Kurimo, Mikko; Enarvi, Seppo; Tilk, Ottokar; Varjokallio, Matti; Mansikkaniemi, André; Alumäe, Tanel

Modeling under-resourced languages for speech recognition

Published in:
LANGUAGE RESOURCES AND EVALUATION

DOI:
10.1007/s10579-016-9336-9

Published: 01/12/2017

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Kurimo, M., Enarvi, S., Tilk, O., Varjokallio, M., Mansikkaniemi, A., & Alumäe, T. (2017). Modeling under-resourced languages for speech recognition. *LANGUAGE RESOURCES AND EVALUATION*, *51*(4), 961-987. https://doi.org/10.1007/s10579-016-9336-9

Modeling under-resourced languages for speech recognition

Mikko Kurimo[1] · Seppo Enarvi[1] · Ottokar Tilk[2] · Matti Varjokallio[1] · André Mansikkaniemi[1] · Tanel Alumäe[2]

The final publication is available at Springer via http://dx.doi.org/10.1007/s10579-016-9336-9.

Abstract One particular problem in large vocabulary continuous speech recognition for low-resourced languages is finding relevant training data for the statistical language models. Large amount of data is required, because models should estimate the probability for all possible word sequences. For Finnish, Estonian and the other fenno-ugric languages a special problem with the data is the huge amount of different word forms that are common in normal speech. The same problem exists also in other language technology applications such as machine translation, information retrieval, and in some extent also in other morphologically rich languages. In this paper we present methods and evaluations in four recent language modeling topics: selecting conversational data from the Internet, adapting models for foreign words, multi-domain and adapted neural network language modeling, and decoding with subword units. Our evaluations show that the same methods work in more than one language and that they scale down to smaller data resources.

Keywords Large vocabulary speech recognition · Statistical language modeling · Subword units · Data filtering · Adaptation

1 Introduction

In agglutinative languages, for example Finnish and Estonian, the number of different word forms is huge, because of derivation, inflection and compounding. This is problematic for statistical language modeling that tries to build

1
Department of Signal Processing and Acoustics,
Aalto University, Espoo, Finland
E-mail: firstname.lastname@aalto.fi

2
Institute of Cybernetics, Tallinn University of Technology, Tallinn, Estonia
E-mail: firstname.lastname@phon.ioc.ee

probabilistic models of word sequences. While modeling the morphology in these languages is complex, modeling the pronunciation of words is rule-based with few exceptions. Thus, splitting words into subwords, such as morphemes or statistical morphs, is a viable and useful tool in applications like automatic speech recognition. However, statistical modeling of morphology, lexicon and word sequences still requires a considerable amount of relevant training data. For under-resourced agglutinative languages, such as variations of Sami and other small fenno-ugric languages, the collection of relevant training data is a significant challenge for language technology development. In this paper we study this resource problem by performing simulations in Finnish and Estonian which include similar morphological properties, but have sufficient resources for carrying out evaluations.

The technical focus of this paper is in large-vocabulary continuous speech recognition (LVCSR) that is essential for automatic processing of dictations, interviews, broadcasts, and all audio-visual recordings. In LVCSR we target on four language modeling topics where we have recently been able to show significant progress: selecting conversational language modeling data from the Internet, adapting pronunciation and language models (LMs) for foreign words, multi-domain and adapted neural network language modeling for improving performance in target topic and style, and decoding with subword lexical units.

For many languages today, large amounts of textual material can be extracted from the World Wide Web. These texts, however, generally provide rather poor match to the targeted style of the language. On the other hand, producing enough accurately transcribed matching training data is expensive. We have faced this problem when developing speech recognition systems for conversational Finnish and Estonian. Huge amounts of Finnish and Estonian data can be crawled from the Internet, but careful filtering is required to obtain a model that matches spontaneous conversations. Several methods have been proposed for selecting segments from an inconsistent collection of texts, so that the selected segments are in some sense similar to in-domain development data [14,21,31]. However, these methods rely on proper development data, but for our Finnish and Estonian tasks there are little carefully transcribed spontaneous conversations available.

A particular problem in lexical modeling is the frequent use of foreign words, which do not follow the same morphological and pronunciation rules as the native words. This becomes a major problem for speech recognition, because a single misrecognized word can severely degrade the modeling of the whole sentence, and the proper names, in particular, are often the most important key words of the content. In many automatic speech recognition (ASR) applications the correct recognition of foreign words relies on hand-made pronunciation rules that are added to the native lexicon. This is a time-consuming solution. An alternative is to automatically generate pronunciation rules for foreign words. Data-driven grapheme-to-phoneme (G2P) converters are often used for this purpose [5]. Focused pronunciation adaptation for foreign words has been previously implemented by automatically detecting the most likely foreign words with letter n-gram models and then generating pronunciation

rules for them with language-specific G2P converters [19,15]. Discriminative pruning of G2P pronunciation variants for foreign proper names has also been applied, to reduce the effect of lexical confusion [1].

The state-of-the-art in statistical language modeling has been pushed forward by the application of neural networks [4]. Neural network models, projecting word sequences into a continuous space, are capable of modeling more complex dependencies, and improve generalization and discrimination. Neural network language models (NNLMs) have also been shown to be useful when training data is very limited [10]. Recently, the methods to improve performance in targeted speaking styles and topics have improved—starting with weighted sampling [29] to more recent work in adaptation [23], multi-domain models [2,37] and curriculum learning [33]. We put our focus on multi-domain models and adaptation in this article.

Subword LMs have many advantages in agglutinative languages with limited data resources. A relatively small lexicon can sufficiently cover an almost unlimited number of words, while still producing models that are capable of accurately predicting words. However, in some cases, the system can also produce words that are very rare or even nonsense. To avoid this we have proposed a new decoder [39], that can efficiently build and use a search network of millions of acceptable words. Thus, new words can be easily added whenever there is a need to recognize some important words that do not exist in the training data.

In our work we mainly present LVCSR evaluations in Finnish and Estonian. Although these two are significantly smaller and less resourced than the main languages of the world, we have fairly good benchmarking tasks to evaluate. For the smaller agglutinative languages, such as Northern Sami, we can not provide such evaluations. However, by artificially reducing Finnish and Estonian training data, we can make simulations that may reveal useful properties of the language modeling methods we propose. The evaluation material in both languages can be divided into broadcast news that suffer from large vocabulary and foreign proper names, and conversations that suffer from the small amount of relevant training data.

2 Methods

2.1 Methods for segmenting words into subwords

Most of the methods described below rely on segmenting the vocabulary into subword units, to address the problems originating from the huge number of different words in Finnish and Estonian. Unless otherwise stated, we have used Morfessor [6] for deriving these segmentations.

The selection algorithms presented in sections 2.2.2 and 2.2.3 need to estimate models from development data, which is less than 100,000 words. We found a Morfessor model to be problematic for the selection algorithms, because with so little training data Morfessor commonly segments unseen words

into single letters that are missing from the LM, which has a significant effect when scoring unseen sentences.

Therefore, in sections 2.2.2 and 2.2.3, we created the subwords using the multigram training algorithms from the freely available software package [38], which avoids setting for any fixed segmentation altogether. By training a multigram model [7] using the forward-backward estimation procedure, the segmentation of words into subwords is probabilistic and all segmentation paths are considered in the model. The multigram formulation is also closely related to Markov models. The model may be written as a unigram model, where the probabilities correspond to fractional frequencies as estimated by the forward-backward training. The model can be used for segmentation of unseen words into subwords, and computation of the probability of any sentence, eliminating the OOV issue.

It should be noted that Morfessor segmentations can still significantly benefit automatic speech recognition of agglutinative languages, even when less than 50,000 words of training data is used [16].

In the decoding experiments in section 3.5, Morfessor was used for the language models trained on the smaller subset. On the larger subset, the subword vocabulary was selected to code the training corpus with high unigram likelihood [40]. This segmentation approach is suitable for reasonably large text corpora.

2.2 Methods for selecting conversational data from the Internet

When modeling under-resourced languages, Internet is often the first place to look for training data. However, the noisy web data requires careful filtering. Several methods exist for selecting LM training data that matches the targeted style of the language, but their computational cost can be high, and the sparsity of development data may pose difficulties especially with agglutinative languages. Furthermore, conversational Finnish is written down phonetically, meaning that also phonetic variation increases vocabulary size and data sparsity [9].

