TOWARDS AUTOMATIC SPEECH RECOGNITION WITHOUT PRONUNCIATION DICTIONARY, TRANSCRIBED SPEECH AND TEXT RESOURCES IN THE TARGET LANGUAGE USING CROSS-LINGUAL WORD-TO-PHONEME ALIGNMENT

Felix Stahlberg*  Tim Schlippe*  Stephan Vogel†  Tanja Schultz*

* Cognitive Systems Lab, Karlsruhe Institute of Technology (KIT), Germany
† Qatar Computing Research Institute, Qatar Foundation, Qatar

ABSTRACT
In this paper we tackle the task of bootstrapping an Automatic Speech Recognition system without an a priori given language model, a pronunciation dictionary, or transcribed speech data for the target language Slovene – only untranscribed speech and translations to other resource-rich source languages of what was said are available. Therefore, our approach is highly relevant for under-resourced and non-written languages. First, we borrow acoustic models from a strongly related language (Croatian) and apply a Croatian phoneme recognizer to the Slovene speech. Second, we segment the recognized phoneme strings into word units using cross-lingual word-to-phoneme alignment. Third, we compensate for phoneme recognition and alignment errors in the segmented phoneme sequences and aggregate the resulting phoneme sequence segments in a pronunciation dictionary for Slovene. Orthographic representations are generated using a Croatian phoneme-to-grapheme model. Finally, we use the resulting dictionary and the Croatian acoustic models to recognize Slovene. Our best recognizer achieves a Character Error Rate of 52% on the BMED corpus.

Index Terms— pronunciation dictionary, non-written languages, word-to-phoneme alignment, language discovery, zero-resource

1. INTRODUCTION
Nowadays the majority of state-of-the-art Automatic Speech Recognition (ASR) systems heavily relies on large amounts of data which is necessary to train such systems. Transcribed speech resources, large amounts of text for language modeling, and pronunciation dictionaries are of great importance to create such systems. Authors in [1] estimate that transcription of 1 hour conversational speech data can take up to 20 hours of effort. Therefore, in recent years ASR research has shifted its focus to low- and under-resourced settings [2] to address less prevalent languages as well, e.g. by exploring new ways to collect data [3, 4, 5, 6], using grapheme-based approaches [7], or sharing information across languages [8]. Zero-resource (ZR) ASR goes one step further and even refrains from assuming the availability of a pronunciation dictionary, transcribed audio data, or a language model (LM) in the target language [9] – only untranscribed audio data are available in the target language. Language discovery for ZR ASR can be subdivided into two steps: 1. Phonetic discovery aims to find subword units suitable for acoustic modeling. 2. Lexical discovery identifies word-like structures and phrases based on phonetic transcriptions of continuous target language speech. Word segmentation describes the task of segmenting the phonetic target language transcriptions into word-like units and thus is a form of lexical discovery. Among a variety of monolingual approaches to word segmentation [9, 10, 11, 12], recent studies have shown [13, 14, 15, 16], that adding written translations in a resource-rich source language can help the word segmentation process.

In our scenario a human translator produces utterances in the target language (Slovene) from prompts in one or many resource-rich source languages (German, English, Croatian) as illustrated in Fig. 1. Since we assume the source languages to be well-studied, we have access to a phoneme recognizer trained on speech data in a related language (Croatian). Since Croatian is phonetically closely related with Slovene, we simply use this phoneme recognizer for Slovene instead of applying phonetic discovery methods that learn from target language speech. In this paper, we combine recent findings in language discovery research, a novel string clustering method, and resources from other resource-rich languages to build an ASR system for the target language Slovene without an a priori given LM, a pronunciation dictionary, or transcribed speech data in the target language – only untranscribed speech in Slovene and translations to other resource-rich source languages of what was said is available. Our approach is in particular intended for non-written languages and dialects since no written form is required for the target language.

To the best of our knowledge, although isolated steps in this process have been studied extensively within restricted experiment settings, the complete pipeline has never been set up before. In particular, research in word segmentation often assumes error-free phonetic transcriptions [12, 9]. In this work, we compensate for alignment and recognition errors.

