



Impact of Word Classing on Shrinkage-Based Language Models

Ruhi Sarikaya, Stanley F. Chen, Abhinav Sethy, Bhuvana Ramabhadran

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

{sarikaya, stanchen, asethy, bhuvana}@us.ibm.com

Abstract

This paper investigates the impact of word classing on a recently proposed shrinkage-based language model, Model M [5]. Model M, a class-based n -gram model, has been shown to significantly outperform word-based n -gram models on a variety of domains. In past work, word classes for Model M were induced automatically from unlabeled text using the algorithm of [2]. We take a closer look at the classing and attempt to find out whether improved classing would also translate to improved performance. In particular, we explore the use of manually-assigned classes, part-of-speech (POS) tags, and dialog state information, considering both hard classing and soft classing. In experiments with a conversational dialog system (human-machine dialog) and a speech-to-speech translation system (human-human dialog), we find that better classing can improve Model M performance by up to 3% absolute in word-error rate.

Index Terms: word classing, exponential models, Model M

1. Introduction

N -gram language models have been commonly used across a wide range of speech, natural language processing and machine translation applications. Even though n -gram models have serious short-comings, it has been difficult to outperform n -gram language models consistently across different domains, tasks and languages. It is well-known that n -gram language models are not effective in modeling long range lexical, syntactic and semantic dependencies. So far, with the exception of several approaches [3, 10], the improvements obtained by some more elaborate language models [9, 11] come from the explicit use of syntactic and semantic knowledge put into the annotated corpus.

Shrinkage based language models [4, 5, 6], particularly Model M have achieved some of the largest gains over word n -gram models reported in the literature on a variety of domains, by just using the lexical information - without using additional syntactic and semantic information - to the language modeling. Model M is a class-based n -gram model where word classes are induced automatically using the algorithm of [2]. While Model M, like word n -gram models, can be estimated from raw text, syntactic or semantic annotation of training data has been shown to improve performance in many contexts, e.g., [9, 11]. In this work, we examine whether we can use manual annotation to generate word classes that improve the performance of Model M.

We consider two speech recognition tasks where most or all of the vocabulary has been assigned a class label by a human annotator. Our goal is to compare unsupervised automatic word

classing to supervised word classing as well as hybrid schemes. We also consider using part-of-speech tags as word classes. In the original Model M work, *hard* word classing was used, i.e., each word is assigned to a single class. However, many words are syntactically or semantically ambiguous, so we discuss how Model M can be extended to support soft word classing.

The rest of the paper is organized as follows: Section 2 describes Model M and how to extend Model M for soft classing. Section 3 introduces the various word classing techniques we evaluate. Section 4 describes our experiments; and conclusions are presented in Section 5.

2. Model M

Shrinkage based exponential language models, namely Model M and regularized minimum discrimination information (rMDI) models were motivated as ways to *shrink* a word n -gram model. That is, when training and test data are drawn from the same distribution, it has been empirically found for many types of exponential language models that

$$\log \text{PP}_{\text{test}} \approx \log \text{PP}_{\text{train}} + \frac{\gamma}{D} \sum_i |\tilde{\lambda}_i| \quad (1)$$

where PP_{test} and PP_{train} denote test and training set perplexity; D is the number of words in the training data; $\tilde{\lambda}_i$ are *regularized* (i.e., smoothed) estimates of the model parameters; and γ is a constant independent of domain, training set size, and model type [4]. Thus, one can improve test performance if one can shrink the sum $\sum_i |\tilde{\lambda}_i|$ while maintaining training set performance.

Model M is a class-based n -gram model that can be viewed as the result of shrinking an exponential word n -gram model using word classes. If we assume each word w is mapped to a single class $c(w)$, we can write

$$p(w_1 \cdots w_l) = \prod_{j=1}^{l+1} p_{n_g}(c_j | c_{j-2} c_{j-1}, w_{j-2} w_{j-1}) \times \prod_{j=1}^l p_{n_g}(w_j | w_{j-2} w_{j-1} c_j)$$

where $p_{n_g}(y|\theta)$ denotes an exponential n -gram model, and where $p_{n_g}(y|\theta_1, \theta_2)$ denotes a model containing all features in $p_{n_g}(y|\theta_1)$ and $p_{n_g}(y|\theta_2)$.

2.1. Soft Classing

In previous work with Model M [4, 5, 6], only hard classing was used; i.e., each word belonged to a single class. In many applications such as POS tagging, it may be useful to allow words to

belong to multiple classes. To extend Model M to soft classing, we need to modify the right-hand side of 2 by summing over all possible class sequences $c_1 \cdots c_l$.

Evaluating Model M is more complex when using soft classing, because we must consider all possible class sequences for each word sequence. Ideally, one should sum the probabilities of all class labels when computing the likelihood of a word sequence, but in this work, we use the Viterbi approximation and consider only the most likely class sequence when calculating the likelihood of a word sequence. We can perform this computation using dynamic programming, and implement this using the IBM Infinite State Machine Toolkit [6]. By using the weighted automata framework, it is straightforward both to compute the likelihood of individual utterances as well as to do lattice rescoring. Model M training is basically unchanged, except that each word in the training data must be explicitly labelled with its class. (In hard classing, the class for each word is unambiguous.) Then, training component word prediction and class prediction models in Model M can be done as in previous work [4].

