Strategy for Future Development of Upper Sorbian Speech Recognition

Ivan Kraljevski, Frank Duckhorn, Constanze Tschöpe, Matthias Wolff*

Fraunhofer Institut für Keramische Technologien und Systeme IKTS
Maria-Reiche-Straße 2, 01109 Dresden

**Brandenburgische Technische Universität Cottbus–Senftenberg, Cottbus*

Cottbus, 2023-24

Table of contents

1 Introduction

This document aims to define general guidelines and recommendations for further development of speech recognition in Upper Sorbian considering possible real-world application and use cases.

1.1 Speech Recognition Technology

Automatic speech recognition (ASR) also known as Speech-to-Text recognition (STT), has a long history spanning over seven decades. Like Natural Language Processing (NLP), the crucial milestones correspond with AI's general advances, but the practical and widespread application became possible only in the last two decades. The technology experienced breakthroughs rapidly dropping the word error rates, capitalizing on deep learning and big data [1].

1.1.1 Approaches

Statistical speech recognition algorithms are comprised of those addressing acoustic and language modeling. Except for pure end-to-end (E2E) approaches, the acoustic and whole or some parts of the language modeling are performed together and cannot be separated.

- **The acoustic model** defines the relationship between an audio signal and the units that make up speech, for instance, phonemes (linguistic) or sub-phonetic (senones). It takes audio recordings with their corresponding transliterations to train an acoustic model containing statistical representations of the sounds that make up each word. Since the raw speech signal is not a suitable input, feature extraction is performed to transform the speech signal into time sequences of acoustic feature vectors.
    - The idea of using neural networks for acoustic modeling in hybrid ASR systems is quite old and this approach is currently the most widespread in commercial ASR applications.
    - Recently, E2E models have been used simultaneously for acoustic and language modeling, effectively omitting the feature extraction phase, and mapping the input audio to a sequence of linguistic tokens (senones, phonemes, sub-words, words). Depending on the nature of the output tokens, additional language modeling is required to improve the word recognition performance.
- **Language models** can be statistical (data-driven) or knowledge-driven (grammar) and can be used for a variety of tasks, not only for speech recognition but also, optical-character or handwriting recognition, and natural language generation.
  Recently large language models (LLM) are becoming more and more ubiquitous with the ability to achieve **general-purpose** language understanding and generation. The language model approaches can be roughly divided into:
    - **N-gram statistical language models** that are based on the probability of occurrence of a word given the previous N-1 words. The model assumes that the probability of a word depends only on the previous N-1 words (Markov assumption). N-gram models are simple and computationally efficient. However, they also have limited context understanding and struggle with long-range dependencies.
    - **Neural Networks** can capture non-linear dependencies and have more context awareness than N-gram models. Language models can be used in the form of Feedforward NN, Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) Networks, and Gated Recurrent Units (GRU) models.
    - **Transformers** revolutionized NLP with an attention mechanism, allowing them to capture long-range dependencies more effectively. The model is based on self-attention mechanisms and operates in parallel, making it highly efficient. They achieve State-of-the-Art (SotA) performance, have parallel processing, and are effective for long-range dependencies. On the other hand, they are computationally expensive and require substantial resources. Bidirectional Encoder Representations from Transformers (BERT) [2]

considers the context in both directions while Generative Pre-trained Transformer (GPT) models focus on generative tasks to predict the next word in a sequence.

Main ASR approaches ordered by their historical occurrence:

- Template recognition based - DTW (Dynamic-Time-Warping). DTW measures the similarity between two temporal sequences, such as time-series data or speech signals. It allows for a flexible alignment between the two sequences by warping the time axis, accounting for variations in speed and timing.
- Gaussian Mixture Model/Hidden-Markov Model (GMM-HMM). GMM models the probability distribution of a multidimensional data point as a weighted sum of several Gaussian distributions. Each Gaussian component represents a cluster in the data, and the model allows for mixed membership, meaning a data point can belong to multiple clusters. HMM is a probabilistic model that represents a system with hidden states, and observable outcomes are probabilistically dependent on these hidden states. HMM consists of a set of hidden states, transition probabilities between states, and emission probabilities governing the observed data given to each hidden state. GMM-HMM is widely used in speech recognition systems, and gesture recognition, and is used for speaker verification and identification in biometric systems.
- Hybrid approach, Deep Neural Networks/ Hidden-Markov Model (DNN/HMM). DNNs can capture complex acoustic patterns, enhancing the modeling of acoustic properties in the hybrid system. HMMs, with their inherent ability to model temporal dependencies, complement DNNs in capturing the sequential nature of data.
- End-to-end (E2E) is a paradigm in automatic speech recognition where a single neural network is trained to directly convert input audio signals into transcriptions without the need for intermediate representations or subsystems.

1.1.2 Applications

There are many diverse areas where speech recognition applications are employed:

- Human-computer interfaces (in-car systems),
- Health care for medical documentation dictation and therapeutic use,
- Administrative, dictation, and protocol transcriptions,
- Real-time captioning of live events,
- Personal Voice Assistants,
- Information retrieval,
- Command-and-control (industry, military),
- Assistive technologies (hearing and sight impaired),
- Computational phonology and linguistics,
- Telephony,
- Computer gaming,
- Computer-aided language learning,
- Emotion recognition,
- Home Automation,
- and many others.

Each application or use case has different requirements in terms of accuracy and speed, streaming, latency, and adaptation capability, which affect the practical system deployment.

ASR systems can be described by different features of both major components (acoustic and linguistic):

Acoustic:

- Speaker dependence versus independence.
- Isolated, discontinuous, or continuous speech.
- Read vs. spontaneous speech.
- Adverse conditions (reverberation, background noise).

Language:

- Vocabulary size and confusability (small, medium, or large).
- Task and language constraints (general (open) or specific domain).

Each of them has a different influence on the speech recognition performance. For instance, ASR with a small vocabulary (command-and-control) is more accurate and robust in adverse conditions. Large Vocabulary Continuous ASR (LVCSR) is better performing as a speaker-dependent for dictation tasks, and so on.

1.1.3 Challenges in Speech Recognition

The main problem in speech recognition is the mismatch between the data used to train/build the ASR system and the one that is used as input for the trained system. The mismatch can occur in both acoustic (speaker, ambient noise) and the language (pronunciation, out-of-vocabulary words, different domain) models or any of the above-mentioned features of the ASR it will greatly reduce the recognition performance.

The "rule of thumb" is that enough amount of representative speech data must be provided for the supervised training and optimization of an ASR system. However, collected speech data needs to be manually transcribed, which is time-consuming and costly, or collected from sources with pre-existing transcriptions, which are harder to find for application domains that lack wide representation.

1.1.4 Security and Explainability

Human speech is a biometrical feature that can be used for the identification of a person; hence the acoustic model contains the features of many speakers, and it is possible that such data could be compromised. This applies more to the GMM/HMM systems than to the E2E systems.

1.1.5 State-of-the-Art

The Dynamic-Time-Warping and pure GMM-HMM-based ASR approaches are becoming obsolete, and they are not able to keep up with the recent deep-learning performance achievements. They are still used in some legacy applications or low-end hardware for simple voice control applications.

For deep learning, the paradigm "There is no data like more data" (*Bob Mercer at Arden House, 1985*) is the key to success. Now it is possible to train end-to-end speech recognition systems, with raw speech signals as input and the transcriptions in the form of a sequence of words as outputs.

During training on hundreds of thousands of hours of speech (some recent models used over a million), the neural network model learns the optimal acoustic features, phono-tactic and linguistic rules, and syntactic and semantic concepts. There is no need for algorithms, linguistics, statistics, error analysis, or anything else that requires human expertise if there is enough data (black-box approach).

The current SotA speech recognition systems demonstrated impressive performances on standard recognition tasks (Librispeech, TIMIT). Just a few examples, the recent framework from the Facebook research team, the wav2vec 2.0 [3] (trained on 960 hours of speech), achieved a word-error-rate (WER) of 1.8/3.3 percent on "test-clean/other" of Librispeech and phoneme-error-rate of 7.4/8.3 percent on "dev/test" on TIMIT.

Such impressive results are achieved on standard speech corpora in restricted conditions, domains, and language, not necessarily directly translated to real-world situations. The study [4] shows that the current SotA of off-the-shelf speech recognizers performs relatively poorly in domain-specific use cases under noisy conditions.

1.1.6 Comparison of Speech Recognition Approaches

Qualitative comparison of the three most common approaches in speech recognition: classic GMM/HMM (Gaussian Mixture Model/Hidden Markov Model), hybrid DNN/HMM (Deep Neural Network/Hidden Markov Model), and E2E (end-to-end) systems:

1. Classic GMM/HMM (Gaussian Mixture Model/Hidden Markov Model):

- Pros:
    - Established: GMM/HMM has been a traditional and well-established approach in speech recognition.
    - Decoding simplicity: the use of HMMs for temporal modeling simplifies the decoding process.
    - Robustness: GMMs can model complex distributions, providing robustness to variations in speech.
- Cons:
    - Lack of context modeling: GMMs struggle to capture long-term dependencies and context effectively.
    - Feature engineering: careful design of handcrafted features to represent the input speech signal.
    - Limited capacity: unable to capture intricate patterns in highly variable speech data.

2. Hybrid DNN/HMM (Deep Neural Network/Hidden Markov Model):

- Pros:
    - Improved context modeling: capturing complex dependencies and context, improving recognition accuracy.
    - E2E feature learning: automatic learning of relevant features from raw input data.
    - Adaptability: fine-tuning for specific tasks, enhancing adaptability to different domains.
- Cons:
    - Training complexity: requires substantial computational resources and time.
    - Data requirements: large amounts of labeled data for effective training.
    - Interpretability: considered as "black box" models, making it challenging to interpret their decisions.

3. E2E (End-to-end) Systems:

- Pros:
    - Simplified architecture: no need for explicit feature engineering or intermediate representations.
    - Reduced manual effort: automatic feature learning process, reducing the need for extensive manual intervention.
    - Better context modeling: capturing long-term dependencies and context more effectively.
- Cons:
    - Training requirements: substantial amounts of labeled data for training, computational resources, and time. Directly reflected in costs.
    - Hardware requirements: most E2E systems require a Graphical-Processing-Unit (GPU) for training and sometimes also for inference.
    - Lack of interpretability: challenging to interpret due to their complex architectures.
    - Limited adaptability: less adaptable to new domains compared to more traditional approaches.

