Improving Automatic Speech Recognition for Lectures through Transformation-based Rules Learned from Minimal Data

Cosmin Munteanu[*†] **Gerald Penn**[†] **Xiaodan Zhu**[†]

[*]National Research Council Canada
46 Dineen Drive
Fredericton E3B 9W4, CANADA
Cosmin.Munteanu@nrc.gc.ca

[†]University of Toronto
Dept. of Computer Science
Toronto M5S 3G4, CANADA
{gpenn,xzhu}@cs.toronto.edu

Abstract

We demonstrate that transformation-based learning can be used to correct noisy speech recognition transcripts in the lecture domain with an average word error rate reduction of 12.9%. Our method is distinguished from earlier related work by its robustness to small amounts of training data, and its resulting efficiency, in spite of its use of true word error rate computations as a rule scoring function.

1 Introduction

Improving access to archives of recorded lectures is a task that, by its very nature, requires research efforts common to both Automatic Speech Recognition (ASR) and Human-Computer Interaction (HCI). One of the main challenges to integrating text transcripts into archives of webcast lectures is the poor performance of ASR systems on lecture transcription. This is in part caused by the mismatch between the language used in a lecture and the predictive language models employed by most ASR systems. Most ASR systems achieve Word Error Rates (WERs) of about 40-45% in realistic and uncontrolled lecture conditions (Leeuwis et al., 2003; Hsu and Glass, 2006).

Progress in ASR for this genre requires both better acoustic modelling (Park et al., 2005; Fügen et al., 2006) and better language modelling (Leeuwis et al., 2003; Kato et al., 2000; Munteanu et al., 2007). In contrast to some unsupervised approaches to language modelling that require large amounts of manual transcription, either from the same instructor or on the same topic (Nanjo and Kawahara, 2003; Niesler and Willett, 2002), the solution proposed by Glass et al. (2007) uses half of the lectures in a semester course to train an ASR system for the other half or for when the course is next offered, and still results in significant WER reductions. And yet even in this scenario, the business case for manually transcribing half of the lecture material in every recorded course is difficult to make, to say the least. Manually transcribing a one-hour recorded lecture requires at least 5 hours in the hands of qualified transcribers (Hazen, 2006) and roughly 10 hours by students enrolled in the course (Munteanu et al., 2008). As argued by Hazen (2006), any ASR improvements that rely on manual transcripts need to offer a balance between the cost of producing those transcripts and the amount of improvement (i.e. WER reductions).

There is some work that specializes in adaptive language modelling with extremely limited amounts of manual transcripts. Klakow (2000) filters the corpus on which language models are trained in order to retain the parts that are more similar to the correct transcripts on a particular topic. This technique resulted in relative WER reductions of between 7% and 10%. Munteanu et al. (2007) use an information retrieval technique that exploits lecture presentation slides, automatically mining the World Wide Web for documents related to the topic as attested by text on the slides, and using these to build a better-matching language model. This yields about an 11% relative WER reduction for lecture-specific language models. Following upon other applications of computer-supported collaborative work to address shortcomings of other systems in artificial intelligence (von Ahn and Dabbish, 2004), a wiki-based technique for collaboratively editing lecture transcripts has been shown to produce entirely cor-

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 764–772,
Suntec, Singapore, 2-7 August 2009.

rected transcripts, given the proper motivation for students to participate (Munteanu et al., 2008). Another approach is active learning, where the goal is to select or generate a subset of the available data that would be the best candidate for ASR adaptation or training (Riccardi and Hakkani-Tur, 2005; Huo and Li, 2007).[1] Even with all of these, however, there remains a significant gap between this WER and the threshold of 25%, at which lecture transcripts have been shown with statistical significance to improve student performance on a typical lecture browsing task (Munteanu et al., 2006).

People have also tried to correct ASR output in a second pass. Ringger and Allen (1996) treated ASR errors as noise produced by an auxiliary noisy channel, and tried to decode back to the perfect transcript. This reduced WER from 41% to 35% on a corpus of train dispatch dialogues. Others combine the transcripts or word lattices (from which transcripts are extracted) of two complementary ASR systems, a technique first proposed in the context of NIST's ROVER system (Fiscus, 1997) with a 12% relative error reduction (RER), and subsequently widely employed in many ASR systems.

This paper tries to correct ASR output using transformation-based learning (TBL). This, too, has been attempted, although on a professional dictation corpus with a 35% initial WER (Peters and Drexel, 2004). They had access to a very large amount of manually transcribed data — so large, in fact, that the computation of true WER in the TBL rule selection loop was computationally infeasible, and so they used a set of faster heuristics instead. Mangu and Padmanabhan (2001) used TBL to improve the word lattices from which the transcripts are decoded, but this method also has efficiency problems (it begins with a reduction of the lattice to a confusion network), is poorly suited to word lattices that have already been heavily domain-adapted because of the language model's low perplexity, and even with higher perplexity models (the SWITCHBOARD corpus using a language model trained over a diverse range of broadcast news and telephone conversation transcripts), was reported to produce only a 5% WER reduction.

