Web augmentation of language models for continuous speech recognition of SMS text messages

Mathias Creutz[1], **Sami Virpioja**[1,2] and **Anna Kovaleva**[1]
[1]Nokia Research Center, Helsinki, Finland
[2]Adaptive Informatics Research Centre, Helsinki University of Technology, Espoo, Finland
mathias.creutz@nokia.com, sami.virpioja@tkk.fi, annakov@gmx.de

Abstract

In this paper, we present an efficient query selection algorithm for the retrieval of web text data to augment a statistical language model (LM). The number of retrieved relevant documents is optimized with respect to the number of queries submitted.

The querying scheme is applied in the domain of SMS text messages. Continuous speech recognition experiments are conducted on three languages: English, Spanish, and French. The web data is utilized for augmenting in-domain LMs in general and for adapting the LMs to a user-specific vocabulary. Word error rate reductions of up to 6.6 % (in LM augmentation) and 26.0 % (in LM adaptation) are obtained in setups, where the size of the web mixture LM is limited to the size of the baseline in-domain LM.

1 Introduction

An automatic speech recognition (ASR) system consists of acoustic models of speech sounds and of a statistical language model (LM). The LM learns the probabilities of word sequences from text corpora available for training. The performance of the model depends on the amount and style of the text. The more text there is, the better the model is, in general. It is also important that the model be trained on text that matches the style of language used in the ASR application. Well matching, in-domain, text may be both difficult and expensive to obtain in the large quantities that are needed.

A popular solution is to utilize the World Wide Web as a source of additional text for LM training. A small in-domain set is used as seed data, and more data of the same kind is retrieved from the web. A decade ago, Berger and Miller (1998) proposed a just-in-time LM that updated the current LM by retrieving data from the web using recent recognition hypotheses as queries submitted to a search engine. Perplexity reductions of up to 10 % were reported.[1] Many other works have followed. Zhu and Rosenfeld (2001) retrieved page and phrase counts from the web in order to update the probabilities of infrequent trigrams that occur in N-best lists. Word error rate (WER) reductions of about 3 % were obtained on TREC-7 data.

In more recent work, the focus has turned to the collection of text rather than n-gram statistics based on page counts. More effort has been put into the selection of query strings. Bulyko et al. (2003; 2007) first extend their baseline vocabulary with words from a small in-domain training corpus. They then use n-grams with these new words in their web queries in order to retrieve text of a certain genre. For instance, they succeed in obtaining conversational style phrases, such as "we were friends but we don't actually have a relationship." In a number of experiments, word error rate reductions of 2-3 % are obtained on English data, and 6 % on Mandarin. The same method for web data collection is applied by Çetin and Stolcke (2005) in meeting and lecture transcription tasks. The web sources reduce perplexity by 10 % and 4.3 %, respectively, and word error rates by 3.5 % and 2.2 %, respectively.

Sarikaya et al. (2005) chunk the in-domain text into "n-gram islands" consisting of only content words and excluding frequently occurring stop words. An island such as "stock fund portfolio" is then extended by adding context, producing "my stock fund portfolio", for instance. Multiple islands are combined using *and* and *or* operations to form web queries. Significant word error reductions between 10 and 20 % are obtained; however, the in-domain data set is very small, 1700 phrases,

[1]All reported percentage differences are *relative* unless explicitly stated otherwise.

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 157–165,
Athens, Greece, 30 March – 3 April 2009.

which makes (any) new data a much needed addition.

Similarly, Misu and Kawahara (2006) obtain very good word error reductions (20 %) in spoken dialogue systems for software support and sightseeing guidance. Nouns that have high tf/idf scores in the in-domain documents are used in the web queries. The existing in-domain data sets poorly match the speaking style of the task and therefore existing dialogue corpora of different domains are included, which improves the performance considerably.

Wan and Hain (2006) select query strings by comparing the n-gram counts within an in-domain topic model to the corresponding counts in an out-of-domain background model. Topic-specific n-grams are used as queries, and perplexity reductions of 5.4 % are obtained.

