Self-Supervised Discriminative Training of Statistical Language Models

Puyang Xu[#], Damianos Karakos[#*], Sanjeev Khudanpur[#*]

[#]*Department of Electrical and Computer Engineering*
[#]*Center for Language and Speech Processing*
[*]*Human Language Technology Center of Excellence*
Johns Hopkins University
Baltimore, MD 21218, USA
email: {puyangxu,damianos,khudanpur}@jhu.edu

***Abstract*—A novel self-supervised discriminative training method for estimating language models for automatic speech recognition (ASR) is proposed. Unlike traditional discriminative training methods that require *transcribed* speech, only untranscribed speech and a large text corpus is required. An exponential form is assumed for the language model, as done in maximum entropy estimation, but the model is trained from the text using a discriminative criterion that targets word *confusions* actually witnessed in first-pass ASR output lattices. Specifically, model parameters are estimated to maximize the *likelihood ratio* between words w in the text corpus and w's *cohorts* in the test speech, i.e. other words that w competes with in the test lattices. Empirical results are presented to demonstrate statistically significant improvements over a 4-gram language model on a large vocabulary ASR task.**

I. Introduction

In language modeling, one is interested in computing a probability distribution over word sequences, such that sequences which are well-formed (in terms of fluency or semantic coherence, for instance) are given a higher likelihood than those which are not. A typical application of language modeling is automatic speech recognition (ASR), whose task is to produce a verbatim transcription of a speech segment presented to it. In a typical ASR system, the recognizer employs an acoustic model to measure the goodness of the stochastic match $P(s|\mathbf{w})$ of speech segment s with every candidate word sequence $\mathbf{w}$. Since many words and word sequences may result in similar speech signal, it is conventional to use language model $P(\mathbf{w})$ for the *a priori* probability of $\mathbf{w}$. The language model is usually estimated via a maximum likelihood criterion (in conjunction with some *smoothing* method) from a large corpus of text in the target language, domain and genre. Because of the nature of human language, data sparsity poses a serious challenge in the estimation of language models; a number of methods [1], [2] have been proposed to mitigate this problem.

Discriminative methods for language modeling have been recently proposed as an effective alternative to maximum likelihood methods [3], [4]. Given a development speech corpus which has been manually transcribed, and a lattice output of that corpus generated by a recognizer (with a baseline language model), the aim of these methods is to increase the likelihood of paths in the lattice which are more faithful to the manual transcription (in terms of WER), than the current most likely path. This is typically done by extracting appropriate features from the lattices, and then slightly changing the probabilities assigned by the baseline language model whenever these features are present. Interestingly, these methods are effective even when they just employ commonly used features, such as n-grams; there seems to be a gain in WER over n-gram maximum-likelihood based language models which have been trained on very large text corpora.

Despite the above gains, discriminative language modeling techniques are limited by the fact that they require a separate *manually transcribed* development corpus. The size of such a corpus plays an important role in the effectiveness of the discriminative model: as is well known, discriminative methods easily get overtrained on small corpora. Such development data are usually set aside for other system tuning task, but if they are used for discriminative training, they cannot be used for such purposes. Therefore, the transcribed speech for discriminative training is usually cannibalized from the acoustic model training corpus, which may be a difficult tradeoff to make, particularly in low resource situations.

This paper aims at overcoming the above limitation; we propose a novel idea for carrying out discriminative training of language models *without* manual transcriptions (self-training). The crux of our method is in learning to resolve residual *confusions* in the output of an ASR system: these are words which tend to be acoustically confusable, tend to appear in similar contexts, and are frequently in competition with each other in the output ASR lattice[1]. Even in the absence of manual transcriptions, we show that it is still possible to learn to disambiguate between such confused words, using contextual cues that are extracted from a text corpus. This line of work is in the spirit of *word sense disambiguation* (WSD) [5]. In WSD the goal is to classify ambiguous words into one of multiple senses (e.g., the word "bank" can refer to a financial institution

[1]Note that language model probabilities involving only such *mutually confusable* words need to be carefully adjusted, as word errors are the result of not being able to efficiently resolve these confusions.

or the bank of a river) using the surrounding context, and we try to do something similar here: we treat each set of frequently confused words as an entity with multiple "senses", and then we use a large text corpus (which contains the ground truth by the mere fact that each "sense" coincides with the word observed in the text) to learn to distinguish between them based on the surrounding context.