What we show in this paper is that a true WER calculation is so valuable that a manual transcription of only about 10 minutes of a one-hour lecture is necessary to learn the TBL rules, and that this smaller amount of transcribed data in turn makes the true WER calculation computationally feasible. With this combination, we achieve a greater average relative error reduction (12.9%) than that reported by Peters and Drexel (2004) on their dictation corpus (9.6%), and an RER over three times greater than that of our reimplementation of their heuristics on our lecture data (3.6%). This is on top of the average 11% RER from language model adaptation on the same data. We also achieve the RER from TBL without the obligatory round of development-set parameter tuning required by their heuristics, and in a manner that is robust to perplexity. Less is more.

Section 2 briefly introduces Transformation-Based Learning (TBL), a method used in various Natural Language Processing tasks to correct the output of a stochastic model, and then introduces a TBL-based solution for improving ASR transcripts for lectures. Section 3 describes our experimental setup, and Section 4 analyses its results.

2 Transformation-Based Learning

Brill's tagger introduced the concept of Transformation-Based Learning (TBL) (Brill, 1992). The fundamental principle of TBL is to employ a set of rules to correct the output of a stochastic model. In contrast to traditional rule-based approaches where rules are manually developed, TBL rules are automatically learned from training data. The training data consist of sample output from the stochastic model, aligned with the correct instances. For example, in Brill's tagger, the system assigns POSs to words in a text, which are later corrected by TBL rules. These rules are learned from manually-tagged sentences that are aligned with the same sentences tagged by the system. Typically, rules take the form of context-dependent transformations, for example "change the tag from verb to noun if one of the two preceding words is tagged as a determiner."

An important aspect of TBL is rule scoring/ranking. While the training data may suggest

[1] This work generally measures progress by reduction in the size of training data rather than relative WER reduction. Riccardi and Hakkani-Tur (2005) achieved a 30% WER with 68% less training data than their baseline. Huo and Li (2007) worked on a small-vocabulary name-selection task that combined active learning with acoustic model adaptation. They reduced the WER from 15% to 3% with 70 syllables of acoustic adaptation, relative to a baseline that reduced the WER to 3% with 300 syllables of acoustic adaptation.

a certain transformation rule, there is no guarantee that the rule will indeed improve the system's accuracy. So a scoring function is used to rank rules. From all the rules learned during training, only those scoring higher than a certain threshold are retained. For a particular task, the scoring function ideally reflects an objective quality function.

Since Brill's tagger was first introduced, TBL has been used for other NLP applications, including ASR transcript correction (Peters and Drexel, 2004). A graphical illustration of this task is presented in Figure 1. Here, the rules consist of word-level transformations that correct n-gram sequences. A typical challenge for TBL is the heavy computational requirements of the rule scoring function (Roche and Schabes, 1995; Ngai and Florian, 2001). This is no less true in large-vocabulary ASR correction, where large training corpora are often needed to learn good rules over a much larger space (larger than POS tagging, for example). The training and development sets are typically up to five times larger than the evaluation test set, and all three sets must be sampled from the same cohesive corpus.

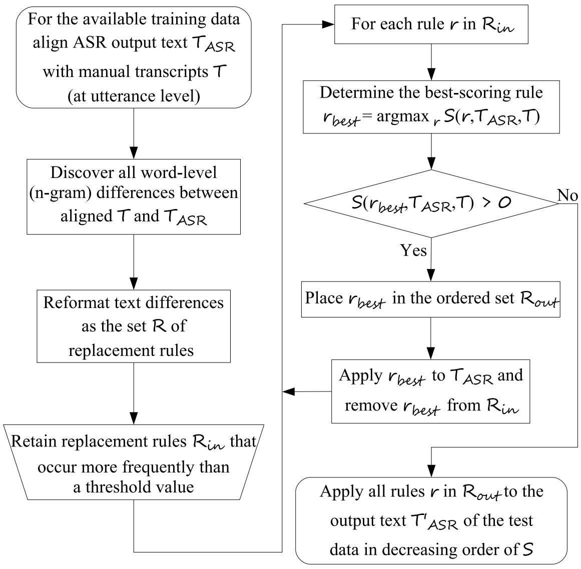
For the available training data align ASR output text T_{ASR} with manual transcripts T (at utterance level)
Discover all word-level (n-gram) differences between aligned T and T_{ASR}
Reformat text differences as the set R of replacement rules
Retain replacement rules R_{in} that occur more frequently than a threshold value
For each rule r in R_{in}
Determine the best-scoring rule r_{best}= argmax $_r$ $S(r,T_{ASR},T)$
$S(r_{best},T_{ASR},T) > 0$
No
Yes
Place r_{best} in the ordered set R_{out}
Apply r_{best} to T_{ASR} and remove r_{best} from R_{in}
Apply all rules r in R_{out} to the output text T'_{ASR} of the test data in decreasing order of S