We have developed tools for effectively applying suitable criteria to select useful segments for language modeling from large data sets, when working with only a handful of development data and a morphologically rich language. The source code is available in GitHub[1]. The selection criteria that we have implemented are summarized below. The first two define a score for a text segment, based on which the segments are filtered independently of each other. The third one defines a criterion for adding a text segment to current selection set: The data is scanned sequentially and each segment is selected if it improves the selection set.

– **devel-lp**. A model is estimated from the unfiltered training data, and with a segment removed. The decrease in development data log probability

[1] `https://github.com/senarvi/senarvi-speech/tree/master/filter-text`

when a segment is removed, is the score of the segment. This is the selection criterion used by Klakow [14].
- **xe-diff**. A model is estimated from the development data, and from the same amount of unfiltered training data. The score of a segment is the difference in cross-entropy given by these two models. This is the selection criterion used by Moore and Lewis [21].
- **devel-re**. A text segment is added to the selection set, if including it reduces relative entropy with respect to the development data. This is the criterion used by Sethy et al. [31].

The implementation of each filtering criterion is explained below. In practice, when the language is agglutinative, the only way is to build the LMs from subword units, or the high number of out-of-vocabulary (OOV) words makes reliable estimation of the probabilities impossible [9]. To make the implementations as fast as possible, unigram subword models are used. Limiting to unigrams does not seem to be harmful, since higher-order LMs tend to overlearn small development sets [14].

2.2.1 Implementation of devel-lp filtering

The filtering method presented by Klakow [14] optimizes the perplexity (or equally log probability) of a model computed from the filtered data, on development data. A naive implementation scores each text segment by removing the text segment from the training data, training a language model, and computing the log probability of the development data. This is compared to the log probability given by an LM trained on all training data, and the difference is the score of the text segment. Models are estimated only from the training data, which makes this approach especially suitable for the situation when we have very limited amount of development data. OOV words or subwords are less of a problem when all the models are estimated from a large data set. Consequently, this was the only one of these filtering methods that we applied in [9].

The naive implementation requires training as many LMs as there are text segments. Even though the computation can be done in parallel, a number of optimizations were needed to make the algorithm scale to tens of millions of text segments. First we note that the log probability given by the LM trained on all training data is constant, so we can equivalently define the score of a text segment as the log probability when a text segment is removed from the training data. The only statistics needed for the computation of unigram probabilities are subword counts. As we only compute probabilities on the development data, we only need the counts of the subwords that exist in the development data , $\{c_1^T..c_N^T\}, C^T$, which are collected only once. For each text segment, the counts, $\{c_1^S..c_N^S\}, C^S$ are collected and the score of the segment is computed as

$$\sum_{i=1}^{N} log(\frac{c_i^T - c_i^S}{C^T - C^S})c_i^D, \quad (1)$$

where c_i^D is the number of times the subword appears in the development data. Thus the running time of the algorithm is proportional to the number of text segments times the number of unique subwords in the development data.

2.2.2 Implementation of xe-diff filtering

In the method proposed by Moore and Lewis [21], two language models are estimated, one from the development data and another from the same amount of unfiltered training data. The score of a text segment is the difference in cross-entropy given by these two models. The method requires only computation of the two LM probabilities for each text segment. Thus, the running time is proportional to the number of words in the unfiltered training data.

2.2.3 Implementation of devel-re filtering

The idea behind the filtering method proposed by Sethy et al. [31] is to match the distribution of the filtered data with the distribution of the development data in terms of relative entropy. First a language model is estimated from the development data, and the same amount of unfiltered training data is used to initialize a model of the selection set. Then the text segments are processed sequentially. It is computed how much relative entropy would change, with respect to the development data model, if a segment was included in the selection set. If the change is negative, the text segment is included and the selection set model is updated.

We used the revised version of the algorithm that uses skew divergence in place of Kullback-Leibler (KL) divergence [32]. Skew divergence contains parameter α, whose value 1 corresponds to KL divergence, and smaller values smooth the maximum-likelihood model of the selection set. We first select the same amount of text as there is in the initial model and then recompute the model from only the selected data.

Sethy et al. present an optimization that runs proportional to the number of words in the unfiltered training data. However, the sequential algorithm itself cannot be parallelized. The authors note that the algorithm is greedy and running it several times with random permutations of the text segments improves the result. They also suggest skipping sentences that have already been included in more than two passes, in order to gain new data faster. We did not enforce that requirement, enabling us to run multiple passes simultaneously. It should be noted that also the generation of a random permutation can be time consuming and I/O intensive, especially when the data set is too large to be loaded into memory, and multiple parallel processes access the same data.

2.3 Methods for adapting models for foreign words

In ASR applications the correct recognition of foreign proper names (FPNs) is a difficult challenge. The problem of recognizing foreign words is especially a problem for smaller languages where influence from other languages is bigger and FPN occurrence more frequent. For Finnish subword-based ASR, foreign names constitute one of the largest error sources [11].

The challenge in recognizing foreign names stems from a combination of many factors. The most obvious is pronunciation modeling. Pronunciation rules that cover native words usually give unreliable results for foreign words. Foreign names are often rare and topic-specific. Background LMs usually give unreliable estimates for FPNs. A third factor that is quite specific to subword LMs is oversegmentation (base form of the word is split into many different parts). Oversegmentation of foreign words complicates the mapping of non-standard pronunciation rules to separate subword units.

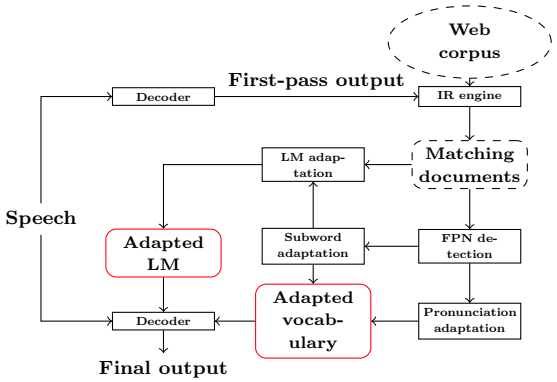

Fig. 1 Adaptation framework for foreign proper name adaptation. Adapted LM and vocabulary are used in second-pass recognition

Previously, FPN recognition for Finnish subword-based speech recognition has been improved using a two-pass adaptation framework, as illustrated in Fig. 1 [20]. Based on first-pass ASR output the language model and lexicon are both adapted in an unsupervised manner. In-domain articles which best match the first-pass output are selected based on latent semantic indexing (LSI). From the selected articles an in-domain LM (P_I) is trained and adapted with the background LM (P_B). In this work linear interpolation is used with a fixed interpolation weight (eq. 2, $\lambda = 0.1$).

$$P_{adap_I}(w|h) = \lambda P_I(w|h) + (1 - \lambda)P_B(w|h) \quad (2)$$

Lexicon adaptation is performed by first screening for foreign word candidates in the in-domain texts. All words starting with an uppercase letter are selected as foreign word candidates. From the candidate list, the most likely

foreign words are chosen using the product of two factors, letter n-gram perplexity $ppl(word)$ and topic similarity $sim(word)$, as a score (eq. 3). $ppl(word)$ is the perplexity given by letter n-gram model estimated from a native word list collected beforehand, on word $word$ in the in-domain article. $sim(word)$ is defined as the cosine similarity between the first-pass output and the article where $word$ occurs.

$$score(word) = ppl(word) * sim(word) \tag{3}$$

The most likely foreign names (with the highest score) are selected and added to the vocabulary. Adapted pronunciation rules for each FPN are generated using a data-driven G2P model [5]. Optionally subword restoration is applied for oversegmented FPN candidate words.

In this work we study how well this adaptation framework can be transferred from Finnish to a related language, Estonian. The phoneme sets of the two languages are quite similar. This gives the option of sharing the foreign word G2P model. The original G2P model was trained on 2000 foreign names retrieved from a Finnish text corpus. The hand-crafted pronunciation rules were made with a Finnish phoneme set and Finnish speakers in mind. The pronunciation rules generated from the G2P model can with some minor modifications be converted to an Estonian phoneme set.

