SEMANTIC STRUCTURED LANGUAGE MODELS

Hakan Erdogan, Ruhi Sarikaya, Yuqing Gao and Michael Picheny

IBM TJ Watson Research Center PO Box 218 Yorktown Heights, NY 10598

{erdogan,sarikaya,yuqing,picheny}@us.ibm.com

ABSTRACT

In this study, we propose two novel semantic language modeling techniques for spoken dialog systems. These methods are called semantic concept based language modeling and semantic structured language modeling. In the concept based language modeling, we propose to use long span semantic units to model meaning sequences in spoken utterances. In the latter technique, we use statistical semantic parsers to extract information from a sentence. This information is then utilized in a maximum entropy based language model. The language models are trained and evaluated in the air travel reservation domain. We obtain improvement over a sophisticated class based N-gram language model both in terms of recognition accuracy and perplexity. Interpolation of the proposed techniques with the class-based N-gram LM provides additional improvement.

1. INTRODUCTION

Language modeling for speech recognition attempts to model the probability $P(W)$ of observing a word sequence W in natural speech. Until recently, simple N-gram language modeling has been the dominant method of modeling natural language for speech recognition.

It has been long argued that the N-gram language models are suboptimal, and one could do better by considering longer range dependencies between words. However, it has been surprisingly difficult to beat the performance of N-gram language models consistently. Recently, there has been studies to enable use of syntactic structure of the sentence for language modeling [1, 2, 3]. This method is named structured language modeling (SLM). In SLM, one uses a syntactic parser to find the most likely syntactic parse of a sentence. To enable long range dependencies in the language models, the phrasal head words that are exposed by the parser before the current word are used as additional history words. The non-terminal labels can also be used in the history [3]. Researchers achieved about 2-3% relative improvement in error rate across different tasks such as switchboard and wall street journal data using SLMs [4].

In this paper, we introduce two new types of methods for language modeling. The idea underlying our first algorithm is to define semantic concepts defined by phrasal CFGs to model the meaning sequence in sentences. This approach is similar to the technique defined in [5, 6]. We estimate the probability of the concept sequence using an N-gram concept model with words as fillers. The difference between our approach and the method in [5] is that, we obtain the word sequence probabilities using N-grams instead of relying on phrasal PCFG probabilities which might be error prone. Moreover, our concepts are designed to maximize the word coverage in a sentence while having as different ways of expressing the same meaning as possible.

The second technique proposed in this study is based on using statistical semantic parses of air travel domain sentences for language modeling. We explore ways to incorporate the rich information in the semantic parses to enhance the language model. We cannot use head words since it might not be clear what should be the head word in a semantic parse. We compute the joint probability of the sentence and the parse and use features such as parent label name, previous finished constituent label name as well as regular N-gram history features for both words and labels. We use maximum entropy (ME) modeling for combining these features in a single effective language model. We provide the details in section 4.

The rest of the paper is organized as follows. In section 2, we present our approach to semantic concept based language modeling. We briefly describe the semantic classer and parser in section 3. Section 4 presents maximum entropy modeling in general and gives details about our maximum entropy language models (MELMs) that use semantic classer information. The experimental results are presented in section 5.

2. CONCEPT BASED LANGUAGE MODELING

The class based language model built at IBM for Darpa Communicator (DC) task [7, 8] is very rich in terms of word classes. It has a total of 41 classes, with commonly used classes such as [city], [month], [dayofweek], and more. Some of these classes are designed to model the structure within a broad class such as [city], by defining detailed sub-classes [bigcity], [mediumcity], [smallcity], [citystate] and [cityairport]. Therefore, we focus on modeling long span information that complements the information in the classes.

An extension of CFGs is the concept language modeling [5] which defines a semantic unit as concept. In [5], each concept is written as a PCFG and compiled into a stochastic recursive transition network (SRTN). Our concepts are also represented by CFGs. In addition, each word can be a concept by itself (as a filler concept if needed). We write a finite state transducer that combines all concept CFGs and words to parse the text data to find the semantic concept parse that has the least number of concepts. The difference between [5] and what we are proposing here is that we are not interested in the probability of word sequence given the concept sequence, $P(W|C)$. The drawback of the approach in [5] is if the spoken utterance is not grammatical with respect to predefined concepts, or if the training data to estimate the within grammar probabilities is not enough, $P(W|C)$ may be unreliable. In our approach, we find the probability of the most likely concept sequence $P(C)$, and rely on regular N-gram to estimate $P(W|C)$ by ignoring the concept sequence.