Figure 1: General TBL algorithm. Transformation rules are learned from the alignment of manually-transcribed text (T) with automatically-generated transcripts (T_{ASR}) of training data, ranked according to a scoring function (S) and applied to the ASR output (T'_{ASR}) of test data.

While the objective function for improving the ASR transcript is WER reduction, the use of this for scoring TBL rules can be computationally prohibitive over large data-sets. Peters and Drexel (2004) address this problem by using an heuristic approximation to WER instead, and it appears that their approximation is indeed adequate when large amounts of training data are available. Our approach stands at the opposite side of this trade-off — restrict the amount of training data to a bare minimum so that true WER can be used in the rule scoring function. As it happens, the minimum amount of data is so small that we can automatically develop highly domain-specific language models for single 1-hour lectures. We show below that the rules selected by this function lead to a significant WER reduction for individual lectures even if a little less than the first ten minutes of the lecture are manually transcribed. This combination of domain-specificity with true WER leads to the superior performance of the present method, at least in the lecture domain (we have not experimented with a dictation corpus).

Another alternative would be to change the scope over which TBL rules are ranked and evaluated, but it is well known that globally-scoped ranking over the entire training set at once is so useful to TBL-based approaches that this is not a feasible option — one must either choose an heuristic approach, such as that of Peters and Drexel (2004) or reduce the amount of training data to learn sufficiently robust rules.

2.1 Algorithm and Rule Discovery

As our proposed TBL adaptation operates directly on ASR transcripts, we employ an adaptation of the specific algorithm proposed by Peters and Drexel (2004), which is schematically represented in Figure 1. This in turn was adapted from the general-purpose algorithm introduced by Brill (1992).

The transformation rules are contextual word-replacement rules to be applied to ASR transcripts, and are learned by performing a word-level alignment between corresponding utterances in the manual and ASR transcripts of training data, and then extracting the mismatched word sequences, anchored by matching words. The matching words serve as contexts for the rules' application. The rule discovery algorithm is outlined in Figure 2; it is applied to every mismatching word sequence between the utterance-aligned manual and ASR transcripts.

For every mismatching sequence of words, a set

```
◇ for every sequence of words $c_0 w_1 \ldots w_n c_1$ in the
      ASR output that is deemed to be aligned with a
      corresponding sequence $c_0 w'_1 \ldots w'_m c_1$ in the
      manual transcript:

      ◇ add the following contextual replacements to the
      set of discovered rules:
            $/\, c_0 w_1 \ldots w_n c_1 \,/\, c_0 w'_1 \ldots w'_m c_1 \,/$
            $/\, c_0 w_1 \ldots w_n \,/\, c_0 w'_1 \ldots w'_m \,/$
            $/\, w_1 \ldots w_n c_1 \,/\, w'_1 \ldots w'_m c_1 \,/$
            $/\, w_1 \ldots w_n \,/\, w'_1 \ldots w'_m \,/$

      ◇ for each $i$ such that $1 \leq i < min(n, m)$, add
      the following contextual replacements to the set of
      discovered rules:
            $/\, c_0 w_1 \ldots w_i \,/\, c_0 w'_1 \ldots w'_{a(i)} \,/$
            $/\, w_{i+1} \ldots w_n c_1 \,/\, w'_{a(i+1)} \ldots w'_m c_1 \,/$
            $/\, w_1 \ldots w_i \,/\, w'_1 \ldots w'_{a(i)} \,/$
            $/\, w_{i+1} \ldots w_n \,/\, w'_{a(i+1)} \ldots w'_m \,/$
```

Figure 2: The discovery of transformation rules.