2. RELATED WORK
Pronunciation dictionaries are used to train speech processing systems by describing the pronunciation of words in manageable units such as phonemes [17]. The production of pronunciation dictionar-

![Fig. 1. Scenario.](image-url)
BMED languages to Slovene given by GIZA++ [23]. Tab. 2 summarizes the word-level IBM-4 perplexities from the recurrences), small vocabularies and low average sentence lengths. Tab. 1 shows high average word frequencies (i.e. frequent word speakers. Each Slovene sentence was read by 3-5 Slovene speakers. We recorded 50 minutes Slovene speech from 5 Slovene native English, German and Slovene in the scope of common medical phrases. The BMED corpus consists of 200 parallel written sentences in Croatian, English and Slovene. The open source tool NanoGong [37] was integrated to provide voice recording functionality via a Java applet. Croatian and Slovene are phonetically and linguistically related. Tab. 3 shows the high similarity of both phoneme sets. Both are South Slavic languages and their orthographies are based on Gaj’s Latin alphabet [38, 39]. This enables us to utilize the Croatian phoneme set, acoustic models, and a phoneme-to-grapheme model for the Slovene language with limited performance degradation.

4. PRONUNCIATION DICTIONARY EXTRACTION

We bootstrap an ASR system for the target language Slovene without a Slovene pronunciation dictionary or transcribed Slovene audio data using resources from Croatian and written translations in Croatian, English, and German. First, we build the pronunciation dictionary for Slovene:

1. Phoneme Recognition: Transform the Slovene speech with a Croatian phoneme recognizer to phoneme sequences.
2. Cross-lingual Word-to-Phoneme Alignment: Align the phoneme sequences to the written translations in Croatian, English, and German. The alignments induce a segmentation of the phoneme sequences in word-like chunks.
3. Phoneme Sequence Clustering: The phoneme sequence segments extracted from the alignments suffer from frequent alignment and phoneme recognition errors. Therefore, group different realizations of the same Slovene word into clusters.

Since our speakers were geographically widely distributed, we developed the web-based recording tool CorpusGong. This tool is designed with major respect to usability and stability so that speakers can record their utterances at home and no supervision is necessary. The user interface (Fig. 2) is available in Croatian, English and Slovene. The open source tool NanoGong [37] was integrated to provide voice recording functionality via a Java applet.

Croatian and Slovene are phonetically and linguistically related. Tab. 3 shows the high similarity of both phoneme sets. Both are South Slavic languages and their orthographies are based on Gaj’s Latin alphabet [38, 39]. This enables us to utilize the Croatian phoneme set, acoustic models, and a phoneme-to-grapheme model for the Slovene language with limited performance degradation.

4. PRONUNCIATION DICTIONARY EXTRACTION

We bootstrap an ASR system for the target language Slovene without a Slovene pronunciation dictionary or transcribed Slovene audio data using resources from Croatian and written translations in Croatian, English, and German. First, we build the pronunciation dictionary for Slovene:

1. Phoneme Recognition: Transform the Slovene speech with a Croatian phoneme recognizer to phoneme sequences.
2. Cross-lingual Word-to-Phoneme Alignment: Align the phoneme sequences to the written translations in Croatian, English, and German. The alignments induce a segmentation of the phoneme sequences in word-like chunks.
3. Phoneme Sequence Clustering: The phoneme sequence segments extracted from the alignments suffer from frequent alignment and phoneme recognition errors. Therefore, group different realizations of the same Slovene word into clusters.

Since our speakers were geographically widely distributed, we developed the web-based recording tool CorpusGong. This tool is designed with major respect to usability and stability so that speakers can record their utterances at home and no supervision is necessary. The user interface (Fig. 2) is available in Croatian, English and Slovene. The open source tool NanoGong [37] was integrated to provide voice recording functionality via a Java applet.

Croatian and Slovene are phonetically and linguistically related. Tab. 3 shows the high similarity of both phoneme sets. Both are South Slavic languages and their orthographies are based on Gaj’s Latin alphabet [38, 39]. This enables us to utilize the Croatian phoneme set, acoustic models, and a phoneme-to-grapheme model for the Slovene language with limited performance degradation.

3. BMED CORPUS

We collected our BMED corpus (Basic Medical Expression Database) to evaluate our methods on short but realistic sentences. The BMED corpus consists of 200 parallel written sentences in Croatian, English, German and Slovene in the scope of common medical phrases. We recorded 50 minutes Slovene speech from 5 Slovene native speakers. Each Slovene sentence was read by 3-5 Slovene speakers. Tab. 1 shows high average word frequencies (i.e. frequent word recurrences), small vocabularies and low average sentence lengths. Tab. 2 summarizes the word-level IBM-4 perplexities from the BMED languages to Slovene given by GIZA++ [23].