It is customary to postprocess and filter the downloaded web texts. Sentence boundaries are detected using some heuristics. Text chunks with a high out-of-vocabulary (OOV) rate are discarded. Additionally, the chunks are often ranked according to their similarity with the in-domain data, and the lowest ranked chunks are discarded. As a similarity measure, the *perplexity* of the sentence according to the in-domain LM can be used; for instance, Bulyko et al. (2007). Another measure for ranking is *relative perplexity* (Weilhammer et al., 2006), where the in-domain perplexity is divided by the perplexity given by an LM trained on the web data. Also the BLEU score familiar from the field of machine translation has been used (Sarikaya et al., 2005).

Some criticism has been raised by Sethy et al. (2007), who claim that sentence ranking has an inherent bias towards the center of the in-domain distribution. They propose a data selection algorithm that selects a sentence from the web set, if adding the sentence to the already selected set reduces the relative entropy with respect to the in-domain data distribution. The algorithm appears efficient in producing a rather small subset (1/11) of the web data, while degrading the WER only marginally.

The current paper describes a new method for query selection and its applications in LM augmentation and adaptation using web data. The language models are part of a continuous speech recognition system that enables users to use speech as an input modality on mobile devices, such as mobile phones. The particular domain of interest is personal communication: The user dictates a message that is automatically transcribed into text and sent to a recipient as an SMS text message. Memory consumption and computational speed are crucial factors in mobile applications. While most studies ignore the sizes of the LMs when comparing models, we aim at improving the LM *without increasing its size* when web data is added.

Another aspect that is typically overlooked is that the collection of web data costs time and computational resources. This applies to the querying, downloading and postprocessing of the data. The query selection scheme proposed in this paper is *economical* in the sense that it strives to download as much relevant text from the web as possible using as few queries as possible avoiding overlap between the set of pages found by different queries.

2 Query selection and web data retrieval

Our query selection scheme involves multiple steps. The assumption is that a batch of queries will be created. These queries are submitted to a search engine and the matching documents are downloaded. This procedure is repeated for multiple query batches.

In particular, our scheme attempts to maximize the number of retrieved relevant documents, when two restrictions apply: (1) queries are not “free”: each query costs some time or money; for instance, the number of queries submitted within a particular period of time is limited, and (2) the number of documents retrieved for a particular query is limited to a particular number of “top hits”.

2.1 N-gram selection and prospection querying

Some text reflecting the target domain must be available. A set of the most frequent n-grams occurring in the text is selected, from unigrams up to five-grams. Some of these n-grams are characteristic of the domain of interest (such as “Hogwarts School of Witchcraft and Wizardry”), others are just frequent in general (“but they did not say”); we do not know yet which ones.

All n-grams are submitted as queries to the web search engine. Exact matches of the n-grams are required; different inflections or matches of the words individually are not accepted.

The search engine returns the total number of hits $h(q_s)$ for each query q_s as well as the URLs of a predefined maximum number of "top hit" web pages. The top hit pages are downloaded and post-processed into plain text, from which duplicate paragraphs and paragraphs with a high OOV rate are removed.

N-gram language models are then trained separately on the in-domain text and the the filtered web text. If the amount of web text is very large, only a subset is used, which consists of the parts of the web data that are the most similar to the in-domain text. As a similarity measure, relative perplexity is used. The LM trained on web data is called a *background LM* to distinguish it from the *in-domain LM*.

2.2 Focused querying

Next, the querying is made more specific and targeted on the domain of interest. New queries are created that consist of n-gram pairs, requiring that a document contain two n-grams ("but they did not say"+"Hogwarts School of Witchcraft and Wizardry").[2]

If all possible n-gram pairs are formed from the n-grams selected in Section 2.1, the number of pairs is very large, and we cannot afford using them all as queries. Typical approaches for query selection include the following: (i) select pairs that include n-grams that are relatively more frequent in the in-domain text than in the background text, (ii) use some extra source of knowledge for selecting the best pairs.

2.2.1 Extra linguistic knowledge

We first tested the second (ii) query selection approach by incorporating some simple linguistic knowledge: In an experiment on English, queries were obtained by combining a highly frequent n-gram with a slightly less frequent n-gram that had to contain a first- or second-person pronoun (I, you, we, me, us, my, your, our). Such n-grams were thought to capture direct speech, which is characteristic for the desired genre of personal communication. (Similar techniques are reported in the literature cited in Section 1.)