We have chosen to train such self-supervised discriminative language models within the exponential family of distributions[2]. As we show in a later section, learning can be done much more efficiently than in the standard maximum likelihood setting, despite the fact that the size of the training text can be very large. The reason for the improved computational complexity lies in the fact that the normalizing constant of the exponential model (which is computationally expensive to compute in standard maximum entropy because it involves all words of the vocabulary) only involves the relatively few words that appear to be confused with each other.

The remainder of the paper is organized as follows. In section II, we present the concept of word *cohorts* and describe methods for extracting them; the parameterization and training of the proposed language model is described in Section III. Experimental results are presented in Section IV along with some analysis in Section V, followed by conclusions in Section VI.

II. Word Cohorts in ASR

A. The Concept of a Cohort Set

Central to our methodology is the concept of a *cohort set* of words. These are words frequently in confusion with each other in the ASR output. We define a cohort set associated with a word w as all the words that are often in competition with w, and we denote it by $C(w)$. Properties of these cohort sets include: (i) Symmetry: if $w \in C(v)$, then $v \in C(w)$. (ii) Non-transitivity in the inclusion relationship; if $v \in C(w)$ and $w \in C(z)$, it is not true in general that $v \in C(z)$. Example cohorts from ASR are shown in Table I.

Note that, except in a few cases involving function words, cohort sets are typically very small; this fact is of great importance in the training of our models. Empirical statistics involving cohort sets appear in Section IV.

This concept of cohorts is general enough to be applicable to other fields such as machine translation, where a cohort set can be thought of as different translations of the same source word.

B. Determination of Cohort Sets $C(w)$

In the context of ASR, cohort sets can be obtained through a variety of means, depending on the format of the automatic recognition output:

[2]Exponential models are also used in *contrastive estimation* [6], a technique closely related to ours for learning discriminative models in an unsupervised way. In contrastive estimation one has to define the "neighborhood" of a training instance to discriminate against, and [6] did this by randomly perturbing the training instances. In our work, on the other hand, the "neighborhoods" are obtained in a more informed way, by using the set of words that the automatic recognizer is frequently confused about.

TABLE I
TYPICAL COHORT WORDS

w	$C(w)$, which always includes w itself
weight	weight wait wage wheat
venture	venture adventure
remained	remained remain remains means main maine
explanation	explanation explosion exploration

- *Lattice*: Each node in the lattice corresponds to a time point and n-gram history in the ASR output; therefore, the arcs that emanate from the same node are words in competition, and can be considered as cohort words.
- *Confusion Network*[7]: Cohorts can be easily defined in this setting, namely, words that appear in the same confusion bin are cohorts of each other.
- *N-best List*: By aligning the hypotheses from the n-best list using minimization of edit distance as a criterion, similar to what is done in ROVER [8], a confusion network can be generated; cohort sets can then be extracted as mentioned above.

In a real-time recognition scenario, where the test data arrive at a high speed and there is no time to extract cohort sets from it and do the subsequent training, one would need to use some pre-existing cohort sets extracted from some other (untranscribed) speech data. The hope would then be that the confusions seen in the test data fall into one of these cohort sets, especially if the amount of untranscribed speech is sufficiently large. We plan to report on this idea in subsequent work.

III. Model Parameterization and Training

One reason we chose the exponential family of language models [9] for self-supervised learning is that these models permit easy inclusion of a number of interesting features within a unified framework, and have a satisfactory axiomatic justification in the maximum entropy framework. Even though only n-gram features are considered in this work, other features (topic, syntactic, etc.) can be easily incorporated. Thus for any words w following a word sequence h,

$$\phi_a(h,w) = \begin{cases} 1 & \text{if } w = a, \\ 0 & \text{otherwise;} \end{cases}$$

$$\phi_{a\,b\,c}(h,w) = \begin{cases} 1 & \text{if } h \text{ ends in } a\,b \text{ and } w = c, \\ 0 & \text{otherwise;} \end{cases}$$

are the features for the unigram a and the trigram $a\,b\,c$ respectively. In our experiments, we only take features from the test set N-best list; more details appear in the next section.