<table>
<thead>
<tr>
<th>Source Language</th>
<th>IBM-4 Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Croatian (hr)</td>
<td>3.25</td>
</tr>
<tr>
<td>English (en)</td>
<td>9.13</td>
</tr>
<tr>
<td>German (de)</td>
<td>7.95</td>
</tr>
</tbody>
</table>

Table 2. IBM Model 4 perplexity according GIZA++ [23] on the BMED corpus (Target language: Slovene).

![Fig. 2. Recording interface of CorpusGong in Slovene.](image-url)
4. Building the Pronunciation Dictionary: For each cluster, find a representative phoneme sequence and write it as word pronunciation to the dictionary. Orthographic representations are generated with a phoneme-to-grapheme model from Croatian.

The next sections discuss these steps in depth. Fig. 3 shows the complete system design and Fig. 4 illustrates the steps with the help of a small example.

4.1. Phoneme Recognition

The phoneme recognition in Slovene is done with a Croatian context-independent phoneme recognizer (50 phonemes) trained on 20 hours Croatian speech data from the GlobalPhone project [40] using the Janus Recognition Toolkit [41]. The GlobalPhone corpus is a collection of read speech in 20 widespread languages in the world. The recognizer uses a uniformly distributed phoneme level LM (0-gram) because we have no knowledge about n-gram phoneme frequencies in the target language. The preprocessing consists of feature extraction applying a Hamming window of 16ms length with a window shift of 10ms. Each feature vector has 143 dimensions by stacking 11 adjacent frames of 13 Mel-scale Frequency Cepstral Co-efficients (MFCC) frames. A Linear Discriminant Analysis (LDA) transformation is computed to reduce the feature vector size to 42 dimensions. The AM uses a fully-continuous 3-state left-to-right HMM with emission probabilities modeled by Gaussian Mixtures with diagonal covariances (64 Gaussians per state). The recognizer achieves a Phoneme Recognition Error Rate of 33.0% on the Croatian GlobalPhone test set, and 55.2% Phoneme Recognition Error Rate on Slovene speech data from the BMED corpus (Tab. 4). The error rate for the Slovene speech was calculated using an IPA based mapping from the Slovene phoneme set.

4.2. Cross-lingual Word-to-Phoneme Alignment

Cross-lingual word-to-phoneme alignments introduced in [20, 13, 14] and tackled by [15] with the alignment model Model 3P are the basis for our pronunciation extraction algorithm. The word segmentation problem describes the task of segmenting phoneme sequences into word units. [15] and [16] show that unsupervised learning of word segmentation is more accurate when information of another language is used. Model 3P (implemented in the PISA Alignment Tool) for cross-lingual word-to-phoneme alignment extends the generative process of IBM Model 3 by a word length step and additional dependencies for the lexical translation probabilities. Alignments are used for the segmentation task as illustrated in Fig. 5.

---

**Table 3.** Phoneme set coverages of the BMED languages according to IPA [36].

<table>
<thead>
<tr>
<th>Related Lang.</th>
<th>Target Lang.</th>
<th>Croatian (33 phonemes)</th>
<th>English (42 phonemes)</th>
<th>German (39 phonemes)</th>
<th>Slovene (34 phonemes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Croatian</td>
<td>100%</td>
<td>45.24%</td>
<td>61.54%</td>
<td>88.24%</td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>75.76%</td>
<td>100%</td>
<td>69.23%</td>
<td>70.58%</td>
<td></td>
</tr>
<tr>
<td>German</td>
<td>72.73%</td>
<td>61.90%</td>
<td>100%</td>
<td>82.35%</td>
<td></td>
</tr>
<tr>
<td>Slovene</td>
<td>90.91%</td>
<td>57.14%</td>
<td>71.79%</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>

---

**Table 4.** Performance of the Croatian phoneme recognizer.

<table>
<thead>
<tr>
<th></th>
<th>Phoneme Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Croatian GlobalPhone test set</td>
<td>33.0%</td>
</tr>
<tr>
<td>Croatian BMED corpus</td>
<td>43.4%</td>
</tr>
<tr>
<td>Slovene BMED corpus</td>
<td>55.2%</td>
</tr>
</tbody>
</table>