The parameters of our concept model are trigram concept probabilities, $P(C_i|C_{i-1}, C_{i-2})$, where C_i is the ith concept obtained from the rule-based semantic parse of the sentence. We define a parse of a sentence to be a sequence of concepts which

generate the sequence of words in the sentence. To obtain the language model score, we interpolate the concept score and the word sequence score:

$$\text{LM score} = \lambda \log P_{\text{N-gram}}(W) + (1-\lambda) \log P_{\text{N-gram}}(C),$$

where $\lambda \in [0,1]$ is an interpolation parameter. Note that this score is not an accurate estimation of the probability of the sentence $P(W)$, but just a language model score to compare sentences.

First, we defined a concept as a set of sequence of words which has a unique semantic meaning. Next, we designed rules to capture the concepts from the training data. The rules are designed to maximize both the coverage of words in a sentence and the variety of ways of expressing a concept while keeping semantic coherence. The examples given below illustrate this point. The reason for this is that the class based language model we use subsumes some of the concepts as its classes. Therefore, the concept language model should complement not only the plain trigram but also a very rich class based trigram language model. In our systems there are 23 concepts. The training data is tokenized to obtain a concept sequence. The words that are not covered by concepts are also assumed as trivial concepts.

- [book_flight] please book me on [/book_flight] [numflt] flight twenty one [/numflt]
- [i_want_to_go] i would like to fly [/i_want_to_go] [city_from] from philadelphia [/city_from] [city_to] to dallas [/city_to]
- [request1] could you please list the [/request1] flights [city_from] from boston [/city_from] [city_to] to denver [/city_to] on [date] july twenty eighth [/date]

The trigram class-based language model is trained using 137K sentences and smoothed using deleted interpolation on a held out data of 18K sentences. Concept language model $P(C)$ is trained on a 100K subset of the training data and again smoothed using deleted interpolation on the same held out set. We evaluate the concept-based language models in Section 5.

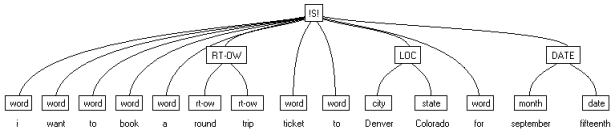

Figure 1: An example of a semantic classer output.

3. STATISTICAL SEMANTIC CLASSING AND PARSING

Semantic classing and parsing are used in IBM for natural language understanding for spoken dialog systems [9]. In semantic classing, we tag words according to their meaning (such as month names, numbers representing hours) instead of their part-of-speech. We have a single level of labels above the tags and no more additional levels are allowed. Thus, our semantic classer only groups together neighboring words that constitute a certain meaning or concept (such as a date expression or time expression). A tree representation of a semantic classer is shown in Figure 1. Semantic parser takes the output of the classer and derives more complex interactions between semantic constituents. An example is shown in Figure 2. The

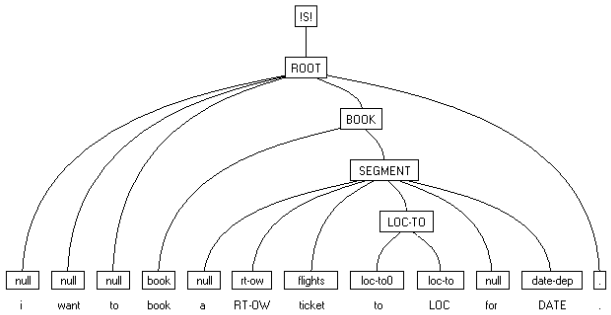

Figure 2: The parser output of the same sentence that is classed in Figure 1.

decision-tree based statistical classer and parser for air travel domain were developed at IBM [9] for the NLU module of the DARPA communicator project. The analysis process is separated into two parts for the ease of training and simplification. However, it is also possible to define a single level semantic parser that will perform the whole analysis in one step.