Although successful for English, this scheme is more difficult to apply to other languages, where person is conveyed as verbal suffixes rather than single words. Linguistic knowledge is needed for every language, and it turns out that many of the queries are "wasted", because they are too specific and return only few (if any) documents.

2.2.2 Statistical approach

The other proposed query selection technique (i) allows for an automatic identification of the n-grams that are characteristic of the in-domain genre. If the relative frequency of an n-gram is higher in the in-domain data than in the background data, then the n-gram is potentially valuable. However, as in the linguistic approach, there is no guarantee that queries are not wasted, since the identified n-gram may be very rare on the Internet. Pairing it with some other n-gram (which may also be rare) often results in very few hits.

To get out the most of the queries, we propose a query selection algorithm that attempts to optimize the relevance of the query to the target domain, but also takes into account the expected amount of data retrieved by the query. Thus, the potential queries are ranked according to the *expected number of retrieved relevant documents*. Only the highest ranked pairs, which are likely to produce the highest number of relevant web pages, are used as queries.

We denote queries that consist of two n-grams s and t by $q_{s\wedge t}$. The expected number of retrieved relevant documents for the query $q_{s\wedge t}$ is $r(q_{s\wedge t})$:

$$r(q_{s\wedge t}) = n(q_{s\wedge t}) \cdot \rho(q_{s\wedge t} \mid Q), \qquad (1)$$

where $n(q_{s\wedge t})$ is the expected number of retrieved documents for the query, and $\rho(q_{s\wedge t} \mid Q)$ is the expected proportion of relevant documents within all documents retrieved by the query. The expected proportion of relevant documents is a value between zero and one, and as explained below, it is dependent on all past queries, the query history Q.

Expected number of retrieved documents $n(q_{s\wedge t})$. From the prospection querying phase (Section 2.1), we know the numbers of hits for the single n-grams s and t, separately: $h(q_s)$ and $h(q_t)$. We make the operational, but overly simplifying, assumption that the n-grams occur evenly distributed over the web collection, independently of each other. The expected size of the intersection $q_{s\wedge t}$ is then:

$$\hat{h}(q_{s\wedge t}) = \frac{h(q_s) \cdot h(q_t)}{N}, \qquad (2)$$

where N is the size of the web collection that our n-gram selection covers (total number of docu-

[2] Higher order tuples could be used as well, but we have only tested n-gram pairs.

ments). N is not known, but different estimates can be used, for instance, $N = \max_{\forall q_s} h(q_s)$, where it is assumed that the most frequent n-gram occurs in every document in the collection (probably an underestimate of the actual value).

Ideally, the expected number of retrieved documents equals the expected number of hits, but since the search engine returns a limited maximum number of "top hit" pages, M, we get:

$$n(q_{s\wedge t}) = \min(\hat{h}(q_{s\wedge t}), M). \quad (3)$$

Expected proportion of relevant documents $\rho(q_{s\wedge t} \,|\, Q)$. As in the case of $n(q_{s\wedge t})$, an independence assumption can be applied in the derivation of the expected proportion of relevant documents for the combined query $q_{s\wedge t}$: We simply put together the chances of obtaining relevant documents by the single n-gram queries q_s and q_t individually. The union equals:

$$\rho(q_{s\wedge t} \,|\, Q) = 1 - \big(1 - \rho(q_s \,|\, Q)\big) \cdot \big(1 - \rho(q_t \,|\, Q)\big). \quad (4)$$

However, we do not know the values for $\rho(q_s \,|\, Q)$ and $\rho(q_t \,|\, Q)$. As mentioned earlier, it is straightforward to obtain a relevance *ranking* for a set of n-grams: For each n-gram s, the LM probability is computed using both the in-domain and the background LM. The in-domain probability is divided by the background probability and the n-grams are sorted, highest relative probability first. The first n-gram is much more prominent in the in-domain than the background data, and we wish to obtain more text with this crucial n-gram. The opposite is true for the last n-gram.