---

**Fig. 3.** Overview of the system design.

**Fig. 4.** Steps for the pronunciation dictionary extraction.

---

4.3. Phoneme Sequence Clustering

Let $\text{PhonemeSet}_{hr}$ be the Croatian phoneme set and $\Sigma \subseteq \text{PhonemeSet}_{hr}^+$ be the set of phoneme sequence segments which we extract from the word-to-phoneme alignments (see Sec. 4.2). The function $c : \Sigma \rightarrow \mathbb{N}_+$ indicates how often a phoneme sequence segment was found in the word-to-phoneme alignments in the previous step. Elements in $\Sigma$ correspond to Slovene words, but are often corrupted by alignment and phoneme recognition errors. In this step, we therefore group different realizations of the same Slovene word into clusters

$$C = \{C_1, C_2, \ldots, C_n\} \subset \mathcal{P}(\Sigma) \text{ with } \Sigma = \bigcup_{i \in \{1, n\}} C_i. \quad (1)$$

In order to build $C$ automatically, we first run $k$-means clustering [42] based on the Levenshtein distance for 8 iterations. The means are initialized with the $k$ most frequent elements in $\Sigma$ (indicated by $c$). To find a mean $\mu(C_i) \in \text{PhonemeSet}_{hr}^+$ for a cluster $C_i$, we use the native-lattice tool [43]. $k$ is an initial guess for the target language vocabulary size, that can be derived from the vocabulary size of Croatian. Therefore we set $k = 280$. However, $k$-means fails to separate phonetically similar Slovene words reliably: For example, different inflections (like $bo$ (“it will”) and $bom$ (“I will”)) or completely different words (like $da$ (“yes”) and $dan$ (“day”)) are often placed in the same cluster. Instead, we want elements in $C_i$ to be realizations of a single Slovene word $\mu(C_i)$. Therefore we introduce the outlier index $\epsilon_{oidx} : C \rightarrow \mathbb{Q}_+$:

$$C_i := \{p \in C_i | p \neq \mu(C_i)\}$$

$$\epsilon_{oidx} : C_i \rightarrow \begin{cases} 1 \\ \max_{c(p) \in C_i} c(p) \end{cases} \frac{\text{outlier index}}{\text{median outlier index}}.$$

$C_i$ denotes the set of all elements in $C_i$ that differ from the mean $\mu(C_i)$. If the elements in $C_i$ are corrupted realizations of $\mu(C_i)$, we assume that they are approximately uniformly distributed (Var$(c(C_i))$ is small). This is only an approximation, because errors made by the phoneme recognizer or the alignment model usually depend on the context and the actually spoken phoneme. However, an element $o \in C_i$ that occurs significantly more often than other elements in $C_i$ is likely to be a different Slovene word rather than a corrupted version of $\mu(C_i)$: Correct pronunciations are assumed to occur more often than incorrect ones and thus result in a high $\epsilon_{oidx}$. Fig. 6 illustrates the $\epsilon_{oidx}$ for seven clusters—i.e. seven possible distributions in $C_i$. Peaks in the distributions indicate Slovene words that are incorrectly grouped to cluster $C_i$. In each iteration, the maximal element in each $C_i$ with $\epsilon_{oidx}$ higher than a threshold $\epsilon_{oidx}$ is moved in a separate cluster and $k$ is incremented (i.e. $k$ is increased to 12 in Fig. 6). We refrained from using other outlier detection methods: Some (e.g. Grubbs’ test or Chauvenet’s criterion [44]) take the standard derivation into account and thus score examples 3 to 5 differently. Some (e.g. Grubbs’ test) only work reliably on larger sample sizes. Dixon’s Q test [45] would penalize example 3 containing two outliers because of the small gap between both. Tests based on the (edit) distance between the cluster mean and the putative outlier are misleading since a small distance does not necessary indicate that both elements belong to the same cluster (e.g. mine and fine in English). On the contrary, the proposed $\epsilon_{oidx}$ is a simple criterion that has proved to be effective in our case. After the initial 8 $k$-means iterations, we therefore run a modified version of $k$-means that searches for clusters with high outlier indices in each iteration, and would propagate $o$ as a new mean by putting it in a separate cluster. Thus, the final number of clusters may differ from $k$ since it is incremented whenever an element is moved to a separate cluster because of the $\epsilon_{oidx}$ criterion. The complete algorithm description is listed in Alg. 1.