Under such a framework, the conditional probability of the word w given context h can be expressed as,

$$P(w \mid h\,;\,\Theta) = \frac{1}{Z(\Theta,h)} \prod_{j=1}^{J} e^{\theta_j\,\phi_j(h,w)} = \frac{1}{Z(\Theta,h)} e^{\sum_{j=1}^{J} \theta_j\,\phi_j(h,w)} \quad (1)$$

where Z is the normalizer for the context h, J is the number of features in the model, and $\Theta = (\theta_1, \ldots, \theta_J)$ are the parameters to be estimated.

A. Estimation of Model Parameters

Consider the standard problem of estimating, from a training corpus $\mathcal{W} = \{\mathbf{w}_k, k = 1, \ldots, K\}$ of K sentences covered by a word-vocabulary V, the conditional probability $P(w_i \,|\, h_i)$ of the word in the i-th position of a sentence $\mathbf{w}$ given the context $h_i \equiv w_1 \ldots w_{i-1}$. The standard practice is to parameterize this conditional probability and estimate the parameters via maximum likelihood. Letting n_k denote the length of the sentence $\mathbf{w}_k$,

$$\begin{aligned} \widehat{\Theta} &= \arg\max_{\Theta} \prod_{k=1}^{K} \prod_{i=1}^{n_k} P(w_i \,|\, h_i \,;\, \Theta) \\ &= \arg\max_{\Theta} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \log P(w_i \,|\, h_i \,;\, \Theta). \end{aligned} \tag{2}$$

On the other hand, the standard practice in discriminative training, in essence, is to require (say) speech s_k corresponding to all the sentences $\mathbf{w}_k$ in $\mathcal{W}$, to then run the best available ASR system on this speech to produce a list of likely word hypotheses $\widehat{\mathbf{w}}_k$, and to adjust the parameters Θ to discriminate between the correct words w_i and the ASR output $\widehat{w}_i$ in the context h_i. In our case, the list of likely word hypotheses are what we have defined as cohort sets. Note that herein lies an important diffference from some of the previous work as in [3], [4], [10], where the target function is the ratio of the likelihood of the correct sentence and the likelihood of the alternative sentences. Here we are only focusing on word confusions and it is not necessary to know which word in the cohort set is the correct word; the language model training text hopefully contains enough occurrences of these words in order to learn which contextual cues are good for disambiguating them. All that is required is to identify $C(w)$, namely which words are in frequent competition with w. With a well-specified cohort set, the objective then becomes the maximization of the ratio between the likelihood of each word and its competing words, i.e.,

$$\begin{aligned} \widehat{\Theta}^* &= \arg\max_{\Theta} \prod_{k=1}^{K} \prod_{i=1}^{n_k} \frac{P(w_i \,|\, h_i \,;\, \Theta)}{\sum_{w \in C(w_i)} P(w \,|\, h_i \,;\, \Theta)} \\ &= \arg\max_{\Theta} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \log \frac{P(w_i \,|\, h_i \,;\, \Theta)}{\sum_{w \in C(w_i)} P(w \,|\, h_i \,;\, \Theta)}. \end{aligned} \tag{3}$$

The parameterization (1) significantly simplifies the estimation of the language model under both maximum likelihood criterion and our discriminative criterion. In particular, maximum likelihood estimation (2) reduces to

$$\begin{aligned} \widehat{\Theta} &= \arg\max_{\Theta} L(\Theta, \mathcal{W}) \\ &= \arg\max_{\Theta} \sum_{i=1}^{I} \Theta \, \Phi(h_i, w_i) \\ &- \log \sum_{w \in V} e^{\Theta \, \Phi(h_i, w)} \; - \; \lambda \, ||\Theta||^2, \end{aligned} \tag{4}$$

where we have reindexed the word-positions in the training text $\mathcal{W}$ to go from 1 to $I = \sum_{k=1}^{K} n_k$, and $\lambda \, ||\Theta||^2$ is a regularization term that helps prevent overfitting [11]. Similarly, the likelihood ratio maximization (3) reduces to

$$\begin{aligned} \widehat{\Theta}^* &= \arg\max_{\Theta} L^*(\Theta, W) \\ &= \arg\max_{\Theta} \sum_{i=1}^{I} \Theta \, \Phi(h_i, w_i) \\ &- \log \sum_{w \in C(w_i)} e^{\Theta \, \Phi(h_i, w)} \; - \; \lambda \, ||\Theta||^2. \end{aligned} \tag{5}$$