We need to transform the ranking into $\rho(\cdot)$ values between zero and one. There is no absolute division into relevant and irrelevant documents from the point of view of LM training. We use a probabilistic query ranking scheme, such that we define that of all documents containing an x % relevant n-gram, x % are relevant. When the n-grams have been ranked into a presumed order of relevance, we decide that the most relevant n-gram is 100 % relevant and the least relevant n-gram is 0 % relevant; finally, we scale the relevances of the other n-grams according to rank.

When scoring the remaining n-grams, linear scaling is avoided, because the majority of the n-grams are irrelevant or neutral with respect to our domain of interest, and many of them would obtain fairly high relevance values. Instead, we fix the relevance value of the "most domain-neutral" n-gram (the one with the relative probability value closest to one); we might assume that only 5 % of all documents containing this n-gram are indeed relevant. We then fit a polynomial curve through the three points with known values (0, 0.05, and 1) to get the missing $\rho(\cdot)$ values for all q_s.

Decay factor $\delta(s \,|\, Q)$. We noticed that if constant relevance values are used, the top ranked queries will consist of a rather small set of top ranked n-grams that are paired with each other in all possible combinations. However, it is likely that each time an n-gram is used in a query, the need for finding more occurrences of this particular n-gram decreases. Therefore, we introduced a decay factor $\delta(s \,|\, Q)$, by which the initial $\rho(\cdot)$ value, written $\rho_0(q_s)$, is multiplied:

$$\rho(q_s \,|\, Q) = \rho_0(q_s) \cdot \delta(s \,|\, Q), \quad (5)$$

The decay is exponential:

$$\delta(s \,|\, Q) = (1 - \epsilon)^{\sum_{\forall s \in Q} 1}. \quad (6)$$

ϵ is a small value between zero and one (for instance 0.05), and $\sum_{\forall s \in Q} 1$ is the number of times the n-gram s has occurred in past queries.

Overlap with previous queries. Some queries are likely to retrieve the same set of documents as other queries. This occurs if two queries share one n-gram and there is strong correlation between the second n-grams (for instance, "we wish you"+"Merry Christmas" vs. "we wish you"+ "and a Happy New Year"). In principle, when assessing the relevance of a query, one should estimate the overlap of that query with all past queries. We have tested an approximate solution that allows for fast computing. However, the real effect of this addition was insignificant, and a further description is omitted in this paper.

Optimal order of the queries. We want to maximize the expected number of retrieved relevant documents while keeping the number of submitted queries as low as possible. Therefore we sort the queries best first and submit as many queries we can afford from the top of the list. However, the relevance of a query is dependent on the sequence of past queries (because of the decay factor). Finding the optimal order of the queries takes $O(n^2)$ operations, if n is the total number of queries.

A faster solution is to apply an iterative algorithm: All queries are put in some initial order. For

each query, its $r(q_{s \wedge t})$ value is computed according to Equation 1. The queries are then rearranged into the order defined by the new $r(\cdot)$ values, best first. These two steps are repeated until convergence.

Repeated focused querying. Focused querying can be run multiple times. Some ten thousands of the top ranked queries are submitted to the search engine and the documents matching the queries are downloaded. A new background LM is trained using the new web data, and a new round of focused querying can take place.

2.2.3 Comparison of the linguistic and statistical focused querying schemes

On one language (German), the statical focused querying algorithm (Section 2.2.2) was shown to retrieve 50 % more unique web pages and 70 % more words than the linguistic scheme (Section 2.2.1) for the same number of queries. Also results from language modeling and speech recognition experiments favored statistical querying.

2.3 Web collections obtained

For the speech recognition experiments described in the current paper, we have collected web texts for three languages: US English, European Spanish, and Canadian French.

As in-domain data we used 230,000 English text messages (4 million words), 65,000 Spanish messages (2 million words), and 60,000 French messages (1 million words). These text messages were obtained in data collection projects involving thousand of participants, who used a web interface to enter messages according to different scenarios of personal communication situations.[3] A few example messages are shown in Figure 1.