A problem with G2P generated pronunciation variants when trying to improve FPN recognition is that many of the variants actually degrade the recognition of native words. In combination with the adaptation framework, we will also evaluate a lattice-based discriminative pronunciation pruning method [8]. The pruning tools are available in GitHub[2]. The algorithm removes those FPN pronunciation variants from the final adapted dictionary that have a negative effect on the total word error rate. Pronunciation variants that have a positive effect on recognition are used to retrain the G2P model by appending them to the foreign word lexicon. This discriminative training procedure is iterated a number of times on the development set before a final G2P model and a list of harmful pronunciations is obtained. The updated G2P model and the list of harmful pronunciations are then used on the evaluation set.

To the authors' knowledge no previous work has used this type of lattice-based discriminative pronunciation pruning for both excluding harmful pronunciation variants and re-training the G2P model with beneficial pronunciation variants.

2.4 Methods for multi-domain and adapted neural network language modeling

When developing a LM for a specific domain it is often the case that the amount of available in-domain data (the data belonging to the target domain) is not sufficient for a good model. This is even more of a problem when dealing

[2] `https://github.com/senarvi/senarvi-speech/tree/master/filter-dictionary`

with under-resourced languages. The scarcity of in-domain data makes it necessary to include out-of-domain sources in the training of the LM. Usually the amount of available out-of-domain data is much bigger than in-domain data. Therefore the LM needs to favour the in-domain data somehow to perform well in the target domain.

NNLMs [4] can achieve this goal in several ways:

- **weighted sampling**. During training the in-domain data is sampled with higher probability than out-of-domain data (e.g. use all in-domain data and only a random subset of out-of-domain data in each epoch [29]).
- **curriculum learning**. The order in which the training data is presented to the network is planned in such a way that more general samples are seen in the beginning of the training while domain-specific samples are kept towards the end of the training so they have more influence on the final model [33].
- **adaptation**. After training the model on out-of-domain data it is adapted for the in-domain data. The adaptation can be done, for example, by adding an adaptation layer and training it on in-domain data while keeping the other parameters fixed [23].
- **multi-domain models**. Most parameters are shared between domains to allow exploiting the inter-domain similarities. A tiny fraction of parameters is reserved to be domain-specific and is switched according to the active domain to take into account the domain-specific differences [2,37]. Unlike with adaptation, the domain-specific and general parameters are trained jointly and the same model can be used in all domains.

In this article we use the adaptation and multi-domain approaches.

For multi-domain approach we use a simplified version of the multi-domain NNLM from [2]. The architecture of our model is shown in Fig. 2. It differs from the architecture described in [2] by omitting the extra linear adaptation layer and applying the multiplicative adaptation factors directly to the pre-activation signal of the hidden layer rectified linear units (ReLU). The hidden layer activations are computed as shown in eq. 4 where y_0 and y_1 are projection and hidden layer activations respectively, W_{1a} and b_{1a} are hidden layer and W_{1b} and b_{1b} are domain adaptation weights and biases respectively. W_{1b} consists of domain-specific row-vectors (domain vectors) while b_{1b} is shared across domains. To prevent the adaptation factors from shrinking the inputs to ReLU from the start of training, the weights W_{1b} or bias b_{1b} can be initialized to ones (we used the latter in our experiments).

$$y_1 = ReLU\big(y_0 W_{1a} \circ (d_t W_{1b} + b_{1b}) + b_{1a}\big) \tag{4}$$

This kind of hidden layer enables each domain to influence the structure of sparsity in the output layer inputs (i.e. which hidden layer units are more or less likely to be exactly zero for each domain) in addition to modulating the nonzero outputs. One can consider the NNLM as a log-linear model on top of an automatically learned feature vector obtained by transforming the input through nonlinear transformations in lower layers as in [30]. In this perspective

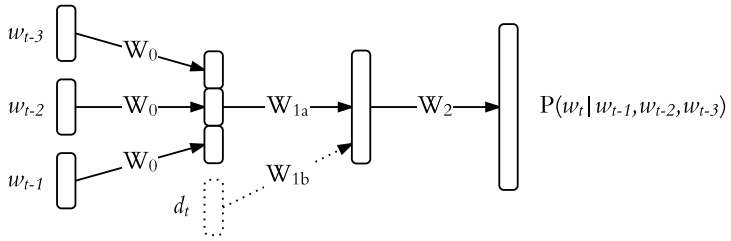

Fig. 2 Description of the NNLM architecture. Dotted lines stress the parts of the network that are characteristic only to the multi-domain and adapted models. The inputs (context word indices w_{t-1}, w_{t-2}, w_{t-3} and the domain index d_t) are one-of-N encoded vectors

the multi-domain model can influence the relevance of the log-linear model input features in the context of different domains. Our experience shows that the simplified model performs just as well or even marginally better than the original one with an additional layer.

The multi-domain model requires the availability of in-domain data in the training set. With limited-resource domains it is possible that there is not enough target domain data for separate training, validation and test set. This means that there might be no in-domain data left for the training phase. We propose an adaptation approach which uses exactly the same model architecture as the multi-domain model to overcome this problem. The advantage of using the multi-domain architecture for adaptation is its resistance to overfitting due to the very small amount of domain specific parameters that need to be trained on the target domain data. The amount of domain-specific parameters is limited to a single vector with a number of elements equal to the hidden layer size (usually several hundred or thousand), which is tiny compared to the total amount of parameters in the network (usually in millions). Thus, the training error on validation data gives a good estimate of the performance on unseen data and all the available in-domain data (except the test data) can be used for adaptation.

The adaptation procedure is as follows:

1. Train a general model on out-of-domain training data using the in-domain validation data for early stopping and hyperparameter selection;
2. After the general model is ready, add the domain-specific parameters W_{1b}, b_{1b} and modify the hidden layer activation according to eq. 4;
3. Train only the domain-specific parameters added in the previous step on the in-domain validation data until convergence, while keeping the rest of the parameters fixed.

Initially, we believed that to effectively utilize the domain vectors, the network should have a multi-domain architecture from the start and be trained as such on non-target domains. However, the preliminary experiments revealed that this is not true. The adapted model works just as well if all the multi-domain architecture specific elements are added right before training the target domain parameters.

This procedure raised a question whether the multi-domain model can also be improved by combining all the in-domain data from both training and validation set and using it to fine-tune the target domain vector as a final step of training. Unfortunately, our preliminary experiments showed that this does not significantly improve the perplexity of the test set.

2.5 Methods for decoding with subword units

The normal approach to language modeling in ASR is to train n-gram LMs over sequences of words. For morphologically rich languages this is often problematic, because the number of OOV words may be high. This is especially the case for less-resourced languages, as considered here. Thus, words are not necessarily the best units for language modeling. By training the n-gram models over sequences of subwords, it is possible to assign probabilities to previously unseen word forms. In our final task we compare different combinations of lexical units and decoders.

A common approach to LVCSR decoding is the dynamic token-passing search [41], where tokens are propagated in a graph containing paths for the allowed recognition output with the corresponding Hidden-Markov-Model (HMM) state sequences. A token contains at least the accumulated likelihood scores, information about the current n-gram state and the recognition history. Many standard techniques [22] like hypothesis recombination, beam pruning and LM lookahead are needed to make the search efficient. Cross-word pronunciation modeling [35] is also important for the speech recognition accuracy in tasks dealing with continuous speech. In Fig. 3, the first graph is a conceptual example of a standard word decoder utilizing triphone HMMs and word n-grams. Silence and cross-word modeling is omitted from the image.

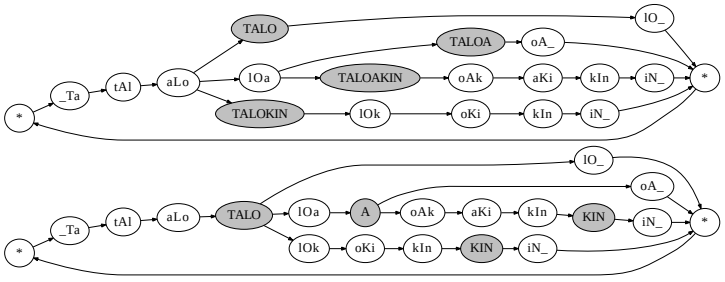

Fig. 3 Example decoding graphs for word n-grams (above) and subword n-grams (below), for the same 4-word recognition vocabulary. Grey nodes depict the n-gram identifiers

In the case of subword n-grams, the same search principles may be applied, but the graph should be constructed differently. Here we consider subword decoders, which are general in the sense, that arbitrary segmentations of words to subwords are allowed. With subword n-grams, it is possible to allow all

possible concatenations of subwords [26], which enables unlimited recognition vocabulary, as all word forms may be created by concatenating the subwords. The requirement for this construction is that the pronunciation of each subword is defined. For the languages considered here, the pronunciation may be easily derived from the grapheme form of the subword.