Utterance-align ASR output and correct transcripts:

ASR: the okay one and you come and get your seats

Correct: ok why don't you come and get your seats

⇓

Insert sentence delimiters (to serve as possible anchors for the rules):

ASR: <s> the okay one and you come and get your seats </s>

Correct: <s> ok why don't you come and get your seats </s>

⇓

Extract the mismatching sequence, enclosed by matching anchors:

ASR: <s> the okay one and you

Correct: <s> ok why don't you

⇓

Output all rules for replacing the incorrect ASR sequence with the correct text, using the entire sequence (a) or splices (b), with or without surrounding anchors:

(a)	the okay one and / ok why don't
(a)	the okay one and you / ok why don't you
(a)	<s> the okay one and / <s> ok why don't
(a)	<s> the okay one and you / <s> ok why don't you
(b)	the okay / ok
(b)	<s> the okay / <s> ok
(b)	one and / why don't
(b)	one and you / why don't you
(b)	the okay one / ok why
(b)	<s> the okay one / <s> ok why
(b)	and / don't
(b)	and you / don't you

Figure 3: An example of rule discovery.

of contextual replacement rules is generated. The set contains the mismatched pair, by themselves and together with three contexts formed from the left, right, and both anchor context words. In addition, all possible splices of the mismatched pair and the surrounding context words are also considered.[2] Rules are shown here as replacement expressions in a sed-like syntax. Given the rule $r = /w_1 \ldots w_n/w'_1 \ldots w'_m/$, every instance of the n-gram $w_1 \ldots w_n$ appearing in the current transcript is replaced with the n-gram $w'_1 \ldots w'_m$. Rules cannot apply to their own output. Rules that would result in arbitrary insertions of single words (e.g. $/\ \ /w_1/$) are discarded. An example of a rule learned from transcripts is presented in Figure 3.

2.2 Scoring Function and Rule Application

The scoring function that ranks rules is the main component of any TBL algorithm. Assuming a relatively small size for the available training data, a TBL scoring function that directly correlates with WER can be conducted globally over the entire training set. In keeping with TBL tradition, however, rule selection itself is still greedily approximated. Our scoring function is defined as:

$$\begin{aligned} S_{WER}(r, T_{ASR}, T) = \quad & WER(T_{ASR}, T) \\ & - WER(\rho(r, T_{ASR}), T), \end{aligned}$$

where $\rho(r, T_{ASR})$ is the result of applying rule r on text T_{ASR}.

As outlined in Figure 1, rules that occur in the training sample more often than an established threshold are ranked according to the scoring function. The ranking process is iterative: in each iteration, the highest-scoring rule r_{best} is selected. In subsequent iterations, the training data T_{ASR} are replaced with the result of applying the selected rule on them ($T_{ASR} \leftarrow \rho(r_{best}, T_{ASR})$) and the remaining rules are scored on the transformed training text. This ensures that the scoring and ranking of remaining rules takes into account the changes brought by the application of the previously selected rules. The iterations stop when the scoring function reaches zero: none of the remaining rules improves the WER on the training data.

On testing data, rules are applied to ASR tran-

[2] The splicing preserves the original order of the word-level utterance alignment, i.e., the output of a typical dynamic programming implementation of the edit distance algorithm (Gusfield, 1997). For this, word insertion and deletion operations are treated as insertions of blanks in either the manual or ASR transcript.

scripts in the same order in which they were selected.

3 Experimental Design

Several combinations of TBL parameters were tested with no tuning or modifications between tests. As the proposed method was not refined during the experiments, and since one of the goals of our proposed approach is to eliminate the need for developmental data sets, the available data were partitioned only into training and test sets, with one additional hour set aside for code development and debugging.

It can be assumed that a one-hour lecture given by the same instructor will exhibit a strong cohesion, both in topic and in speaking style, between its parts. Therefore, in contrast to typical TBL solutions, we have evaluated our TBL-based approach by partitioning each 50 minute lecture into a training and a test set, where the training set is smaller than the test set. As mentioned in the introduction, it is feasible to obtain manual transcripts for the first 10 to 15 minutes of a lecture. As such, the evaluation was carried out with two values for the training size: the first fifth ($TS = 20\%$) and the first third ($TS = 33\%$) of the lecture being manually transcribed.

Besides the training size parameter, during all experimental tests a second parameter was also considered: the rule pruning threshold (RT). As described in Section 2.2, of all the rules learned during the rule discovery step, only those that occur more often than the threshold are scored and ranked. This parameter can be set as low as 1 (consider all rules) or 2 (consider all rules that occur at least twice over the training set). For larger-scale tasks, the threshold serves as a pruning alternative to the computational burden of scoring several thousand rules. A large threshold could potentially lead to discrediting low-frequency but high-scoring rules. Due to the intentionally small size of our training data for lecture TBL, the lowest threshold was set to $RT = 2$. When a development set is available, several values for the RT parameter could be tested and the optimal one chosen for the evaluation task. Since we used no development set, we tested two more values for the rule pruning threshold: $RT = 5$ and $RT = 10$.