(4) differs from (5) only in the normalizer, where in the discriminative case the sum is over the cohort instead of the entire vocabulary. Remember that the complexity of training exponential models mostly comes from the enormous amount of time of computing the normalizer. In contrast, (5) makes the training considerably more manageable, because the cohort set size is much smaller than the vocabulary size.

The above criterion is a well studied maximization problem, and several algorithms are available for the solution. It is easy to see that the only difference about this problem is in the model expectation calculation, where each history in the training data can only be followed by a set of cohort words. We use a variant of generalized iterative scaling [12] for the optimization.

Obviously, the fact that (5) maximizes likelihood *ratios* (instead of likelihoods) can result in a situation where, in the presence of certain contexts, almost all of the probability mass is concentrated on certain words only. Thus, to make (5) return a reasonable probability for *any* plausible sequence of words (as is the usual objective of language modeling), maximum likelihood initialization can be used. In other words, before applying our discriminative criterion, we can build a standard maximum entropy language model, and then use some kind of regularization to make sure our model does not deviate too much from the maxmimum likelihood solution. Or alternatively, do linear interpolation with a standard n-gram model. We have experimented with both approaches in our experiments and have found that linear interpolation gives the best results.

To summarize, our methodology for self-supervised discriminative language modeling boils down to the following: (i) extract features and cohort sets from the test set (lattice or n-best list); (ii) apply the discriminative optimization (5), and, finally, (iii) interpolate with a standard n-gram language model.

IV. Experiments and Results

We evaluated our model with an n-best rescoring task using 100-best lists from the DARPA WSJ'92 and WSJ'93 20k open vocabulary data sets. The acoustic model used to generate the n-best list can be found in [13]. We used the sets 93et and 93dt for evaluation, and 92et for development (parameter tuning). The evaluation set contained 465 utterances, and the development set contained 333 utterances. The development experiment extracts features and cohort set from the development n-best list, the evaluation experiment takes them from the evaluation n-best list.

The baseline language model was a 4-gram modified Kneser-Ney smoothed language model built from 37M words in the NYT section of English Gigaword. We took the top 20k words from the corpus together with the words in the n-best list as the vocabulary; all out-of-vocabulary words were mapped to a special symbol *<unk>*.

The discriminative language model only used up to 4-gram features extracted from the test set n-best list; this was done to reduce computation, since no other features can "fire" during rescoring anyway. This caused the number of features to be significantly smaller than in the standard maximum likelihood case; for example, our language model built on the evaluation n-best list only contained 41577 features (among which only 17419 features are associated with words in the cohort set vocabulary and therefore updated in training), instead of the millions of features required in maximum likelihood. Furthermore, the cohort sets were also computed from the test set n-best list, after aligning them together into confusion networks (as mentioned in Section III). Function words (determined by imposing a cutoff on unigram counts from the English Gigaword) were excluded. The average size of the cohort sets ended up being slightly above 4 words.

We also computed a standard maximum entropy language model, using a speedup trick [14], and the same set of features from the test set n-best list. Even with the speedup, training took much longer than the training of the self-supervised language model.

For the baseline 4-gram language model, as well as the maximum entropy language model, we optimized the language model scaling factor α on the development data. For the self-supervised language model, we did a grid search for the best combination of scaling factor α, regularization contant λ, and interpolation weight β on the development data. The grid details are the following, α from 0 to 10, with step size 0.5, β from 0 to 1 (weights for the standard n-gram language model), with stepsize 0.05, λ is taken from $\{0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 1, 5\}$.

The best parameter setting for the standard n-gram language model on the development data turned out to be $\alpha = 6.5$, while for standard maximum entropy model it was $\alpha = 5.5$. For the self-supervised language model, the best parameters were $\alpha = 5.5$, $\beta = 0.9$, $\lambda = 0.1$.

Table II shows the results in terms of word error rate.

