TRIGGER-BASED LANGUAGE MODELING USING A LOSS-SENSITIVE PERCEPTRON ALGORITHM

Natasha Singh-Miller

MIT CSAIL
32 Vassar St., Cambridge, MA 02139
natashas@csail.mit.edu

Michael Collins

MIT CSAIL
32 Vassar St., Cambridge, MA 02139
mcollins@csail.mit.edu

ABSTRACT

Discriminative language models using n-gram features have been shown to be effective in reducing speech recognition word error rates. In this paper we describe a method for incorporating discourse-level triggers into a discriminative language model. Triggers are features identifying re-occurrence of words within a conversation. We introduce triggers that are specific to particular unigrams and bigrams, as well as "back off" trigger features that allow generalizations to be made across different unigrams. We train our model using a new loss-sensitive variant of the perceptron algorithm that makes effective use of information from multiple hypotheses in an n-best list. We train and test on the Switchboard data set and show a 0.5 absolute reduction in WER over a baseline discriminative model which uses n-gram features alone, and a 1.5 absolute reduction in WER over the baseline recognizer.

***Index Terms*—** Perceptrons, Speech recognition, Natural languages

1. INTRODUCTION

Previous work on discriminative language modeling [1] has considered models where the optimal string $\mathbf{w}^*$ for a given acoustic input $\mathbf{a}$ is defined as follows:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \left(\beta \log P_l(\mathbf{w}) + \log P_a(\mathbf{a}|\mathbf{w}) + \langle \bar{\alpha}, \Phi(\mathbf{a}, \mathbf{w}) \rangle\right)$$

In this approach, a standard language model, P_l, and an acoustic model, P_a, are used alongside a linear correction term $\langle \bar{\alpha}, \Phi(\mathbf{a}, \mathbf{w}) \rangle$.[1] $\Phi(\mathbf{a}, \mathbf{w})$ is a *feature-vector representation* of the pair $(\mathbf{a}, \mathbf{w})$, and $\bar{\alpha}$ is a parameter vector of the same dimensionality as $\Phi(\mathbf{a}, \mathbf{w})$. The parameters $\bar{\alpha}$ are estimated using discriminative methods (e.g. the perceptron algorithm). Improvements in word error rate (WER) have been observed by incorporating both n-gram and syntactic features within $\Phi(\mathbf{a}, \mathbf{w})$ [1, 2].

In this paper we consider two extensions to the discriminative language modeling approach. Our first contribution is to describe a method for including *trigger features* [3, 4] within the definition of $\Phi(\mathbf{a}, \mathbf{w})$. Trigger features are designed to model the fact that content words are more likely to be used repeatedly within a single conversation than to occur evenly spread throughout all speech. For example the word "Uzbekistan" may occur very rarely, but within the context of a conversation where it has already occurred, the likelihood of seeing "Uzbekistan" again increases dramatically. To capture this phenomenon in our model, a trigger feature can be defined that indicates the number of times in a conversation that "Uzbekistan" is seen preceded by a previous instance of "Uzbekistan". In addition to lexically-specific trigger features, we also introduce *backoff* trigger features where content words are placed into different equivalence classes based on their TF-IDF scores [5]. The use of lexicalized trigger features within a generative language model, i.e., a model that attempts to estimate $P_l(\mathbf{w})$, is described in [3, 4]. However, our use of trigger features in a discriminative language model is arguably simpler and more direct—in particular, the parameter estimation method is more closely related to optimizing WER.

Our second contribution is to introduce a new loss-sensitive variant of the perceptron algorithm for the estimation of $\bar{\alpha}$. This perceptron is similar in form to that proposed by [7] for multiclass classification, however it explicitly models the loss of selecting different hypotheses, and also takes into account the fact that multiple hypotheses may be considered optimal. In contrast to the work in [1], this perceptron algorithm makes updates based on averaging the contribution from a larger number of hypotheses, potentially making much better use of the information in the hypothesis set.

We tested our model on the Switchboard corpus using the recognizer of [6] and the discriminative language model of [1] as baselines. Our model demonstrates a 0.5 absolute reduction in WER over the model in [1], and a 1.5 absolute reduction in WER over the baseline recognizer of [6].

