LANGUAGE MODEL ADAPTATION IN MACHINE TRANSLATION FROM SPEECH

Ivan Bulyko, Spyros Matsoukas, Richard Schwartz, Long Nguyen and John Makhoul

BBN Technologies
10 Moulton St., Cambridge, MA 02138
ibulyko@bbn.com

ABSTRACT

This paper investigates the use of several language model adaptation techniques applied to the task of machine translation from Arabic broadcast speech. Unsupervised and discriminative approaches slightly outperform the traditional perplexity-based optimization technique. Language model adaptation, when used for n-best rescoring, improves machine translation performance by 0.3-0.4 BLEU and reduces translation edit rate (TER) by 0.2-0.5% compared to an unadapted LM.

Index Terms— Speech translation, language modeling, domain adaptation

1. INTRODUCTION

Language model (LM) is one of primary components in any statistical machine translation (MT) system. LMs used by most MT systems are n-gram models – the type of language model commonly used by many speech and language applications. While domain adaptation of n-gram LMs has been widely used for speech recognition (e.g. [1, 2]), the existing adaptation techniques so far have seen little application to the task of machine translation. This paper explores the use of several LM adaptation methods in an Arabic-English machine translation system applied to broadcast speech.

N-gram language models can be adapted to a particular style of data by means of interpolation. The assumption is that each corpus has some unique style characteristics captured in its n-gram counts. If we were to merge the counts from all corpora, many of these style characteristics may be lost, thus producing some generic n-gram statistics. Alternatively, we can estimate a separate language model from each individual corpus and then combine these LMs by interpolating n-gram probabilities. This technique, also referred to as mixtures of language models, was reported by IBM in 1995 [1] in the context of the speech recognition task, and has been used by many sites since then. In this work we compare count merging and interpolation in terms of machine translation performance.

This work was supported under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

Language model interpolation weights are chosen to reflect the target data style, typically by minimizing the LM mixture perplexity on a held-out data set selected to be representative of the test data. In some cases, selection of such held-out data may not be feasible either due to high cost or lack of prior knowledge of the test domain. In such cases one can use a generic optimization set. The output from a generic model can be refined by applying a second pass of either decoding or rescoring with an adapted model, where the mixture weights are optimized using the test set hypotheses (e.g. [2]). In this paper we apply this technique (often referred to as "unsupervised" or "dynamic" adaptation) to the task of machine translation.

While perplexity minimization is a commonly used mixture weight optimization criterion, perplexity is a measure of language model's generative power which is not a direct measure of system's performance in most situations. In this work we explore the use of discriminative estimation of language model weights by optimizing for machine translation performance directly, i.e. by minimizing the translation edit rate (TER)[3] or by maximizing BLEU[4] score.

2. UNSUPERVISED ADAPTATION

Unsupervised adaptation described in this paper is accomplished by the following sequence of steps: 1) decoding with an unadapted language model and generating n-best lists; 2) optimizing LM interpolation weights by minimizing perplexity of the 1-best translation output; and 3) rescoring the n-best lists with the adapted interpolated LM.

In this work we experiment with unsupervised LM adaptation at two levels of granularity: 1) whole test set, and 2) each individual document. The former is accomplished by estimating LM interpolation weights on the translation output for the entire test set. In the second approach we estimate a separate set of mixture weights for each test document (i.e. a broadcast episode). One can envision also applying this at other levels of granularity: finer (e.g. paragraph) or coarser (e.g. source); however, smaller units may not give us robust estimates, while under the test conditions (see Section 5) no prior knowledge of the source was available to allow us make groups of documents.

3. DISCRIMINATIVE ADAPTATION

Discriminative adaptation described in this paper involves optimizing language model mixture weights using an MT performance measure (TER or BLEU) directly. For this purpose we generate n-best lists ($n = 300$) with a development set and then tune the weights using Powell's hill climbing algorithm [5] with the objective of minimizing TER (or maximizing BLEU) on this set of n-best lists.

Here we investigate two methods of combining language model probabilities from different LM components. In the first case, LM components are treated as independent knowledge sources and their probabilities are combined log-linearly (i.e. a weighted sum of log scores). In the second case, we use interpolation in probability space, identical to the standard LM interpolation procedure.