Another recently suggested possibility is to use subword n-grams, but still restrict the recognition vocabulary to the desired set of words [39]. In Fig. 3, the second graph is a conceptual example of a decoding graph, which is constructed in this way. As also in this case the n-gram model has probabilities for all word forms, unseen words may be segmented with the n-gram model, and the corresponding paths added to the graph. This opens up new possibilities for augmenting and adapting the vocabulary, especially in cases, when the training data does not cover enough word forms. For analysis purposes, the recognition performance of the word n-gram and the subword n-gram estimates may be compared for the same recognition vocabulary. This is useful in assessing, whether the improvement in using subwords models is caused by the better n-gram estimates or the reduced OOV rate.

3 Experiments

3.1 Data

The speech data sets used in our experiments are listed in Table 1 and Table 2. Finnish acoustic models for all experiments (except the conversational speech experiment) were trained on the Finnish Speecon database [13], from which 31 hours of clean dictated wideband speech from 310 speakers (*fi-std-train*) was used for training. Estonian acoustic models for the conversational speech and neural network language modeling experiments were trained on the full *ee-conv-train* set. It consists of a small amount of spontaneous Estonian conversations, but mostly less spontaneous radio broadcasts and lecture recordings. Estonian acoustic models for the foreign proper name adaptation and subword decoding experiments were trained on a 30 hour subset of the *ee-conv-train* set, consisting of only broadcast news recordings.

Finnish conversational speech experiments were carried out on data collected at Aalto University by recording and transcribing pair-wise conversations between students. Finnish acoustic models for web text filtering experi-

Table 1 Finnish speech data sets

Data set	Words	Hours
fi-std-train	131,005	31.4
fi-conv-train	200,415	15.2
fi-conv-eval	6,268	0.73
fi-news-dev	35,439	5.38
fi-news-eval	37,196	5.58

Table 2 Estonian speech data sets

Data set	Words	Hours
ee-conv-train	1,251,638	165
ee-conv-eval	25,942	2.90
ee-news-dev	15,961	2.13
ee-news-eval	15,335	2.03

Table 3 Sizes of Finnish text data sets after preprocessing

Data set	Words
fi-web-1	766,918
fi-web-2	1,035,043
fi-web-3	561,489
fi-web-4	25,175,069
fi-web-5	46,207,390
fi-web-6	2,618,084,259
fi-conv-dev1	98,956
fi-conv-dev2	8,853
fi-general	153,535,459
fi-webnews	12,675,262
fi-newswire	31,809,529

Table 4 Sizes of Estonian text data sets after preprocessing

Data set	Words
ee-web-1	28,490,011
ee-web-2	4,189,681
ee-web-3	273,413,272
ee-web-4	30,599,060
ee-conv-dev1	187,436
ee-conv-dev2	21,202
ee-newspapers	20,423,775
ee-webnews	76,235,530
ee-news-train	133,171

ments were trained on the *fi-conv-train* set. It consists of student conversations, transcribed radio shows, FinDialogue part of the FinINTAS [17] corpus, and free spontaneous speech from Finnish SPEECON [13] corpus. The extent to which the speech is spontaneous varies between the recordings, as well as the dialect and style. The evaluation set *fi-conv-eval* consists of transcribed radio conversations and student conversations from unseen speakers. *ee-conv-eval* consists of transcribed conversations from the Phonetic Corpus of Estonian Spontaneous Speech[3].

Text data sets are listed in Table 3 and Table 4. Training data for conversational LMs were crawled from four Estonian conversation sites (*ee-web-1* to *ee-web-4*) and six Finnish sites (*fi-web-1* to *fi-web-6*). These sites contain active discussions in various topics, such as technology, sports, relationships, and culture. The most important tool we have used is the Python library Scrapy. The web data filtering experiments required two development sets for each language. *ee-conv-dev1* and *ee-conv-dev2* consist of transcripts from the Phonetic Corpus of Estonian Spontaneous Speech. *fi-conv-dev1* and *fi-conv-dev2* contain partly the same data that was used in acoustic model training: student conversations, transcribed radio shows, and FinDialogue.

Foreign proper name adaptation experiments were conducted on broadcast news data. The development and evaluation sets *fi-news-dev* and *fi-news-eval* were used in the Finnish experiment and the sets *ee-news-dev* and *ee-news-eval* in the Estonian experiment. The *fi-general* set from the Finnish Text Collection[4] corpus was used for Finnish baseline LM training. It contains texts from books, magazines and newspapers. For Estonian baseline LM training, the full *ee-newspapers* and *ee-news-train* sets were used, and a random 75% subset of *ee-webnews*.

NNLM experiments for Finnish were carried out on the development and evaluation sets *fi-news-dev* and *fi-news-eval*, which consist of Finnish broadcast news recordings collected in 2011 and 2012. For training the LMs, three data

[3] `http://www.keel.ut.ee/et/foneetikakorpus`

[4] `https://research.csc.fi/-/finnish-text-collection`

sources were used: a random subset of 23 million words from *fi-general*, a corpus of texts from Finnish web news portals (*fi-webnews*), and a corpus of newswire texts from a Finnish news agency STT (*fi-newswire*). The Estonian experiment was based on the development and evaluation sets *ee-news-dev* and *ee-news-eval* that contain broadcast news speech from 2005. For language modeling we used three data sources: newspaper texts (*ee-newspapers*), texts from web news portals (*ee-webnews*) and broadcast news transcripts (*ee-news-train*).

Finnish LMs for subword decoding experiments were trained on two subsets from *fi-general*. The larger subset contained 50M word tokens with 2.2M distinct word types and the smaller 10M word tokens with 850k word types. Estonian LMs were trained on the *ee-webnews*, *ee-newspapers* and *ee-news-train* data sets. A larger model was trained on all the training data of around 80M words with 1.6M distinct word types and a smaller model from a 10M word subset with 550k word types.

3.2 Experiments in selecting conversational data from the Internet

In this section we experiment how the most important filtering criteria perform when filtering large amounts of Internet data, when there is only very little in-domain development data available. Our motivation has been development of automatic speech recognition for conversational Finnish and Estonian. We have a small amount of transcribed Finnish and Estonian conversations that are enough for development and evaluation. For LM training data we crawled large amounts of multi-domain data from Internet conversation sites. The segments used as the unit of filtering are conversation site messages.

For the baseline experiments, the sizes of the largest data sets were limited by random selection. In total the number of words in Finnish training data was reduced to 9.9 % and in Estonian data to 49 % of the original. *devel-lp* and *xe-diff* methods define a score for each text segment. The filtering threshold is optimized to minimize the perplexity of a bigram subword model on the second development set (*fi-conv-dev2* or *ee-conv-dev2*). *devel-re* does not define a score for each segment. Instead, whether a segment is included depends on what has been included earlier. We found running multiple passes with random permutations of the input text segments to be crucial for collecting enough data. The number of passes is limited by the high computational cost. We ran 100 passes, but also tried using data from only so many passes that unigram subword model perplexity on the second development set was minimized. We selected the value 0.975 for the smoothing parameter α, based on observations of the original author [32], without trying to optimize the value.

Filtering was performed, and the filtering threshold and the number of passes was optimized, on each data set (conversation site) separately. However, sets *fi-web-1* to *fi-web-3* were pooled together during filtering, and the set *fi-web-6* was split into 48 parts during *devel-lp* and *xe-diff* filtering.

