammatical (ex. 1), lexical (ex. 2) or non-linguistical features (ex. 3 refers not to the utterance text, but application-specific annotation such as slot tagging).

Table 22. AQE model results for *beta* parameter threshold maximizing.

Beta	Direction	Accuracy	Precision	Recall	F_1-score	Treshold
1.0	en-es	0.567970	0.442971	0.888298	0.591150	0.186132
	en-fr	0.659218	0.599462	0.871094	0.710191	0.290161
	en-it	0.683426	0.661932	0.823322	0.733858	0.328626
	en-de	0.672253	0.623145	0.813953	0.705882	0.351728
0.5	en-es	0.698324	0.593103	0.457447	0.559896	0.408210
	en-fr	0.718808	0.788043	0.566406	0.730847	0.514182
	en-it	0.681564	0.666667	0.798587	0.689445	0.351847
	en-de	0.692737	0.702586	0.631783	0.687184	0.471980
0.1	en-es	0.662942	1.000000	0.042553	0.817814	0.786935
	en-fr	0.577281	1.000000	0.117188	0.930590	0.792456
	en-it	0.528864	0.969697	0.113074	0.902037	0.777117
	en-de	0.560521	0.896552	0.100775	0.831539	0.791546

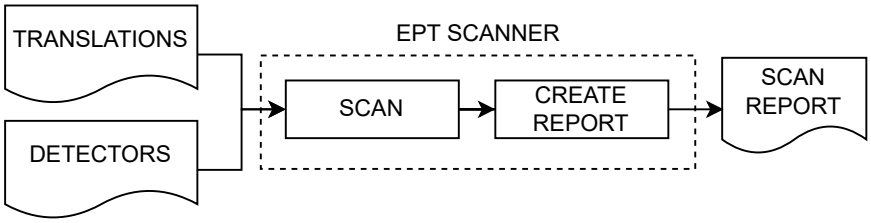

Figure 33. EPT scanner pipeline processing translations from TSV files, resulting in error reports for source-translation pairs.

Figure 33 shows the processing pipeline of each EPT scanner of the system. Scanner input is translations in TSV file that consists of source sentence (e.g., English), translated sentence and column with context (e.g. device state), and detector program. An output scanner report is generated with errors found in each source-translation pair.

Detectors are small programs (technically, boolean-valued Python functions decorated with ept.detector) that detect a specific error pattern that occurs in a given translation pair. Here is a particularly simple detector function for the pattern en-US_es-ES/car-translation:

```
@ept.detector('en', 'es', 'car-translation')
def noun_car_translation(domain, goal, source, translation) -> bool:
    """Detect word 'car' translated as auto (should be 'coche')"""
    return (re.search(r'\bcar(?:s)?\b', untag(source).lower()) \
        and re.search(r'\bauto(?:s)?\b', untag(translation).lower()))
```

String arguments domain and goal provide additional application-specific context for the detector (not needed in this example). Detectors are organized in Python modules, which makes it easier to select only a particular module or package (for example, all detectors for a specific language pair) every time detectors are used.

Table 23 shows the quality (measured by precision, recall and F_1-score) of the rule-based error detection system, separately for each group of errors. The evaluation is performed with the EPT Evaluation test set used as a reference.

Table 23. Results of EPT rule-based detectors on internal testset.

Detector Name	**Precision**	**Recall**
agreement error	80.00%	29.60%
article error	89.90%	44.70%
incompatible with dictionary	100.00%	39.10%
missing tag	100.00%	100.00%
not translated	89.10%	70.40%
polite form used	100.00%	80.00%
preposition error	100.00%	16.40%
spelling error	100.00%	94.10%
word sense disambiguation error	100.00%	41.30%
wrong part of speech	100.00%	44.70%
wrong verb form	77.80%	74.20%
overall (rejection)	93.50%	49.50%

6.8. Conclusions

In this chapter, I have outlined the Bixby Capsule Translator, an auxiliary system designed to assist in translating NLU resources. This tool complements the Bixby IVA ecosystem developed by Samsung Electronics. The system comprises two main elements: a domain-adapted MT and a set of post-editing rules created over the course of six development rounds. It achieves an 82.48% Sentence Acceptance Rate, a metric that measures the average accuracy of individual translated sentences. In simpler terms, if we have 10 sentences and 5 are accepted while 5 are not, the Sentence Acceptance Rate would be 50%.

The Bixby Capsule Translator is a support tool for NLU developers, becoming particularly useful once an English NLU capsule has been developed. Developers can then translate this capsule into a new language and make specific adjustments as needed. While the system has achieved high-performance rates—such as an 88.20% accuracy for the English-to-Spanish model—it is not fully automated and still requires human oversight. To assist with this, the system includes an automatic quality estimation module. However, this additional model has its limitations, achieving no more than a 93.05% F_1-score, leaving the final decision to the user.

7. Academic Achievements

Below is a list of my academic achievements.

7.1. Articles

7.1.1. International conferences

Publications at international conferences, including 1 article at CORE A conference, 1 article at CORE A conference and 3 articles at CORE B conferences:

1. M. Kubis, M. Sowański, P. Skórzewski and T. Ziętkiewicz, "Back Transcription as a Method for Evaluating Robustness of Natural Language Understanding Models to Speech Recognition Errors". to be presented at *EMNLP 2023*, December 2023. (CORE A*)
2. M. Sowański and A. Janicki, "Leyzer: A dataset for multilingual virtual assistants", in *Proc. Conference on Text, Speech, and Dialogue (TSD2020)*, P. Sojka, I. Kopeček, K. Pala, and A. Horák, Eds., Brno, Czechia: Springer International Publishing, 2020, pp. 477–486, ISBN: 978-3-030-58323-1.
3. M. Sowański and A. Janicki, "Slot lost in translation? not anymore: A machine translation model for virtual assistants with type-independent slot transfer", in *Proc. 30th International Conference on Systems, Signals and Image Processing, (IWSSIP 2023)*, Ohrid, North Macedonia: IEEE, 2023, pp. 1–4.
DOI: 10.1109/IWSSIP58668.2023.10180229.
4. M. Sowański and A. Janicki, "Optimizing machine translation for virtual assistants: Multi-variant generation with VerbNet and conditional beam search", in *Proc. 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, M. Ganzha, L. Maciaszek, M. Paprzycki, D. Ślęzak (eds). Warsaw, Poland: ACSIS, Vol. 35, pp. 1149–1154 (2023). DOI: 10.15439/2023B8601 (CORE B)
5. J. Hosciłowicz, M. Sowański, P. Czubowski, and A. Janicki, "Can we use probing to better understand fine-tuning and knowledge distillation of the BERT NLU?", in *International Conference on Agents and Artificial Intelligence*, Lisbon, Portugal, 2023. (CORE B)
6. K. Gabor-Siatkowska, M. Sowański, M. Pudo, et al., "Therapeutic spoken dialogue system in clinical settings: Initial experiments", in *Proc. 30th International Conference on Systems, Signals and Image Processing (IWSSIP 2023)*, Ohrid, North Macedonia: IEEE, 2023, pp. 1–4. DOI: 10.1109/IWSSIP58668.2023.
10180265.
7. M. Kubis, P. Skórzewski, M. Sowański, and T. Ziętkiewicz, "Center for artificial intelligence challenge on conversational AI correctness", in *Proc. 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, M. Ganzha, L. Maciaszek, M.

Paprzycki, D. Ślęzak (eds). Warsaw, Poland: ACSIS, Vol. 35, pp. 1319–1324 (2023) 10.15439/2023B6058 (CORE B)

8. M. Kozłowski, K. Gabor-Siatkowska, I. Stefaniak, M. Sowański, and A. Janicki, "Enhanced emotion and sentiment recognition for empathetic dialogue system using big data and deep learning methods", in *Proc. International Conference on Computational Science (ICCS 2023)*, Prague, Czechia, 2023. (CORE A)

7.1.2. Domestic conference and chapters in monographs

8. M. Sowański, M. Pudo, and A. Janicki, "Wykrywanie nieprzetłumaczalnych fraz w tekstach naukowych z dziedziny chemii, biologii i fizyki", in *Kopernikańskie Seminarium Doktoranckie. Na pograniczu chemii, biologii i fizyki – rozwój nauk*, vol. 4, Prezentacja: XV Kopernikańskie Seminarium Doktoranckie, 20-22.06.2022, Toruń, Poland, Toruń, Poland: Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika, 2022.
9. M. Pudo, M. Sowański, and A. Janicki, "Metody uczenia częściowo nadzorowanego w automatycznym rozpoznawaniu mowy", in *Kopernikańskie Seminarium Doktoranckie. Na pograniczu chemii, biologii i fizyki – rozwój nauk*, vol. 4, Prezentacja: XV Kopernikańskie Seminarium Doktoranckie, 20-22.06.2022, Toruń, Poland, Toruń, Poland: Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika, 2022.

7.2. Patents

7.2.1. Patents received

1. E. Wesołowska, M. Sowański, R. Paprocki, and R. Frączek, "Electronic device and control method thereof", pat. WO2022075591A1, Awarded 14 Apr 2022.
   The patent describes the NLU system that is part of the IVA system.

7.3. Speeches and Presentations

1. WMI Talks 2022, 8.03.2022r., "Współczesne modele rozumienia języka – jak powstały i dokąd zmierzają?", Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań
2. Shape the Future with Samsung Research Poland, 2.12.2021r., "Shape the Future with Samsung Research Poland", online conference

7.4. Other activities

1. Co-organized the "CAICCAIC: Center for Artificial Intelligence Challenge on Conversational AI Correctness" shared task at the FedCSIS 2023 conference, scheduled for September 20, 2023. The objective of the task was to develop an NLU system capable of robustly mitigating errors generated by Automatic Speech Recognition systems, thereby improving intent classification and slot-filling accuracy. This initiative is

of significant industrial relevance, as it addresses challenges commonly faced in commercial settings.

8. Summary

In this dissertation, I demonstrated that machine translation (MT) is an effective tool for translating training datasets for dialogue agents. Particularly, it can be employed as a strategy for developing natural language understanding (NLU) within intelligent virtual assistants (IVA) for new languages. I formulated three theses:

1. **[T1]** Machine translation, when adapted to the language of intelligent virtual assistants, serves as an efficient tool for localizing natural language understanding models,
2. **[T2]** To translate natural language understanding training resources, which comprise semantic annotations, machine translation must preserve and appropriately translate named entity locations,
3. **[T3]** Generating multiple variants when translating training data for intelligent virtual assistants improves the NLU accuracy.

Although dialogue agents have been widely studied since the 2000s, in early 2019, when I began this research, there were only five corpora available for NLU, collectively containing fewer than 150.000 utterances. These corpora were limited in scope, covering a narrow range of domains and intents and most of them were available only for English. While commercial NLU systems developed by companies such as Google, Apple, and Samsung have rapidly grown, offering their users thousands of features (intents) open-source resources used in the research focused on only a few domains and several intents only. Additionally, most of the resources were available only in English. To foster research and to be able to work on MT for NLU, I began my research by creating the Leyzer, a multilingual dataset designed for IVAs. Leyzer, presented in Chapter 3, consists of 20 domains and offers a wide intent selection. Its unique feature is that all utterances have been classified by naturalness level and verb pattern.