We used the statistical semantic classer output for our language modeling experiments. We plan to use semantic parser information in our future studies.

4. MAXIMUM ENTROPY LANGUAGE MODELING USING SEMANTIC FEATURES

Maximum entropy method is an effective method to combine multiple information sources (features) in statistical modeling. Maximum entropy model generates a probability model that matches the empirical feature probabilities exactly, but assumes no additional information about unseen events, effectively smoothing the probability distribution for them. For this work, we used maximum entropy modeling to incorporate lexical and semantic information sources for language modeling. ME models have been used in language modeling before, in the context of N-gram models, whole sentence models and syntactic structural language models [10]. Khudanpur and Wu [3] also added global apriori semantics (the topic) as another feature to include in the maximum entropy model. However, there has not been any study to use sentence-based higher level semantic features such as the information obtained through a semantic classer. Such semantic features could be very important especially in domain specific conversational dialog systems, such as conversational telephony , or domain dependent speech-to-speech translation applications.

We propose to use higher level semantic features in our air travel domain language model. We achieve this by using the semantic classer we introduced in Section 3.

We used the maximum entropy models to rescore N-best hypothesis, so the parser is not required to be left-to-right as opposed to the one used in [2]. In fact, our classers work on the full sentence and not left-to-right. However, especially for telephony applications, we believe it is feasible to perform an N-best rescoring step after the N-gram based decoding is done, since it costs much less to rescore N-best hypotheses as compared to the decoding time.

4.1. MAXIMUM ENTROPY MODELS

We can combine multiple features in a maximum entropy model in the following way:

$$P(o|h) = \frac{e^{\sum_i \lambda_i f_i(o,h)}}{\sum_{o'} e^{\sum_i \lambda_i f_i(o',h)}},$$

where o is the outcome (*e.g.*current word), h represents the history or context and f_i are feature indicator functions which are "activated" when a certain outcome $o_{j(i)}$ occurs in a certain context where $q_{k(i)}(h) = 1$. Here $q_{k(i)}$ is also an indicator function that is activated only when the context h has a certain property or in other words, when h is in an equivalence class $H_{k(i)}$. Mathematically:

$$f_i(o,h) = \begin{cases} 1 & \text{if } o = o_{j(i)} \text{ and } q_{k(i)}(h) = 1 \\ 0 & \text{otherwise} \end{cases}$$

For example, a bigram feature f_i representing the word sequence "IN THE" in maximum entropy modeling would have $o_{j(i)}$ = "THE" and $q_{k(i)}$ would be the question "Does the context h contain the word "IN" as the previous word of the current word ?".

4.2. JOINT ME PROBABILITY MODEL

We are interested in computing the probability of the word sequence $P(W)$ using the semantic features present in the parse C. A correct formulation would compute:

$$P(W) = \sum_{C'} P(W, C'),$$

where C' are all possible parses of the sentence W. To simplify the problem, we only use the most likely parse C,

$$P(W) \approx P(W, C).$$

To compute $P(W, C)$, one could think about decomposing and modeling $P(W|C)P(C)$ or $P(C|W)P(W)$. However, we built a direct joint ME model.

Consider the following equivalent textual representation of the semantic classer tree output shown in Figure 1:

- [!S! i_word want_word to_word book_word a_word [RT-OW round_rt-ow trip_rt-ow RT-OW] ticket_word to_word [LOC Denver_city Colorado_state LOC] for_word [DATE september_month fifteenth_date DATE] !S!]

We used the tokens in this representation for our joint model. Here "[LABEL" denotes begin label and "LABEL]" denotes end label and the tags are separated from the word by "_". We decided to ignore the tags, since our LM classes were very rich and tags had only limited information in addition to the LM classes. Labels, however had more longer range information, so we focused on labels and words. Our outcome vocabulary is $\mathcal{T} = \mathcal{W} \cup \mathcal{B}_l \cup \mathcal{E}_l$, where $\mathcal{W}$ is the word (or class) vocabulary and $\mathcal{B}_l$ and $\mathcal{E}_l$ are begin and end label vocabularies. So, we represent the joint probability

$$P(W, C) = \sum_{i=1}^{N} P(t_i|t_1, \ldots, t_{i-1}),$$

where $t_i \in \mathcal{T}$ are individual tokens in the textual representation presented above.