The queries were submitted to Yahoo!'s web search engine. The web pages that were retrieved by the queries were filtered and cleaned and divided into chunks consisting of single paragraphs. For English, we obtained 210 million paragraphs and 13 billion words, for Spanish 160 million paragraphs and 12 billion words, and for French 44 million paragraphs and 3 billion words.

[3]Real messages sent from mobile phones would be the best data, but are hard to get because of privacy protection. The postprocessing of authentic messages would, however, require proper handling of artifacts resulting from the limited input capacities on keypads of mobile devices, such as specific acronyms: *i'll c u l8er*. In our setup, we did not have to face such issues.

I hope you have a long and happy marriage. Congratulations!
Remember to pick up Billy at practice at five o'clock!
Hey Eric, how was the trip with the kids over winter vacation? Did you go to Texas?

Figure 1: Example text messages (US English).

The linguistic focused querying method was applied in the US English task (because the statistical method did not yet exist). The Spanish and Canadian French web collections were obtained using statistical querying. Since the French set was smaller than the other sets ("only" 3 billion words), web crawling was performed, such that those web sites that had provided us with the most valuable data (measured by relative perplexity) were downloaded entirely. As a result, the number of paragraphs increased to 110 million and the number of words to 8 billion.

3 Speech Recognition Experiments

We have trained language models on the in-domain data together with web data, and these models have been used in speech recognition experiments. Two kinds of experiments have been performed: (1) the in-domain LM is augmented with web data, and (2) the LM is adapted to a user-specific vocabulary utilizing web data as an additional data source.

One hundred native speakers for each language were recorded reading held-out subsets of the in-domain text data. The speech data was partitioned into training and test sets, such that around one fourth of the speakers were reserved for testing.

We use a continuous speech recognizer optimized for low memory footprint and fast recognition (Olsen et al., 2008). The recognizer runs on a server (Core2 2.33 GHz) in about one fourth of real time. The LM probabilities are quantized and precompiled together with the speaker-independent acoustic models (intra-word triphones) into a finite state transducer (FST).

3.1 Language model augmentation

Each paragraph in the web data is treated as a potential text message and scored according to its similarity to the in-domain data. Relative perplexity is used as the similarity measure. The paragraphs are sorted, lowest relative perplexity first,

US English				
FST size [MB]	10	20	40	70
In-domain	42.7	40.1	39.1	–
Web mixture	42.0	37.6	35.7	33.8
Ppl reduction [%]	1.6	6.2	8.7	13.6

European Spanish				
FST size [MB]	10	20	25	40
In-domain	68.0	64.6	64.3	–
Web mixture	63.9	58.4	55.0	52.1
Ppl reduction [%]	6.0	9.6	14.5	19.0

Canadian French				
FST size [MB]	10	20	25	50
In-domain	57.6	–	–	–
Web mixture	51.7	47.9	45.9	44.6
Ppl reduction [%]	10.2	16.8	20.3	22.6

Table 1: *Perplexities.*

US English				
FST size [MB]	10	20	40	70
In-domain	17.9	17.5	17.3	–
Web mixture	17.5	16.7	16.4	15.8
WER reduction	2.2	4.4	5.2	8.4

European Spanish				
FST size [MB]	10	20	25	40
In-domain	18.9	18.7	18.6	–
Web mixture	18.7	17.9	17.4	16.8
WER reduction	1.4	4.1	6.6	9.7

Canadian French				
FST size [MB]	10	20	25	50
In-domain	22.6	–	–	–
Web mixture	22.1	21.7	21.3	20.9
WER reduction	2.3	4.1	5.8	7.5

Table 2: *Word error rates* [%].

In the tables, the perplexity and word error rate reductions of the web mixtures are computed with respect to the in-domain models of the same size, if such models exist; otherwise the comparison is made to the largest in-domain model available.

and the highest ranked paragraphs are used as LM training data. The optimal size of the set depends on the test, but the largest chosen set contains 15 million paragraphs and 500 million words.

Separate LMs are trained on the in-domain data and web data. The two LMs are then linearly interpolated into a mixture model. Roughly the same interpolation weights (0.5) are obtained for the LMs, when the optimal value is chosen based on a held-out in-domain development test set.