Once the Leyzer dataset was established to enhance NLU testing, I shifted my focus to MT for NLU, fulfilling the objectives of my first thesis, **T1**. Specifically, the MT model was adapted by fine-tuning the M2M100 model on a custom IVA dataset. This dataset comprises parallel NLU corpora and out-of-domain data, which were selectively filtered to align with the IVA domain, thereby preserving the model's ability to generalize. The fine-tuning process was conducted lightly over the 10 epochs, yielding optimal results. In terms of performance, the fine-tuned model showed an improvement of +19.62 ± 1.6 BLEU points for the English-to-Polish model and +10.45 ± 1.92 BLEU points for the English-to-Spanish model, respectively. A BLEURT analysis on the WMT dataset substantiated that the adapted model, while marginally less effective in general contexts, displayed superior performance in the IVA domain.

As my research progressed, I realized that the translation and transfer of entities in MT were crucial for the effective development of dialogue agents. In Chapter 4, I delved into

the concept of slot transfer and introduced a parallel dataset with slot annotations for this task. The findings from this chapter provided valuable insights into the translation and transfer of entities, which are essential components of dialogue agents. I have trained the NLU model from translated data. The efficacy of the created model is evident with a +17.21 BLEU improvement in the IVA domain and slot F1 of 65.47% for sentences with multiple slot types and 87.54% for sentences with single slot types. Therefore, the presented results defend thesis **T2**.

At this stage of the project, I realized I had at my disposal an adapted MT model and a suitable dataset, which provided me with the tools to analyze NLU performance more effectively than with existing resources. This allowed me to observe that while MT models produce high-quality translations, NLU training must encompass as many grammatical variants as possible. Typically, when developing NLU training resources, this is achieved by collecting sentences that convey the same meaning but exhibit different grammatical forms. However, MT models often exhibit a bias toward producing the same translation for different inputs. Guided by these observations, I investigated the potential of multiverb and multivariant MT to enhance NLU in IVAs. I constructed a verb ontology for IVA NLU and mapped IVA verbs to the Levin classes and VerbNet. The influence of multiverb translation on NLU was a central focus of this chapter, and the outcomes were encouraging. In my experiments, I showed that multi-verb translation improves intent classification accuracy by 3.8% relative compared to single-best translation. This defends thesis **T3**.

Throughout my research journey, I encountered numerous challenges and obstacles. However, the findings from each chapter provided valuable insights and contributed to the development of dialogue agents in new languages. The creation of the Leyzer dataset, the exploration of slot transfer, the development of multiverb and multivariant MT, and the focus on quality estimation and translation sieving were all crucial components of my research. The industrial applications of MT further highlighted the potential of my research for real-world applications.

8.1. Contribution to the development of the scientific field

During my Ph.D., my research interests focused on MT and NLU for emerging languages. This dissertation outlines several key contributions to these fields. First and foremost, I developed the Leyzer dataset, designed to train and evaluate multilingual MT and NLU models. Since its inception, Leyzer has been employed in various research endeavors, most notably in the Challenge on Conversational AI Correctness [102]. As one of four co-authors, I also had the distinct privilege of participating in the organization of this challenge, which aimed to develop robust language comprehension models. Hosted within the framework of the “18th Conference on Computer Science and Intelligence Systems FedCSIS 2023”, this event facilitated invaluable knowledge exchange with field ex-

perts. The challenge and the dataset have jointly contributed to improving the robustness of NLU models, addressing a critical need in the field.

My second major contribution lies in creating MT models that are specifically tailored for the IVA domain. These adapted models significantly outperform existing state-of-the-art models, filling a notable gap as there are no publicly available MT models adapted for the IVA domain. These models support translations between English and ten different languages, thereby extending the reach and applicability of IVAs. These models support translations from English to ten different languages, including Polish, Spanish, German, French, Portuguese, Swedish, Chinese, Japanese, Turkish, and Hindi.

My third contribution is a Python library that incorporates these adapted MT models and a verb ontology from VerbNet to enable multivariant translations. All contributions are released under an MIT license, facilitating their adoption in both academic and commercial settings.

In addition to these contributions, I have also engaged in research that, while not covered in this dissertation, holds relevance to the field. This includes investigations into the linguistic capabilities of NLU models [103], employing diagnostic classifiers to probe the transformer architecture commonly used in NLU tasks. Another significant milestone was co-authoring the article "Back Transcription as a Method for Evaluating Robustness of Natural Language Understanding Models to Speech Recognition Errors." This paper, accepted at the prestigious “The 2023 Conference on Empirical Methods in Natural Language Processing”, introduces a novel method for assessing NLU models in the context of speech recognition errors, a topic of increasing importance as dialogue systems become more robust.

8.2. Contribution to the industry

Industrial Ph.D., by definition, should emphasize industrial implications. In this work, I described a successful implementation of the MT system for Cloud Bixby NLU. This is a culmination of my work, as I applied my research findings to real-world applications. In this dissertation, I have presented how NLU systems used in various dialogue systems can be localized to new languages and markets. I shared methods, system architecture, domain adaptation techniques, and tools that both researchers and engineers can use.

One of the main goals of my research from the beginning was to create tools that could allow faster development of non-English-only NLU systems. To reach this goal, I researched different methods within MT and concluded that MT can be successfully used for that purpose. In my research, I tried to choose the most straightforward solutions and present them in this dissertation in a new context. The conclusions I have presented in this dissertation allow me to state that MT trained from specialized NLU corpus and used to translate training resources of dialogue agents is a good strategy as long as it is applied to work in a pipeline where NLU developers use it as an extension to their work and not

only as a sole solution. Although still lacking, I believe that the quality of MT models will, in the next few years, surpass the level of human annotators and developers of localized NLU.

My research also has an economic impact on the industry as the tools presented in this dissertation can save costs, increase revenue, or improve efficiency. Although I have not measured the exact impact of MT on the development time and cost of the NLU system because it is beyond the scope of this project, I was able to observe how NLU developers used MT in a commercial setting. My observations allow me to say that MT has a strong positive impact on the development time of NLU systems. The introduction of MT to the NLU development pipeline comes, however, with a cost for developers. Many NLU developers I have worked with claimed that MT tends to produce frequent (but small) errors that are tedious to fix. These errors usually are caused by model overfitting or underfitting, as well as from specific patterns in the training data. For instance, the model might choose an incorrect word translation over the correct one because it appears more frequently in the dataset. Thankfully, these errors can often be easily corrected with post-editing modules. Addressing these issues could be a valuable area for future research.