2. FEATURES

In this section, we describe how to extend the discriminative model described above in order to include trigger features in the model. We will use the following definitions:

- $\mathbf{a}_1 \ldots \mathbf{a}_n$ represents a sequence of acoustic inputs constituting a single conversation.
- GEN$(\mathbf{a}_i)$ denotes the set of n-best hypotheses produced by the baseline recognizer for the acoustic input $\mathbf{a}_i$.
- $\mathbf{v}_i$ designates the transcription of $\mathbf{a}_i$ we use to construct histories for identifying triggering events.
- $\mathbf{h}_i = \{\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}\}$ is the history of $\mathbf{a}_i$.
- $\Phi(\mathbf{a}_i, \mathbf{w}, \mathbf{h}_i)$ is a feature-vector representation. We assume that the score assigned by the generative model is the first feature in this vector (i.e., $\Phi_1(\mathbf{a}, \mathbf{w}, \mathbf{h}) = \log P_a(\mathbf{a}|\mathbf{w}) + \beta P_l(\mathbf{w})$).
- The resulting decoding model is:

$$\mathbf{w}_i^* = \arg\max_{\mathbf{w}} \langle \bar{\alpha}, \Phi(\mathbf{a}_i, \mathbf{w}, \mathbf{h}_i) \rangle$$

[1] β is a positive constant that determines the relative weight of the language and acoustic models. We use $\langle x, y \rangle$ to denote the inner product of two vectors x and y.

For training $\bar{\alpha}$, we assume that the baseline speech recognizer can be used to generate an n-best list of candidate hypotheses for any acoustic input. During training, $\mathbf{v}_i$ is the least errorful hypothesis in GEN($\mathbf{a}_i$). During decoding, $\mathbf{v}_i$ is the best scoring hypothesis under the generative model for each $\mathbf{a}_i$. We also experimented with defining $\mathbf{v}_i$ to be the hypotheses selected while decoding, but this gave neglible differences in performance.

The baseline discriminative model and our new model both include the following features. The first feature is the score assigned by the recognizer as described above. The remaining features include unigram, bigram, and trigram features. As one example, a trigram feature would be

$$\Phi_2(\mathbf{a}, \mathbf{w}, \mathbf{h}) = \text{ number of times } \textit{the dog barked} \text{ appears in } \mathbf{w}$$

Similar features are defined for all unigrams, bigrams, and trigrams seen in the n-best lists of the training data.

2.1. Trigger Features

We augment the baseline model with trigger features designed to capture information about the re-occurrence of words. These features operate at the discourse level in that they depend upon the words of the current candidate hypothesis as well as all other words that have occurred in previous utterances in the conversation. The *unigram trigger* features, created for all unigrams seen in the training data, are of the following form.

$\Phi_3(\mathbf{a}, \mathbf{w}, \mathbf{h}) =$
1 iff: (a) *Uzbekistan* is seen in $\mathbf{w}$ at least twice; or (b) *Uzbekistan* is seen in $\mathbf{w}$ once and is seen at least once in the history $\mathbf{h}$
0 otherwise

In addition to unigram features, we include *bigram trigger* features. For example, we might have a feature that is similar to Φ_3 above, but tests for the bigram *San Francisco*. Features of this form are created for all bigrams seen in training data.

Since the above features are lexicalized—i.e., there is a separate feature for each distinct unigram or bigram—some may be very sparse within our training set. To counteract this shortcoming we introduce a set of *backoff trigger* features. Each word w in the vocabulary is assigned to one of eleven *bins* based on its TF-IDF score [5]. The TF-IDF score is defined as follows for any word w and conversation d, where w is seen in d:

$$score(w, d) = (1 + \log(tf_{wd}))(\log \frac{n}{df_w})$$

Here df_w is the number of conversations in which the word w occurs, n is the total number of conversations in training data, and tf_{wd} is the number of times word w occurs in conversation d. The score for a word w, which we will denote as $score(w)$, is the average of $score(w, d)$ over all conversations d that contain w. The function $score(w)$ attempts to measure the degree to which the word w is a content word (and thus is likely to be a good trigger feature). We calculated TF-IDF scores for each word seen in the training data, using the 4,800 transcribed conversation sides in the Switchboard training set as documents.[2]