As expected, the discriminative language model performs poorly by itself. However, when combined with the standard n-gram, as we are suggesting, there's a 0.6% absolute improvement over baseline on both development and evaluation data, and the MAPSSWE test of the NIST *sclite* toolkit indicates that the improvement is statistically significant (with a p-value of 0.02 and 0.001 for the development and evaluation sets, respectively). Notice the standard maximum entropy model also gives extra gains when combined with the 4-gram (the interpolation weight is optimized on development set).

TABLE II
WORD ERROR RATE RESULTS

WER	92et (dev)	93et and 93dt (eval)
ASR output	12.9	18.4
4-gram LM rescoring (baseline)	12.1	17.5
Standard Maxent rescoring	12.5	17.9
Standard Maxent + 4-gram	11.9	17.2
Self-supervised LM rescoring	17.4	23.0
*Self-supervised LM + 4-gram	**11.5**	**16.9**
Oracle	6.1	9.5

V. Discussion and Future Work

In order to better understand where the WER improvement of the interpolated self-supervised language model comes from, we performed the following on the dev data. Align the two hypotheses obtained by n-gram language model and interpolated self-supervised language model. The resulting confusion network is then aligned to the reference. Therefore, all the changes in WER come from confusion bins where two words differ and one of them is the correct one.

Among all these bins, close to one third involve insertions or deletions, and most often, these are insertions or deletions of short function words. Since our model training does not involve function words, we excluded bins that contain function words or epsilons in the analysis.

Table III lists all the correct confusion resolution, and Table IV lists the incorrect ones. Also shown in the table are the probabilities for the confusable words given by n-gram language model (P_{Ng}) and interpolated self-supervised language model (P_{Ss}). The last column shows how much our method increases/decreases the n-gram probabilities. For all the successful cases, our method boosts the correct words probabilities by an average factor of 6.11, while the incorrect words also get amplified by 1.95. For the unsuccessful cases, the incorrect words probabilities are increased by an average factor of 6.71, the correct words are boosted by only 1.44. Probably what's more interesting than the average number is the fact that there are quite some cases where the word probabilities are increased drastically, overall, the degree of change to the word probabilities vary greatly. For example, in Table III, the probability for '(no doubt with) diminished' is increased by a factor a 31.61! A major reason for this is that function words are excluded in our training, there are no features that capture the fact that a function word is probably the most likely word following 'with'(with a, with the, with that ...). Therefore, these words contribute very little to the normalizer. The resulting distribution may put too much probability mass to some words based on limited

information in the context. Such overly confident decision can be dangerous because it may affect the word choices in its neighborhood, especially given the fact that the context is often very noisy and the cues are spurious. This indicates that regularizing the discriminative model parameters using the maximum likelihood estimate is vital as already mentioned in Section III-A. We achieve such smoothing by interpolating with a standard n-gram language model.

Speaking of noisy context, they are most likely never seen in the training text, it is very likely that our model could hardly find any meaningful cues in its neighboring words, and the decision relies solely upon some less informative features. For example, in Table IV, '(to the zone) homeland' vs.'(to resolve on) land', the context is so confusing that it becomes very hard to give a probability to 'homeland' or 'land' in either context. Another example where 'exploration' and 'explosion' are in confusion following 'eight six challenger', without the knowledge of this event, there's probably no way to figure out which one is the correct word by only lexical evidence within the trigram context, we would probably have to search for longer distance dependency, a trigger word such as 'fire' or 'die' could probably make the decision easier.

TABLE III
CASES OF *SUCCESSFUL* COHORT WORDS RESOLUTION. WORDS IN BOLD ARE THE CORRECT WORDS AS CHOSEN BY OUR METHOD. BRACKETED WORDS ARE THE TRIGRAM CONTEXT.