E2E models perform better than the classic and hybrid ASR when training data is abundant, while not scaling well to low-resource conditions. A domain change requires a flexible exchange of language models, which is natural for classical ASR models based on a separation of acoustic and language models.

There are several major advantages of E2E models over traditional hybrid models. First, E2E models use a single objective function which is consistent with the ASR objective to optimize the whole network, while traditional hybrid models optimize individual components separately, which cannot guarantee the global optimum [5].

E2E models simplify the ASR pipeline because they output directly tokens, such as characters or words, in contrast, the design of traditional hybrid models is complicated, requiring lots of expert knowledge with years of ASR experience. E2E models are much more compact than traditional hybrid models because a single network is used for ASR.

On the other hand, hybrid models are still used in a large proportion of commercial ASR systems, because the ASR accuracy is not the only factor for the production choice between hybrid and E2E models. Factors such as latency, responsiveness, ease of adaptation on speaker, environment, and domain, affect the commercial model deployment decision. Traditional hybrid models, highly optimized on these factors, are usually outperforming the E2E model in highly specialized commercial applications [5].

Top E2E ASR systems usually require orders of magnitude more training iterations (epochs) than comparable classical and hybrid ASR systems. The high level of integration of E2E models also involves a loss in modularity, which might support the explainability and reusability of models. One assumed advantage of E2E models is that everything is trained from data and secondary knowledge sources (e.g., pronunciation lexica and phoneme sets) are not necessary [6] in the training process.

In summary, GMM/HMM is a classic approach that is simple and robust, the hybrid DNN/HMM systems are better in context modeling while preserving adaptability. E2E systems have even further simpler architectures and they automate feature learning, however at the cost of interpretability and adaptability in certain use cases. The choice between these approaches depends on the specific requirements of the application and the available resources.

1.2 Current State of Speech Recognition in Upper Sorbian

1.2.1 HSB-I: Feasibility Study on Automatic Speech Recognition in Upper Sorbian Language

The first project was a feasibility study (noted as HSB-I), where the objective was to investigate possibilities for transfer learning by using existing speech resources (German), in the case of Upper Sorbian (ISO language code: hsb) as an example of an endangered and under-resourced (UR) language.

This was achieved by the development of a voice application demonstrator, a prototypical Automatic Speech Recognition (ASR) system on a limited language domain ("Smart Lamp").

The main work consisted of a definition of the main objectives:

- Definition of grapheme and phoneme inventory using all available sources provided by the project partner (Stiftung für das sorbische Volk).
- Definition of a simple speech application (demonstrator of a voice-controlled smart lamp).
- Preparing and organizing data collection of speech recordings used to test and develop the speech application.
- Finally, the realization of the speech application by:
    - Post-processing of the speech corpus.
    - Phoneme recognition for acoustic model evaluation.
    - Optimization of the phoneme inventory and the grapheme-phoneme mappings.
    - Language modeling realized with a Finite-States-Grammars.
    - Acoustic modeling, adaptation with the collected data of the basic AM model (German).
    - Evaluation of the models on the test speech data.

The main impact of the feasibility study was the fulfillment of requirements for the development of more sophisticated speech applications in the Upper Sorbian language: phoneme inventory, speech corpus (with around 11 hours duration), simple acoustic (mono-phone), and language models (grammars).
Also, the created tools and the established procedures can be used to collect data and develop speech applications in the Lower Sorbian language by redefining the grapheme to phoneme inventory and the mappings.

1.2.2 HSB-II: Improving Speech Recognition in Upper Sorbian

The second project (noted as HSB-II) aimed to use the collected data and perform acoustic modeling with the native phoneme inventory, instead of using acoustic models of other languages (German):

- Acoustic modeling:
    - Pronunciation modeling.
    - Forced alignment of the speech corpus with a pre-trained model.
    - Training of new, native Upper Sorbian acoustic models with (dLabPro[1]/UASR[2]).
    - Speaker-dependent adaptation.
    - Performance evaluation of speaker-independent and dependent acoustic models.
- Lexicon modeling:
    - Statistical Grapheme-to-Phoneme (G2P) supervised modeling.
    - Open-source toolkits with available training data in the form of word – phonemes.
- Language modeling:
    - Context free grammars (Finite-State-Grammars) realized as Finite-State-Transducers.
    - Word-class grammar (time, date, numbers).
    - Statistical language model with word classes.
- Guidelines of best practices for creating a speech corpus.

This project further improved the tools and established procedures for acoustic, lexicon, and language modeling that can be used for the creation of domain-specific, yet flexible speech applications with a small to medium vocabulary.

[1] https://github.com/matthias-wolff/dLabPro
[2] https://github.com/matthias-wolff/UASR

The used tools and the technologies are still very dependent on the amount of available speech and text data. For instance, with the amount of speech, only mono-phone acoustic model training is feasible. The tri-phone model in this case will underperform, and to use of pre-trained tri-phone or hybrid Deep Neural Networks with Hidden Markov models (DNN/HMM) on other languages will introduce a mismatch in the observed triphones in the adaptation data.

The text data is insufficient to train statistical language models with a large vocabulary, therefore context-free-grammars combined with word classes are optimal solutions for speech applications such as voice-controlled personal assistants or smart home devices limited on one or more speakers (adapted acoustic models).

Additionally, small domain-specific text data with limited vocabulary can be successfully employed for statistical language modeling with word classes which as a result will provide voice applications with more flexible expression, and where even the utterance is not completely correctly recognized the meaning (semantics) could be robustly inferred.

1.2.3 HSB-III: Speech Recognition in Upper Sorbian for Dictation

The project aims to investigate the possible development of a domain-independent voice dictation system by expanding and improving the following technologies:

- Triphone-based acoustic models.
- Sub-word-based statistical language models (syllable or morpheme sub-word units)

This project builds upon the foundations laid in the previous two projects, intending to further expand and develop resources and tools necessary to advance toward Large Vocabulary and Continuous Speech Recognition (LVCSR) in Upper Sorbian.

At first, it is restricted to the application domain (domain-dependent), and in perspective to be employed in domain-independent applications.

The activities were divided into three work packages targeting the main components of a traditional LVCSR system, the acoustic (work package 1), the language modeling (work package 2), and the decoder within the speech recognizer (work package 3).

The acoustic modeling employing tri-phones was established and achieved a level of robustness ensuring speaker-independent recognition in real and adverse conditions. The acoustic modeling procedure does not need any further development and the models will be significantly improved with more data.

Speech recognition for highly inflected languages (such as Upper Sorbian) poses challenges for language modeling due to the many word forms that must be included in the vocabulary. The size of the vocabulary and the quality of the language modes directly influence speech recognition performance.

In the language modeling work package, tools and procedures for text processing and normalization, word-class modeling with recognition of named entities (NER), and tokenization in sub-word units (e.g., morphemes) were developed. The approaches were evaluated, and the best performing were employed for the domain-dependent speech recognition configuration.

The ideal case would be the one, where the language model is built on the domain-dependent texts with, as much as, possible small vocabulary, and where the spoken sentences match the application domain.

In the third work package, the decoder in the speech recognizer application was modified to support word-class modeling along with the sub-word tokens. The recognizer is configurable and flexible, providing robust and real-time recognition for the given acoustic and language models.

1.2.4 HSB-IV: Improvement of the Domain-Independent Upper Sorbian ASR for Dictation

The created speech and language resources are valuable assets that contribute to the digitalization and preservation of the Upper Sorbian language.

The experiences, the developed procedures, and the software are foundations for other applications in computational linguistics and speech technologies (speech synthesis, dialog management, natural language understanding and generation, and machine translation). Also, they will speed up acquiring new data, that will further improve the speech and language models.

The project is focused on further improvement and optimizations of the technologies by collecting and organizing speech and language data.

Exploration of the possibilities for employment of the latest State-of-the-Art technologies and investigating new and inventive speech applications in education, office and business administration, live captioning, and transcription of media, personalized virtual assistants and chatbots, improvement of life quality of people with disabilities.

The objectives of this project are:

- Implementation of new training recipes of the acoustic models (monophonic and tri-phone) including new speech recordings and changes in phonetics.
- Optimization of the phonetic inventory. Further resources for the creation of the lexicon were investigated and integrated.
- Creation of domain-specific language models based on the already created normalized texts.
- Normalization of new domain-specific texts and their integration.
- Improvement of the approach to the use of morphological units, possibly using other NLP tokenization approaches.
- Code review of the adaptation of the recognizer to the "VOSK" API.
- Integration/improvement/parameterization of voice detection ("VAD").
- Based on the currently available technology and resources, a strategy for future speech recognition development for Upper Sorbian.

1.2.5 Publications

As an outcome of the activities in the projects, some of the results are presented at international conferences and their proceedings.

Word Class-Based Language Modeling: A Case of Upper Sorbian

Authors
Isidor Maier, Johannes Kuhn, Frank Duckhorn, Ivan Kraljevski, Daniel Sobe, Matthias Wolff, Constanze Tschöpe

Publication date
2022/6

Conference
Proceedings of the Workshop on Resources and Technologies for Indigenous, Endangered, and Lesser-resourced Languages in Eurasia within the 13th Language Resources and Evaluation Conference

Pages
28-35

Description
In this paper, we show how word class-based language modeling can support the integration of a small language in modern applications of speech technology. The methods described in this paper can be applied to any language. We demonstrate the methods on Upper Sorbian. The word classes model the semantic expressions of numerals, date, and time of day. The implementation of the created grammars

was realized in the form of finite-state transducers (FSTs) and minimalist grammars (MGs). We practically demonstrate the usage of the FSTs in a simple smart-home speech application, that can set wake-up alarms and appointments expressed in a variety of spontaneous and natural sentences. While the created MGs are not integrated into an application for practical use yet, they provide evidence that MGs could potentially work more efficiently than FSTs in built-on applications. In particular, MGs can work with a significantly smaller lexicon size, since their more complex structure lets them generate more expressions with fewer items, while still avoiding wrong expressions.

Glottal Stops in Upper Sorbian: A Data-Driven Approach

Authors
Ivan Kraljevski, Maria Paola Bissiri, Frank Duckhorn, Constanze Tschöpe, Matthias Wolff

Publication date
2021/9/3

Conference
Interspeech 2021

Issue
Proc. Interspeech 2021

Pages
1001-1005

Publisher
doi: 10.21437/Interspeech.2021-1101

Description
We present a data-driven approach for the quantitative analysis of glottal stops before word-initial vowels in Upper Sorbian, a West Slavic minority language spoken in Germany. Glottal stops are word-boundary markers, and their detection can improve the performance of automatic speech recognition and speech synthesis systems.
We employed cross-language transfer using an acoustic model in German to develop a forced-alignment method for the phonetic segmentation of a read-speech corpus in Upper Sorbian. The missing phonemic units were created by combining the existing phoneme models. In the forced alignment procedure, the glottal stops were considered optional in front of word-initial vowels.