Words are then placed into bins according to their score. Words with $score(w)$ less than 1.0 are assigned to bin_0. All remaining words are sorted by increasing score and divided into ten equal-sized bins. In practice bin_0 consists of roughly the hundred most common words in speech (e.g. *a*, *with*, *go*, etc.). Since these words are so frequent, we anticipate that their trigger features will behave differently from the other words in the vocabulary. We create the other ten bins in this graded manner because we anticipate that different content levels will result in different trigger behavior.

One feature for each bin is then added to the model. Suppose Φ_w for any word w in the vocabulary is a trigger feature for that word (for example, $\Phi_{Uzbekistan}$ would be defined as in the example above). For each bin_b, for $b = 0 \ldots 10$, we define a feature as follows:

$$\Phi_b(\mathbf{a}, \mathbf{w}, \mathbf{h}) = \sum_{v \in bin_b} \Phi_v(\mathbf{a}, \mathbf{w}, \mathbf{h})$$

The feature Φ_b counts the number of triggering events involving the words in bin_b. These features allow the model to learn a general preference for triggering events involving each of the 11 bins.

3. TRAINING: PERCEPTRON

Figure 1 show the loss-sensitive perceptron algorithm we use for training $\bar{\alpha}$. This perceptron is similar in form to the perceptrons proposed by [7] for multiclass classification and by [8] for reranking. The perceptron is loss-sensitive in two ways. First, the perceptron enforces a margin that scales linearly with increases in loss. Second, the perceptron recognizes that there may be multiple hypotheses with minimal loss that should all be considered optimal.

In a given n-best list, GEN($\mathbf{a}_i$), there may be one or more optimal hypotheses. For example, the correct transcription may not be present in the list, but there may be several hypotheses each with only one error, while all the other hypotheses have two or more errors. We denote the set of lowest error hypotheses of GEN($\mathbf{a}_i$) by G_i. In terms of performance, all members of G_i are considered optimal choices by the discriminative model.

Let B_i = GEN($\mathbf{a}_i$) $- G_i$, i.e. the set of all non-optimal hypotheses in GEN($\mathbf{a}_i$). Each hypothesis in B_i will display different numbers and types of errors. The following loss function is used to indicate the badness of each member of B_i:

$$\Delta_i(b) = edits(b) - edits(g) \text{ where } g \text{ is any member of } G_i$$

This loss function is simply the additional number of errors introduced by a hypothesis over the number of errors present in an optimal hypothesis. Note that all members of G_i have a loss of 0, while all members of B_i have a loss of 1 or greater.

We define a margin that scales as $\lambda\Delta_i(b)$ where $\lambda \geq 0$ is a parameter we select. Scaling the margin with the loss was originally proposed by [9], who give statistical bounds justifying this. Intuitively, the idea is to ensure that hypotheses with a large number of errors are more strongly separated from the members of G_i. In the experiments presented in this paper λ is always set to 1.0. We define the two sets $C_i \subseteq G_i$ and $E_i \subseteq B_i$ in Figure 1 which consist of optimal and non-optimal hypotheses, respectively, that violate the scaled margin. We then construct two new vectors $\sum_{c \in C_i} \tau(c)\Phi_i(c)$ and $\sum_{e \in E_i} \tau(e)\Phi_i(e)$, which are used to train the perceptron in the usual way. The values of τ must meet the constraints described in Figure 1. The first four constraints insure that the weights used to create the representative samples are all non-negative and sum to 1. The final constraint insures that the newly constructed average samples still violate the margin constraint in an averaged sense.

[2]Note that we used the reference transcriptions for calculating TF-IDF scores, as opposed to the outputs from the baseline recognizer.