Competing hypotheses	P_{Ng}	P_{Ss}	$\frac{P_{Ss}}{P_{Ng}}$
(were mixed with) **beans**	$4.72 * 10^{-6}$	$2.49 * 10^{-5}$	5.28
(were mixed with) gains	$2.80 * 10^{-2}$	$2.70 * 10^{-2}$	0.96
(no doubt with) **diminished**	$1.49 * 10^{-5}$	$4.71 * 10^{-4}$	31.61
(no doubt would) diminish	$2.73 * 10^{-4}$	$2.76 * 10^{-4}$	1.17
(<s> we) **felt**	$2.95 * 10^{-3}$	$3.25 * 10^{-3}$	1.10
(<s> we) thought	$4.33 * 10^{-3}$	$4.75 * 10^{-3}$	1.10
(bank holding companies) **slated**	$3.40 * 10^{-5}$	$3.11 * 10^{-4}$	9.15
(bank holding companies) waited	$9.37 * 10^{-6}$	$6.79 * 10^{-5}$	7.25
(<s> when flights) **arrive**	$1.17 * 10^{-5}$	$3.33 * 10^{-4}$	2.85
(<s> when flights) arrived	$3.80 * 10^{-4}$	$5.35 * 10^{-4}$	1.41
(wait for a) **gate**	$2.29 * 10^{-5}$	$2.64 * 10^{-5}$	1.15
(wait for a) date	$1.60 * 10^{-4}$	$1.59 * 10^{-4}$	0.99
(<s> his) **m.**	$6.62 * 10^{-5}$	$9.75 * 10^{-4}$	14.73
(<s> his) n.	$2.89 * 10^{-4}$	$8.75 * 10^{-4}$	3.03
(a few clearly) **thoughts**	$1.95 * 10^{-5}$	$2.40 * 10^{-5}$	1.23
(no doubt would) thought	$5.13 * 10^{-4}$	$4.79 * 10^{-4}$	0.93
(not at all) **unhappy**	$4.31 * 10^{-6}$	$2.95 * 10^{-5}$	6.84
(at all and) happy	$8.44 * 10^{-5}$	$1.16 * 10^{-4}$	1.37
(slowdown because they) **continue**	$2.81 * 10^{-4}$	$3.22 * 10^{-3}$	11.46
(slowdown because they) continued	$2.31 * 10^{-4}$	$6.42 * 10^{-4}$	2.78
('s continued group) **rate**	$1.94 * 10^{-4}$	$3.52 * 10^{-4}$	1.81
(employers continue to) pray	$1.22 * 10^{-4}$	$1.48 * 10^{-4}$	1.21
(<s> the) **worries**	$1.14 * 10^{-5}$	$1.45 * 10^{-5}$	1.27
(<s> the) warriors	$1.54 * 10^{-4}$	$1.78 * 10^{-4}$	1.14
(at c. p.) **pulled**	$5.23 * 10^{-5}$	$5.40 * 10^{-5}$	1.03
(or at sea) people	$2.49 * 10^{-4}$	$4.14 * 10^{-4}$	1.66
(the employer 's) **continued**	$2.21 * 10^{-4}$	$1.40 * 10^{-3}$	6.33
(outside the employers) continue	$7.90 * 10^{-4}$	$1.27 * 10^{-3}$	1.61
(blue chip economists) **expect**	$7.60 * 10^{-3}$	$6.90 * 10^{-3}$	0.91
(blue chip economists) expected	$1.63 * 10^{-2}$	$1.48 * 10^{-2}$	0.91
(that has 'nt) **stopped**	$6.30 * 10^{-2}$	$7.19 * 10^{-2}$	1.14
(that has 'nt) stop	$9.35 * 10^{-5}$	$3.46 * 10^{-4}$	3.70

TABLE IV
CASES OF *UNSUCCESSFUL* COHORT WORDS RESOLUTION. WORDS IN BOLD ARE THE CORRECT WORDS AS CHOSEN *NOT* BY OUR METHOD. BRACKETED WORDS ARE THE TRIGRAM CONTEXT.