Finally, I would like to emphasize again that most of the models, methods, and tools described in this dissertation are freely available under the MIT license, which allows engineers to use them freely in commercial projects.

References

[1] Voicebot.ai. "Smartphone voice assistant use stalls out, but consumers want more voice features in mobile apps: New report". Accessed: 2023-08-30. (2022), [Online]. Available: https://voicebot.ai/2022/02/09/smartphone-voice-assistant-use-stalls-out-but-consumers-want-more-voice-features-in-mobile-apps-new-report.

[2] Voicebot.ai. "Voice assistant use on smartphones rise, siri maintains top spot for total users in the u.s." Accessed: 2023-08-30. (2020), [Online]. Available: https://voicebot.ai/2020/11/05/voice-assistant-use-on-smartphones-rise-siri-maintains-top-spot-for-total-users-in-the-u-s/.

[3] P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. Choudhury, "The state and fate of linguistic diversity and inclusion in the NLP world", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 6282–6293. DOI: 10.18653/v1/2020.acl-main.560. [Online]. Available: https://aclanthology.org/2020.acl-main.560.

[4] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, "A survey of joint intent detection and slot-filling models in natural language understanding", *arXiv preprint arXiv:2101.08091*, 2021.

[5] Y. Tada, Y. Hagiwara, H. Tanaka, and T. Taniguchi, "Robust understanding of robot-directed speech commands using sequence to sequence with noise injection", *Frontiers in Robotics and AI*, vol. 6, p. 144, 2020.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks", *Advances in neural information processing systems*, vol. 27, 2014.

[7] C. Sin-wai, *Routledge encyclopedia of translation technology*. Routledge, 2014.

[8] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer, *et al.*, "The mathematics of statistical machine translation: Parameter estimation", 1993.

[9] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate", in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[10] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation", in *EMNLP*, 2014.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation", in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[12] T. Sellam, D. Das, and A. Parikh, "Bleurt: Learning robust metrics for text generation", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7881–7892.

[13] F. García, E. Segarra, C. Millán, E. S. Arnal, and L. F. Hurtado, "A train-on-target strategy for multilingual spoken language understanding", in *IberSPEECH Conference*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:34935891.

[14] M. Cettolo, A. Corazza, and R. De Mori, "Language portability of a speech understanding system", *Computer Speech & Language*, vol. 12, no. 1, pp. 1–21, 1998.

[15] J. Glass, G. Flammia, D. Goodine, *et al.*, "Multilingual spoken-language understanding in the mit voyager system", *Speech communication*, vol. 17, no. 1-2, pp. 1–18, 1995.

[16] A. Waibel, "Interactive translation of conversational speech", *Computer*, vol. 29, no. 7, pp. 41–48, 1996.

[17] B. Jabaian, L. Besacier, and F. Lefèvre, "Investigating multiple approaches for slu portability to a new language", in *Interspeech*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:15196393.

[18] C. Servan, N. Camelin, C. Raymond, F. Béchet, and R. D. Mori, "On the use of machine translation for spoken language understanding portability", *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5330–5333, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:458721.

[19] E. A. Stepanov, I. Kashkarev, A. O. Bayer, G. Riccardi, and A. Ghosh, "Language style and domain adaptation for cross-language slu porting", *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 144–149, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:37478891.

[20] A. Abujabal, C. D. Bovi, S.-R. Ryu, T. Gojayev, F. Triefenbach, and Y. Versley, "Continuous model improvement for language understanding with machine translation", in *North American Chapter of the Association for Computational Linguistics*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235097671.

[21] C. L. Hench, C. S. Peris, J. G. M. FitzGerald, and K. Rottmann, "Massively multilingual natural language understanding 2022 (mmnlu-22) workshop and competition", *ArXiv*, vol. abs/2212.06346, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254591769.

[22] B. Zheng, Z. Li, F. Wei, Q. Chen, L. Qin, and W. Che, "Hit-scir at mmnlu-22: Consistency regularization for multilingual spoken language understanding", *ArXiv*, vol. abs/2301.02010, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:255440691.

[23] M. Nicosia and F. Piccinno, "Evaluating byte and wordpiece level models for massively multilingual semantic parsing", *ArXiv*, vol. abs/2212.07223, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254636123.

[24] M. De Bruyn, E. Lotfi, J. Buhmann, and W. Daelemans, "Machine translation for multilingual intent detection and slots filling", in *Proceedings of the Massively Mul-*

tilingual Natural Language Understanding Workshop (MMNLU-22), 2022, pp. 69–82.

[25] S. Rentschler, M. Riedl, C. Stab, and M. Rückert, "Data augmentation for intent classification of german conversational agents in the finance domain", in *Conference on Natural Language Processing*, 2022. [Online]. Available: `https://api.semanticscholar.org/CorpusID:252091139`.

[26] J. Quan and D. Xiong, "Effective data augmentation approaches to end-to-end task-oriented dialogue", *2019 International Conference on Asian Language Processing (IALP)*, pp. 47–52, 2019. [Online]. Available: `https://api.semanticscholar.org/CorpusID:208636952`.

[27] W. Zhu, H. Liu, Q. Dong, *et al.*, "Multilingual machine translation with large language models: Empirical results and analysis", *arXiv preprint arXiv:2304.04675*, 2023.

[28] X. Wei, H. Wei, H. Lin, *et al.*, "Polylm: An open source polyglot large language model", *arXiv preprint arXiv:2307.06018*, 2023.

[29] A. Rosenbaum, S. Soltan, W. Hamza, Y. Versley, and M. Boese, "Linguist: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging", in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 218–241.