Cross-lingual acoustic modeling in Upper Sorbian-preliminary study

Authors
Ivan Kraljevski, Marek Rjelka, Frank Duckhorn, Constanze Tschöpe, Matthias Wolff

Publication date
2021

Journal
Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung

Volume
2021

Pages
43-50

Description
In this paper, we present a preliminary study for acoustic modeling in Upper Sorbian, where a model of German was used in cross-lingual transfer learning. At first, we define the grapheme and phoneme inventories and map the target phonemes from the most similar German source equivalents. Phonetically

balanced sentences for the recording prompts were selected from a combination of general and domain-specific textual data. The speech corpora with a total duration of around 11 hours were collected in controlled recording sessions involving an equal number of females, males, and children. The baseline acoustic model was employed to force-align the speech corpora given the knowledge-based phoneme mappings. How well the mappings were, was evaluated by the phoneme confusions in free-phoneme recognition. The newly derived data-driven model with a reduced phoneme set was included in the adaptation and evaluation along with the baseline acoustic model. The model adaptation performance was cross validated with the "Leave One Group Out" strategy. We observed major improvements in phoneme error rates after adaptation to the knowledge-based and data-driven phoneme mappings. The study confirmed the feasibility of transfer learning for acoustic model adaptation in the case of Upper Sorbian, at the same time demonstrating practical usability with a small vocabulary speech recognition application (Smart Lamp).

1.3 Strategy Document Considerations

This document provides an overview of the current state of speech resources and technologies and tries to give relevant answers and directions to the following questions and dilemmas regarding prospective speech applications.

1.3.1 Applications

- Toys and smart home applications are covered by existing technology. Namely, the dLabPro/UASR and the recIKTS speech recognition systems.

- Other prospective applications will be client-server based, where one or more instances of the ASR system will recognize the client's audio recordings and serve back the results.

1.3.2 Licensing (OSS vs Proprietary)

The choice between open-source and proprietary software depends on specific organizational needs, preferences, and the nature of the tasks the software is intended to perform. A combination of both types of software can be used to leverage the advantages each brings to the table.

- The value/strategic importance of Open-Source Software (OOS)., extends beyond cost savings, encompassing flexibility, security, innovation, and community-driven collaboration. Embracing open-source principles can be a strategic decision that positively impacts various aspects of an organization's operations and development.

- Benefits of proprietary software. Control and consistency, dedicated customer support services, including unique features and functionality to specific industry needs, a more focused and streamlined development process, comprehensive documentation, user guides, and training materials, comply with industry standards and regulations.

1.3.3 Transfer Learning

Transfer learning for speech recognition in Slavonic languages involves employing pre-existing models or knowledge from one Slavonic language to achieve recognition or to improve the performance of a speech recognition system in another Slavonic language. In the given case it would be the pairs: Czech and Upper Sorbian or Polish and Lower Sorbian.

- Shared phonetics and acoustics. Transfer learning can be effective when the source and target languages have similar sound patterns and pronunciation.
- Multilingual acoustic models are capable of recognizing speech across multiple Slavonic languages. Pre-training a model on a dataset containing diverse Slavonic languages can help the model capture general acoustic features.
- Shared vocabulary and language models could be beneficial for pre-training a language model on one language and adapting it to another.

- Data augmentation of training data for the target language by creating variations of existing audio data.
- Transfer learning from multilingual ASR Models. Pre-trained multilingual models designed to handle multiple languages to capture general acoustic features and language patterns.
- Cross-lingual acoustic embeddings. Extracted cross-lingual acoustic embeddings from a pre-trained model on a source Slavonic language to be used as features for training a new acoustic model on the target (Upper Sorbian) language.
- Adapting language models. Adapting pre-trained language models to the target language by incorporating domain-specific vocabulary and language characteristics.
- Fine-tuning with limited target language data on a related, higher-resource Slavonic language and then adapting it to the target language using the available data.

Transfer learning for speech recognition in Slavonic languages requires careful consideration of linguistic similarities, available resources, and the specific characteristics of the target language.

1.3.4 LLMs and Translation to German/English

LLMs may not always outperform dedicated translation models for specific language pairs. The effectiveness of immediate translation using LLMs depends on factors like the quality of the pre-training, the availability of parallel data for fine-tuning, and the linguistic similarity between the source and target languages. Additionally, for critical applications, evaluating the performance of these models on specific translation tasks is recommended. Several approaches can be employed:

- Multilingual LLMs can understand and generate text in multiple languages.
- Zero-shot translation, a capability of a model to translate between language pairs it has not been explicitly trained on.
- Fine-tuning for translation of a pre-trained LLM between German and another Slavonic language.
- Pre-trained translation models are specifically designed for translating between a pair of languages. For instance, MarianMT[3] or mBART (multilingual BART) [7].
- Hybrid approaches, use pre-trained LLM for initial understanding and then fine-tune or use a dedicated translation model for more accurate and context-aware translations.
- Interactive translation systems use LLMs to understand the input sentence and generate a context-aware representation to be passed to a translation module that performs the actual translation.
- Context-aware translation, some LMMS like GPT-3 [8], can understand the context in a sentence for more accurate and context-aware translations.

Performing Upper Sorbian to German translation using BERT (Bidirectional Encoder Representations from Transformers) involves fine-tuning a pre-trained BERT model on a parallel corpus of Upper Sorbian and German sentences. To achieve that the following steps must be executed:

- Data preparation. Parallel corpus containing pairs of sentences in Upper Sorbian and their corresponding translations in German, must be aligned at the sentence level.
- Tokenization of sentences in both languages into sub-word and word-level tokens.
- Pre-processing the data into a format suitable for training, including encoding the tokens, creating attention masks, and preparing input sequences for the BERT model.
- Model selection of a pre-trained BERT model. Either a general-purpose multilingual BERT model or a model specifically pre-trained for translation tasks if available.
- Fine-tuning of the selected BERT model on the prepared parallel corpus.
- Evaluation of the performance of the fine-tuned model on a separate validation set to ensure that it is effectively learning the translation task.
- Inference on new Upper Sorbian sentences to obtain their German translations.
- Post-processing of the model's output to handle any specific formatting or linguistic differences between Upper Sorbian and German. This step ensures that the translated sentences are grammatically correct and contextually appropriate.

[3] https://huggingface.co/docs/transformers/model_doc/marian

Fine-tuning BERT for translation requires careful consideration of the available data and computational resources. Several machine translation frameworks and libraries, such as HuggingFace[4] Transformers or TensorFlow[5]'s official implementation, provide tools and pre-trained models that can simplify the process of fine-tuning BERT for translation tasks.

1.3.5 Requirements for Language Resources

The amount of textual material required to train a language model depends on the complexity of the language, the size and architecture of the model, and the specific task:

- Larger models with more parameters generally require more training data to perform well. For example, GPT-3 has hundreds of billions of parameters [8], and it needs massive amounts of diverse textual data for training.

- The complexity of the language task influences the amount of training data needed. Simple tasks, such as predicting the next word in a sentence (simple language modeling), may require less data compared to more complex tasks like machine translation or question answering.

- Languages with complex structures, rich vocabularies, and complex grammatical rules may require more training data to capture the distinctions effectively. For example, training a language model for a highly inflected language (such as the Upper Sorbian) might demand more data compared to a less complex language.

- If the language model is intended for a specific domain (e.g., legal documents, medical texts), training data should be representative of that domain where the amount of available domain-specific data directly influences the model performance.

- Pre-training a language model on a large corpus of diverse data before fine-tuning on a task-specific dataset is a usual practice. The amount of data used for pre-training can significantly affect the model's ability to generalize to new tasks or domains during fine-tuning.

- The quality of the training data is as important as the quantity. Clean, diverse, and representative data contribute to better model performance. Noisy or biased data can adversely impact the model's ability to generalize.

- Training large language models requires substantial computational resources. The availability of GPUs or TPUs and the training time budget can influence the decision on how much data to use.

To train LMMs (like GPT-3 or BERT), hundreds of gigabytes to terabytes of diverse textual data are often used. However, smaller models for simpler tasks may achieve satisfactory results with smaller datasets. It is important to experiment and monitor the model performance with different amounts of data to find the balance between computational resources, task requirements, and data availability. Additionally, continual evaluation and refinement are essential for achieving optimal performance.

[4] https://huggingface.co/docs/transformers/index
[5] https://www.tensorflow.org/

2 Speech Resources

2.1 Current Resources

The listed speech resources are collected during the projects HSB- I- IV. To the best of our knowledge, there are no other speech corpora in Upper Sorbian with this amount of data with corresponding transliterations that are ready to be used within any ASR framework with or without any preprocessing. Speech corpora created in other various projects for Upper Sorbian have different purposes and cannot be directly employed for the development of speech technologies (e.g., "Colloquial Upper Sorbian (Catholic vernacular SWR, Germany)"[6].

2.1.1 Speech Corpus

The HSB corpus is divided into train, dev, test sets as well as additional recordings.

1. Training (131 speakers, 62.32 hours, 52895 recordings) including augmented versions:

- HSB-1	Recordings from the HSB-I project	(3:33:50)
- HSB-2	Recordings from the HSB-I project	(3:22:30)
- HSB-3	Recordings from the HSB-I project	(4:38:34)
- SCF_9A	speech_corpus_film_9_a_pol	(0:08:29)
- SCF_G	speech_corpus_film_gilles	(0:53:12)
- SCF_KK	speech_corpus_film_karla_a_katrina	(0:40:15)
- SCF_MI	speech_corpus_film_mpz_insekten	(0:28:08)
- SCF_MR	speech_corpus_film_mpz_reise	(0:26:38)
- SCF_MW	speech_corpus_film_mpz_wjedro	(0:31:25)
- SCF_P	speech_corpus_film_peeweeje	(0:48:07)
- SCF_SW	speech_corpus_film_syn_winnetouwa	(1:43:09)
- SCF_MWJERA	speech_corpus_film_mala_wjera	(1:34:28)
- SCM_R1	speech_corpus_mic_recordings_1	(4:24:18)
- SCM_R2	speech_corpus_mic_recordings_2	(1:41:41)
- SCM_R3	speech_corpus_mic_recordings_3	(3:10:03)
- SCM_R4	speech_corpus_mic_recordings_4	(3:09:00)
- SCM_R5	speech_corpus_mic_recordings_5	(3:28:14)

2. Development (cross-validation) (17 speakers, 2.28 hours):

   - CV common voice dataset version 5.1

3. Test (8 speakers, 6.63 hours):

   - AABT
   - HSB_1_0001
   - HSB_1_0010
   - HSB_2_0001
   - HSB_2_0011
   - HSB_3_0001
   - HSB_3_0006
   - HSB_3_0007

[6] https://pangloss.cnrs.fr/corpus/Sorabe_sup%C3%A9rieur_(courant)?lang=en

4. Miscellaneous recordings:

   - ADP adaptation files (0:11:02)
   - ADP_B adaptation files (0:11:02)
   - GRM grammar testing (0:01:30)

5. Domain-Specific Recordings:
   The recordings are collected from YouTube videos of Sunday church services and serve for performance evaluation.