Note that the training examples used as input to the algorithm are constructed in the following way. $\mathbf{a}_1 \ldots \mathbf{a}_m$ is a sequence of acoustic representations formed by concatenating all conversations in the training data. The histories $\mathbf{h}_i$ are constructed as follows. We take $\mathbf{w}_i^*$ to be the member of G_i that is scored highest by the generative model. We define the history, $\mathbf{h}_i$, for utterance $\mathbf{a}_i$ to be the sequence $\mathbf{w}_{i-l}^*, \mathbf{w}_{i-l+1}^*, \ldots, \mathbf{w}_{i-1}^*$ where l is the number of previous utterances which belong in the current conversation.

There are many methods for selecting the values of τ. In this paper we consider the following simple definition:

$$\forall c \in C_i, \tau(c) = \frac{1}{|C_i|}$$

$$\forall e \in E_i, \tau(e) = \sum_{c \in C_i} \frac{v_c(e)}{|C_i| v_c^{total}}$$

$$v_c(e) = \begin{cases} 1 & \text{if } \langle \bar{\alpha}, (\Phi_i(c) - \Phi_i(e)) \rangle < \lambda \Delta_i(e) \\ 0 & \text{otherwise} \end{cases}$$

$$v_c^{total} = \sum_{e \in E_i} v_c(e)$$

Essentially all the hypotheses in C_i receive an equal positive weight. The weights of the hypotheses in E_i are assigned based on the values $v_c(e)$. If $v_c(e)$ is 1 for many correct hypotheses c, $\tau(e)$ will be relatively high.

The more standard perceptron used in the baseline model can be thought of as a special case of this perceptron in which $\lambda = 0$ and the τ values are assigned as follows. We designate some $c' \in G_i$ as the single best hypothesis (for the baseline, the hypothesis in G_i with the best recognizer score). We update only if $c' \in C_i$. We set $\tau(c') = 1$ and $\tau(e) = 1$ where e is the member of E_i for which $\langle \bar{\alpha}, (\Phi_i(c') - \Phi_i(e)) \rangle$ is the lowest. All other τ values are set to 0.

We can prove some useful properties for the perceptron in Figure 1. Consider the case where the training data is linearly separable, or more specifically there exists some vector $\mathbf{U}$ and some maximal margin $\delta > 0$ such that $||\mathbf{U}|| = 1$ and the following constraint holds for all i:

$$\langle \mathbf{U}, (\Phi_i(g) - \Phi_i(b)) \rangle \geq \delta \Delta_i(b) \quad \forall b \in B_i, \forall g \in G_i$$

It can be shown that in a finite number of iterations, given that the values for τ satisfy the given constraints, the perceptron in Figure 1 learns a model $\bar{\alpha}$ that separates the data as follows:[3]

$$\langle \frac{\bar{\alpha}}{||\bar{\alpha}||}, (\Phi_i(g) - \Phi_i(b)) \rangle \geq \gamma \Delta_i(b) \quad \forall b \in B_i, \forall g \in G_i$$

where $\gamma = \frac{\lambda}{2\lambda + \frac{4R^2}{s}} \times \delta$, R is an upper bound on the maximum length of a sample feature vector, and s is the minimum size of the loss seen on an error. (For our loss function we have $s = 1$.) Note that as $\lambda \to \infty$, $\gamma \to \frac{\delta}{2}$.

Input: An integer T specifying the number of training iterations. A sequence of inputs $\mathbf{a}_1 \ldots \mathbf{a}_m$. A function GEN($\mathbf{a}_i$) that produces an n-best list of outputs for the input $\mathbf{a}_i$. A mapping $\mathbf{h}_i$ that represents the history for $\mathbf{a}_i$. A function $\Delta_i(\mathbf{w})$ that represents the loss of selecting output $\mathbf{w}$ for the sample $\mathbf{a}_i$. Δ_i must always be non-negative and there must be at least one member of GEN($\mathbf{a}_i$) with a loss equal to 0.