Since our TBL solution is an extension of the solution proposed in Peters and Drexel (2004), their heuristic is our baseline. Their scoring function is the *expected error reduction*:

$$XER = ErrLen \cdot (GoodCnt - BadCnt),$$

a WER approximation computed over all instances of rules applicable to the training set which reflects the difference between true positives (the number of times a rule is correctly applied to errorful transcripts – $GoodCnt$) and false positives (the instances of correct text being unnecessarily "corrected" by a rule – $BadCnt$). These are weighted by the length in words ($ErrLen$) of the text area that matches the left-hand side of the replacement.

3.1 Acoustic Model

The experiments were conducted using the SONIC toolkit (Pellom, 2001). We used the acoustic model distributed with the toolkit, which was trained on 30 hours of data from 283 speakers from the WSJ0 and WSJ1 subsets of the 1992 development set of the Wall Street Journal (WSJ) Dictation Corpus. Our own lectures consist of eleven lectures of approximately 50 minutes each, recorded in three separate courses, each taught by a different instructor. For each course, the recordings were performed in different weeks of the same term. They were collected in a large, amphitheatre-style, 200-seat lecture hall using the AKG C420 head-mounted directional microphone. The recordings were not intrusive, and no alterations to the lecture environment or proceedings were made. The 1-channel recordings were digitized using a TASCAM US-122 audio interface as uncompressed audio files with a 16KHz sampling rate and 16-bit samples. The audio recordings were segmented at pauses longer than 200ms, manually for one instructor and automatically for the other two, using the silence detection algorithm described in Placeway et al. (1997). Our implementation was manually fine-tuned for every instructor in order to detect all pauses longer than 200ms while allowing a maximum of 20 seconds in between pauses.

The evaluation data are described in Table 1. Four evaluations tasks were carried out; for instructor *R*, two separate evaluation sessions, *R-1* and *R-2*, were conducted, using two different language models.

The pronunciation dictionary was custom-built to include all words appearing in the corpus on which the language model was trained. Pronunciations were extracted from the 5K-word WSJ dictionary included with the SONIC toolkit and from

Evaluation task name	R-1	R-2	G-1	K-1
Instructor	R.		G.	K.
Gender	Male		Male	Female
Age	Early 60s		Mid 40s	Early 40s
Segmentation	manual		automatic	automatic
# lectures	4		3	4
Lecture topic	Interactive media design		Software design	Unix programming
Language model	WSJ-5K	WEB	ICSISWB	WSJ-5K

Table 1: The evaluation data.

the 100K-word CMU pronunciation dictionary. For all models, we allowed one non-dictionary word per utterance, but only for lines longer than four words. For allowable non-dictionary words, SONIC's `sspell` lexicon access tool was used to generate pronunciations using letter-to-sound predictions. The language models were trained using the CMU-CAM Language Modelling Toolkit (Clarkson and R., 1997) with a training vocabulary size of 40K words.

3.2 Language Models

The four evaluations were carried out using the language models given in Table 1, either custom-built for a particular topic or the baseline models included in the SONIC toolkit, as follows:

WSJ-5K is the baseline model of the SONIC toolkit. It is a 5K-word model built using the same corpus as the base acoustic model included in the toolkit.

ICSISWB is a 40K-word model created through the interpolation of language models built on the entire transcripts of the ICSI Meeting corpus and the Switchboard corpus. The ICSI Meeting corpus consists of recordings of university-based multi-speaker research meetings, totaling about 72 hours from 75 meetings (Janin et al., 2003). The Switchboard (SWB) corpus (Godfrey et al., 1992) is a large collection of about 2500 scripted telephone conversations between approximately 500 English-native speakers, suitable for the conversational style of lectures, as also suggested in (Park et al., 2005).

WEB is a language model built for each particular lecture, using information retrieval techniques that exploit the lecture slides to automatically mine the World Wide Web for documents related to the presented topic. WEB adapts ICSISWB using these documents to build a language model that better matches the lecture topic. It is also a 40K-word model built on training corpora with an average file size of approximately 200 MB per lecture, and an average of 35 million word tokens per lecture.

It is appropriate to take the difference between ICSISWB and WSJ-5K to be one of greater genre specificity, whereas the difference between WEB and ICSISWB is one of greater topic-specificity. Our experiments on these three models (Munteanu et al., 2007) shows that the topic adaptation provides nearly all of the benefit.

4 Results

Tables 2, 3 and 4[3] present the evaluation results for instructors R and G. The transcripts were obtained through ASR runs using three different language models. The TBL implementation with our scoring function S_{WER} brings relative WER reductions ranging from 10.5% to 14.9%, with an average of 12.9%.