[30] S. Soltan, S. Ananthakrishnan, J. G. M. FitzGerald, *et al.*, "Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model", *arXiv*, 2022. [Online]. Available: `https://www.amazon.science/publications/alexatm-20b-few-shot-learning-using-a-large-scale-multilingual-seq2seq-model`.

[31] V. D. Lai, N. T. Ngo, A. P. B. Veyseh, *et al.*, "Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning", *arXiv preprint arXiv:2304.05613*, 2023.

[32] H. Huang, T. Tang, D. Zhang, *et al.*, "Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting", *arXiv preprint arXiv:2305.07004*, 2023.

[33] T. L. Scao, A. Fan, C. Akiki, *et al.*, "Bloom: A 176b-parameter open-access multilingual language model", *arXiv preprint arXiv:2211.05100*, 2022.

[34] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?", in *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 2021, pp. 610–623.

[35] M. Sowański and A. Janicki, "Leyzer: A dataset for multilingual virtual assistants", in *Proc. Conference on Text, Speech, and Dialogue (TSD2020)*, P. Sojka, I. Kopeček, K. Pala, and A. Horák, Eds., Brno, Czechia: Springer International Publishing, 2020, pp. 477–486, ISBN: 978-3-030-58323-1.

[36] M. Sowański and A. Janicki, "Slot lost in translation? not anymore: A machine translation model for virtual assistants with type-independent slot transfer", in *Proc. 30th International Conference on Systems, Signals and Image Processing, (IWSSIP*

2023), Ohrid, North Macedonia: IEEE, 2023, pp. 1–4. DOI: 10.1109/IWSSIP55020.2023.00000.

[37] M. Sowański and A. Janicki, "Optimizing machine translation for virtual assistants: Multi-variant generation with verbnet and conditional beam search", in *Proceedings of the 18th Conference on Computer Science and Intelligence Systems*, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., ser. Annals of Computer Science and Information Systems, vol. 35, IEEE, 2023, 1149–1154. DOI: 10.15439/2023B8601. [Online]. Available: http://dx.doi.org/10.15439/2023B8601.

[38] P. Price, "Evaluation of spoken language systems: The ATIS domain", in *Proc. of the Speech and Natural Language Workshop, Hidden Valley, PA*, 1990.

[39] I. Casanueva, I. Vulić, G. Spithourakis, and P. Budzianowski, "Nlu++: A multi-label, slot-rich, generalisable dataset for natural language understanding in task-oriented dialogue", in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 1998–2013.

[40] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "SLURP: A Spoken Language Understanding Resource Package", in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[41] S. Larson, A. Mahendran, J. Peper, *et al.*, "An evaluation dataset for intent classification and out-of-scope prediction", in *Proc. 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*, 2019.

[42] X. Liu, A. Eshghi, P. Swietojanski, and V. Rieser, "Benchmarking natural language understanding services for building conversational agents", *arXiv preprint arXiv:1903.05566*, 2019.

[43] S. Gupta, R. Shah, M. Mohit, A. Kumar, and M. Lewis, "Semantic parsing for task oriented dialog using hierarchical representations", in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2787–2792.

[44] A. Coucke, A. Saade, A. Ball, *et al.*, "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces", *arXiv preprint arXiv:1805.10190*, 2018.

[45] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual transfer learning for multilingual task oriented dialog", in *Proc. of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019), Minneapolis, MN*, 2019.

[46] H. Li, A. Arora, S. Chen, A. Gupta, S. Gupta, and Y. Mehdad, "Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark", in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 2950–2962.

[47] R. Goel, W. Ammar, A. Gupta, *et al.*, "Presto: A multilingual dataset for parsing realistic task-oriented dialogs", *arXiv preprint arXiv:2303.08954*, 2023.

[48] W. Xu, B. Haider, and S. Mansour, "End-to-end slot alignment and recognition for cross-lingual nlu", in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 5052–5063.

[49] J. FitzGerald, C. Hench, C. Peris, *et al.*, "MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages", *arXiv preprint arXiv:2204.08582*, 2022.

[50] P. Budzianowski, T.-H. Wen, B.-H. Tseng, *et al.*, "MultiWOZ – a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling", in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 5016–5026. DOI: 10.18653/v1/D18-1547. [Online]. Available: https://www.aclweb.org/anthology/D18-1547.

[51] G. Campagna, R. Ramesh, S. Xu, M. Fischer, and M. S. Lam, "Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant", in *Proc. of the 26th International Conference on World Wide Web*, 2017, pp. 341–350.

[52] T. S. Wu, M. T. Ribeiro, J. Heer, and D. S. Weld, "Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models", in *Annual Meeting of the Association for Computational Linguistics*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235266322.

[53] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domains", in *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, 2015, pp. 76–79.

[54] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation", in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2013, pp. 678–683.

[55] R. Wang, A. Finch, M. Utiyama, and E. Sumita, "Sentence embedding for neural machine translation domain adaptation", in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 560–566.

[56] Y. Lü, J. Huang, and Q. Liu, "Improving statistical machine translation performance by training data selection and optimization", in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 343–350.

[57] M. Freitag and Y. Al-Onaizan, "Fast domain adaptation for neural machine translation", *arXiv preprint arXiv:1612.06897*, 2016.

[58] I. J. Goodfellow, M. Mirza, X. Da, A. C. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgeting in gradient-based neural networks", *CoRR*, vol. abs/1312.6211, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:12730344.

[59] B. Thompson, J. Gwinnup, H. Khayrallah, K. Duh, and P. Koehn, "Overcoming catastrophic forgetting during domain adaptation of neural machine translation", 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)*, 2019, pp. 2062–2068.

[60] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, *et al.*, "Overcoming catastrophic forgetting in neural networks", *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[61] A. Bapna and O. Firat, "Simple, scalable adaptation for neural machine translation", 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)*, Association for Computational Linguistics, 2019.

[62] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network", in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: `http://arxiv.org/abs/1503.02531`.