3.1.1 Test set perplexities

In Table 1, the prediction abilities of the in-domain and web mixture language models are compared. As an evaluation measure we use perplexity calculated on test sets consisting of in-domain text. The comparison is performed on FSTs of different sizes. The FSTs contain the acoustic models, language model and lexicon, but the LM makes up for most of the size. The availability of data varies for the different languages, and therefore the FST sizes are not exactly the same across languages.

The LMs have been created using the SRI LM toolkit (Stolcke, 2002). Good-Turing smoothing with Katz backoff (Katz, 1987) has been used, and the different model sizes are obtained by pruning down the full models using entropy-based pruning (Stolcke, 1998). N-gram orders up to five have been tested: 5-grams always work best on the mixture models, whereas the best in-domain models are 4- or 5-grams.

For every language and model size, the web mixture model performs better than the corresponding in-domain model. The perplexity reductions obtained increase with the size of the model. Since it is possible to create larger mixture models than in-domain models, there are no in-domain results for the largest model sizes.

Especially if large models can be afforded, the perplexity reductions are considerable. The largest improvements are observed for French (between 10.2 % and 22.6 % relative). This is not surprising, as the French in-domain set is the smallest, which leaves much room for improvement.

3.1.2 Word error rates

Speech recognition results for the different LMs are given in Table 2. The results are consistent in the sense that the web mixture models outperform the in-domain models, and augmentation helps more with larger models. The largest word error rate reduction is observed for the largest Spanish model (9.7 % relative). All WER reductions are statistically significant (one-sided Wilcoxon signed-rank test; level 0.05) except the 10 MB Spanish setup.

Although the observed word error rate reductions are mostly smaller than the corresponding

perplexity reductions, the results are actually very good, when we consider the fact that considerable reductions in perplexity may typically translate into meager word error reductions; see, for instance, Rosenfeld (2000), Goodman (2001). This suggests that the web texts are very welcome complementary data that improve on the robustness of the recognition.

3.1.3 Modified Kneser-Ney smoothing

In the above experiments, Good-Turing (GT) smoothing with Katz backoff was used, although modified Kneser-Ney (KN) interpolation has been shown to outperform other smoothing methods (Chen and Goodman, 1999). However, as demonstrated by Siivola et al. (2007), KN smoothing is not compatible with simple pruning methods such as entropy-based pruning. In order to make a meaningful comparison, we used the revised Kneser pruning and Kneser-Ney growing techniques proposed by Siivola et al. (2007). For the three languages, we built KN models that resulted in FSTs of the same sizes as the largest GT in-domain models. The perplexities decreased 4–8%, but in speech recognition, the improvements were mostly negligible: the error rates were 17.0 for English, 18.7 for Spanish, and 22.5 for French.

For English, we also created web mixture models with KN smoothing. The error rates were 16.5, 15.9 and 15.7 for the 20 MB, 40 MB and 70 MB models, respectively. Thus, Kneser-Ney outperformed Good-Turing, but the improvements were small, and a statistically significant difference was measured only for the 40 MB LMs. This was expected, as it has been observed before that very simple smoothing techniques can perform well on large data sets, such as web data (Brants et al., 2007).

For the purpose of demonstrating the usefulness of our web data retrieval system, we concluded that there was no significant difference between GT and KN smoothing in our current setup.

3.2 Language model adaptation

In the second set of experiments we envisage a system that adapts to the user's own vocabulary. Some words that the user needs may not be included in the built-in vocabulary of the device, such as names in the user's contact list, names of places or words related to some specific hobby or other focus of interest.

Two adaptation techniques have been tested: (1) *Unigram adaptation* is a simple technique, in which user-specific words (for instance, names from the contact list) are added to the vocabulary. No context information is available, and thus only unigram probabilities are created for these words. (2) In *message adaptation*, the LM is augmented selectively with paragraphs of web data that contain user-specific words. Now, higher order n-grams can be estimated, since the words occur within passages of running text. This idea is not new: information retrieval has been suggested as a solution by Bigi et al. (2004) among others.