Competing hypotheses	P_{Ng}	P_{Ss}	$\frac{P_{Ss}}{P_{Ng}}$
(<s>) fiat	$6.54 * 10^{-5}$	$1.07 * 10^{-4}$	1.67
(<s>) **five**	$3.94 * 10^{-4}$	$4.96 * 10^{-4}$	1.26
(eighty six challenger) exploration	$1.42 * 10^{-5}$	$4.85 * 10^{-5}$	3.42
(eighty six challenger) **explosion**	$9.01 * 10^{-4}$	$9.95 * 10^{-4}$	1.10
(the weak dollar) meaning	$5.93 * 10^{-5}$	$9.91 * 10^{-4}$	16.71
(weak dollar may) **mean**	$2.87 * 10^{-3}$	$2.71 * 10^{-3}$	0.94
(years old mister) wayne	$1.87 * 10^{-5}$	$4.47 * 10^{-5}$	2.39
(years old mister) **wang**	$1.43 * 10^{-4}$	$1.59 * 10^{-4}$	1.11
(probably the largest) contributed	$4.75 * 10^{-5}$	$1.03 * 10^{-4}$	2.17
(probably the largest) **contributor**	$1.23 * 10^{-3}$	$2.66 * 10^{-3}$	2.16
(u. s. as) cooperation	$1.94 * 10^{-6}$	$1.87 * 10^{-5}$	9.64
(s. as quite) **operation**	$6.86 * 10^{-5}$	$8.97 * 10^{-5}$	1.31
(too small and) identified	$2.15 * 10^{-5}$	$1.21 * 10^{-4}$	5.63
(at two small) **unidentified**	$5.52 * 10^{-6}$	$9.78 * 10^{-6}$	1.77
(belgique 's shares) appear	$2.01 * 10^{-5}$	$3.27 * 10^{-4}$	16.27
(belgique 's shares) **appeared**	$2.97 * 10^{-5}$	$6.83 * 10^{-5}$	2.30
(to the zone) homeland	$3.77 * 10^{-6}$	$9.22 * 10^{-6}$	2.45
(to resolve on) **land**	$5.37 * 10^{-4}$	$5.20 * 10^{-4}$	0.97

VI. CONCLUSIONS

In this work, we make a first attempt to build a self-supervised discriminative language model. The work differs from previous research on discriminative language modeling in that we do not need an extra amount of manual transcriptions of speech, in addition of what is provided for training a baseline ASR system. We propose the idea of word cohorts, and design an optimization criterion that makes the language model more discriminative among those words. Significant improvement in WER is obtained on a n-best list rescoring task from WSJ93.

ACKNOWLEDGMENT

The authors are grateful to Denis Filimonov and Mary Harper for providing the n-best lists and for pre-processing the language model training text used in the experiments reported here. This work was partially supported by National Science Foundation Grant No 0840112.

REFERENCES

[1] F. Jelinek, *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[2] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th Annual Meeting of the ACL*, 1996, pp. 310–318.

[3] B.Roark, M.Saraclar, and M.Colins, "Corrective language modeling for large vocabulary asr with the perceptron algorithm," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Quebec, Canada, May 2004, pp. 749–752.

[4] B.Roark, M.Saraclar, M.Colins, and M.Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," in *Proc. the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Barcelona, Spain, Jul. 2004, pp. 47–54.

[5] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Prentice Hall, 2000.

[6] N. A. Smith and J. Eisner, "Contrastive estimation: Training log-linear models on unlabeled data," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, Michigan, June 2005, pp. 354–362.

[7] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," in *Proc. Eurospeech*, 1999, pp. 495–498.

[8] J. Fiscus, "A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (ROVER)," in *Proc. 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA, 1997, pp. 347–352.

[9] R.Rosenfeld, "A maximum entropy approach to adaptive statistical language modeling," *Computer Speech and Language*, vol. 10, pp. 187–228, Mar. 1996.

[10] M.Colins, M.Saraclar, and B.Roark, "Discriminative syntactic language modeling for speech recognition," in *Proc. the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Sydney, Australia, Jun. 2005, pp. 507–514.

[11] S. F. Chen and R. Rosenfeld, "A Gaussian prior for smoothing maximum entropy models," Tech. Rep., 1999.

[12] J.Darroch and D.Ratcliff, "Generalized iterative scaling for log-linear models," *The Annals of Mathematical Statistics*, vol. 24, pp. 413–421, 1972.

[13] W. Wang and M. P. Harper, "The SuperARV language model: Investigating the effectiveness of tightly integrating multiple knowledge sources," in *Proc. the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, Jul. 2002, pp. 238–247.

[14] J. Wu and S. Khudanpur, "Efficient training methods for maximum entropy language modeling," in *Proc. the 6th International Conference on Spoken Language Technologies (ICSLP-00*, 2000, pp. 114–117.