4. LM TRAINING DATA

Data used for training language models in MT consists of monolingual text in the target language (English in our case). In the experiments described is Section 5, we used about 6B words of training data consisting predominantly of news articles, with the exception of CNN talk show transcripts (CNNtrans) and the University of Washington web data (ConvWeb) which contain text of conversation-like style.

Roughly half of the training data (2.8B words) came from the LDC's Gigaword corpus containing articles from four major news agencies published between 1995 and 2004. In order to supplement the data available from LDC, we downloaded additional news articles, published within past 6 years, from a variety of on-line news publishers (e.g. BBC) that offer free access to their archives. Collectively these articles (NewsArchives) totaled to 1.3B words.

We also included two corpora that we typically use for training our speech-to-text (STT) language models, namely, transcripts of CNN talk shows (CNNtrans, 60M words) downloaded from the CNN website and news articles from a variety of publishers, downloaded daily within the past year (DailyNews, 1B words).

In order to broaden our coverage of data styles we included the UW web data corpus – a corpus collected by the University of Washington, featuring text that resembles conversation-like style [6] (ConvWeb, 650M words). Finally, we used the English side of the broadcast news portion of the parallel data (Parallel, 14M words).

5. EXPERIMENTAL SETUP

Our experiments focus on translating Arabic speech to English text. We report machine translation performance on an internal test set that consists of Arabic broadcast news (BN) and broadcast conversational (BC) material in roughly equal proportions (around 30K words each). The evaluation paradigm is similar to that of GALE evaluations [7] where no prior knowledge of the genre (BN or BC) is available, hence we use the same translation system to process both genres, even though we report separate results for BN and BC.

The BBN Arabic STT system uses a similar modeling and search strategy as described in [8]. The multi-pass recognizer first does a fast match of the data to produce scores for numerous word endings (aka word graphs) using a coarse state-tied mixture acoustic model (AM) and a bigram language model (LM). Next, a state-clustered tied-mixture (SCTM) AM and a trigram LM are used to decode the word graphs to produce lattices. The lattices are then rescored using a cross-word SCTM AM and a 4- gram LM. The best path of the rescored lattice is the recognition results. The decoding process is repeated two (or three) times with speaker-independent AMs used in the first stage while subsequent decoding stages use speaker-adaptively-trained AMs . All AMs were trained on about 1300 hours of speech data with the largest model having about 6k states and 800k Gaussians. All LMs were estimated based on a training corpus of almost 1B words.

The BBN translation engine employs statistical phrase-based translation models, with a decoding strategy similar to [9]. Phrase translations are extracted from word alignments obtained by running GIZA++ [10] on a bilingual parallel training corpus (139M words of Arabic/English). A significant portion of the phrase translations are generalized through the use of part of speech classes, for improved performance on unseen data [11]. Both forward and backward phrase translation probabilities are estimated and used in decoding along with a pruned trigram English LM, a penalty for phrase reordering, a phrase segmentation score, and a word insertion penalty.

A separate tuning set (referred to as *bnc-tune*) with data sources and epoch similar to the test set was used for the MT system optimization. The system weights were optimized using reference transcriptions of the tuning data, and the same set of weights was then used to translate the test set from both reference transcriptions and STT hypotheses (see [12] for more details about optimizing MT for speech input). Two sets of system weights were computed using two different optimization criteria: minimum TER and maximum BLEU. In our experiments we report both TER and BLEU scores, each obtained with the appropriately optimized system.

We trained 5-gram Kneser-Ney smoothed language models using 6B tokens of English text data described in Section 4. No pruning was used during training, i.e. all ngrams including singletons were retained, which led to models of very large size (tens of gigabytes). To make experimentation with LMs of such size practical we implemented a two-pass decoding architecture, where 1) n-best lists ($n = 300$) were generated with a pruned 3-gram language model, and 2) large 5-gram models were used to rescore the n-best lists.