Definitions: $G_i = \{\mathbf{w} | \mathbf{w} \in \text{GEN}(\mathbf{a}_i) \text{ and } \Delta_i(\mathbf{w}) = 0\}$
$B_i = \text{GEN}(\mathbf{a}_i) - G_i$
Let $\Phi_i(\mathbf{w})$ be shorthand for $\Phi(\mathbf{a}_i, \mathbf{w}, \mathbf{h}_i)$

Algorithm:
$\bar{\alpha} \leftarrow 0$ $\lambda \leftarrow 1.0$
For $t = 1$ to T, $i = 1$ to m

- $C_i = \{c | c \in G_i \text{ and } \exists z \text{ such that } z \in B_i \text{ and } \langle \bar{\alpha}, \Phi_i(c) - \Phi_i(z) \rangle < \lambda \Delta_i(z)\}$
- $E_i = \{e | e \in B_i \text{ and } \exists y \text{ such that } y \in G_i \text{ and } \langle \bar{\alpha}, \Phi_i(y) - \Phi_i(e) \rangle < \lambda \Delta_i(e)\}$
- If $|C_i| \neq 0$, define a function τ over $C_i \cup E_i$ such that the following constraints hold:
  - $\sum_{c \in C_i} \tau(c) = 1$
  - $\sum_{e \in E_i} \tau(e) = 1$
  - $\forall c \in C_i, \tau(c) \geq 0$
  - $\forall e \in E_i, \tau(e) \geq 0$
  - $\langle \bar{\alpha}, \left(\sum_{c \in C_i} \tau(c) \Phi_i(c) - \sum_{e \in E_i} \tau(e) \Phi_i(e) \right) \rangle < \lambda \left(\sum_{e \in E_i} \tau(e) \Delta_i(e) \right)$

  Update the parameters:
  $\bar{\alpha} \leftarrow \bar{\alpha} + \sum_{c \in C_i} \tau(c) \Phi_i(c) - \sum_{e \in E_i} \tau(e) \Phi_i(e)$

Output: The parameters $\bar{\alpha}$.

Fig. 1. The perceptron algorithm we propose for reranking speech recognition output. In our experiments we used the averaged parameters from the perceptron, see [1] for details.

4. EXPERIMENTS

We use the recognizer of [6] as our baseline recognizer (base-G) and to generate 1000-best lists used by the discriminative models. The discriminative model used in [1] also serves as a baseline (base-D). We train the rerankers using Switchboard [10], Switchboard Cellular [11], and CallHome [12] data. Rich Transcription 2002 (rt02) [13] data was used for development. Rich Transcription 2003 (rt03) [14] data was used for testing. The training set consisted of 5533 conversation sides (individual speakers in a conversation), or about 3.3 million words. The development set consisted of 120 conversation sides (6081 sentences) and the test set consisted of 144 conversation sides (9050 sentences).

The perceptron trains very quickly, usually converging within three passes over the training data, and we optimize the exact number of iterations using the development set. We report results for the test set only for the baseline models base-D and base-G, and for the model that produces the best results on rt02.

We tested several combinations of the trigger features and report results in Table 1. We find that including all three types of trigger features—unigram self-triggers, bigram self-triggers, and back-off triggers—gives us the best results on the development set. This model gives us a 0.4% absolute reduction in WER over base-D and a 1.2% absolute reduction in WER over base-G on the development set. This optimal model also achieves a 0.5% absolute reduction in WER over base-D for the test set and a 1.5% absolute reduction in WER over base-G. The results on rt03 are significant with $p < 0.01$

[3] For a proof of this result see the appendix in the online version of this paper.

Features	rt02	rt03
Base-G	37.0	36.4
Base-D	36.2	35.4
Loss-sensitive perceptron: n-grams	36.0	35.3
+ unigram self-triggers	36.0	
+ bigram self-triggers	36.0	
+ backoff unigram self-triggers	35.9	
+ unigram and bigram self-triggers	35.8	
+ unigram and backoff unigram self-triggers	35.9	
+ unigram, bigram, and backoff unigram self-triggers	**35.8**	**34.9**

Table 1. Results of base-G, base-D, and our discriminative model on the development set (rt02) and the test set (rt03).

bin_0	0.18	bin_4	13.51	bin_8	3.61
bin_1	6.96	bin_5	14.62	bin_9	16.00
bin_2	12.21	bin_6	18.44	bin_{10}	8.76
bin_3	13.64	bin_7	2.65		

Table 2. Weights of bin features in final model.

using the sign test at the conversation level.