[63] S. Gu, Y. Feng, and W. Xie, "Pruning-then-expanding model for domain adaptation of neural machine translation", in *North American Chapter of the Association for Computational Linguistics*, 2021. [Online]. Available: `https://api.semanticscholar.org/CorpusID:232352519`.

[64] E. J. Hu, P. Wallis, Z. Allen-Zhu, *et al.*, "Lora: Low-rank adaptation of large language models", in *International Conference on Learning Representations*, 2021.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", in *Proc. of the 6th International Conference on Learning Representations (ICRL 2015), San Diego, CA*, 2015.

[66] M. Clinciu, A. Eshghi, and H. Hastie, "A study of automatic metrics for the evaluation of natural language explanations", in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 2376–2387.

[67] T. Kocmi, H. Matsushita, and C. Federmann, "Ms-comet: More and better human judgements improve metric performance", in *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 2022, pp. 541–548.

[68] S. Giorgi, S. Havaldar, F. Ahmed, *et al.*, "Human-centered metrics for dialog system evaluation", *arXiv preprint arXiv:2305.14757*, 2023.

[69] J. Tiedemann, "Parallel data, tools and interfaces in opus.", in *Lrec*, vol. 2012, 2012, pp. 2214–2218.

[70] Y. Yang, D. Cer, A. Ahmad, *et al.*, "Multilingual universal sentence encoder for semantic retrieval", *arXiv preprint arXiv:1907.04307*, 2019.

[71] A. Fan, S. Bhosale, H. Schwenk, *et al.*, "Beyond english-centric multilingual machine translation", *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 4839–4886, 2021.

[72] T. Kocmi, R. Bawden, O. Bojar, *et al.*, "Findings of the 2022 conference on machine translation (wmt22)", in *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 2022, pp. 1–45.

[73] K. Kipper, H. T. Dang, M. Palmer, *et al.*, "Class-based construction of a verb lexicon", *AAAI/IAAI*, vol. 691, p. 696, 2000.

[74] K. K. Schuler, *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.

[75] G. A. Miller, "Wordnet: A lexical database for english", *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[76] O. Majewska and A. Korhonen, "Verb classification across languages", *Annual Review of Linguistics*, vol. 9, 2023.

[77] A. Huminski, F. Liausvia, and A. Goel, "Semantic roles in verbnet and framenet: Statistical analysis and evaluation", in *Computational Linguistics and Intelligent Text Processing: 20th International Conference, CICLing 2019, La Rochelle, France, April 7–13, 2019, Revised Selected Papers, Part II*, Springer, 2023, pp. 135–147.

[78] B. Levin, *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993.

[79] L. Sun, A. Korhonen, and Y. Krymolowski, "Verb class discovery from rich syntactic data", *Lecture Notes in Computer Science*, vol. 4919, p. 16, 2008.

[80] D. R. Traum, "Speech acts for dialogue agents", in *Foundations of rational agency*, Springer, 1999, pp. 169–201.

[81] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual transfer learning for multilingual task oriented dialog", in *Proceedings of NAACL-HLT*, 2019, pp. 3795–3805.

[82] E. Loper and S. Bird, "Nltk: The natural language toolkit", in *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, 2002, pp. 63–70.

[83] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data", in *54th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics (ACL), 2016, pp. 86–96.

[84] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need", *Advances in neural information processing systems*, vol. 30, 2017.

[85] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners", *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[86] M. Sowański. "Iva_mt_wslot-m2m100_418m-en-es". Hugging Face Model Hub. (2023), [Online]. Available: `https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-es`.

[87] M. Sowański. "Iva_mt_wslot-m2m100_418m-en-fr". Hugging Face Model Hub. (2023), [Online]. Available: `https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-fr`.

[88] M. Sowański. “Iva_mt_wslot-m2m100_418m-en-pl”. Hugging Face Model Hub. (2023), [Online]. Available: https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-pl.

[89] M. Sowański. “Iva_mt_wslot-m2m100_418m-en-pt”. Hugging Face Model Hub. (2023), [Online]. Available: https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-pt.

[90] M. Sowański. “Iva_mt_wslot-m2m100_418m-en-sv”. Hugging Face Model Hub. (2023), [Online]. Available: https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-sv.

[91] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder–decoder approaches”, *Syntax, Semantics and Structure in Statistical Translation*, p. 103, 2014.

[92] P. Gage, “A new algorithm for data compression”, *C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.

[93] R. Sennrich, O. Firat, K. Cho, *et al.*, “Nematus: A toolkit for neural machine translation”, in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 65–68.

[94] A. Axelrod, X. He, and J. Gao, “Domain adaptation via pseudo in-domain data selection”, in *Proceedings of the 2011 conference on empirical methods in natural language processing*, 2011, pp. 355–362.

[95] Y. Wu, M. Schuster, Z. Chen, *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation”, *ArXiv*, vol. abs/1609.08144, 2016.

[96] C. Chu, R. Dabre, and S. Kurohashi, “An empirical comparison of domain adaptation methods for neural machine translation”, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 385–391.

[97] P. Koehn, H. Hoang, A. Birch, *et al.*, “Moses: Open source toolkit for statistical machine translation”, in *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, 2007, pp. 177–180.

[98] L. Macken, D. Prou, and A. Tezcan, “Quantifying the effect of machine translation in a high-quality human translation production process”, in *Informatics*, MDPI, vol. 7, 2020, p. 12.

[99] J. Nitzke and S. Hansen-Schirra, *A short guide to post-editing (Volume 16)*. Language Science Press, 2021.

[100] F. do Carmo, D. Shterionov, J. Moorkens, *et al.*, “A review of the state-of-the-art in automatic post-editing”, *Machine Translation*, vol. 35, pp. 101–143, 2021.

[101] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.”, *Journal of machine learning research*, vol. 9, no. 11, 2008.