ICSISWB		Lecture 1		Lecture 2		Lecture 3	
	TS = %	20	33	20	33	20	33
Initial WER		**50.93**	**50.75**	**54.10**	**53.93**	**48.79**	**49.35**
XER	RT = 10	46.63	49.38	49.93	48.61	49.52	50.43
	RT = 5	48.34	49.75	49.32	48.81	49.58	49.26
	RT = 2	54.05	56.84	52.01	49.11	50.37	51.66
XER-NoS	RT = 10	49.54	49.38	54.10	53.93	48.79	48.24
	RT = 5	49.54	49.31	56.70	55.50	48.51	48.42
	RT = 2	59.00	59.28	57.61	55.03	50.41	52.67
S_{WER}	RT = 10	46.63	46.53	49.80	48.44	45.83	45.42
	RT = 5	46.63	45.60	47.75	47.23	44.76	44.44
	RT = 2	**44.48**	**44.30**	**47.46**	**47.02**	**43.60**	**44.13**

Table 4: Experimental evaluation: WER values for instructor G using the ICSISWB language model.

These WER reductions are greater than those produced by the XER baseline approach. It is not possible to provide confidence intervals since the proposed method does not tune parameters from sampled data (which we regard as a very positive quality for such a method to have). Our speculative experimentation with several values for TS and RT, however, leads us to conclude that this method is significantly less sensitive to variations in both the training size TS and the rule pruning threshold RT than earlier work, making it suitable for application to tasks with limited training data – a result somewhat expected since rules are validated through direct WER reductions over the entire training set.

[3] Although WSJ-5K and ICSISWB exhibited nearly the same WER in our earlier experiments on all lecturers, we did find upon inspection of the transcripts in question that ICSISWB was better interpretable on speakers that had more casual speaking styles, whereas WSJ-5K was better on speakers with more rehearsed styles. We have used whichever of these baselines was the best interpretable in our experiments here (WSJ-5K for R and K, ICSISWB for G).

WSJ-5K		Lecture 1		Lecture 2		Lecture 3		Lecture 4	
	TS = %	20	33	20	33	20	33	20	33
Initial WER		**50.48**	**50.93**	**51.31**	**51.90**	**50.28**	**49.23**	**54.39**	**54.04**
XER	RT = 10	49.97	49.82	49.27	49.77	46.85	48.08	52.17	50.58
	RT = 5	50.01	50.07	49.99	51.13	48.39	47.37	50.91	49.62
	RT = 2	49.87	51.75	49.52	51.13	47.13	47.31	52.70	50.56
XER-NoS	RT = 10	47.25	46.82	49.98	48.72	48.44	45.21	51.37	49.73
	RT = 5	49.03	48.78	47.37	51.25	47.84	44.07	49.54	48.97
	RT = 2	52.21	53.47	49.31	52.29	50.85	49.41	50.63	51.81
S_{WER}	RT = 10	45.18	44.58	49.06	45.97	46.49	45.30	49.60	47.95
	RT = 5	44.82	43.82	46.73	45.52	45.64	43.18	47.79	46.74
	RT = 2	**44.04**	**43.99**	**45.81**	**45.16**	**44.35**	**41.49**	**46.89**	44.28

Table 2: Experimental evaluation: WER values for instructor R using the WSJ-5K language model.

WEB		Lecture 1		Lecture 2		Lecture 3		Lecture 4	
	TS = %	20	33	20	33	20	33	20	33
Initial WER		**45.54**	**45.85**	**43.36**	**43.87**	**46.69**	**47.14**	**49.78**	**49.38**
XER	RT = 10	42.91	43.90	42.44	43.81	46.78	45.35	46.92	49.65
	RT = 5	43.45	43.81	42.65	44.37	46.90	42.12	47.34	46.04
	RT = 2	43.26	45.46	44.19	44.66	43.77	45.12	61.54	60.40
XER-NoS	RT = 10	43.51	42.97	42.11	41.98	44.66	46.59	47.24	46.30
	RT = 5	44.96	42.98	40.01	40.52	44.66	41.74	47.23	44.35
	RT = 2	46.72	48.16	44.79	45.87	40.44	44.32	61.84	64.40
S_{WER}	RT = 10	41.98	41.44	42.11	40.75	44.66	45.27	47.24	45.85
	RT = 5	40.97	40.56	38.85	39.08	44.66	40.84	45.27	42.39
	RT = 2	**40.67**	**40.47**	**38.00**	**38.07**	**40.00**	**40.08**	**43.31**	**41.52**

Table 3: Experimental evaluation: WER values for instructor R using the WEB language models.

As for how the transcripts improve, words with lower information content (e.g., a lower tf.idf score) are corrected more often and with more improvement than words with higher information content. The topic-specific language model adaptation that the TBL follows upon benefits words with higher information content more. It is possible that the favour observed in TBL with S_{WER} towards lower information content is a bias produced by the preceding round of language model adaptation, but regardless, it provides a much-needed complementary effect. This can be observed in Tables 2 and 3, in which TBL produces nearly the same RER in either table for any lecture. We have also extensively experimented with the usability of lecture transcripts on human subjects (Munteanu et al., 2006), and have found that task-based usability varies in linear relation to WER.