In our message adaptation, we have not created web queries dynamically on demand. Instead, we used the large web collections described in Section 2.3, from which we selected paragraphs containing user-specific words. We have tested both adaptation by pooling (adding the paragraphs to the original training data), and adaptation by interpolation (using the new data to train a separate LM, which is interpolated with the original LM). One million words from the web data were selected for each language. The adaptation was thought to take place off-line on a server.

3.2.1 Data sets

For each language, the adaptation takes place on two baseline models, which are the in-domain and web mixture LMs of Section 3.1; however, the amount of in-domain training data is reduced slightly (as explained below).

In order to evaluate the success of the adaptation, a simulated user-specific test set is created. This set is obtained by selecting a subset of a larger potential test set. Words that occur both in the training set and the potential test set and that are infrequent in the training set are chosen as the user-specific vocabulary. For Spanish and French, a training set frequency threshold of one is used, resulting in 606 and 275 user-specific words, respectively. For English the threshold is 5, which results in 99 words. All messages in the potential test set containing any of these words are selected into the user-specific test set. Any message containing user-specific words is removed from the in-domain training set. In this manner, we obtain a test set with a certain over-representation of a specific vocabulary, without biasing the word frequency distribution of the training set to any noticeable degree.

For comparison, performance is additionally computed on a generic in-domain test set, as be-

US English, 23 MB models				
Model	WER (reduction)			
	user-specific		in-domain	
In-domain	29.1	(–)	17.9	(–)
+unigram adapt.	*24.4*	(16.3)	*17.1*	(4.7)
+message adapt.	*21.6*	(26.0)	16.8	(6.0)
Web mixture	25.7	(11.8)	16.9	(5.9)
+unigram adapt.	*23.1*	(20.6)	*16.3*	(8.8)
+message adapt.	*22.2*	(23.8)	16.4	(8.5)

European Spanish, 23 MB models				
Model	WER (reduction)			
	user-specific		in-domain	
In-domain	25.3	(–)	18.6	(–)
+unigram adapt.	*23.4*	(7.7)	18.5	(0.3)
+message adapt.	*21.7*	(14.4)	*18.0*	(3.2)
Web mixture	21.9	(13.7)	17.5	(5.8)
+unigram adapt.	*21.5*	(15.3)	17.7	(5.0)
+message adapt.	*21.2*	(16.5)	17.7	(4.7)

Canadian French, 21 MB models				
Model	WER (reduction)			
	user-specific		in-domain	
In-domain	30.3	(–)	22.6	(–)
+unigram adapt.	*28.3*	(6.4)	22.5	(0.4)
+message adapt.	*26.6*	(12.1)	22.2	(1.8)
Web mixture	26.7	(11.8)	21.4	(5.1)
+unigram adapt.	*26.0*	(14.3)	21.4	(5.4)
+message adapt.	26.0	(14.2)	21.6	(4.3)

Table 3: *Adaptation, word error rates* [%]. Six models have been evaluated on two types of test sets: a user-specific test set with a higher number of user-specific words and a generic in-domain test set. The numbers in brackets are relative WER reductions [%] compared to the in-domain model. WER values for the unigram adaptation are rendered in italics, if the improvement obtained is statistically significant compared to the corresponding non-adapted model. WER values for the message adaptation are in italics, if there is a statistically significant reduction with respect to unigram adaptation.

fore. User-specific and generic development test sets are used for the estimation of optimal interpolation weights.

3.2.2 Results

The adaptation experiments are summarized in Table 3. Only medium sized FSTs (21–23 MB) have been tested. The two baseline models have been adapted using the simple unigram reweighting scheme and using selective web message augmentation. For the in-domain baseline, pooling works the best, that is, adding the web messages to the original in-domain training set. For the web mixture baseline, a mixture model is the only option; that is, one more layer of interpolation is added.

In the adaptation of the in-domain LMs, message selection is almost twice as effective as unigram adaptation for all data sets. Also the performance on the generic in-domain test set is slightly improved, because more training data is available.

Except for English, the best results on the user-specific test sets are produced by the adaptation of the web mixture models. The benefit of using message adaptation instead of simple unigram adaptation is smaller when we have a web mixture model as a baseline rather than an in-domain-only LM.