Table 5 Filtered data sizes and speech recognition results. The best results in terms of WER are in bold type

	Finnish		Estonian	
Algorithm	Words	WER	Words	WER
baseline	266M	55.6 %	167M	53.4 %
devel-lp	192M	54.3 %	82.2M	**52.7 %**
xe-diff	169M	54.4 %	38.9M	53.2 %
devel-re [passes = 1]	5.08M	57.9 %	13.1M	54.4 %
devel-re [passes = 50]	53.9M	54.6 %	93.8M	53.1 %
devel-re [passes = 100]	79.5M	54.2 %	125M	53.1 %
devel-re [optimized]	75.9M	**54.1 %**	117M	53.1 %

The experiments were carried out using Aalto ASR system [12] and GMM-HMM-based acoustic models. Language models were 4-gram word models interpolated from models of individual data sets. The vocabulary was created after filtering by selecting 200,000 top words based on weighted word counts in order to maximize the likelihood of the combined development data. The number of n-grams in every LM was reduced by pruning all n-grams whose removal caused less than 5×10^{-10} increase in training data perplexity.

3.2.1 Results

Results for web text filtering are shown in Table 5. Large phonetic variation in conversational Finnish creates challenges when measuring recognition accuracy. As most of the words can be pronounced in several slightly different ways, and the words are written out as they are pronounced, it would be harsh to compare recognition against the verbatim phonetic transcription. Thus word forms that are simply phonetic variation have been added as alternatives in the reference transcriptions.

devel-re selection resulted in the smallest data size. The amount of data that will be selected depends on the size of the development set. The small development set used in these experiments caused only a minimal amount of data to be selected during the first *devel-re* pass, resulting in poor word error rate. Combining selected data from 100 passes improved word error rate to 54.2 % with Finnish data. The other methods gave very similar results in terms of WER, but more than double the amount of data. However, running 100 passes was computationally very demanding.

Optimizing the number of passes of *devel-re* filtering, in terms of perplexity on held-out development data, gave still a slight improvement. The resulting 54.1 % WER is good, given that only web data was used to build the LM. In our previous state-of-the-art of conversational Finnish ASR, we obtained 57.5 % WER using only web data, and 55.6 % when combined with other corpora, while using only other than web data WER was 59.8 % [9]. One can conclude that significant improvement can be gained by using web data, in the absence of accurately transcribed conversational corpora. However, in this paper we have also used better acoustic models.

Overall, filtering Estonian data did not improve speech recognition compared to the baseline as much as with Finnish data. The best result, 52.7 % WER, was given by *devel-lp* filtering. Compared to the Finnish language results, the advantage to the other methods was surprisingly clear. *devel-re* method gained new data faster than in the Finnish language experiments, probably due to the larger development set, and as many passes were not needed. We are not aware of any earlier research on recognition of spontaneous Estonian conversations.

3.3 Experiments in adapting models for foreign words

Foreign proper name adaptation experiments are conducted with the adaptation framework described in the methods section (Fig. 1). The occurrence of foreign names in the data sets is of importance since we are focusing adaptation efforts on improving their recognition. For Finnish, FPN rate is 4.3 % for the development set (*fi-news-dev*) and 3.5 % for the evaluation set (*fi-news-eval*). For Estonian, FPN rate is 1.6 % for the development set (*ee-news-dev*) and 1.7 % for the evaluation set (*ee-news-eval*).

Experiments are run on the Aalto ASR system [12] and GMM-HMM-based acoustic models. For Finnish, a Kneser-Ney smoothed varigram LM (n=12) with a 45k subword lexicon was trained on the LM training data using variKN language modeling toolkit [34] and Morfessor [6]. A letter bigram model was trained on the same LM training data for the foreign name detection algorithm.

A subword-based baseline LM for Estonian was trained, similarly to Finnish using Morfessor and variKN toolkit. The resulting model was a Kneser-Ney smoothed varigram LM (n=8) with a 40k subword lexicon. A letter bigram model for foreign name detection was trained on a word list extracted from the LM training data.

First set of experiments are run with the baseline LMs to retrieve the first-pass ASR output. After that unsupervised LM adaptation experiments are run. The background LM is adapted with 6,000 of the best matching articles compared to the ASR output. The retrieval corpus is a collection of articles retrieved from the Web. The Finnish retrieval corpus consists of 44,000 articles (*fi-webnews*). The Estonian retrieval corpus consists of 80,000 articles (25 % subset of *ee-webnews*).

In the third adaptation layer we apply vocabulary adaptation. Foreign proper name candidates are selected based on the letter-gram perplexity and cosine similarity score. A threshold is set so that only 30 % of the best scoring FPN candidates are selected for adaptation. Furthermore an additional constraint is set so that the number of new words added can not exceed 4 % of the original vocabulary size. Four new pronunciation rules are generated for each selected FPN candidate and added to the lexicon. The pronunciation rules are generated with a data-driven G2P model which has been trained on 2,000 foreign names found in Finnish texts. The same G2P model is used for both Finnish and Estonian. Subword restoration is applied on oversegmented

FPN candidate words to enable one-to-one mapping between pronunciation rule and vocabulary unit.

In the final adaptation layer we implement discriminative pronunciation pruning based on the ASR output lattices when using the adapted LM and lexicon. Harmful FPN pronunciation variants that degrade overall recognition accuracy by five word errors or more are excluded in the next run. Beneficial FPN pronunciation variants that decrease word error by one word or more are added to the 2,000 word foreign name lexicon. A new G2P model is re-trained with the updated lexicon. This procedure is iterated a couple of times on the development set before get a final list of harmful pronunciation variants and an updated G2P model which are then used on the evaluation set.

3.3.1 Results

Results of the FPN adaptation experiments are presented in Table 6. Performance is measured in average word error rate (WER) and foreign proper name error rate (FER).

First set of experiments were run on the Finnish development set (*fi-news-dev*). Compared to the baseline model, unsupervised LM adaptation reduces average WER with 3 % and FER with 10 %. Vocabulary adaptation (pronunciation and subword adaptation) reduces FER with another 7 % but average WER remains unchanged, compared to only using unsupervised LM adaptation. After three iterations discriminative pronunciation pruning is able to further reduce WER with 1 % and FER with 2 %. It does seem that pronunciation pruning, in excluding some of the most harmful pronunciation variants, is able to correct the misrecognition of some native words.

For the Finnish evaluation set (*fi-news-eval*) results are similar compared to the development set, when applying unsupervised LM and vocabulary adaptation. Average WER is reduced with around 3 % compared to the baseline LM. Vocabulary adaptation reduces FER with 7 % compared to only using unsupervised LM adaptation. Discriminative pronunciation pruning was tested with the list of harmful pronunciation variants and re-trained G2P model obtained after three iterations on the development set. In terms of average WER, which remains unchanged, results are not as good as on the development set. There is probably not enough overlap between harmful pronunciation variants introduced in the development set that are also relevant for the evaluation set. We might see a more significant impact over larger data sets. The re-trained G2P model reduces FER with around 2 %. The change is small but it does indicate that it is possible to improve G2P modeling through discriminative pronunciation pruning on development data.

For the Estonian broadcast news development set, unsupervised LM adaptation reduced average WER with nearly 2 % and FER with under 1 %. Vocabulary adaptation increases average WER, but reduces FER with over 1 %, compared to using only unsupervised LM adaptation. Discriminative pronunciation pruning does manage to improve recognition of foreign names

Table 6 FPN adaptation results for Finnish and Estonian. Baseline results are followed by results for unsupervised LM adaptation (Adapted LM), combination of unsupervised LM and vocabulary adaptation (Adapted LM + VOC), and iterations of discriminative pronunciation pruning (Adapted LM + VOC [pruned, iter = x]). On the evaluation sets discriminative pronunciation pruning is tested with the pruning data and models obtained after the third iteration on the development set (Adapted LM + VOC [pruned, dev. iter = 3])

	fi-news-dev		*ee-news-dev*	
Adaptation	WER	FER	WER	FER
Baseline	29.6 %	73.5 %	19.2 %	51.8 %
Adapted LM	28.6 %	66.4 %	18.9 %	51.4 %
Adapted LM + VOC	28.6 %	61.8 %	19.5 %	50.7 %
Adapted LM + VOC [pruned, iter = 1]	28.5 %	60.6 %	19.4 %	50.0 %
Adapted LM + VOC [pruned, iter = 2]	28.4 %	60.6 %	19.4 %	49.3 %
Adapted LM + VOC [pruned, iter = 3]	28.4 %	60.5 %	19.4 %	49.3 %
	fi-news-eval		*ee-news-eval*	
Adaptation	WER	FER	WER	FER
Baseline	30.5 %	71.6 %	19.6 %	49.3 %
Adapted LM	29.7 %	64.8 %	19.2 %	47.1 %
Adapted LM + VOC	29.6 %	60.0 %	19.5 %	46.0 %
Adapted LM + VOC [pruned, dev. iter = 3]	29.6 %	59.0 %	19.4 %	46.0 %

with almost 3 % but average WER is still higher than compared to only using unsupervised LM adaptation.