   - MISA speech_corpus_boze_mse_chroscicy_1 (3:23:10)

2.1.2 Textual Corpus

The combined textual corpus was collected from different sources with different domains.
Here under the term textual corpus, we denote a collection of normalized sentences, with a common letter casing, without punctuations and any other meta-data. The format is suitable for training N-gram statistical language models.

1. **HSB corpus**. All texts collected during HSB- I-IV projects (HSB-I, CV, V4.1):

   - HSB Common Voice v5.1
   - sorbian_institute_monolingual
   - web_monolingual
   - witaj_monolingual
   - adaptation
   - V4.1 (filmy_Normiert, lektorizowane_hs_teksty_bobr_Normiert, myto_cisinskeho_Normiert, wselcizny_Normiert)

   Sentences 591,101, vocabulary 389,407, and size of 42MB.

2. **Soblex Dictionary**. The dictionary contains pairs of words with their hyphenations.
   Vocabulary of 119,629 and size of 1.3MB.

3. **Lexemes Vocabulary**. A large list of words with all generated word forms.
   Vocabulary of 2,676,419 and size of 36MB.

4. **The Bible in Upper Sorbian.** Electronic version of the bible in Upper Sorbian in PDF format. The document is converted, processed, and normalized with numbers as the only used word class.
   Sentences 25,503, vocabulary 34,321, and size of 4.6MB.

5. **Choir Songs Book (spewarske2012).** Electronic version of church choir songs in PDF and DOCX format. The document is converted, processed, and normalized with numbers as the only used word class.
   Sentences 5,114, vocabulary 11,282, and size of 458KB.

6. **MCN Corpus**. Collection of transcribed eulogies, normalized without word classes.
   Sentences 892, vocabulary 5,332, and size of 103KB.

7. **Kaldi Corpus (V4).** Created from the transliterations of the train, test, and dev parts of the HSB-speech corpus.
   Sentences 58,007, vocabulary 29,603, and size of 2.8MB.

8. **Smart-Lamp Corpus.** It is used to demonstrate the word class combined with sub-word modeling.
   Sentences 375, vocabulary 106, and size of 12KB.

2.2 Deficiencies in the Current Speech Resources

2.2.1 Audio Data

The audio corpus with its augmented counterpart has a duration of total **74:58:47** hours. This is enough for acoustic modeling with GMM-HMM and DNN/HMM approaches, however, more data with appropriate quality is always better.

It might be possible to perform even a full E2E training from scratch with such an amount of data. However, by using state-of-the-art models such as wav2vec [3] or OpenAI Whisper [9], there is a danger of overfitting on the, in this case, relatively small amount of data. The solution would be either to reduce the capacity of the neural network model (fewer hidden layers or nodes per layer, e.g., OpenAI Whisper-small vs. -medium model) or artificially increase the amount of speech with acoustic augmentation. Acoustic augmentation would improve acoustic robustness but cannot improve language modeling because the amount of data is smaller than the text corpora used for separate language models.

On the other hand, the existing audio corpus can be successfully used for fine-tuning the pre-trained models in different or similar languages. The more recordings with more diverse textual content, the better.

2.2.2 Text Data

The available text corpus (in total) has a diverse size in different domains, just mixing them without any weighting will produce a relatively noisy corpus that is not suitable for statistical N-gram language modeling.

There will be disjointed contexts with unreliable probabilities that will not match the spoken language.

The whole speech application (recognizer) should be trained with the speech corpus that matches the intended domain. Namely, with audio recordings and their transcriptions from the target domain.

It is practically quite challenging to collect the required amount of in-domain data, sometimes impossible. Therefore, commonly the acoustic model is trained on different data and combined in the ASR system with the linguistical resources, pronunciation lexicon, and the language model of different domains.

However, there will be a high probability of a mismatch between the acoustic model (the seen tri-phones, phones, and senones) and the in-domain textual content. The impact on the performance is less than using a recognizer trained only on the available in-domain data.

At the same time, such an approach increases the re-usability of the components for different tasks, keeping the same robust and reliable acoustic model, and exchanging the language models with the lexicons accordingly for the intended application.

In the case of an E2E system, in-domain training requires a much larger amount of data and here fine tuning is the recommended approach. Depending on the output tokens, additional language modeling might be necessary to correct the misspellings and improve the accuracy. For instance, many E2E systems provide an output sequence of tokens (graphemes, phonemes) or sub-word tokens, hence it is quite possible to get non-existing and irregular words that should be corrected.

In the case where the E2E system is trained to output words, there is a possibility of Out-of-Vocabulary words rendering the fine-tuning more complicated.

2.3 Additional Speech Resources

From our experience, the amount of speech in the current corpus is sufficient for reliable and robust speaker-independent acoustic modeling. When necessary, the model can be always fine-tuned (adapted) to a specific speaker and acoustic environment regardless of the used approach (GMM/HMM, DNN/HMM, or E2E).

A more critical requirement is the number and size of textual resources. General purpose textual corpus should reflect the usual way how people communicate, while in-domain corpus contains specific and limited vocabulary and sentence patterns.

2.3.1 Text Collection Strategies

The most common approach to establishing a general-purpose corpus is to collect data from publicly available sources on the internet. Technically, this can be done with ***web spider bots*** that collect content for a given list of public internet domains and with specific topics, such as news portals, social networks, encyclopedias, movie subtitles, etc. This still can be a problem in the case of under-resourced languages, and the collected data will be sparsely covering different domains.

General strategies for text content collection:

1. Publicly accessible content, usage of existing digital texts in Upper Sorbian, such as books, articles, websites, and social media content. Government documents, educational materials, and local publications.

2. WEB scrapping, employment of tools to extract relevant content from websites, forums, and social media. It is important to consider and respect the terms of service and legal requirements.

3. Crowdsourcing, usage of platforms like Amazon Mechanical Turk or specialized linguistic crowdsourcing services for collecting and annotating data.

4. Scanning and digitizing of manuscripts and public documents, as found in libraries, private collections, academia, language schools, and similar.

5. Parallel texts (translated from other languages) of the same content in both the Upper Sorbian language and a more widely spoken language, such as German.

6. Speech-to-Text: employment of existing speech-to-text technology to transcribe audio content, to include colloquial language.

7. Cooperation with relevant institutions, linguists, and universities.

In the case of under-resourced languages, one feasible approach is to use a Speech-To-Text approach with an E2E multilingual acoustic model (such as fairseq MMS [10]) and consequently correct the misspellings by a human expert (native speaker).

2.3.2 Data Protection Considerations

When collecting and using data, ethical considerations should be respected, proper permissions obtained, and privacy laws adhered to. Especially when dealing with personal or sensitive information. Additionally, it is important to document the data sources and any pre-processing steps taken to maintain transparency and reproducibility.

If the licensing does not provide the possibility of use of the complete texts because of copyright infringements, one should check if there is a possibility to use derived data of such content that will not violate the copyright or the local laws. Source texts can be used to perform statistical analysis and provide the N-gram counts of the text tokens (graphemes, sub-words, or words) in such a way that the original content cannot be reconstructed.

3 Technology Improvements

3.1 Current State of the HSB-ASR Technology (recIKTS)

IKTS proprietary ASR is a toolkit that allows easy configuration of speech recognition applications and consists of a configurator and a recognizer (decoder).
The target operating system should be Windows or Linux and either x86 or ARM as the architecture, in each case as a 32-bit or 64-bit variant. The decoder (testrec) could also be built on other systems with a C compiler, e.g., Raspberry-Pi or DSPs. The storage and computing requirements mainly depend on the size of the acoustic model, the lexicon, and the language model. It is expected to be in the range of tens to hundreds of megabytes of memory.

Configurator (testbld) supports:

- Acoustic models (Kaldi mono-, tri-phone, dLabPro mono-phone).
- Language model creation (text corpus).
- Language models (ARPA, FST).
- Forced alignment configurations.
- Acoustic model adaptation.
- Parameter values modification without recompiling the configuration.
- Sub-word modeling.
- Word classes in the language model.

Recognizer (testrec) supports:

- Folders, files, and file lists as an input.
- Timestamps with confidence values in the output.
- Evaluation and conversion scripts.

3.2 SotA Technologies for Upper Sorbian ASR Tools

3.2.1 Robust Named Entity Recognition (pre-, post-processing)

Currently, the named entity recognition (NER) is solved with a combination of a simple dictionary and rule-based matching. The rules are defined as Finite-State-Transducers (FST). The FST definitions can be seamlessly used for NER parsing as well as for world-class grammar creation.

Other possible approaches, such as machine learning or deep learning are less feasible in the case of under-resourced languages, where for the training an annotated textual corpus is required, if not for training, at least for fine-tuning of Large Language models (LLMs).

There are studies [11] showing that low-resourced languages can benefit from closely related languages and their models on the NER downstream task.

In [12], the authors present a development of a cross-lingual name tagging and linking framework for 282 languages that exist in Wikipedia.
Such a database can be used for fine-tuning the Upper Sorbian language of a BERT-NER model.

Other frameworks and libraries that can be used for NER and word-class modeling, considering the existence of training/tuning data or pre-trained models:

- SpaCy[7] is an open-source natural language processing (NLP) library in Python that provides efficient tools for various NLP tasks, including NER for identifying and classifying entities such as names of people, organizations, locations, dates, and other predefined categories in text.