A regular token N-gram model can be built based on this representation. We built an N-gram type ME probability model on tokens t_i. We call this model MELM1. MELM1 uses the following question types for token t_i: (1) unigram, default question (2) Bigram, t_{i-1}=? (3) Trigram, (t_{i-2}, t_{i-1})=? (4) 4-gram, $(t_{i-3}, t_{i-2}, t_{i-1})$=?.

Furthermore, it is possible to use more intelligent features that will capture more longer range and high level information. Considering data sparsity and computation requirements, we came up with the following sublist of context question types for individual token probability computations:

- Default question (for unigram feature)
- previous word p_i (for bigram feature, skip label tokens)
- Two previous words, pp_i and p_i (for trigram feature, skip label tokens)
- Current active parent label for the token (L_i)
- L_i and N_i= the number of words to the left since starting the current constituent
- L_i, N_i and p_i
- O_i, M_i: the previos completed constituent and number of words to the left since completing O_i

We call this model MELM2. In this model, the context h_i can be represented by the six-tuple $(p_i, pp_i, L_i, N_i, O_i, M_i)$ introduced above and the LM probability can be computed by:

$$P(t_i|h_i) = e^{\sum_{j=1}^{N_f} \lambda_j f_j(t_i, h_i)}/Z,$$

where each f_j is associated with one outcome token and one question of the kinds listed above, and Z is the normalization term.

Note that, these are the questions we chose for the ME model. There might be many other possible features that utilize other information such as tags, grandparent labels *etc.* The choices could be dependent on the domain or the type of semantic parsing employed. The maximum entropy framework enables one to incorporate any type of features as long as they are computable.

We evaluated the ME LMs by rescoring N-best hypotheses and computing the perplexity on two different testsets. The results are presented in the next section.

5. EXPERIMENTS AND RESULTS

We performed experiments in the air travel domain. The acoustic models are trained using air travel and generic telephony data [8]. The language model training data consisted of about 137K sentences in air travel domain. A held out set of 18K sentences was used for smoothing. The class-based trigram and concept language models are trained using maximum likelihood N-gram training with deleted-interpolation smoothing. For maximum entropy training, all features extracted from training data are kept as model features. The ME models are trained using the improved iterative scaling algorithm with fuzzy ME smoothing [11] with a single variance parameter of 3.0. The heldout data set was used to determine the optimum value of this parameter.

We used two testsets to evaluate the new language models. Test1 has 1173 utterances from the calls received by IBM in DARPA communicator evaluation in June 2000. Test2 is a subset of calls received from 8 different sites during the same evaluation. This subset is chosen randomly among all utterances that were complete and meaningful (unchopped speech, grammatical). Test2 has 1458 sentences. Our motivation for making testset Test2 was to have a testset that was free of chopped speech and ungrammatical sentences.

Methods	Test1	Test2
word trigram	64.57	42.42
class trigram	45.25	26.55
MELM1	36.53	27.45
MELM2	35.86	26.82

Table 1: Perplexity values obtained for two testsets using word and class trigrams and new MELMs.

We have already worked very hard to improve the language model for DARPA communicator and achieved good performance with class-based and compound word language models [7]. These improvements are included in the baseline class trigram language model that we used. Table 1 shows the perplexity results for word and class trigram LMs and the new MELMs. Note that the concept based LM is not normalized the same way and it does not provide comparable perplexity results, so we only compare perplexities for word/class trigrams, MELM1 and MELM2. It should also be noted that the MELMs use the probability $P(W, C)$ which is bounded from above by $P(W)$, so the perplexity values are slightly overestimated. For Test1, MELM2 achieved 45% and 21% reductions in perplexity as compared to word and class trigram LMs respectively. Note that MELM1 and MELM2 had lower perplexity than the class trigram in Test1, but not in Test2, possibly due to the grammatical sentences that are well modeled by the class trigram.