Results for the Estonian evaluation set are quite similar to the development set. Unsupervised LM adaptation reduces WER with 2 % and FER with 4 %. Again, vocabulary adaptation degrades recognition of native words. Average WER increases but FER is reduced with 2 %. Discriminative pronunciation pruning (data and models obtained from the development set's third iteration) does lower average WER slightly but FER is not further improved.

There seems to be more acoustic confusion added to Estonian ASR when augmenting the lexicon with G2P generated pronunciation variants. It is not clear whether this is because of the low FPN rate in Estonian speech data or if the Finnish G2P model has negative effects on the recognition of some native Estonian words. Discriminative pronunciation pruning is not able to significantly lessen the effect of lexical confusion.

3.4 Experiments in multi-domain and adapted neural network language modeling

In multi-domain and adapted NNLM experiments we evaluate the models in terms of perplexity (PPL) and WER. The models are evaluated on two broadcast news data sets: a Finnish data set consisting of subwords (morphs) and an Estonian data set consisting of compound-split words. The PPL scores are calculated on their respective lexical units, WER scores are computed on words.

Our baseline LM is a back-off 4-gram model with modified Kneser-Ney discounting constructed over all available training data. Surprisingly, interpolating domain-specific models results in an inferior model.

It has been recently verified that NNLMs perform better than back-off n-gram models on under-resourced languages [10]. One of our goals is to check whether the multi-domain and adapted NNLMs bring additional improvements and what is the relationship between their relative improvement and training set size.

Four experiments are performed on both languages. We start by training all the models on all available text data and continue by halving the training data for each consecutive experiment by taking every second line of the previous data set. NNLM hidden and projection layer size is divided by $\sqrt{2}$ every time the training data is halved. The initial hidden layer size is 500 for Finnish and 1400 for Estonian NNLM; initial projection layer size is 3×100 for Finnish and fixed to 3×128 for Estonian. Both Finnish and Estonian models use a shortlist [28] of 1024 most frequent units (subwords or compound-split words respectively) plus an additional end of sentence token. The input vocabulary consists of 50,410 most frequent subwords and 199,861 most frequent compound-split words for Finnish and Estonian data set respectively. Both input vocabularies contain an additional token for the beginning of sentence and unknown units. When interpolating the n-gram and NNLM model outputs we use an equal weight of 0.5 for both models. Out-of-shortlist units are evaluated only by the n-gram model. All NNLMs are trained with backpropagation and mini-batch stochastic gradient descent using batch size of 200 samples and learning rate of 0.1 until the best model according to validation perplexity is not within the last 5 epochs. We use our NNLM adaptation method on Finnish data set, because there we have no in-domain training data. Estonian data set has in-domain training data, so we use the multi-domain NNLM there.

In speech recognition experiments recognition lattices were generated using systems based on the Kaldi toolkit [24], and the lattices were rescored using the NNLMs. Finnish acoustic models are triphones, built using fMLLR-based speaker-adaptive training (SAT) and optimized using the boosted MMI criterion [25]. Lattices are obtained after two decoding passes: first pass uses speaker-independent models, and the second pass fMLLR-transformed features with SAT-based models. Estonian acoustic models are hybrid deep neural networks based hidden Markov models (DNN-HMMs) that use speaker identity vectors (i-vectors) as additional input features to the DNNs in parallel with the regular acoustic features, thus performing unsupervised transcript-free speaker adaptation [27]. The output hypotheses of the speech recognition systems consist of subword units for Finnish and compounds-split words for Estonian. These were converted to word hypotheses using a hidden event LM that treats a word break (for Finnish) or an inter-compound unit (for Estonian) as a hidden word that needs to be recovered. More details about the Estonian system are available in [3].

3.4.1 Results

The results of PPL and WER evaluations on the test set can be seen in Table 7. All NNLMs consistently outperform back-off n-gram models in PPL and WER. Utilizing NNLMs in addition to n-gram models gives a similar effect as using about twice as much training data: the PPL improves 7.1–17.5 % relative, statistically significant WER improvement is about 2.1–4.9 % relative. The type of lexical units used in vocabulary and baseline WER (largely determined by the acoustic model quality) don't seem to affect the relative WER improvement brought by NNLMs. Both, the multi-domain and adapted, NNLMs consistently beat the simple NNLM in PPL evaluation (0.6–7.1 % relative). Unfortunately this makes no significant difference in WER for neither case. This holds true for all languages and training set sizes we tested.

Table 7 LM test set PPL and WER with different sized training sets. Comparison with the n-gram baseline in parentheses. *a-nnlm* is the adapted and *md-nnlm* is the multi-domain NNLM

		Finnish			
		1	1/2	1/4	1/8
PPL	n-gram	197	222	256	298
	nnlm + n-gram	183 (-7.1%)	205 (-7.7%)	236 (-7.8%)	274 (-8.1%)
	a-nnlm + n-gram	177 (-10.2%)	200 (-9.9%)	230 (-10.2%)	268 (-10.1%)
WER	n-gram	33.3	34.0	34.9	35.7
	nnlm + n-gram	32.3 (-3.0%)	32.9 (-3.2%)	33.4 (-4.3%)	34.3 (-3.9%)
	a-nnlm + n-gram	32.4 (-2.7%)	32.8 (-3.5%)	33.4 (-4.3%)	34.3 (-3.9%)
		Estonian			
		1	1/2	1/4	1/8
PPL	n-gram	223	252	301	366
	nnlm + n-gram	198 (-11.2%)	216 (-14.3%)	257 (-14.6%)	315 (-13.9%)
	md-nnlm + n-gram	184 (-17.5%)	208 (-17.5%)	250 (-16.9%)	313 (-14.5%)
WER	n-gram	9.2	9.6	10.3	10.8
	nnlm + n-gram	9.0 (-2.2%)	9.4 (-2.1%)	9.8 (-4.9%)	10.3 (-4.6%)
	md-nnlm + n-gram	9.0 (-2.2%)	9.4 (-2.1%)	9.9 (-3.9%)	10.3 (-4.6%)

The small PPL gap and no significant WER improvement between the simple and multi-domain NNLM architecture seems to indicate that the single static domain vector has too little capacity to alter the model sufficiently to reflect all the domain differences. This problem can be solved by either reducing the domain sizes—by clustering them into subdomains for example—or by using adaptation with more capacity and influence over the model.

3.5 Experiments in decoding with subword units

In this section we experiment with different combinations of lexical units and decoders. N-gram LMs used modified Kneser-Ney smoothing and were trained using the VariKN package [34]. Maximum order of the n-grams was 3 for word n-grams and 6 for subword n-grams. Relatively large n-gram models with respect to the corpus sizes were used in all the experiments. Word error rates for the models trained on the larger training corpora may be found in Table 8 and for the smaller training corpora in Table 9.

Table 8 Word error rates for the models trained on the larger training corpora

	Finnish		Estonian	
Units	Vocabulary size	WER	Vocabulary size	WER
Words	2.2M	32.1 %	1.6M	16.2 %
Subwords	2.2M	32.1 %	1.6M	15.6 %
Subwords	-	31.2 %	-	15.1 %
Subwords	2.2M + OOV	31.0 %	1.6M + OOV	14.9 %

Table 9 Word error rates for the models trained on the smaller training corpora

	Finnish		Estonian	
Units	Vocabulary size	WER	Vocabulary size	WER
Words	850k	35.2 %	550k	19.6 %
Subwords	850k	35.2 %	550k	18.7 %
Subwords	-	33.6 %	-	17.7 %
Subwords	2.2M	34.0 %	1.6M	17.8 %
Subwords	2.2M + OOV	32.8 %	1.6M + OOV	17.1 %

The first observation from the results is that effectively very large vocabularies are needed to obtain good ASR performance on the broadcast news task for both languages, irrespective of the way of modeling. If more was known about the topics to be recognized, more limited vocabularies could be utilized. Accurate topic modelling, however, would likely require more resources than assumed to be available here. The results also show, that the standard dynamic token-passing decoding can effectively operate with very large vocabularies, if care is taken in the implementation [36, 39].