[7] https://spacy.io

- NLTK[8] is a comprehensive library for natural language processing in Python. While it provides tools for many NLP tasks, it also includes modules for named entity recognition.
- Stanford NER[9] was developed by the Stanford Natural Language Processing Group; Stanford NER is a Java implementation that allows users to perform named entity recognition using pre-trained models. It supports multiple languages.
- AllenNLP[10] is a deep learning library for natural language processing tasks. It provides pre-built models and components for various NLP tasks, including named entity recognition.
- Transformers (by Hugging Face[11]) library is widely used for natural language processing tasks, leveraging pre-trained transformer-based models. It includes models fine-tuned for NER on various datasets.
- BERT-based Models (e.g., Hugging Face Transformers) have achieved state-of-the-art performance in NLP tasks, including named entity recognition. They offer pre-trained models that can be fine-tuned for specific domains.
- GATE (General Architecture for Text Engineering)[12] is an open-source framework for text processing that supports various NLP tasks, including named entity recognition. It provides a graphical development environment.

3.2.2 Feature Engineering

Improvement of the acoustic model performance by using other features than the dLabPro based. For instance, filter bank features, or Mel-frequency cepstral coefficients (MFCCs), are commonly used for training DNNs in ASR.

3.2.3 Proprietary recIKTS Support for TDNN

Kaldi supports feedforward neural networks and time-delay neural networks (TDNN) for acoustic modeling. DNN-based models in Kaldi can be seamlessly integrated with other components, such as the lattice-based decoder for decoding and generating transcriptions.
Tools for speaker adaptation are also provided, allowing DNN models to adapt to specific speakers or acoustic conditions. Kaldi is designed to be scalable and can handle large-scale ASR tasks using DNNs.
recIKTS will be upgraded to support nnet3 acoustic models trained with Kaldi.

3.3 Technology Updates and Upgrades

Regular technology updates and upgrades can be achieved by organizing the software modules within a git repository:

- Git repository with submodules (open-source modules).
- Submodules forked from the original repositories.
- Docker automatic installation and compilation.
- Self-testing, unit tests.

There will be always a danger that some of the open-source submodules become abandoned and obsolete. In such cases it is advisable to find a replacement, for instance, the original phonetisaurus project was abandoned, however, several other projects were further developed based on the original code.

[8] https://www.nltk.org
[9] https://nlp.stanford.edu/software/CRF-NER.shtml
[10] https://allenai.org/allennlp
[11] https://huggingface.co/docs/transformers/index
[12] https://gate.ac.uk

4 Open-Source Solutions

Overview of some open-source ASR solutions as alternatives to the **recIKTS** toolkit and recognizer.

4.1 Kaldi

Kaldi[13] [13] provides various decoders for different usage and model types (mono-, tri-phone, or chain TDNN). There are offline and online decoders, online decoders process the frames in real time as they become available.

The program *online2-wav-gmm-latgen-faster* is currently the primary example program for the GMM-based online-decoding setup. It reads in whole wave files but internally it processes them chunk by chunk with no dependency on the future.

Some main points from the Kaldi documentation[14] cover various aspects of online decoding, including adaptation methods, model types, language models, and practical examples for using and testing online decoding setups.

1. Adaptation in online decoding:
   - The standard adaptation method for ASR is feature-space Maximum Likelihood Linear Regression (fMLLR).
   - fMLLR involves an affine transform of features, and its estimation is done periodically due to lattice posterior computation.
   - Configuration variables determine when to re-estimate fMLLR, typically after specific time intervals.

2. Use of multiple models in GMM-Based online decoding:
   - Up to three models can be supplied: a speaker-independent model, a speaker-adapted model using fMLLR, and a discriminatively trained version of the speaker-adapted model.
   - Maximum Likelihood estimated models are preferred for adaptation parameters.

3. Neural net-based online decoding with iVectors:
   - The recommended setup involves using neural networks with un-adapted features and iVectors representing speaker properties.
   - iVectors are estimated in a left-to-right manner using Maximum Likelihood with Gaussian Mixture Models.

4. Example for using already-built online-nnet2 models:
   - Instructions are provided for downloading pre-built online-nnet2 models and evaluating them on custom data.
   - The process involves downloading specific directories, extracting archives, and modifying pathnames in config files.

5. Using your language model with existing online-nnet2 models:
   - Users can incorporate their language models into existing online-nnet2 models.
   - Steps include building an ARPA format language model, compiling it into WFST format, and updating the decoding graph.

6. Using a different vocabulary with existing online-nnet2 models:
   - Instructions are provided for changing the vocabulary of existing models by creating a new pronunciation lexicon and updating the lang directory.

[13] https://kaldi-asr.org
[14] https://kaldi-asr.org/doc/online_decoding.html

7. Online decoding with nnet3 models:
    - Online decoding with nnet3 models is like nnet2 models, but with some limitations on the model types supported.
    - Considerations for recurrent models are discussed, and examples of scripts and downloadable models are mentioned.

8. TCP server for nnet3 online decoding:
    - The program *online2-tcp-nnet3-decode-faster* is introduced for running a TCP server for nnet3 online decoding.
    - Usage instructions, options, and considerations for testing with *netcat* and *sox* are provided.

Kaldi is a powerful speech framework that is intended to be used by experienced users with a deep understanding of speech technologies. Configuring training and using existing trained models is not straightforward and involves using a collection of different tools (such as OpenFst[15], OpenNgram[16], IRSTLM[17], SRILM[18], KenLM[19], phonetisaurus[20], and many others).
Trained models that do not need to be updated or adapted can be easily incorporated into other frameworks like the one presented in the next section. Python wrapper PyKaldi[21] is also available providing functions to read, write, inspect, manipulate, or visualize Kaldi and OpenFst objects in Python. It includes Python wrappers for most functions and methods that are part of the public APIs of Kaldi and OpenFst C++ libraries.

4.2 Alphacephei VOSK

Vosk[22] is an offline open-source speech recognition toolkit. It enables speech recognition for more than 20 world languages and dialects. Vosk accepts Kaldi-trained models that are small while providing LVCSR transcription, quick response with streaming API, reconfigurable vocabulary, and speaker identification. API bindings are implemented for various programming languages like Python, Java, Node.JS, C#, C++, Rust, Go, and others.
Vosk can be used for speech recognition for chatbots, smart home appliances, and virtual assistants. It can also create subtitles for movies and transcription for lectures and interviews.
Vosk scales from small devices like Raspberry Pi or Android smartphones to big clusters.
The toolkit does not provide its tools for model training and adaptation, the Kaldi framework is used for vocabulary and language model adaptation.

5 End-To-End Transfer Learning

We consider the feasibility of employing the latest E2E OSS frameworks for transfer learning in Upper Sorbian or other under-resourced languages. We give an overview of recent relevant open-source E2E frameworks that might be employed for fine-tuning with Upper Sorbian speech resources.
Some of them are abandoned and obsolete (Coqui, Deepspeech2, or similar) others are supported by the tech giants (Google, Meta, OpenAI) providing models trained on an unparalleled amount of speech data in many languages. There are many E2E speech toolkits and frameworks developed in academic and research institutions that allow interoperability with already available pre-trained models.

[15] https://www.openfst.org/
[16] https://www.openfst.org/twiki/bin/view/GRM/NGramLibrary
[17] https://github.com/irstlm-team/irstlm
[18] http://www.speech.sri.com/projects/srilm/
[19] https://github.com/kpu/kenlm
[20] https://github.com/AdolfVonKleist/Phonetisaurus
[21] https://github.com/pykaldi/pykaldi
[22] https://github.com/alphacep/vosk-api

There are many ongoing E2E ASR projects, many of which are one-language centric (such as PaddleSpeech[23] and FunASR[24] for Chinese) here we include the most influential ones.

5.1 Coqui (abandoned)

We include this framework only because it has a model for Upper Sorbian trained on the Common Voice data. The speech-to-text framework is no longer maintained[25]. The HSB-trained model is available here https://github.com/coqui-ai/STT-models/tree/main/upper-sorbian/itml/v0.1.0.

5.2 Nvidia NeMO

NVIDIA NeMo[26] is a conversational AI toolkit built for researchers working on automatic speech recognition (ASR), text-to-speech synthesis (TTS), large language models (LLMs), and natural language processing (NLP). The primary objective of NeMo is to help researchers from industry and academia reuse prior work (code and pre-trained models) and make it easier to create new conversational AI models.

State-of-the-art pre-trained NeMo models are freely available on HuggingFace Hub and NVIDIA NGC. These models can be used to transcribe audio, synthesize speech, or translate text in just a few lines of code. The project is current, and it has the Apache-2.0 license.

5.3 NVIDIA OpenSeq2Seq

OpenSeq2Seq[27] allows effective exploration of various sequence-to-sequence models. The efficiency is achieved by fully supporting distributed and mixed-precision training. OpenSeq2Seq is built using TensorFlow and provides all the necessary building blocks for training encoder-decoder models for neural machine translation, automatic speech recognition, speech synthesis, and language modeling. The project is stalled, the last update 5 years ago, and it has the Apache-2.0 license.

5.4 Speechbrain

The SpeechBrain[28] project aims to build a novel speech toolkit fully based on PyTorch. With SpeechBrain users can easily create speech processing systems, ranging from speech recognition (both HMM/DNN and end-to-end), speaker recognition, speech enhancement, speech separation, multi-microphone speech processing, and many others.
The toolkit provides both training from scratch and fine-tuning of pre-trained models such as Whisper, Wav2Vec2, WavLM, Hubert, GPT3, Llama2, and beyond. The models on HuggingFace can be easily plugged in and fine-tuned. The project is current, and it has the Apache-2.0 license.

5.5 Mozilla DeepSpeech

DeepSpeech[29] is an open-source Speech-To-Text engine, using a model trained by machine learning techniques based on Baidu's Deep Speech research paper [14]. Project DeepSpeech uses Google's TensorFlow to make the implementation easier. The project is stalled, the last update 5 years ago, and it has the MPL-2.0 license.

[23] https://github.com/PaddlePaddle/PaddleSpeech
[24] https://github.com/alibaba-damo-academy/FunASR
[25] https://github.com/coqui-ai/STT
[26] https://github.com/NVIDIA/NeMo
[27] https://github.com/NVIDIA/OpenSeq2Seq
[28] https://github.com/speechbrain/speechbrain
[29] https://github.com/mozilla/DeepSpeech

5.6 Tensorflow - Lingvo

Lingvo[30] is a framework for building neural networks in Tensorflow (Google), particularly sequence models. The extensive list of related publications is given on the project repository[31]. The project is current, and it has the Apache-2.0 license.