**Algorithm 1** $\text{kmeansOidx}(\Sigma, k \in \mathbb{N}_+, \epsilon_{oidx} \in \mathbb{Q}_+)$$$

**Require:** $\Sigma \neq \emptyset$

**Require:** $M \subset \text{PhonemeSet}_{hr}^+ \times \mathcal{P}(\Sigma)$

1: $M \leftarrow$ initializeMeans$(\Sigma, k)$
2: for $i \leftarrow 1 \text{ to } 8$ do
3: assignmentStep$(M)$
4: updateStep$(M)$
5: end for
6: for $i \leftarrow 1 \text{ to } 8$ do
7: assignmentStep$(M)$
8: updateStep$(M)$
9: for all $\{(\mu, C) \in M | \epsilon_{oidx}(C) \geq \epsilon_{oidx}\}$ do
10: $o \leftarrow \text{arg max}_{p \neq \mu \in C} c(p)$
11: $M \leftarrow M \cup \{o, \{o\}\}$
12: end for
13: end for
14: end for

---

Fig. 5. Word segmentation through word-to-phoneme alignment.

Fig. 6. The outlier indices for different distributions in $C_i$. 

---

76
4.4. Building the Pronunciation Dictionary

The previous step results in a set of clusters \( C = \{C_1, C_2, \ldots, C_n\} \), where each cluster \( C_i \) stands for a single Slovene word. We use its mean \( \mu(C_i) \) as pronunciation in the pronunciation dictionary for Slovene. Consequently, the number of clusters \(|C|\) is reflected by the size of the extracted dictionary in Tab. 5 and 7. To find an orthographic representation for \( \mu(C_i) \), we transform the phoneme sequence to a written form with a phone-to-grapheme model from a related language. Croatian is closely related to Slovene and is even written with the same script (Sec. 3). Both languages have a good phone-to-grapheme relation. We trained a phone-to-grapheme model \( p2g\) on the Croatian GlobalPhone pronunciation dictionary using Sequitur G2P [18]. This model is used for generating the written form from a pronunciation \( \mu(C_i) \). Applying this model to the correct pronunciations in the Slovene reference dictionary and comparing the generated written forms with the correct Slovene written words results in 5.4% character error rate.

5. LANGUAGE MODEL EXTRACTION

Language modeling for the target language is especially hard because we do not assume the availability of text data in the target language. In initial experiments, we apply a uniformly distributed LM (0-gram). However, Sec. 6.3 shows that our best results are achieved with a unigram LM. The unigram word probabilities are estimated using the sum of occurrences of elements in a cluster:

\[
\hat{P}(p2g_{\mu}(\mu(C_i))) = \frac{\sum_{p \in C_i} c(p)}{\sum_{p \in C} c(p)}, \quad i \in [1, n].
\]

6. EXPERIMENTS

6.1. Evaluation Measures

To evaluate the extracted pronunciation dictionaries, we apply the evaluation measures for the pronunciation extraction from phoneme sequences introduced in [16]: The mapping \( m \) maps each entry in the extracted dictionary to the most similar pronunciation in the reference dictionary containing the correct pronunciations as shown in Fig. 7. The Phoneme Error Rate (PER) is the average edit distance between pairs mapped by \( m \). The Out-Of-Vocabulary rate (OOV) is calculated using the set of all reference dictionary entries mapped by \( m \). The Hypo/Ref ratio indicates how many hypothesis entries in the extracted dictionary are mapped to a single reference dictionary entry on average. The higher the Hypo/Ref ratio, the more pronunciations are extracted unnecessarily.

We calculate the Character Error Rate (CER) to evaluate the final word recognizer for the target language Slovene because it is more robust against minor spelling or segmentation errors than the Word Error Rate (WER). Blanks are treated as separate characters. In all experiments, the word recognizer uses context-dependent acoustic models trained on 20 hours Croatian speech from the GlobalPhone project [40] to recognize Slovene. The preprocessing is described in Sec. 4.1. The AM uses a fully-continuous 3-state left-to-right HMM with emission probabilities modeled by Gaussian Mixtures with diagonal covariances. For our context-dependent AMs with different context sizes, we stopped the decision tree splitting process at 2000 triphones. After context clustering, a merge-and-split training was applied, which selects the number of Gaussians according to the amount of data (19 on average, 38k in total). We did not apply adaptation techniques to improve the acoustic models since Slovene transcriptions are not given. This recognizer achieves a CER of 13.6% on the Slovene portion of the BMED corpus with the correct Slovene pronunciation dictionary and an 1-gram LM trained on the 200 BMED sentences (Tab. 6).