An analysis of the rules selected by both TBL implementations revealed that using the XER approximation leads to several single-word rules being selected, such as rules removing all instances of frequent stop-words such as “the” and “for” or pronouns such as “he.” Therefore, an empirical improvement ($XER - NoS$) of the baseline was implemented that, beside pruning rules below the RT threshold, omits such single-word rules from being selected. As shown in Tables 2, 3 and 4, this restriction slightly improves the performance of the approximation-based TBL for some values of the RT and TS parameters, although it still does not consistently match the WER reductions of our scoring function.

Although the experimental evaluation shows positive improvements in transcript quality through TBL, in particular when using the S_{WER} scoring function, an exception is illustrated in Table 5. The recordings for this evaluation were collected from a course on Unix programming, and lectures were highly interactive. Instructor K used numerous examples of C or Shell code, many of them being developed and tested in class. While the keywords from a programming language can be easily added to the ASR lexicon, the pronunciation of such abbreviated forms (especially for Shell programming) and of mostly all variable and custom function names proved to be a significant difficulty for the ASR system. This, combined with a high speaking rate and often inconsistently truncated words, led to few TBL rules occurring even above the lowest $RT = 2$ threshold (despite many TBL rules being initially discovered).

As previously mentioned, one of the drawbacks of global TBL rule scoring is the heavy computational burden. The experiments conducted here, however, showed an average learning time of one hour per one-hour lecture, reaching at most three

WSJ-5K		Lecture 1		Lecture 2		Lecture 3		Lecture 4	
	TS = %	20	33	20	33	20	33	20	33
Initial WER		44.31	44.06	46.12	45.80	51.10	51.19	53.92	54.89
XER	RT = 10	44.31	44.06	46.12	46.55	51.10	51.19	53.92	54.89
	RT = 5	44.31	44.87	46.82	47.47	51.10	51.19	53.96	55.56
	RT = 2	47.46	55.21	50.54	51.01	52.60	54.93	57.48	60.46
XER-NoS	RT = 10	44.31	44.06	46.12	46.55	51.10	51.19	53.92	54.89
	RT = 5	44.31	44.87	46.82	47.47	51.10	51.19	53.96	55.56
	RT = 2	46.43	54.41	50.54	51.01	53.01	55.02	57.47	60.02
S_{WER}	RT = 10	44.31	44.06	46.12	45.80	51.10	51.19	53.92	54.89
	RT = 5	44.31	44.05	46.11	45.88	51.10	51.19	53.92	54.89
	RT = 2	44.34	44.07	46.03	45.89	50.96	50.93	54.01	55.16

Table 5: Experimental evaluation: WER values for instructor K using the WSJ-5K language model.

hours[4] for a threshold of 2 when training over transcripts for one third of a lecture. Therefore, it can be concluded that, despite being computationally more intensive than a heuristic approximation (for which the learning time is on the order of just a few minutes), a TBL system using a global, WER-correlated scoring function not only produces better transcripts, but also produces them in a feasible amount of time with only a small amount of manual transcription for each lecture.

5 Summary and Discussion

One of the challenges to reducing the WER of ASR transcriptions of lecture recordings is the lack of manual transcripts on which to train various ASR improvements. In particular, for one-hour lectures given by different lecturers (such as, for example, invited presentations), it is often impractical to manually transcribe parts of the lecture that would be useful as training or development data. However, transcripts for the first 10-15 minutes of a particular lecture can be easily obtained.

In this paper, we presented a solution that improves the quality of ASR transcripts for lectures. WER is reduced by 10% to 14%, with an average reduction of 12.9%, relative to initial values. This is achieved by making use of manual transcripts from as little as the first 10 minutes of a one-hour lecture. The proposed solution learns word-level transformation-based rules that attempt to replace parts of the ASR transcript with possible corrections. The experimental evaluation carried out over eleven lectures from three different courses and instructors shows that this amount of manual transcription can be sufficient to further improve a lecture-specific ASR system.

In particular, we demonstrated that a true WER-based scoring function for the TBL algorithm is both feasible and effective with a limited amount of training data and no development data. The proposed function assigns scores to TBL rules that directly correlate with reductions in the WER of the entire training set, leading to a better performance than that of a heuristic approximation. Furthermore, a scoring function that directly optimizes for WER reductions is more robust to variations in training size as well as to the value of the rule pruning threshold. As little as a value of 2 can be used for the threshold (scoring all rules that occur at least twice), with limited impact on the computational burden of learning the transformation rules.