5.7 TensorflowASR

TensorFlowASR[32] implements some automatic speech recognition architectures such as DeepSpeech2, Jasper, RNN Transducer, ContextNet, Conformer, etc. These models can be converted to TFLite to reduce memory and computation for deployment. The project is stalled, the latest release was 2 years ago, and it has the Apache-2.0 license.

5.8 OpenSpeech

OpenSpeech[33] is a framework for making end-to-end speech recognizers. E2E ASR is a single integrated approach with a much simpler training pipeline with models that operate at low audio frame rates. This reduces the training time, and decoding time, and allows joint optimization with downstream processing such as natural language understanding.
Many E2E speech recognition-related open-source frameworks are based on basic PyTorch or Tensorflow, it is very difficult to use various functions such as mixed-precision, multi-node training, and TPU training, etc. Openspeech is a speech recognition framework that introduced PyTorch-Lightning and Hydra for easy use of these advanced features. The project is stale, the last release was 2 years ago and it has an unknown license (MIT).

5.9 Athena

Athena[34] is an open-source implementation of an end-to-end speech processing engine. Our vision is to empower both industrial application and academic research on end-to-end models for speech processing. To make speech processing available to everyone, we're also releasing example implementations and recipes on some open-source datasets for various tasks (ASR, TTS, Voice Conversion, Speaker Recognition, etc). The project is stalled, the last update 2 years ago, and it has the Apache-2.0 license.

5.10 Espresso

Espresso[35] is an open-source, modular, extensible end-to-end neural automatic speech recognition (ASR) toolkit based on the deep learning library PyTorch and the popular neural machine translation toolkit fairseq. Espresso supports distributed training across GPUs and computing nodes and features various decoding approaches commonly employed in ASR, including look-ahead word-based language model fusion, for which a fast, parallelized decoder is implemented. The project is stalled, the last update was 6 months ago, and it has the MIT license.

5.11 openAI Whisper

The Whisper[36] approach was proposed in the paper "Robust Speech Recognition via Large-Scale Weak Supervision" by Alec Radford et al. [9].

---

[30] https://github.com/tensorflow/lingvo
[31] https://github.com/tensorflow/lingvo/blob/master/PUBLICATIONS.md#speech-recognition
[32] https://github.com/TensorSpeech/TensorFlowASR
[33] https://github.com/openspeech-team/openspeech
[34] https://github.com/athena-team/athena
[35] https://github.com/freewym/espresso
[36] https://github.com/openai/whisper

The authors studied the capabilities of speech processing systems trained simply to predict large amounts of transcripts of audio on the internet. When scaled to 680 thousand hours of multilingual and multitask supervision, the resulting models generalize well to standard benchmarks and are often competitive with prior fully supervised results but in a zero-shot transfer setting without the need for any finetuning.

The provided model usually performs well without requiring any finetuning. The inference is currently only possible with pre-segmented audio (max 30s segments).

There is no direct support for Upper Sorbian, however fine-tuning on a new language is possible and projects supporting high-performance inference in C++ implementation exist (e.g., whisper.cpp[37]). The project is current, and it has an MIT license.

5.12 Facebook AI Research (FAIR)

FAIR stands for Facebook AI Research, which is the artificial intelligence research division at Facebook (now Meta) known for its contributions to the field of artificial intelligence and machine learning. FAIR is committed to open research, and many of its research papers, projects, and code implementations are made publicly available.

5.12.1 Flashlight

Flashlight[38] is a new open-source machine learning (ML) library, written entirely in C++, that was built by FAIR to power groundbreaking research by enabling teams to modify deep rapidly and easily and ML frameworks to better fit their needs.

Flashlight's ASR application (formerly the wav2letter project) provides training and inference capabilities for end-to-end speech recognition systems. The project is current, and it has an MIT license.

5.12.2 Wav2Vec

Wav2Vec[39] refers to a family of models developed by Facebook AI Research (FAIR) for self-supervised learning of speech representations. The primary goal of Wav2Vec models is to learn meaningful representations directly from raw audio data without requiring labeled transcriptions.

These models have been successful in various speech-related tasks, including automatic speech recognition (ASR).

- Wav2Vec 1.0 was introduced with the objective of pretraining a deep neural network to predict the quantized speech signal from non-overlapping context windows of the waveform. The model consists of a convolutional neural network (CNN) that processes the raw audio waveform.
- Wav2Vec 2.0 [3] improves upon the original model by introducing a new self-supervised learning task known as "contrastive predictive coding" (CPC). It uses a transformer-based architecture and learns representations by predicting future context within the sequence of features extracted from the audio signal.

The advantage of Wav2Vec models is their ability to perform transfer learning. Pretrained models can be fine-tuned on specific downstream tasks with smaller amounts of labeled data. Wav2Vec models are typically implemented using PyTorch, and the pre-trained models and code are made available by Facebook AI Research[40]. The project is stalled, the latest release was 2 years ago, and it has the MIT license.

[37] https://github.com/ggerganov/whisper.cpp
[38] https://github.com/flashlight/flashlight
[39] https://github.com/pytorch/fairseq/tree/main/examples/wav2vec
[40] https://huggingface.co/blog/fine-tune-xlsr-wav2vec2

5.12.3 Fairseq

Fairseq[41] is a sequence-to-sequence learning toolkit developed by Facebook AI Research (FAIR). It is primarily used for training and evaluating neural network models in natural language processing (NLP) tasks. Fairseq supports various tasks such as machine translation, summarization, and speech recognition.

5.12.3.1 The Massively Multilingual Speech (MMS)

Fairseq MMS[42] project expands speech technology from about 100 languages to over 1000 by building a single multilingual speech recognition model supporting over 1100 languages. Upper Sorbian as language in pre-trained model and fine-tuning is supported. Since MMS uses the Connectionist Temporal Classification (CTC) model, the accuracy can be further improved by running beam search decoding using a language model (such as KenLM)[43]. The project is current, and it has an MIT license. Examples of direct recognition on recording are given in the Annex of this document.

5.13 HuggingFace

Hugging Face is a company and a platform known for its contributions to natural language processing (NLP) and machine learning. They provide a variety of tools and resources, and one of their notable contributions is the Transformers library. The project is current, and it has the Apache-2.0 license.

5.13.1 Transformers

Transformers[44] is an open-source library that offers pre-trained models and various utilities for natural language processing tasks, including text generation, translation, summarization, and sentiment analysis. It supports a wide range of state-of-the-art models, such as BERT, GPT, RoBERTa, and more.

Key Features:

1. Pre-trained models, a large collection of pre-trained models for a variety of NLP tasks.
2. Model hub, users can share, discover, and use pre-trained models.
3. Easy integration into existing machine learning pipelines.
4. Community support, researchers, and developers contribute to the library and share their models.

5.13.2 Datasets

Hugging Face **Datasets[45]** is another library that provides easy access to a vast collection of datasets for NLP tasks. It simplifies the process of loading and preprocessing datasets for training and evaluation.

5.13.3 Tokenizers

Tokenizers[46] is a library developed by Hugging Face for fast and efficient tokenization of text. It is used internally in the Transformers library but can be used independently as well.

5.13.4 Website

The **Hugging Face website[47]** serves as a hub for accessing pre-trained models through the Model Hub, exploring datasets, and using various tools and resources related to natural language processing. Hugging Face has become a central hub for NLP practitioners, offering state-of-the-art models, datasets, and tools that facilitate research and development in the field.

---

[41] https://github.com/pytorch/fairseq
[42] https://github.com/facebookresearch/fairseq/tree/main/examples/mms
[43] https://huggingface.co/blog/mms_adapters
[44] https://github.com/huggingface/transformers
[45] https://github.com/huggingface/datasets
[46] https://github.com/huggingface/tokenizers
[47] https://huggingface.co

5.14 ESPnet

ESPnet[48] [15] is an end-to-end speech processing toolkit covering end-to-end speech recognition, text-to-speech, speech translation, speech enhancement, speaker diarization, and spoken language understanding.

ESPnet uses Pytorch[49] as a deep learning engine and follows Kaldi-style data processing, feature extraction/format, and recipes to provide a complete setup for various speech processing experiments. The project is current, and it has the Apache-2.0 license.

Key Features:

1. End-to-end ASR, covering both training and inference stages.
2. Neural network architectures, including hybrid systems combining neural networks and Hidden Markov Models (HMMs).
3. Flexible configuration of different components of ASR systems, allowing users to experiment with different model architectures and training strategies.
4. Multi-language support for diverse speech processing applications.
5. Extensive documentation, including tutorials and examples, to help users get started and understand the toolkit's capabilities.

Components of ESPnet:

1. e2e-ASR: The end-to-end ASR recipe includes tools for training end-to-end ASR systems using various neural network architectures.
2. e2e-TTS: end-to-end Text-to-Speech (TTS) systems, covering both training and synthesis.
3. The Speaker State Transfer (SST) toolkit focuses on speaker adaptation and enables training models with a small amount of adaptation data.

[48] https://github.com/espnet/espnet/
[49] https://pytorch.org/

6 Conclusions

6.1 Summary

The document outlines the current state of the Upper Sorbian speech technologies with a focus on speech recognition and the available resources. It presents the state of the art in this field comparing the classic, hybrid, and end-to-end approaches (Section 1.1.6).
In the context of the Upper Sorbian, there is a question of which approach would most efficiently provide a practically usable speech recognizer that could be employed in real-world applications, helping speakers in their daily or professional tasks.
Each approach has its advantages and disadvantages considering the required speech and computational resources for the training also for the usage.

6.1.1 Classic and Hybrid Systems Rationale

The proprietary software, the recIKTS speech recognizer using the classic approach, has the advantage that the observed issues can be corrected in any component, acoustic or language modeling. For instance, adding new graphemes, and new words with their pronunciation variants in the lexicon and the language model. Each of the modules can be optimized separately and the system can be improved incrementally when more data becomes available, as it was demonstrated in the previous HBS-I-III projects.

Contrary, systematic problems occurring with an E2E system are much more difficult to correct without whole re-training or fine-tuning, where annotated speech is required, the desired performance or solving the issues cannot be guaranteed. For instance, hallucinations in E2E speech recognition systems occur when the system generates incorrect transcriptions or recognizes words or sounds that do not exist in the input audio. These errors can occur due to various reasons, including noise in the audio data, variability in speakers' accents, or limitations in the training data. Another advantage is that the classic and hybrid approaches (such as recIKTS, and Kaldi) use less computational and storage resources compared to E2E. E2E systems usually need GPU units for training and inference which could be a major limiting factor in their deployment in some use cases.