To evalute the error rate performance of the new LMs, a lattice with low oracle error rate was generated by a Viterbi decoder using a class trigram language model. From the lattice, we generated at most 300 sentences for each utterance to form an N-best list. We rescored these utterances using the new proposed language models and our baseline word and class trigram language models. The results are presented in Table 2.

In Table 2, the first entry is the oracle error rate in the N-best list. The second and third rows show the error rate for the word and class trigram LMs [7, 8]. The error rate for word trigram LM is artificially low due to N-best list rescoring compared to direct decoding with the word trigram. The next entry is the concept based LM score which is an interpolation of the concept trigram and the class trigram. This approach reduces the error rate about 2-3% relative. The following entries are the error rates for maximum entropy language models and their sentence level interpolation with the class trigram LM. The results show that MELM1 can achieve the same error rate as the class trigram LM and when the two are interpolated, further reduction can be achieved. MELM2 performs better by itself, and slightly better after interpolation. Further improvement is observed in Test2 when all three LMs are interpolated, as shown in the last entry in the table. Overall, we can achieve about 3% relative reduction in error rate for both testsets after interpolating with the class trigram language model. These results are similar to the amount of improvement that was achieved with SLMs in other domains [2, 10].

Methods	Test1	Test2
N-best oracle	8.8%	4.1%
word trigram	18.9%	10.8%
class trigram	17.7%	9.9%
Concept+class trigram	17.3%	9.6%
MELM1	17.7%	10.0%
MELM2	17.4%	10.0%
MELM1+class trigram	17.3%	9.6%
MELM2+class trigram	17.2%	9.6%
MELM2+concept+class trigram	17.2%	9.5%

Table 2: Word error rates obtained using various language modeling methods.

6. CONCLUSION

We introduced two new language modeling techniques that use higher level long range semantic information to improve the language modeling for speech recognition. Both models use parsers that will group together neighboring words that constitute a semantic class (or concept). In concept based LM, the semantic information is used in an N-gram maximum likelihood model. In semantic structured language modeling, many sources of semantic information is compiled into a single efficient model using maximum entropy. The results indicate that, some modest gain could be achieved by using semantic information in language modeling. In the future, we plan to utilize the dialog state in our new language models. The semantic structured language model does not have to use ME modeling. Other modeling techniques, such as N-gram like maximum likelihood can be used as well.

7. ACKNOWLEDGEMENTS

The authors would like to thank Stanley F. Chen for his suggestions and valuable discussions about the maximum entropy language modeling. We also thank Adwait Ratnaparkhi for writing the original code base for maximum entropy training and testing algorithms.

REFERENCES

[1] Frederick Jelinek and Ciprian Chelba, "Putting language into language modeling," in *Eurospeech*, 1999.

[2] Ciprian Chelba and Frederick Jelinek, "Recognition performance of a structured language model," in *Eurospeech*, 1999.

[3] Sanjeev Khudanpur and Jun Wu, "Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling," *Computer Speech and Language*, pp. 355–72, Oct. 2000.

[4] Jun Wu and Sanjeev Khudanpur, "Syntactic heads in statistical language modeling," in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, 2000.

[5] Kadri Hacioglu and Wayne Ward, "A word graph interface for a flexible concept based speech understanding framework," in *Eurospeech*, 2001.

[6] Kadri Hacioglu and Wayne Ward, "Dialog context dependent language modeling combining n-grams and stochastic context-free grammars," in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, 2001.

[7] Hakan Erdogan, "Speech recognition for a travel reservation system," in *Intl. Conf. on Artificial Intelligence*, 2001.

[8] Yuqing Gao, Hakan Erdogan, Yongxin Li, Vaibhava Goel, and Michael Picheny, "Recent advances in speech recognition system for IBM Darpa communicator," in *Eurospeech*, 2001, pp. 503–6.

[9] Todd Ward, "How long until a high school student can build a language understanding system," in *ICSLP*, 2000.

[10] Jun Wu and Sanjeev Khudanpur, "Combining nonlocal syntactic and n-gram dependencies in language modeling," in *Eurospeech*, 1999.

[11] Stanley F Chen and Ronald Rosenfeld, "A survey of smoothing techniques for ME models," *IEEE Tr. Speech and Audio Proc.*, vol. 8, no. 1, pp. 37–50, Jan. 2000.