[102] M. Kubis, P. Skórzewski, M. Sowański, and T. Ziętkiewicz, "Center for artificial intelligence challenge on conversational ai correctness", in *Proceedings of the 18th Conference on Computer Science and Intelligence Systems*, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., ser. Annals of Computer Science and Information Systems, vol. 35, IEEE, 2023, 1319–1324. DOI: 10.15439/2023B6058. [Online]. Available: http://dx.doi.org/10.15439/2023B6058.

[103] J. Hościłowicz, M. Sowański, P. Czubowski, and A. Janicki, "Can we use probing to better understand fine-tuning and knowledge distillation of the BERT NLU?", in *International Conference on Agents and Artificial Intelligence*, 2023.

List of Appendices

Appendix 1. Samples from Leyzer corpus

Table 24 presents a selection of Leyzer utterances from the English part of corpora. Presented examples cover all domains and 84 out of 194 intents.

Table 24. Selected utterances from the Leyzer corpus.

Domain	Intent	Utterance
Airconditioner	GetHumidity	check humidity
Airconditioner	GetTemperature	check the temperature on ac
Airconditioner	TurnOff	turn off ac
Airconditioner	TurnOn	turn on air conditioning
Calendar	AddEventWithName	add an event lunch with dulcie oja
Calendar	CheckCalendarEventName	check event meeting with donald
Calendar	OpenCalendar	open calendar
Console	ConsoleCD	go to path fedex4
Console	ConsoleEdit	open siamese archi jpeg
Console	ConsoleLS	show me files in scikit_learn_data
Console	ConsoleRM	remove FNP B 38 pdf
Contacts	EditContactWithName	edit contact hervey
Contacts	OpenContacts	open contacts
Contacts	OpenMyContact	open my contact's info
Contacts	ShowContactWithName	show contact viola aris
Email	OpenEmail	check email
Email	ReplyToEmailFromAddress	reply to papagena@enron.com
Email	SendEmail	send email
Email	SendEmailToAddress	send email to shanta@hotmail.com
Email	ShowEmailFromSender	show emails from lenee@kpmg.com
Email	ShowEmailFromTime	show me emails that arrived on friday
Email	ShowEmailWithLabel	show me emails labelled offers
Email	ShowEmailWithPriority	show me important emails
Facebook	OpenFacebook	open facebook
Facebook	PostPicture	post a picture on facebook
Facebook	PostStatus	write so tired today on facebook
Facebook	ShowAlbumWithName	show photos in my album kittens
Fitbit	AddWeight	save my weight on fitbit
Fitbit	NotifyOnWeight	tell me if my weight goes over 81 kg
Fitbit	ShowSteps	tell me the number of steps i took
Gdrive	CreateFile	create google drive file

Continued on next page

Table 24 – continued from previous page

Domain	Intent	Utterance
Gdrive	OpenFileWithName	edit video 32 on google drive
Gdrive	OpenGdrive	open my google drive
Gdrive	ShowFilesWithStar	show my starred google files
Gdrive	ShowNewestFiles	show my newest google files
Instagram	OpenInstagram	open instagram
Instagram	ShowPictures	show my instagram pictures
Instagram	TakePicture	take a picture using instagram
News	NotifyWhenPortalUpdates	follow news from fox news
News	ShowNews	open ny times
News	ShowNewsFromSection	open the nyt sport section
Phone	CallEmergency	call 911
Phone	CallNumber	call +34316855297
Phone	CallContact	call briney
Phone	ShowSMS	check my sms
Phone	SMSToContact	send a text to coleen
Slack	CheckChannelHistory	check slack channel history
Slack	CheckLastMessages	check slack messages
Slack	CheckUserStatus	check the presence of jack on slack
Slack	OpenSlack	open slack
Slack	SetStatusAway	change status on slack to inactive
Speaker	DecreaseVolume	volume down
Speaker	DecreaseVolumeByPercent	volume down by 69
Speaker	IncreaseVolume	volume up
Speaker	MuteOff	unmute speaker
Speaker	MuteOn	mute speaker
Spotify	AddAlbumToPlaylist	add this single to punk unleashed
Spotify	AddSongToPlaylist	save current song
Spotify	CreatePlaylist	create playlist
Spotify	NextSong	next song
Spotify	OpenSpotify	play some music
Spotify	Pause	pause this song
Spotify	PlayPlaylist	listen to global music playlist
Translate	DetectLanguage	determine language of des huîtres
Translate	SetDefaultLanguage	set language to italian
Translate	TranslateText	translate do you have this in my size
Twitter	FollowUser	follow amilee110 on twitter

Continued on next page

Table 24 – continued from previous page

Domain	Intent	Utterance
Twitter	OpenTwitter	open twitter
Weather	MoonphaseInLocation	check moon phase in berlin
Weather	OpenWeather	what's the weather
Weather	SunriseInLocation	check sunrise in stearns
Weather	WeatherTomorrow	check weather for tomorrow
Websearch	OpenEngine	search on google
Websearch	SearchTextOnEngine	search for crosssite on bing
Websearch	SearchText	search for agario on web
Wikipedia	DownloadAsPdf	download page as pdf
Wikipedia	GoToElementNumber	go to first element from contents
Wikipedia	OpenWikipedia	open wiki
Yelp	OpenRestaurants	find open restaurants nearby
Yelp	SearchByCategory	find salvadoran food around here
Yelp	SearchByQuery	find craft breweries and pubs on yelp
Youtube	FindQuery	find katy perry on youtube
Youtube	NextVideo	play next video
Youtube	OpenYT	open yt

Table 25 presents differences in the sub-intent modalities of the Leyzer corpus. Examples *001* and *002* show a difference between *Naturalness Level* for the same intent. Examples *003* and *004* show a difference between *Verb Pattern* for the same *Intent* and *Naturalness Level*. Examples *005* and *006* present utterances with *IOB* annotation in multi-word slots and multi-slot scenarios.

Table 25. Selected utterances from the Leyzer corpus that show the difference between sub-intent modalities (naturalness level and verb patterns) and slot annotations.