6.2. Experiments with Error-Free Phonetic Transcriptions

First we replace Step 1 in our approach in Sec. 4 and simulate a perfect phoneme recognizer with 0% Phoneme Recognition Error Rate by replacing the words in the Slovene text with their canonical pronunciation and removing word boundary markers. Thereby we initially refrain from dealing with pronunciation variants and phoneme recognition errors.

Tab. 5 summarizes the results when the pronunciation dictionaries are extracted from perfect phonetic transcriptions. As reference we also report the performance of the method presented in [16] that uses a different clustering algorithm and combines elements only if they are aligned to the same source language word. In contrast to this, we cluster solely on the basis of phonetic similarities. Setting \( c_{\text{oidx}} = \infty \) constrains \( \text{kmeansOidx} \) to the standard k-means algorithm. \( k \) is fixed to 280 in all experiments, which is an initial guess for the target language vocabulary size derived from the Croatian vocabulary size. This initialization is not required to be exact since in general the final size of the extracted dictionary differs from \( k \) (as shown in the third column of Tab. 5 and 7): On the one hand, as described in Sec. 4.3, the dictionary size is increased for each new cluster identified by the \( \text{oidx} \) criterion. On the other hand, \( \text{nbest-lattice} \) occasionally calculates the same mean for two separate clusters causing them to merge in \( k \)-means’ assignment step.

We observe that on the designated task, \( \text{kmeansOidx} \) generally results in lower CERs than the method from [16], although the Hypo/Ref ratio is higher. Limiting the maximum \( \text{oidx} \) with \( c_{\text{oidx}} \) effectively reduces both the OOV rate and the CER and tends to produce larger dictionaries. The best recognizer has a CER of 44.2%. When blanks are ignored in the evaluation so that segmentation errors do not affect the error rate, this system achieves a CER of 35.3%.

6.3. Experiments with Recognized Phoneme Sequences

In our scenario we use a Croatian phoneme recognizer to recognize the Slovene target language speech in order to build the pronunci-
Table 5. Results on error-free phonetic transcriptions (oracle experiments).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Method from [16]</td>
<td>de</td>
<td>80</td>
<td>57.8</td>
<td>76.4</td>
<td>1.13</td>
<td>66.0</td>
</tr>
<tr>
<td>Method from [16]</td>
<td>en</td>
<td>74</td>
<td>49.8</td>
<td>76.4</td>
<td>1.04</td>
<td>62.4</td>
</tr>
<tr>
<td>Method from [16]</td>
<td>hr</td>
<td>164</td>
<td>56.8</td>
<td>53.9</td>
<td>1.22</td>
<td>51.3</td>
</tr>
<tr>
<td>kmeansOidx, ϵoidx = ∞</td>
<td>de</td>
<td>280</td>
<td>55.6</td>
<td>39.3</td>
<td>25.2</td>
<td>52.1</td>
</tr>
<tr>
<td>kmeansOidx, ϵoidx = ∞</td>
<td>en</td>
<td>279</td>
<td>51.4</td>
<td>33.9</td>
<td>22.7</td>
<td>51.4</td>
</tr>
<tr>
<td>kmeansOidx, ϵoidx = ∞</td>
<td>hr</td>
<td>278</td>
<td>54.3</td>
<td>36.1</td>
<td>20.9</td>
<td>49.9</td>
</tr>
<tr>
<td>kmeansOidx, ϵoidx = 3</td>
<td>All</td>
<td>275</td>
<td>47.3</td>
<td>42.1</td>
<td>25.4</td>
<td>47.3</td>
</tr>
<tr>
<td>kmeansOidx, ϵoidx = 2</td>
<td>All</td>
<td>318</td>
<td>46.3</td>
<td>42.5</td>
<td>25.2</td>
<td>46.6</td>
</tr>
<tr>
<td>kmeansOidx, ϵoidx = 1.5</td>
<td>All</td>
<td>324</td>
<td>49.0</td>
<td>36.1</td>
<td>20.5</td>
<td>45.9</td>
</tr>
<tr>
<td>kmeansOidx, ϵoidx = 1.1</td>
<td>All</td>
<td>322</td>
<td>48.3</td>
<td>34.6</td>
<td>19.7</td>
<td>45.8</td>
</tr>
</tbody>
</table>

Fig. 8. Performance of the final word recognizer using different dictionaries.