As an illustration the fairseq MMS MMS-1B:FL102 model covering 102 languages trained on the Fleurs [16] dataset has a size of 4.5 GB, while the largest MMS-1B-all around 14GB, which limits its application to systems that have large memory and storage and GPUs.

6.1.2 End-to-End Systems Rationale

The main advantage of the E2E systems is that they employ the "black-box" approach when data amount requirements are fulfilled. They might be more robust in the general-purpose task than the classic and hybrid systems since they can produce the grapheme sequence of unseen words that are understandable and in most cases are proper words (see Section 8. Annex for some examples). However, for acceptable performance language modeling is still required. Opposite, general-purpose domain recognition with classic and hybrid systems requires the collection of representative textual data for the statistical language modeling with limited vocabulary where out-of-vocabulary words can still occur. E2E approaches are based on the neural network architecture, enabling parallelization and task distribution, and when provided with the proper computational resources they can process a much larger amount of speech data in the training and inference compared to the classic and hybrid approaches.

Multilingual E2E systems are heavily biased and provide the best performance with languages that occupied a major portion of the training data while the recognition of minority languages has at best mediocre performance. E2E models that are trained on similar languages can be fine-tuned to a new language of the same family and provide improved performance compared to the case when the E2E is trained from scratch on a new language with less speech data.

Having more data for fine-tuning of pre-trained models is beneficial, but it does not guarantee that the performance will be improved proportionally with the data amount incrementation like in the classic and hybrid systems.

6.2 Recommendations

Considering the current development of the E2E and the hybrid ASR systems, and taking into consideration the low-resource nature of the available electronic speech resources, the main strategy for further development of the Upper Sorbian speech technologies (specifically speech recognition) should consider the following general recommendations:

6.2.1 Speech Corpus (Acoustic Modeling)

- Continuous collection of new speech resources employing available speech technologies in Upper Sorbian and semi-supervised transcription of audio recordings.
  This is not of the highest priority since the currently available data is enough for reliable training of GMM/HMM and DNN/HMM models.
- As soon as there is a significant amount of new data collected, the acoustic models can be updated.

6.2.2 Lexicon (Pronunciation Modeling)

- Currently, the phoneme inventory and the pronunciation rules are well-defined and no further significant improvements can be expected by reducing or expanding the definitions.

6.2.3 Textual Corpus (Language Modeling)

- Continuous effort to collect textual content by contacting and engaging relevant organizations where such resources can be obtained (news portals, government documents, education materials). Since there are very sparse textual data for language modeling, this should be of the highest importance.
- Enhancing the textual corpus with Part-Of-Speech (POS) tags using SotA NER.
- Selection of POS tags for word-class modeling.
- Language modeling will be necessary, regardless of the used technology (N-gram, Recurrent Neural Networks – RNNs, etc.).

6.2.4 Speech Technology

- Upgrading the proprietary ASR system to handle other speech features and hybrid TDNN models (DNN/HMM) will significantly improve the recognition performance.

6.2.5 End-to-End Systems

- Investigate the feasibility of practical employment of the recommended E2E frameworks available through *HuggingFace, such as OpenAI Whisper* and *fairseq MMS*.
- Establishing and continuously updating experimental frameworks based on the SotA E2E ASR systems for transfer learning using existing speech resources.
- Evaluation of adopted E2E systems against the classic or hybrid ASR for general (Jitsi) and domain-specific use cases (Misa, Smart Lamp, etc.)

6.3 Future Work

- Further improvement of the recIKTS ASR system by more robust features and introducing DNN/HMM models.
- Support with textual data collection and validation for reliable language modeling.
- Investigating the feasibility of practical employment and evaluation of suitable E2E frameworks.
- Dissemination of the knowledge on conferences and among relevant stakeholders (the Foundation, Institute, newspapers, media …).

6.4 Technology and Applications

The table presents some possible applications and the available technologies, both open-source and proprietary.

Application	Lingufino (Alphaspeech µ)	dlabpro on ARM (Raspberry)	recikts (proprietary)	Kaldi (OSS)	whisper (OSS)	Facebook (OSS)	Microsoft Translator (proprietary) only text HSB	Linguwerk Alphaspeech (proprietary)
smart home	?	x	x	x	?	?	no	x
toys	x	?	x	?	no	no	no	no
dictation system	no	no	x	x	?	?	no	x
simultaneous translation	no	no	x	x	x	x	x	x
offline transcription	no	no	x	x	x	x	no	x

The applications have different requirements regarding the vocabulary size and intended usage.

- Smart Home usually covers a narrow domain with a handful of devices or services to manage. The devices are simple with few functionalities, some also can set timers and alarms and numerical properties (temperature, percentage). The vocabulary is rather limited, and the speech recognizer should allow freely constructed utterances and not simple command-and-control. Context-free grammars and statistical language models can be employed separately or combined.
- Toys, similarly, to the Smart Home use case, have a limited vocabulary, and specific for this application is that the ASR should be able to recognize children's speech reliably. It is important to have an ASR that works offline due to privacy concerns and personal data protection. The system must have very low computational and storage requirements and to able to run on SoC or embedded hardware.
- Dictation system, here a large vocabulary continuous speech recognition system is necessary that could run offline and online in real-time. Since it is for personal use, it can be adapted to the user's speech and audio conditions (microphones, headphones, room acoustics) improving the recognition performance. This system needs to be flexible so that the user can add missing or specific words to the vocabulary and the language model. The language model can be trained on the existing user's textual documents (e.g., journalist's articles) to narrow the domain and capture the writing/speaking style.
- Offline transcription of speech recordings also involves the LVCSR system, the intended usage is for transcription of audio data in media archives. The ASR system is not required to work in real-time, it should be speaker-independent with a general-purpose domain language model. The correctness of the transcriptions will be checked by a human expert (native speaker of the language) and the final transcriptions can be included to improve the language and lexicon models.
- Simultaneous translation involves two speech interfaces in different languages, ASR and Text-To-Speech systems in the corresponding languages. The ASR should be speaker-independent LVCSR with a general-purpose language model and open vocabulary and must be running in real-time with minimal latency.

7 References

1. Kraljevski, I., Tschöpe, C., & Wolff, M. (2023). Limits and prospects of big data and small data approaches in AI applications. KI-Kritik/AI Critique Volume 4, 115 (https://library.oapen.org/bitstream/handle/20.500.12657/76279/1/9783839457320.pdf#page=116).
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (https://arxiv.org/pdf/1810.04805.pdf).
3. Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). Wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems, 33, 12449-12460 (https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf).
4. Georgila, K., Leuski, A., Yanov, V., & Traum, D. (2020, May). Evaluation of off-the-shelf speech recognizers across diverse dialogue domains. In *Proceedings 32ft the Twelfth Language Resources and Evaluation Conference* (pp. 6469-6476) (https://aclanthology.org/2020.lrec-1.797.pdf).
5. Li, J. (2022). Recent advances in end-to-end automatic speech recognition. APSIPA Transactions on Signal and Information Processing, 11(1) (https://arxiv.org/abs/2111.01690v2).
6. Prabhavalkar, R., Hori, T., Sainath, T. N., Schlüter, R., & Watanabe, S. (2023). End-to-End Speech Recognition: A Survey. arXiv preprint arXiv:2303.03329 (https://arxiv.org/abs/2303.03329).
7. Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., … & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 8, 726-742 (https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00343/96484).
8. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., … & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901 (https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf).
9. Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023, July). Robust speech recognition via large-scale weak supervision. In International Conference on Machine Learning (pp. 28492-28518). PMLR (https://proceedings.mlr.press/v202/radford23a/radford23a.pdf).
10. Pratap, V., Tjandra, A., Shi, B., Tomasello, P., Babu, A., Kundu, S., … & Auli, M. (2023). Scaling speech technology to 1,000+ languages. arXiv preprint arXiv:2305.13516 (https://arxiv.org/pdf/2310.17448).
11. Torge, S., Politov, A., Lehmann, C., Saffar, B., & Tao, Z. (2023, May). Named Entity Recognition for Low-Resource Languages-Profiting from Language Families. In Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023) (pp. 1-10) (https://aclanthology.org/2023.bsnlp-1.1.pdf).
12. Pan, X., Zhang, B., May, J., Nothman, J., Knight, K., & Ji, H. (2017, July). Cross-lingual name tagging and linking for 282 languages. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1946-1958) (https://aclanthology.org/P17-1178.pdf).
13. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Vesely, K. (2011). The Kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding (No. CONF). IEEE Signal Processing Society (https://infoscience.epfl.ch/record/192584/files/Povey_ASRU2011_2011.pdf).
14. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... & Ng, A. Y. (2014). Deep speech: Scaling up end-to-end speech recognition. arXiv preprint (https://arxiv.org/abs/1412.5567).
15. Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., ... & Ochiai, T. (2018). Espnet: End-to-end speech processing toolkit. arXiv preprint arXiv:1804.00015 (https://arxiv.org/pdf/1804.00015).
16. Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., ... & Bapna, A. (2023, January). Fleurs: Few-shot learning evaluation of universal representations of speech. In 2022 IEEE Spoken Language Technology Workshop (SLT) (pp. 798-805). IEEE. (https://arxiv.org/pdf/2205.12446v1.pdf)