References

E. Brill. 1992. A simple rule-based part of speech tagger. In *Proc. 3rd Conf. on Applied NLP (ANLP)*, pages 152 – 155.

P.R. Clarkson and Rosenfeld R. 1997. Statistical language modeling using the CMU-Cambridge Toolkit. In *Proc. Eurospeech*, volume 1, pages 2707–2710.

J.G. Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 347–354.

C. Fügen, M. Kolss, D. Bernreuther, M. Paulik, S. Stüker, S. Vogel, and A. Waibel. 2006. Open domain speech recognition & translation: Lectures and speeches. In *Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 569–572.

J. Glass, T.J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay. 2007. Recent progress in the MIT spoken lecture processing project. In *Proc. 10th EuroSpeech / 8th InterSpeech*, pages 2553–2556.

[4] It should be noted that, in order to preserve compatibility with other software tools, the code developed for these experiments was not optimized for speed. It is expected that a dedicated implementation would result in even lower runtimes.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pages 517–520.

D. Gusfield. 1997. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press.

T.J. Hazen. 2006. Automatic alignment and error correction of human generated transcripts for long speech recordings. In *Proc. 9th Intl. Conf. on Spoken Language Processing (ICSLP) / InterSpeech*, pages 1606–1609.

B-J. Hsu and J. Glass. 2006. Style & topic language model adaptation using HMM-LDA. In *Proc. ACL Conf. on Empirical Methods in NLP (EMNLP)*, pages 373–381.

Q. Huo and W. Li. 2007. An active approach to speaker and task adaptation based on automatic analysis of vocabulary confusability. In *Proc. 10th EuroSpeech / 8th InterSpeech*, pages 1569–1572.

A. Janin, Baron D., J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The ICSI meeting corpus. In *Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 364–367.

K. Kato, H. Nanjo, and T. Kawahara. 2000. Automatic transcription of lecture speech using topic-independent language modeling. In *Proc. Intl. Conf. on Spoken Language Processing (ICSLP)*, volume 1, pages 162–165.

D. Klakow. 2000. Selecting articles from the language model training corpus. In *Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1695–1698.

E. Leeuwis, M. Federico, and M. Cettolo. 2003. Language modeling and transcription of the TED corpus lectures. In *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 232–235.

L. Mangu and M. Padmanabhan. 2001. Error corrective mechanisms for speech recognition. In *Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 29–32.

C. Munteanu, R. Baecker, and G. Penn. 2008. Collaborative editing for improved usefulness and usability of transcript-enhanced webcasts. In *Proc. ACM SIGCHI Conf. (CHI)*, pages 373–382.

C. Munteanu, R. Baecker, G. Penn, E. Toms, and D. James. 2006. The effect of speech recognition accuracy rates on the usefulness and usability of webcast archives. In *Proc. ACM SIGCHI Conf. (CHI)*, pages 493–502.

C. Munteanu, G. Penn, and R. Baecker. 2007. Web-based language modelling for automatic lecture transcription. In *Proc. 10th EuroSpeech / 8th InterSpeech*, pages 2353–2356.

H. Nanjo and T. Kawahara. 2003. Unsupervised language model adaptation for lecture speech recognition. In *Proc. ISCA / IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR)*.

G. Ngai and R. Florian. 2001. Transformation-based learning in the fast lane. In *Proc. 2nd NAACL*, pages 1–8.

T. Niesler and D. Willett. 2002. Unsupervised language model adaptation for lecture speech transcription. In *Proc. Intl. Conf. on Spoken Language Processing (ICSLP/Interspeech)*, pages 1413–1416.

A. Park, T. J. Hazen, and J. R. Glass. 2005. Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling. In *Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*.

B. L. Pellom. 2001. SONIC: The university of colorado continuous speech recognizer. Technical Report #TR-CSLR-2001-01, University of Colorado.

J. Peters and C. Drexel. 2004. Transformation-based error correction for speech-to-text systems. In *Proc. Intl. Conf. on Spoken Language Processing (ICSLP/Interspeech)*, pages 1449–1452.

P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, and M. Siegler. 1997. The 1996 HUB-4 Sphinx-3 system. In *Proc. DARPA Speech Recognition Workshop*.

G. Riccardi and D. Hakkani-Tur. 2005. Active learning: Theory and applications to automatic speech recognition. *IEEE Trans. Speech and Audio Processing*, 13(4):504–511.

E. K. Ringger and J. F. Allen. 1996. Error correction via a post-processor for continuous speech recognition. In *Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 427–430.

E. Roche and Y. Schabes. 1995. Deterministic part-of-speech tagging with finite-state transducers. *Computational Linguistics*, 21(2):227–253.

L. von Ahn and L. Dabbish. 2004. Labeling images with a computer game. In *Proc. ACM SIGCHI Conf. (CHI)*, pages 319–326.