6.4. Human Evaluation

In addition to the automatic evaluation measures we perform a human evaluation for our best systems based on error-free and recognized phonetic transcriptions (44.2% and 52.3% CER): For 100 randomly selected recognition hypotheses from each system we ask a Slovene native speaker to select one sentence from a alphabetically sorted list of all 200 BMED sentences that she thinks is the correct output. If the answer (including reading, understanding, and finding the sentence in the list) takes longer than Maximum Answering Time (MAT), we consider it as wrong. A correct answer with a MAT of 15 seconds usually implies instant understanding. Tab. 9 shows the percentage of correct answers for both systems with different MATs. Surprisingly, the system using recognized phonetic transcriptions outperforms the system using error-free transcriptions when MAT ≥ 30, indicating that a low OOV is more important for understanding than a low PER or even CER. The speaker is able to identify the correct output for 87.9% of the hypotheses from the system based on recognized transcriptions within 1 minute.

7. CONCLUSION AND FUTURE WORK

We have tackled the task of bootstrapping an Automatic Speech Recognition (ASR) system without an a priori given language model (LM), a pronunciation dictionary, or transcribed speech data for the target language Slovene – only untranscribed speech and translations to other resource-rich source languages of what was said were
available. In our scenario a human translator produced utterances in the target language (Slovene) from prompts in resource-rich source languages (Croatian, English, German). First, we cross-lingually aligned the target language speech to the written translations and obtained phoneme sequences segments corresponding to Slovene words, but corrupted by recognition and alignment errors. Second, we introduced a new clustering method kmeansOidx and grouped the segments into clusters. The means of these clusters were written as word pronunciations to the pronunciation dictionary. A Croatian phoneme-to-grapheme model provided orthographic representations. The sizes of the clusters were used to estimate unigram LM probabilities. Both the dictionary and the LM together with Croatian acoustic models were then used to recognize Slovene.

We collected a small corpus (BMED) in four languages consisting of 200 parallel sentences and 50 minutes Slovene speech to evaluate our methods. Our best system achieved a CER of 52% on this corpus, using a unigram LM and all three translations to extract Slovene word pronunciations. For 87.9% of the recognizer hypotheses a Slovene native speaker was able to spot the correct output in a list of 200 sentences within 1 minute.

In the future, we plan to focus on acoustic modeling and apply phonetic discovery methods as in [30, 31, 32] on the target language speech rather than a phoneme recognizer of a related language. Acoustic models could be further improved by iteratively recognizing the speech to provide target language transcriptions, and then using the transcriptions to adapt the models. When it comes to evaluating speech recognizers with partially misspelled words, a automatic measure that is more meaningful than the CER, but less erratic than the Word Error Rate is to be found. Although our results on a limited domain and a small vocabulary are encouraging, the evidence for the applicability of our method on a larger vocabulary and a truly under-resourced or non-written language is still pending.

8. REFERENCES


Table 7: Quality of pronunciation dictionaries extracted from recognized phoneme sequences with the kmeansOidx algorithm.

<table>
<thead>
<tr>
<th>Source Lang.</th>
<th>CER</th>
<th>Dict. Size</th>
<th>PER</th>
<th>OOV</th>
<th>Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>de</td>
<td>1</td>
<td>255</td>
<td>35.8</td>
<td>21.3</td>
<td>57.3</td>
</tr>
<tr>
<td>en</td>
<td>1</td>
<td>255</td>
<td>35.8</td>
<td>21.3</td>
<td>57.3</td>
</tr>
<tr>
<td>hr</td>
<td>1</td>
<td>255</td>
<td>35.8</td>
<td>21.3</td>
<td>57.3</td>
</tr>
<tr>
<td>all</td>
<td>1</td>
<td>255</td>
<td>35.8</td>
<td>21.3</td>
<td>57.3</td>
</tr>
</tbody>
</table>

Table 8: Example hypotheses and their CER (on recognized phoneme sequences, unigram LM, $\epsilon_{oidx} = 2$). Blanks are represented by the underscore character (_).