On the generic test sets, the adaptation of the web mixture makes a difference only for English. Since there were practically no singleton words in the English in-domain data, the user-specific vocabulary consists of words occurring at most five times. Thus, the English user-specific words are more frequent than their Spanish and French equivalents, which shows in larger WER reductions for English in all types of adaptation.

4 Discussion and conclusion

Mobile applications need to run in small memory, but not much attention is usually paid to memory consumption in related LM work. We have shown that LM augmentation using web data can be successful, even when the resulting mixture model is not allowed to grow any larger than the initial in-domain model. Yet, the benefit of the web data is larger, the larger model can be used.

The largest WER reductions were observed in the adaptation to a user-specific vocabulary. This can be compared to Misu and Kawahara (2006), who obtained similar accuracy improvements with clever selection of web data, when there was initially no in-domain data available with both the correct topic and speaking style.

We used relative perplexity ranking to filter the downloaded web data. More elaborate algorithms could be exploited, such as the one proposed by Sethy et al. (2007). Initially, we have experimented along those lines, but it did not pay off; maybe future refinements will be more successful.

References

Adam Berger and Robert Miller. 1998. Just-in-time language modeling. In *In ICASSP-98*, pages 705–708.

Brigitte Bigi, Yan Huang, and Renato De Mori. 2004. Vocabulary and language model adaptation using information retrieval. In *Proc. Interspeech 2004 – ICSLP*, pages 1361–1364, Jeju Island, Korea.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867.

Ivan Bulyko, Mari Ostendorf, and Andreas Stolcke. 2003. Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 7–9, Morristown, NJ, USA. Association for Computational Linguistics.

Ivan Bulyko, Mari Ostendorf, Manhung Siu, Tim Ng, Andreas Stolcke, and Özgür Çetin. 2007. Web resources for language modeling in conversational speech recognition. *ACM Trans. Speech Lang. Process.*, 5(1):1–25.

Özgür Çetin and Andreas Stolcke. 2005. Language modeling in the ICSI-SRI spring 2005 meeting speech recognition evaluation system. Technical Report 05-006, International Computer Science Institute, Berkeley, CA, USA, July.

S. F. Chen and J. Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13:359–394.

Joshua T. Goodman. 2001. A bit of progress in language modeling. *Computer Speech and Language*, 15:403–434.

Slava M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400–401, March.

Teruhisa Misu and Tatsuya Kawahara. 2006. A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts. In *Proc. INTERSPEECH '06*, pages 9–13, Pittsburgh, PA, USA, September, 17–21.

Jesper Olsen, Yang Cao, Guohong Ding, and Xinxing Yang. 2008. A decoder for large vocabulary continuous short message dictation on embedded devices. In *Proc. ICASSP 2008*, Las Vegas, Nevada.

Ronald Rosenfeld. 2000. Two decades of language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278.

Ruhi Sarikaya, Augustin Gravano, and Yuqing Gao. 2005. Rapid language model development using external resources for new spoken dialog domains. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, volume I, pages 573–576.

Abhinav Sethy, Shrikanth Narayanan, and Bhuvana Ramabhadran. 2007. Data driven approach for language model adaptation using stepwise relative entropy minimization. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '07)*, volume IV, pages 177–180.

Vesa Siivola, Teemu Hirsimäki, and Sami Virpioja. 2007. On growing and pruning Kneser-Ney smoothed n-gram models. *IEEE Transactions on Audio, Speech and Language Processing*, 15(5):1617–1624.

A. Stolcke. 1998. Entropy-based pruning of backoff language models. In *Proc. DARPA BNTU Workshop*, pages 270–274, Lansdowne, VA, USA.

A. Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proc. ICSLP*, pages 901–904. `http://www.speech.sri.com/projects/srilm/`.

Vincent Wan and Thomas Hain. 2006. Strategies for language model web-data collection. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '06)*, volume I, pages 1069–1072.

Karl Weilhammer, Matthew N. Stuttle, and Steve Young. 2006. Bootstrapping language models for dialogue systems. In *Proc. INTERSPEECH 2006 - ICSLP Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA, USA, September 17–21.

Xiaojin Zhu and R. Rosenfeld. 2001. Improving trigram language modeling with the world wide web. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01).*, volume 1, pages 533–536.