In terms of error rates, including all the word forms from the LM training data to the vocabulary seems to give reasonable initial results. In the Finnish experiments, word n-grams and subword n-grams performed equally well with these very large vocabularies in both the settings. The OOV-rates were still 3.2 % and 5.3 %, indicating some mismatch between the training corpus and the recognition task. In the Estonian experiments, the subword n-grams outperformed the word n-grams with the same vocabulary in both the settings. It thus seems, that subword n-grams provide better probability estimates in some cases. The OOV-rates in the Estonian experiments were 1.2 % and 2.5 %.

We also experimented with a subword decoder, which enables an unlimited recognition vocabulary and did simulated experiments, where the recognition vocabulary was augmented by the remaining OOV words and in the smaller corpus setting using the vocabulary from the larger corpus instead. The words were segmented using the n-gram model and added to the decoding graph. The subword n-gram model was not modified.

In the large corpus setting, the relative error rate reductions for the unlimited recognition vocabulary were 2.8 % and 3.2 %, compared to the best restricted vocabulary recognizer. The corresponding numbers for the closed vocabulary experiment were 3.4 % and 4.5 %. The results show, that the OOV words were still causing many recognition errors. In this case opting for unlimited vocabulary recognition was quite effective in bridging the gap between the initial and the closed vocabulary.

In the small corpus setting, the relative improvements for unlimited vocabulary recognition were 4.5 % for Finnish and 5.3 % for Estonian. By using the vocabulary from the large corpus, the corresponding results were 3.5 % for Finnish and 4.8 % for Estonian. Adding the remaining OOV-words further improved WER by 3.5 % and 3.9 %. In this setting, it may be seen that the OOV-rate had quite a big impact on the recognition rates. Also, the difference between the unlimited and the closed vocabulary results increased, indicating that the quality of the n-gram estimates started to suffer.

In unlimited vocabulary recognition, also some non-words will be recognized. This may be an annoyance in some ASR use cases. The rate of the non-words will depend much on the task at hand. The results further show, that a restricted vocabulary which is closed or nearly closed, should give the best recognition results. In this case also non-words will be avoided. The question then becomes, in which cases is this a realistic goal? The subword n-gram decoder with a restricted vocabulary opens some new possibilities towards this end, as the vocabulary may be augmented without having all the word forms in the training text corpus. Other data sources, like dictionaries and morphological analyzers (generators), can be used to enrich the vocabulary. This could be especially helpful for less-resourced languages, for which sufficiently large text corpora are mostly not available. It has been estimated, that with entry generators [18], a native linguist may annotate 300–400 new words in an hour to a morphological analyzer lexicon. For the initial lexicon, around 5000 annotated words may suffice. Also in use cases, where the ASR system will be used repeatedly, it may be possible to cover the most important missing words over time.

4 Conclusion

In this work several recently developed language modeling methods were evaluated in LVCSR. The evaluations were performed in two agglutinative languages, Finnish and Estonian. Although language technology in these two languages have not been very widely developed, most of the benchmarking

tasks we used are almost directly comparable to previous work. For the smaller agglutinative languages that are extremely under-resourced, such as Northern Sami, proper evaluations are still impossible. However, by verifying the same evaluations in parallel for both Finnish and Estonian, and by artificially reducing the training data, we managed to make simulations that are realistic for less resourced languages. This allows us to conclude how to collect new data and what methods are suitable for languages with a limited amount of language model training data.

The first task we evaluated was LM training data collection. Although training data for planned speech is relatively easy to collect e.g. from news wire, conversational speech pose a more difficult problem. The best training data would be real conversations, but they are expensive to transcribe. However, we managed to demonstrate a reasonable performance by clever filtering of Internet discussion forums. Reducing data size is essential, not only from the perspective of improving LM accuracy, but also because it makes modeling easier. The most compact training set can be obtained by relative entropy minimization based filtering. The vast reduction in data size may enable new approaches to language modeling, such as NNLMs.

The second evaluation was dealing with the pronunciation and language modeling of foreign words. It is very typical for small languages to borrow new words from English and other large languages. However, the pronunciation of these words do not usually follow the same pronunciation rules as native words and the pronunciation used in practice is often unpredictable. Furthermore foreign words are often topic-specific and poorly estimated by the baseline LM. Our results indicate the we can successfully improve recognition of foreign words with unsupervised LM and vocabulary adaptation. However, generating multiple pronunciation variants for foreign names negatively affects the recognition of some native words. Discriminative pronunciation pruning did improve recognition slightly over the development sets but the pruned models didn't have as much effect on unseen data in the form of the evaluation sets. It is possible that discriminative pronunciation pruning is more effective over larger data sets. We evaluated a shared resource by using a G2P model originally trained for Finnish on Estonian. Results indicate that the model does improve recognition of foreign words in Estonian as well but the added lexical confusion which impacts the recognition of native words seems to be worse than in Finnish. Improving pruning methods and testing over larger data sets need to be done in the future to better understand the feasibility of G2P model sharing between languages.

The results of the third evaluation show that the proposed multi-domain and adapted NNLMs consistently outperform the n-gram baseline and simple NNLMs in terms of PPL. The proposed model provides statistically significant WER improvements compared to the n-gram baseline, but fails to improve upon simple NNLMs. The results appear to be similar in both multi-domain and adaptation modes. Finding better and more clever methods, rather than just more data, to improve the target-domain performance is important for under-resourced languages, because it is not expected that sufficient amount

of in-domain data can be collected for any particular topic or style alone. In our future work we plan to address the lack of WER improvements of multi-domain and adapted models over simple NNLMs by exploring sub-domain level multi-domain models and more powerful adaptation methods.

The last evaluation concerned the different combinations of lexical units and decoding approaches. For agglutinative languages, such as Finnish, Estonian and Sami, subword LMs have many advantages. In the broadcast news experiments, n-gram models trained over subwords performed equally well or better than word n-grams with the same recognition vocabulary. Further advantage is that the subword n-grams are able to assign probabilities to unseen word forms. Decoding with unlimited vocabulary improved recognition accuracy for both languages. Using subword n-grams but still opting for a restricted vocabulary is also a viable alternative, which avoids the recognition of nonsense words. We expect that the ability of quickly adding new words for the search network may become useful if there are important OOV words that the system should recognize better. Also, the results indicated, that in the cases where the recognition vocabulary is closed or nearly closed, better results will be reached with a restricted vocabulary. Much depends on the recognition task and the available resources, if this is a realistic goal.

The next step in our project is to gather and build the resources for constructing and evaluating LVCSR in Northern Sami, where all the results of this paper should become useful. The word error rates from conversational Finnish and Estonian speech recognition experiments are still above 50 %. One area where we still clearly need to improve is acoustic modeling. Accurately transcribed spontaneous conversations are hard to find, so we have had to combine data from many small corpora of varying quality. More intelligent combination of these data sources by model adaptation or neural network models would certainly help, and will be done in the future.

Acknowledgements This work was partially funded by the Estonian Ministry of Education and Research target-financed research theme no. 0140007s12, by the Tallinn University of Technology project Estonian Speech Recognition System for Medical Applications, by the Academy of Finland under the grant number 251170 (Finnish Centre of Excellence Program (2012–2017)), and by Finnish Cultural Foundation. We acknowledge the computational resources provided by Aalto Science-IT project.