The decoding 3-gram LM was kept constant throughout all experiments, hence the results reported in the following

LM component weights optimization	Reference transcriptions				STT hypotheses			
	BN		BC		BN WER=20.1%		BC WER=29.7%	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
1) Merged counts (no interpolation)	58.08	22.47	59.46	21.45	63.26	19.22	69.01	16.51
2) Min-perplexity on MT02	57.74	22.60	59.35	21.66	63.16	19.27	68.96	16.59
3) Min-perplexity on bnc-tune	57.63	22.67	59.23	21.75	63.14	19.42	68.57	16.52
4) Unsupervised, all documents jointly	57.56	22.82	59.16	21.96	63.03	19.53	68.67	16.72
5) Unsupervised, each document separately	57.64	22.76	59.19	21.97	63.10	19.40	68.70	16.73
6) Discriminative, log-linear space	57.71	22.73	59.43	21.84	63.17	19.56	68.87	16.84
7) Discriminative, probability space	57.56	22.83	59.10	21.91	62.90	19.64	68.55	16.85

Fig. 1. Test set translation performance (lower case TER and BLEU) with 5-gram LMs adapted using different methods. The performance is measured on two types of input: reference transcriptions and STT hypotheses.

section are obtained by rescoring the same set of n-best lists[1] with different 5-gram LMs. The decoding 3-gram LM was estimated from merged counts without any tuning or adaptation. While keeping the decoding LM constant simplifies experimentation, the gains that we get from adapting only the rescoring LM underestimate the possible gains from adapting both LMs.

6. RESULTS

The table of results in Figure 1 compares different types of LM adaptation methods in terms of machine translation performance obtained on the test set. Each model is used to translate two types of input: 1) human-generated reference transcriptions of speech, and 2) hypotheses produced by the automatic STT system described in Section 5. Performance is measured using two metrics: TER and BLEU. Translation edit rate (TER) is a measure of error, similar to word error rate (WER), where lower numbers correspond to better results. BLEU, on the other hand, is a measure of similarity, hence higher numbers indicate better performance.

The baseline performance in Figure 1 corresponds to the unadapted language model (row 1), where all training corpora are merged together to produce a single LM. Note that this baseline is not entirely unbiased as our choice of the training data was driven by the knowledge of the target genre, resulting in training data dominated by news materials. Nevertheless, this baseline is consistently outperformed by the adapted LMs.

Rows 2 and 3 in Figure 1 represent traditional interpolation-based LM adaptation where mixture weights are optimized by minimizing perplexity on a held-out set. Results in row 2 are obtained with an LM optimized on the translation references of the NIST 2002 Arabic MT evaluation set (MT02). Row 3 shows an LM optimized on the *bnc-tune* tuning set, similar in content and style to the test set. Rows 2 and 3 represent cases of using a generic[2] and a matched optimization sets respectively. Even though differences in performance between these two models are small, the LM optimized on the matched tuning set performs slightly better.

Use of unsupervised adaptation techniques is illustrated in rows 4 and 5 of Figure 1. Row 4 shows an LM with one set of interpolation weights optimized on the first pass hypotheses from all test documents combined. The language model in row 5 uses a separate set of mixture weights for each test document. Both unsupervised methods perform well, generally, achieving better results than the LM optimized on a matched tuning set. Gains from unsupervised adaptation are larger when translating reference transcriptions as opposed to STT hypotheses. This is not surprising considering that hypotheses produced by the first pass of MT decoding (which unsupervised adaptation relies on) contain fewer errors when they originate from reference transcriptions.

Finally, rows 6 and 7 in Figure 1 show the use of discriminative adaptation with log-linear and probability interpolation-based combining of LM components respectively. Interpolation of LM probabilities gives better performance than combining LM component scores log-linearly. Discriminative adaptation of LM interpolation weights gave the best result on STT hypotheses. It also tied with unsupervised adaptation for the best result on reference transcriptions.

Figure 2 lists the weights assigned to the LM components by different adaptation methods. Weights optimized (using min-perplexity criterion) on MT02 reflect a bias toward the two news sources (AFP and XIN) present in the optimization set. Switching to the matched *bnc-tune* optimization set shifts the weight to the parallel data component (similar in style) and to the DailyNews component (similar in epoch). The ConvWeb and CNNtrans components also get increased weights, due to the "conversational" part of the tuning set, though their relative contribution in the mixture remains small. Interpolation weights used in unsupervised adaptation show little change between the two types of input (reference transcriptions and STT hypotheses) and they are fairly similar to the weights optimized on the *bnc-tune* set.

[1] There are actually two sets of n-best lists, produced with two systems – one optimized for TER and the other for BLEU.