We created several bins for the backoff trigger features with the expectation that words with different frequencies and content levels would have different trigger behavior. The learned weights of these features for the final discriminative model are listed in Table 2. From bin_0 to bin_6 the learned weights increase. This confirms our hypothesis that words with increasing content levels (or bin numbers) are more influenced by triggering events. We see approximately a three-fold increase in weight between bin_1 and bin_6, suggesting that the difference in behavior between the words in the two bins is quite large, and therefore it may be worthwhile to try to create a backoff scheme that is more sensitive to these differences. Finally, somewhat erratic weights are seen for bin_7 through bin_{10}. One reason for this may be that these are the rarest words in the training set, and therefore weights for these bins are not adequately trained.

The words which have the 20 highest weights for their associated unigram trigger features are listed in Table 3. These include content words such as *truck*, as well as stylistic words such as *gonna*. We posit that words such as *gonna* get high trigger weights because they are more heavily used by some speakers than others. Additionally, we see that some of the words in the list are homonyms of other words, such as *wear* and *where*, *wood* and *would*, and *weather* and *whether*. It seems likely that the occurrence of one of these words earlier in a discourse should make it more likely to see it later and help distinguish between homonyms.

Finally we see that the perceptron algorithm we present provides additional gains over the baseline perceptron algorithm. Future work might consider alternative ways to select the parameters τ as this might lead to further gains.

5. CONCLUSION

In this paper we use the discriminative language model of [1] to create a reranker that includes discourse–level features. Specifically, we introduce trigger features that help the discriminative language model to adapt to discourse context. We use lexicalized and backoff trigger features that each show individual improvements and together make a substantial gain over the baseline model. Additionally we present a perceptron algorithm used to train the discriminative model that shows improvements as well. Overall, the WER on the test set was reduced by 0.5 over the baseline discriminative model, and by 1.5 over the baseline recognizer. This work provides evidence that discriminative language modeling has the potential to deliver significant gains for speech recognition tasks. The success of the trigger features also shows how important discourse level information can be to transcribing spoken language.

GONNA	WEATHER	WANNA	WOOD
OIL	WEAR	LAKE	WAR
ONE'S	SOMEONE	ICE	ALRIGHT
TRUST	PARTS	TRUCK	RIDE
READING	SOMEPLACE	UH-HUH	WOMEN

Table 3. The 20 unigram trigger features with highest weight in the final model.

6. REFERENCES

[1] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," in *Proceeding of the 42nd Annual Meeting of the ACL*, 2004, pp. 48–55.

[2] M. Collins, B. Roark, and M. Saraclar, "Discriminative Syntactic Language Modeling for Speech Recognition," in *Proceeding of the 43rd Annual Meeting of the ACL*, 2005, pp. 507–514.

[3] R. Lau, R. Rosenfeld, and S. Roukos, "Trigger-based language models: a maximum entropy approach," in *proc. ICASSP-93*, 1993, vol. 2, pp. 45–58.

[4] R. Rosenfeld, *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*, Ph.D. thesis, Carnegie Mellon University, 1994.

[5] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[6] A. Ljolje, E. Bocchieri, M. Riley, B. Roark, M. Saraclar, and I. Shafran, "The AT&T 1xRT CTS System," in *Rich Transcription Workshop*, 2003.

[7] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 951–991, 2003.

[8] L. Shen and A. K. Joshi, "Ranking and reranking with perceptron," *Machine Learning*, vol. 60, pp. 73–96, 2005.

[9] B. Tasker, C. Guestrin, and D. Koller., "Max-margin markov networks," in *Neural Information Processing Systems Conference*, 2003.

[10] J.J. Godfrey and E. Holliman, "Switchboard-1 release 2," 1997.

[11] D. Graff, K. Walker, and D. Miller, "Switchboard cellular," 2001.

[12] A. Canavan, D. Graff, and G. Zipperlen, "Callhome american english speech," 1997.

[13] J.S. Garofolo, J. Fiscus, and A. Le, "2002 rich transcription broadcast news and conversational telephone speech," 2004.

[14] S. Strassel, C. Walker, and H. Lee, "Rt-03 mdetraining data speech," 2004.