Columns	**Row Values**
ID	001
Domain & Intent	Airconditioner | ChangeTemperature
Naturalness Level & Verb Pattern	L0TC | verb_pattern_01
Utterance	change the temperature on my thermostat
IOB	o o o o o o
ID	002
Domain & Intent	Airconditioner | ChangeTemperature
Naturalness Level & Verb Pattern	REPHRASE | verb_pattern_01
Utterance	it is too hot in here
IOB	o o o o o o
ID	003
Domain & Intent	Contacts | ShowContactWithEmail
Naturalness Level & Verb Pattern	L1TC | verb_pattern_01
Utterance	display contact with an email karin@schwab.com
IOB	o o o o o b-email
ID	004
Domain & Intent	Contacts | ShowContactWithName
Naturalness Level & Verb Pattern	L1TC | verb_pattern_02
Utterance	open contact with an email eadie@outlook.com
IOB	o o o o o b-email
ID	005
Domain & Intent	Fitbit | ShowStepsOnDate
Naturalness Level & Verb Pattern	L0TC | verb_pattern_01
Utterance	tell me the number of steps i took on 22rd July
IOB	o o o o o o o o o b-date i-date
ID	006
Domain & Intent	News | ShowNewsFromSection
Naturalness Level & Verb Pattern	L0TC | verb_pattern_01
Utterance	ead the health section of the time magazine
IOB	o o b-section o o o b-portal i-portal
ID	007
Domain & Intent	Youtube | ShowSubscribedChannels
Naturalness Level & Verb Pattern	L1TC | verb_pattern_01
Utterance	show subscribed channels on youtube
IOB	o o o o o
ID	008
Domain & Intent	Slack | CheckMessagesInChannel
Naturalness Level & Verb Pattern	L1TC | verb_pattern_01
Utterance	display recent slack messages in transportation
IOB	o o o o o b-channel

Appendix 2. Sample translations from IVA_MT model

Table 26 shows the multivariant translations generated by the iva_mt-en-pl model [88]. The library multiverb_iva_mt produces additional translation variants when the verb in the input sentence matches an entry in the verb ontology. If no suitable variant can be generated, the library returns a single translation.

Table 26. Multiple variant translations generated by IVA_MT model's and multiverb_iva_mt library.

Columns	Row Values
ID	001
Input	play my rock playlist
Translation 1	graj moją playlistę rockową
Translation 2	odtwórz moją playlistę rockową
Translation 3	odegraj moją playlistę rockową
Translation 4	odtwarzaj moją playlistę rockową
Translation 5	puść moją rockową playlistę
ID	002
Input	tell me if i have new emails
Translation 1	powiedz czy dostałem nowe maile
Translation 2	opowiedz czy dostałem nowe maile
Translation 3	mów czy dostałem nowe maile
ę ID	003
Input	show me events in sacramento
Translation 1	wyświetl wydarzenia w sacramento
Translation 2	pokaż mi wydarzenia w sacramento
ID	004
Input	delete item from list
Translation 1	usuń pozycję z listy
Translation 2	skasuj pozycję z listy

Table 27 shows translations from three different NLU corpora using the iva_mt-en-pl model [88]. In the Leyzer corpus, slots are annotated using the IOB format and stored in a separate column. For instance, the annotations for the sentence "send an answer to shandeigh" (*ID 001*) are stored in a separate column as "o o o o b-to". The IVA_MT model uses XML annotations, so the input sentence is converted to "send an answer to <a>shandeigh<a>". The "Slot Dictionary" column provides a mapping between the input annotations and the actual corpus annotations. In the MTOD corpus, slots are annotated by specifying their start and end positions in the text, followed by the slot name after a colon. For example, the slot annotation for sentence *ID 004* is "11:15:datetime,16:21:weather/attribute". The SLURP corpus uses another annotation format, but only the final version, which avoids XML conversion, is shown here.

Table 27. IVA_MT model's translation of selected examples across three NLU corpora.

Columns	Row Values
ID	001
Direction	en-pl
Corpus	Leyzer
Input	send an answer to <a>shandeigh<a>
Translation	wyślij odpowiedź do <a>shandeigh<a>
Slot Dictionary	{"a": "to"}
ID	002
Direction	en-pl
Corpus	Leyzer
Input	i want to eat <a>latin american<a> food in <b>milwaukee<b>
Translation	chcę zjeść <a>latynoską<a> kuchnię <b>w krakowie<b>
Slot Dictionary	{"a": "category", "b": "location"}
ID	003
Direction	en-pl
Corpus	MTOD
Input	set an alarm <a>for thursday at 7 am<a>
Translation	nastaw alarm na <a>czwartek na 7 rano<a>
Slot Dictionary	{"a": "datetime"}
ID	004
Direction	en-pl
Corpus	MTOD
Input	will it <a>rain<a> <b>tonight<b>
Translation	czy będzie <a>padać<a> <b>dziś wieczorem<b>
Slot Dictionary	{"a": "weather/attribute", "b": "datetime"}
ID	005
Direction	en-pl
Corpus	MTOD
Input	remind me <a>the day before<a> my <b>doctor's appointment<b>
Translation	przypomnij mi <a>dzień wcześniej<a> o <b>wizycie u lekarza<b>
Slot Dictionary	{"a": "datetime", "b": "todo"}
ID	006
Direction	en-pl
Corpus	SLURP
Input	please turn off the light of [house_place : my son's room]
Translation	proszę wyłącz światło [house_place : w pokoju mojego syna]
ID	007
Direction	en-pl
Corpus	SLURP
Input	what is the definition of [definition_word : logic]
Translation	jaka jest definicja słowa [definition_word : logika]
ID	008
Direction	en-pl
Corpus	SLURP
Input	order [food_type : pizza] for [order_type : delivery]
Translation	zamów [food_type : pizzę] z [order_type : dowozem]