References

1. Adde, L., Svendsen, T.: Pronunciation variation modeling of non-native proper names by discriminative tree search. In: Proc. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4928–4931. Prague, Czech Republic (2011)
2. Alumäe, T.: Multi-domain neural network language model. In: Proc. 14th Annual Conference of the International Speech Communication Association (INTERSPEECH 2013), pp. 2182–2186 (2013)
3. Alumäe, T.: Recent improvements in Estonian LVCSR. In: Spoken Language Technologies for Under-Resourced Languages. St. Petersburg, Russia (2014)
4. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of Machine Learning Research **3**, 1137–1155 (2003)

5. Bisani, M., Ney, H.: Joint-sequence models for grapheme-to-phoneme conversion. Speech Communication **50**(5), 434–451 (2008). DOI 10.1016/j.specom.2008.01.002. URL http://dx.doi.org/10.1016/j.specom.2008.01.002
6. Creutz, M., Lagus, K.: Unsupervised discovery of morphemes. In: Proc. ACL 2002 workshop on morphological and phonological learning, *MPL '02*, vol. 6, pp. 21–30. Association for Computational Linguistics, Stroudsburg, PA, USA (2002). DOI 10.3115/1118647.1118650. URL http://dx.doi.org/10.3115/1118647.1118650
7. Deligne, S., Bimbot, F.: Inference of variable-length linguistic and acoustic units by multigrams. Speech Communication **23**(3), 223–241 (1997)
8. Enarvi, S., Kurimo, M.: A novel discriminative method for pruning pronunciation dictionary entries. In: Proc. 7th International Conference on Speech Technology and Human-Computer Dialogue, pp. 113–116. Cluj-Napoca, Romania (2013)
9. Enarvi, S., Kurimo, M.: Studies on training text selection for conversational finnish language modeling. In: Proc. 10th International Workshop on Spoken Language Translation (IWSLT 2013), pp. 256–263. Heidelberg, Germany (2013)
10. Gandhe, A., Metze, F., Lane, I.: Neural network language models for low resource languages. In: Proc. 15th Annual Conference of the International Speech Communication Association (INTERSPEECH 2014) (2014)
11. Hirsimäki, T., Kurimo, M.: Analysing recognition errors in unlimited-vocabulary speech recognition. In: Proc. North American Chapter of the Association for Computational Linguistics - Human Language Technologies 2009 conference (NAACL 2009), pp. 193–196. Boulder, Colorado, USA (2009)
12. Hirsimäki, T., Pylkkönen, J., Kurimo, M.: Importance of high-order n-gram models in morph-based speech recognition. IEEE Transactions on Audio, Speech & Language Processing **17**(4), 724–732 (2009). DOI 10.1109/TASL.2008.2012323. URL http://dx.doi.org/10.1109/TASL.2008.2012323
13. Iskra, D.J., Grosskopf, B., Marasek, K., van den Heuvel, H., Diehl, F., Kießling, A.: SPEECON - speech databases for consumer devices: Database specification and validation. In: Proc. Third International Conference on Language Resources and Evaluation (LREC 2002). Canary Islands, Spain (2002)
14. Klakow, D.: Selecting articles from the language model training corpus. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2000), vol. 3, pp. 1695–1698. IEEE Computer Society (2000). DOI 10.1109/ICASSP.2000.862077
15. Lehecka, J., Svec, J.: Improving speech recognition by detecting foreign inclusions and generating pronunciations. Text, Speech, and Dialogue, Lecture Notes in Computer Science **8082**, 295–302 (2013)
16. Leinonen, J.: Automatic speech recognition for human-robot interaction using an under-resourced language. Master's thesis, Aalto University, School of Electrical Engineering, Department of Signal Processing and Acoustics, Espoo (2015)
17. Lennes, M.: Segmental features in spontaneous and read-aloud Finnish. In: V. de Silva, R. Ullakonoja (eds.) Phonetics of Russian and Finnish. General Introduction. Spontaneous and Read-Aloud Speech, pp. 145–166. Peter Lang GmbH (2009)
18. Linden, K.: Entry generation for new words by analogy for morphological lexicons. Northern European Journal of Language Technology **1**, 1–25 (2009)
19. Maison, B., Chen, S., Cohen, P.S.: Pronunciation modeling for names of foreign origin. In: Proc. 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 429–434 (2003)
20. Mansikkaniemi, A., Kurimo, M.: Unsupervised topic adaptation for morph-based speech recognition. In: Proc. 14th Annual Conference of the International Speech Communication Association (INTERSPEECH 2013), pp. 2693–2697. Lyon, France (2013)
21. Moore, R.C., Lewis, W.: Intelligent selection of language model training data. In: Proc. ACL 2010 Conference Short Papers, ACLShort '10, pp. 220–224. Association for Computational Linguistics, Stroudsburg, PA, USA (2010). URL http://dl.acm.org/citation.cfm?id=1858842.1858883
22. Ney, H., Ortmanns, S.: Progress in dynamic programming search for LVCSR. Proc. IEEE **88**(8), 1224–1240 (2000)

23. Park, J., Liu, X., Gales, M.J., Woodland, P.C.: Improved neural network based language modelling and adaptation. In: Proc. 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010), pp. 1041–1044 (2010)
24. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: Proc. 2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE Signal Processing Society (2011). IEEE Catalog No.: CFP11SRW-USB
25. Povey, D., Kanevsky, D., Kingsbury, B., Ramabhadran, B., Saon, G., Visweswariah, K.: Boosted mmi for model and feature-space discriminative training. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008), pp. 4057–4060. IEEE (2008)
26. Pylkkönen, J.: An efficient one-pass decoder for finnish large vocabulary continuous speech recognition. In: Proc. 2nd Baltic Conference on Human Language Technologies (2005)
27. Saon, G., Soltau, H., Nahamoo, D., Picheny, M.: Speaker adaptation of neural network acoustic models using i-vectors. In: Proc. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 55–59. IEEE (2013)
28. Schwenk, H., Gauvain, J.L.: Neural network language models for conversational speech recognition. In: Proc. 8th International Conference on Spoken Language Processing (INTERSPEECH 2004) (2004)
29. Schwenk, H., Gauvain, J.L.: Training neural network language models on very large corpora. In: Proc. conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 201–208. Association for Computational Linguistics (2005)
30. Seide, F., Li, G., Chen, X., Yu, D.: Feature engineering in context-dependent deep neural networks for conversational speech transcription. In: Proc. 2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 24–29. IEEE (2011)
31. Sethy, A., Georgiou, P.G., Narayanan, S.: Text data acquisition for domain-specific language models. In: Proc. 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06, pp. 382–389. Association for Computational Linguistics, Stroudsburg, PA, USA (2006). URL `http://dl.acm.org/citation.cfm?id=1610075.1610129`
32. Sethy, A., Georgiou, P.G., Ramabhadran, B., Narayanan, S.S.: An iterative relative entropy minimization-based data selection approach for n-gram model adaptation. IEEE Transactions on Audio, Speech, and Language Processing **17**(1), 13–23 (2009)
33. Shi, Y., Larson, M., Jonker, C.M.: Recurrent neural network language model adaptation with curriculum learning. Computer Speech & Language (2014). DOI http://dx.doi.org/10.1016/j.csl.2014.11.004
34. Siivola, V., Hirsimäki, T., Virpioja, S.: On growing and pruning kneser-ney smoothed n-gram models. IEEE Transactions on Speech, Audio and Language Processing **15**(5), 1617–1624 (2007)
35. Sixtus, A., Ney, H.: From within-word model search to across-word model search in large vocabulary continuous speech recognition. Computer Speech and Language **16**(2), 245–271 (2002)
36. Soltau, H., Saon, G.: Dynamic network decoding revisited. In: Proc. 2009 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 276–281 (2009)
37. Tilk, O., Alumäe, T.: Multi-domain recurrent neural network language model for medical speech recognition. In: Human Language Technologies – The Baltic Perspective, vol. 268, pp. 149–152. IOS Press (2014)
38. Varjokallio, M., Kurimo, M.: A toolkit for efficient learning of lexical units for speech recognition. In: Proc. Ninth International Conference on Language Resources and Evaluation (LREC 2014). Reykjavik, Iceland (2014)
39. Varjokallio, M., Kurimo, M.: A word-level token-passing decoder for subword n-gram LVCSR. In: Proc. 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 495–500. South Lake Tahoe, California and Nevada (2014). DOI 10.1109/SLT.2014.7078624
40. Varjokallio, M., Kurimo, M., Virpioja, S.: Learning a subword vocabulary based on unigram likelihood. In: Proc. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). Olomouc, Czech Republic (2013)

41. Young, S.J., Russell, N.H., Thornton, J.H.S.: Token passing: a simple conceptual model for connected speech recognition system. Tech. rep., Cambridge University Engineering Department (1989)

Erratum

The Finnish NNLM models were based on subword units and the Estonian on compound-split words, while few sentences in the published paper in Section 3.4 erroneously claimed the opposite.