8 Annex

Examples of recognition in Upper Sorbian using a trained model of **fairseq MMS**. The recognition results are given with HYPO and the reference transliteration as REF.

```
HYPO: w wanem času powědaše jězu swojim wučomnikam tute přirunanje z njewjetskim kralewstwom je to takaž z mužom kotrežsy na jězbu poda mus zwowa na puć hotołwy swojich wotročkow a přepodaj jim swoje kubła
REF: we wonym času powědaše jezus swojim wučomnikam tute přirunanje z njebjeskim kralestwom je to tak kaž z mužom kotryž so na jězbu poda muž złowa na puč hotowy swojich wotročkow a přepoda jim swoje kubła
---------------------
Processed 1 sentences (200 tokens) in 2.0s 0.51 sentences per second, 102.10 tokens per second)
Word error rate: 40.0000
```

```
HYPO: jednomu da pjeć talentu druhemu dwaj a třećemu jedyn kóžemu po jeho mocach a potom wotpućoła keiž bě pjeć talentow dóstał wotandźenydom a wichowaše z nimi a dybe k tomu pjeć druhich
REF: jednomu da pjeć talentow druhemu dwaj a třećemu jedyn kóždemu po jeho mocach a potom wotpućowa kiž by pjeć talentow dóstał woteńdźe hnydom a wikowaše z nimi a doby k tomu pjeć druhich
---------------------
Processed 1 sentences (182 tokens) in 2.1s 0.47 sentences per second, 85.31 tokens per second)
Word error rate: 27.2727
```

```
HYPO: runje tak doby tón kež bi dwaj dóstał druhej dwaje kž pak bě jedyn dóstał w botenźe zahrjěba jón do zemje a skoha pjenjez swojeho knjeza w podołhem času róče so knjez tych wotročkow a ličbowaše z nimi
REF: runje tak doby tón kiž by dwaj dóstał druhej dwaj kiž pak by jedyn dóstał woteńdźe zarywa jón do zemje a schowa pjenjez swojeho knjeza po dołhim času wróći so knjez tych wotročkow a ličbowaše z nimi
---------------------
Processed 1 sentences (201 tokens) in 2.0s 0.51 sentences per second, 101.53 tokens per second)
Word error rate: 32.4324
```

```
HYPO: tak přistupi tón kež bě pjet talentow dóstał přinese druhich pjet talentow k tomu a praj knježe pjed talentow sym jej přepodał hlej druhich pjedsym přidołbył
REF: tak přistupi tón kiž by pjeć talentow dóstał přinjese druhich pjeć talentow k tomu a praji knježe pjeć talentow sy mi přepodał hlej druhich pjeć sym přidobył
---------------------
Processed 1 sentences (158 tokens) in 2.1s 0.48 sentences per second, 75.21 tokens per second)
Word error rate: 44.4444
```

```
HYPO: knjez prajj jemu dobry a swěrne wotročko dokelž sej był swěrny nad małym postajujće nad norim zastup do radosće swojeho knjezy přistypi potom tejž štón keiž bě dwaj talenta je dóstała rěkny
REF: knjez praji jemu dobry a swěrny wotročko dokelž sy był swěrny nad małym postaju će nad mnohim zastup do radosće swojeho knjeza přistupi potom tež tón kiž by dwaj talentaj dóstał a rjeknje
---------------------
Processed 1 sentences (190 tokens) in 2.0s 0.51 sentences per second, 96.25 tokens per second)
Word error rate: 48.4848
```

```
HYPO: knježe dwaj talentaj sym mi dowěrił leddwaj druhej sym dobył jho knjezpaj jemu dobre a swěrne wotročko doke sej nad małym swěrne postajujće nad mnohim
REF: knježe dwaj talentaj sy mi dowěrił hlej dwaj druhej sym dobył jeho knjez praji jemu dobry a swěrny wotročko dokelž sy nad małym swěrny postaju će nad mnohim
---------------------
Processed 1 sentences (151 tokens) in 2.0s 0.49 sentences per second, 73.94 tokens per second)
Word error rate: 46.4286
```

```
HYPO: zastup do wjesela swojeho knjeza přistupi pak tež štónkež bi jedyn talent dóstała praji knježe wěm zo se kruty človjek hnjeješ hdźež njejcy sył a zběraš hdźež njejcy sypał
REF: zastup do wjesela swojeho knjeza přistupi pak tež tón kiž by jedyn talent dóstał a praji knježe wěm zo sy kruty človjek žněješ hdźež njejsy sył a zběraš hdźež njejsy sypał
---------------------
```

```
HYPO: w bojach so a tuž wotendźek a skowach swój talent dozem lej tumaš štojše twoj ale j ho knjez jemu wotmołwi zwy aleni wotwočko sy wědźał zo dźneju hdźež njejsym sył a zběram hdźež njejsym sypał
REF: bojach so a tuž woteńdźech a schowach swój talent do zemje hlej tu maš štož je twoje ale jeho knjez jemu wotmołwi zły a lěni wotročko sy wědźał zo žněju hdźež njejsym sył a zběram hdźež njejsym sypał
---------------------
Processed 1 sentences (193 tokens) in 2.0s 0.49 sentences per second, 94.89 tokens per second)
Word error rate: 47.3684
```

```
HYPO: hdydźe bě moje pinjezy bankowcam datžměł a ja bych po mojim nawroće swoje zdanju dósta zmiće tohodla wot njeho talent a dajće jón tomu kiž ma dźesać talent
```

REF: ty dźe by moje pjenjezy bankowcam dać měł a ja bych po mojim nawróće swoje zdanje dóstał wzmiće tohodla wot njeho talent a dajće jón tomu kiž ma dźesać talentow
---

HYPO: dokelž kóžde mu kišma budźe daty a změje wele čuž pak nima tomu wodmje so tež to štoójšma čisće njełužitno wotročka al wonkornjeje čmy tam budźe płać a křipjenje zubow ewangelij našeho knjeza jezusa christu s
REF: dokelž kóždemu kiž ma budźe daty a změje wjele štóž pak nima tomu wozmje so tež to štož ma čisće njewužitneho wotročka do wonkowneje ćmy tam budźe płać a křipjenje zubow ewangelij našeho knjeza jezusa chrystusa
---
Processed 1 sentences (209 tokens) in 2.1s 0.48 sentences per second, 100.70 tokens per second)
Word error rate: 38.8889

HYPO: pód sněholinki a palčikow tedwelewele lětama bydleše jena rjana krasna holca sneběła z čerwjelnymi ličkama taj bódliwa na jenom rjanom rodźa hdež je bydliwjen krala jena kralowna to bě je dźowłka toreka wona bě je na princesna jednoho dnja pajo ta mać zemrěła a tukeže tón kralše jenu mać za tón dźołku trebał jo sun znowa łoženjł a takje ta sněholinka zasjenu now mamuměłe tapa so sej snjejolinku njejho znjesła hajo hajnike prawjiła dowhjedź tón snjholinku won dolesa a zaceliła hajnik na přichodny ranje šoł ze snjolinku do wubokoléša celedalokonuc doléša alnjejhodźł zacelił unjho jenože jeno sornika celił a je tón wutrobutej kralowne dones a praju sym wšo zdokonjała snjolinka pakjašła daloko do wubokols přez hory a dohpřez sydom hory přez sydom dołha a jo namakało jenu kěžkwjetej kěsce jebje něchtó doma wonaje kłapała zwoknane spohladała ale tam něchtó njebě snjolinka woběk tak mučna a wódna a lačna a jako widźeše zo by wšitko přihotowane wupis jedno nopažka wod z je stamnotaloka klěbužka a lene so do najwjetšowóžka kižbě namakałokcele prawy wodpalčika to bě wšoł od palčikow a nadobo su woni přišla jen dwajtř štri pjeć šćsydom palčikow jen wjetši hadźtón druh tónajwjetši hdźeže prieni do dómskow a widźeše zo bě ho stolc přesunen a wón wótcłołaše študajo no mojim stólcu sejdźał a tón sredźanski hižołhołoše študajo z mojo nopažka pił a přichodnywwoše študajhw mój klébožk řěrd a nadobo so něchtu zadźiła študa we mojim wóžku spinka a nadobo snjeholinka wodtuči a so dźiłaše zoběše sreč palčikow něcseonikhašachu štułowna ja ahdež b wona řišć ssowjedała woni reknjěchu zo móže wona polanich hłostaća ta njejowjac přišła a něcje ta snjejolinka popalčikow bódliła ajim stajne pomała a pomazki mazała a dómčka jenje wurědźiła a tak su wšitce wjeseło buli sponcynejene nowo wulke wóžkow za snjholinkey natwarilatamajana pepleajakpil samšl to su ni sami šili jakubjpoahajepalčik atikaaadałanotao hladawa ta sama pomała ta wušiwała
REF: wot sněholinki ha palčikow před wěle wěle lětami bydleše jena rjana krasna holca sněhběła z čerwenymi ličkami ta jo bódliła na jenom rjanom hrode hde jo bydlił jen kral ha jena kralowna to bě jeje dowka to rěka wona bě jena princesna jedno dnja pa jo ta mać zemrěła ha dókejš jo tón kral šće jenu mać za tón dowku trebał jo so wón znowa woěnił a tak jo ta sněholinka zas jenu now mamu měła ta pa so z tej sněholinku nejo znesła ha jo hajnikej prajła doweć tón sněholinku won do lěsa a zatřěl ju hajnik jo na přichodne ranje šoł zes sněholinku do hłuboko lěsa cyle daloko nutř do lěsa ale nejo ju zatřělił wón jo jeno jeno sornika třělił a je tón wutrobu tej kralowne dones a prajuł sym šo zdokonjał sněholinka pak jo šła daloko do hłuboko lěsa přez hory a doły přez sydom hory přez sydom doły a jo namakała jenu chěku we tej chěce nebě nichtó doma wona je chłapała z wókna nutř pohladała ale tam nichtó nebě sněholinka pa bě tak mučna a hłódna a lačna a jako wideše zo bě šitko přihotowane wupi z jedno nópaška wodu zjě z tamno talerka chlěbušk a lehny so do najwetšo łóka ki bě namakała wot palčika cyle prawe wot palčika to bě šo wot palčikow ha nadobo su woni přišli jen dwaj tři štyri peć šěšć sydom palčikow jen wetši hač tón druj tón najwetši děše prěni do dómsko a wideše zo bě jeho stólc přesunene a wón wótře wołaše štó da jo na mojim stólcu sejdał a tón srejdanski hio wołaše štó da jo z mojo nópaška pił a přichodne wołaše štó da jo mój chlěbušk zjěd a nadobo so nichtó zadiwa štó da we mojim łóku spinka a nadobo sněholinka wotući a so diwaše zo běše srejć palčikow něk so woni prašachu štó wona jo ha hdy bě wona šitko powědała woni reknychu zo móe wona pola nich wostać a ta lóza ćeta ta nejo jac přišła ha něk jo ta sněholinka po palčikow bódliła ha jim stajne pómała ha pómazki mazała ha dómčk rejne wurediła ha tak su šitcy wesoło bóli pon su wone jene nowo wulke łóko za sněholinku natwarili haj ha jen jene jen poslešćo ha jen zawk kupili sami šili to su woni sami šili jakubje su te šě pómali haj šě palčiki ha ta sněholinka jo hladała no ta jo hladała ta jo sama pómała ta jo wušiwała
---
Processed 1 sentences (1943 tokens) in 19.3s 0.05 sentences per second, 100.89 tokens per second)
Word error rate: 66.4234