[2] Note that MT02 cannot be treated as an entirely generic held-out set as it is somewhat similar in style to the broadcast news portion of the test set

LM component	size (tokens)	Min Perplexity				Discriminative	
		Optimization set		Unsupervised from		log-linear combination	probability interpolation
		MT02	bnc-tune	ref trans	stt hyps		
Gigaword (NYT)	1.3B	0.027	0.021	0.070	0.082	0.117	0.202
Gigaword (AFP)	400M	0.137	0.053	0.070	0.078	0.144	0.009
Gigaword (APW)	900M	0.081	0.034	0.099	0.111	0.083	0.038
Gigaword (XIN)	200M	0.344	0.060	0.073	0.084	0.118	0.024
NewsArchives	1.3B	0.307	0.337	0.326	0.344	0.023	0.123
ConvWeb	650M	0.011	0.082	0.036	0.018	0.079	0.036
CNNtrans	60M	0.001	0.030	0.020	0.014	0.016	0.078
DailyNews	1.0B	0.037	0.126	0.036	0.035	0.229	0.051
Parallel	14M	0.056	0.256	0.271	0.234	0.190	0.440

Fig. 2. LM interpolation weights estimated using different methods. The log-linear combination weights are normalized for ease of comparison.

The two sets of discriminatively estimated weights show drastic difference with respect to the weights optimized using perplexity. The set of weights used in the log-linear combination has been normalized (to sum to 1) to allow easier comparison within Figure 2, although there is still no clear interpretation to the log-linear weights. The probability interpolation weights, that are estimated discriminatively, show a strong preference for the parallel data component. This may be due to a high degree of overlap in terms of phrases that appear in the n-best lists used in weight optimization, since the same parallel data was used to train the translation model. [3]

7. CONCLUSIONS AND FUTURE WORK

In summary, we have compared several language model adaptation techniques applied to the task of machine translation from speech. Use of adaptation improved machine translation performance by 0.2-0.5% TER and 0.3-0.4 points BLUE when compared to an unadapted LM. Discriminative adaptation of LM interpolation weights gave the best performance when translating STT hypotheses and tied with unsupervised adaptation for the best performance on reference transcriptions.

Gains from unsupervised adaptation diminish as the error rates increase, hence the unsupervised methods work better when translating reference transcriptions as opposed to STT hypotheses. Adaptation at the document level did not outperform the full test set adaptation, perhaps, due to its inability to robustly estimate mixture weights from small amounts of text. Combining documents into groups (e.g. via clustering) for weight optimization may lead to better results.

In this work, adaptation was applied only to the language model used in rescoring of n-best lists. One may expect larger gains in performance if the decoding LM is also adapted. Furthermore, adaptation may have stronger effect when entropy-based pruning is applied to the LM, which is often true for the decoding LM. In our experiments, all 5-gram LMs used in rescoring of n-best lists were unpruned, hence they did not differ in terms of ngrams present in the model.

[3]The training data used for the translation model also included other sources, e.g. UN documents, that differ from broadcast news in style.

8. REFERENCES

[1] F. Liu et al., "IBM Switchboard progress and evaluation site report," in *LVCSR Workshop*, Gaithersburg, MD, 1995, NIST.

[2] M. Bacchiani and B. Roark, "Unsupervised language model adaptation," in *Proc. ICASSP*, 2003, pp. 224–227.

[3] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. AMTA*, 2006.

[4] K. Papineni et al., "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL*, 2001, pp. 311–318.

[5] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical recipes in C: the art of scientific computing*, Cambridge University Press, second edition, 1994, pp. 416-420.

[6] I. Bulyko, M. Ostendorf, and A. Stolcke, "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures," in *Proc. HLT/NAACL*, 2003, pp. 7–9.

[7] NIST, "NIST machine translation evaluation for GALE," http://www.nist.gov/speech/tests/gale/index.htm, 2006.

[8] L. Nguyen, B. Xiang, M. Afify, S. Abdou, S. Matsoukas, R. Schwartz, and J. Makhoul, "The BBN RT04 English broadcast news transcription system," in *Proc. Eurospeech*, 2005.

[9] P. Koehn et al., "Statistical phrase-based translation," in *Proc. HLT/NAACL*, 2003.

[10] "Giza+++," http://www.fjoch.com/GIZA++.html.

[11] B. Xiang et al., "The BBN machine translation system for the NIST 2006 MT evaluation," presentation, NIST MT06 Workshop Washington DC, 2006.

[12] S. Matsoukas, I. Bulyko, B. Xiang, R. Schwartz, and J. Makhoul, "Intergrating speech recognition and machine translation," submitted to ICASSP, 2007.