3. Classing for Model M

3.1. Automatic Classing

The automatic word classing algorithm used in the baseline Model M is based on the bigram mutual information between word classes [2]. The algorithm collects the bigram counts from the corpus and partitions the vocabulary into a specified number of classes to maximize the bigram mutual information between classes. The algorithm starts by assigning each word to a distinct class and computes the average mutual information between adjacent classes using a greedy algorithm. Pairs of classes with the least average loss in mutual information loss are merged. This process is repeated until the predetermined number of classes are reached. The final step of the algorithm cycles through the vocabulary moving each word to the class for which the resulting partition has the greatest mutual information.

3.2. Predefined Classes

Most of the spoken dialog systems have domain specific vocabulary, which naturally cluster into distinct classes. For example, in an air travel reservation task, all the cities, states, countries, airlines, flight numbers, days, months, etc. are separate classes. Similarly, in a financial transaction application, stocks, plan names, bonds etc., make up their own classes. Typically, the domain vocabulary contains words that are not observed in training data used to build the language model. However, the unobserved words can easily be substituted for words that share the same class. For example, in the air travel reservation task, we may not observe the city “Boise”, but we can easily replace “Denver” in a training utterance, “I want to book a ticket to Denver”, with “Boise”, to create a legitimate sentence in this domain.

3.3. POS Classing

Part-of-speech (POS) tags can be considered as natural syntactic classes for words. POS tags have been used in class-based language modeling in several studies [13, 14, 12, 15]. However, there was little, if any, improvement obtained over automatic classing methods. Nevertheless, POS tagging is an alternative word classing mechanism and it should be investigated with

application to Model M. We used POS tagging in both, hard and soft classing modes when building Model M LMs. In hard classing, a word with multiple POS tags in the training corpus is assigned the most frequent tag an SVM-tagger [7], which reports a tagging accuracy of around 97% using 35 tags on the WSJ corpus. We neither trained nor tuned the POS tagger on any of our data sets.

4. Experimental Task Description and Results

We evaluated the proposed word classing methods for Model M on two spoken dialog tasks. The first task is from the Air Travel Reservation domain and involves human-machine dialogs, while the second one is from a military/medical domain and has human-human dialogs from a speech-to-speech translation application. We studied the air travel reservation task in detail. Subsequently, we repeated a relevant subset of these experiments on the military/medical domain.

4.1. Air Travel Reservation Domain

Spoken dialog systems in the air travel reservation domain have been studied extensively during the past two decades. We consider the air travel reservation task primarily because large amounts of annotated data comprising of word classes that are tuned to this task and domain are available. We believe these predefined word classes can be considered as gold standards against which automatic classing methods should be compared. Language model rescoring experiments are conducted for the speech recognition task in this domain. The air travel domain has 39 predefined word classes that cover 20.6K of the 23.2K words in the vocabulary. The acoustic models were trained using air travel and conversational telephony data using the approach described in [11]. The language model training data consisted of about 137K sentences in the air travel domain and it does not cover all the words in the predefined classes, which is typical for such spoken dialog applications. The test set has 1173 utterances [11] from calls received by IBM in the official DARPA Communicator evaluation in June 2000. An annotated example from this task is given below. The semantic tags are shown within square brackets.

*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
SanAntonio_city Texas.state LOC] for_word [DATE septem-
ber_month twenty_third.date DATE].*

Words that are part of a semantic concept are assigned semantic tags, whereas the remaining words (e.g. “want”) are assigned the same generic tag (i.e. “word”). This information is also included in the traditional class based trigram language model that we present in this paper.

4.2. Military Domain

The second data set was collected for the initial phase of a speech-to-speech translation project in a military force-protection/medical domain where soldiers interact with civilians [7]. Unlike DARPA Communicator, this task does not have a clear dialog structure. The dialogs can vary over a diverse set of topics. For example, some dialogs cover the interaction between a soldier and a civilian during a vehicle search, while, other dialogs can involve requests to enter a

military base. If a civilian wants to enter a base, then a soldier can ask questions related to who the civilian wishes to see, or if he has any weapons in his possession. Part of the corpus also covers the medical domain where doctors and patients interact in a military setting. Thus, building class based language models is not straightforward compared to the air travel reservation task. The adopted annotation strategy uses a mixture of semantic and syntactic concepts. For example, in this example, some words are annotated using semantic concepts, while others are annotated using syntactic concepts.

[*WHQ what=WHQ time=WHQ WHQ*] [*AUX did=AUX AUX*] [*PRON you=PRON-SUB PRON*] [*VERB see=VERB VERB*] [*PERSON the=ART person=PERSON PERSON*]

Unlike the annotation of air travel reservation task, which focused on annotating the concept words, in the annotation of this data all the words in the training data are labeled. However, it was difficult to be consistent in the annotation of this data even with a single annotator. This domain contains a fairly small amount of data for training and test. The LM training data comprises of only 5882 sentences, of which 5382 sentences are used for training, and 500 form the heldout set. The vocabulary size is 1887. The test set consists of 718 sentences. The baseline LM for this task is a trigram model. Acoustic models trained for generic domain independent tasks were adapted to the military domain using the approach described in [11]. The speech data used for acoustic model adaptation was obtained from the same 5882 sentences used for language modeling.

4.3. Experimental Tasks and Results

In order to evaluate the impact of the various word classing methods on Model M performance, a lattice with low oracle error rate was generated by a Viterbi decoder using a trigram language model. These lattices were used to generate a 300-best list for subsequent rescoring experiments. The results for the Air Travel Reservation task are presented in Table 1. The first line in the table is the N-best oracle Word Error Rate (WER) of 8.8%, which provides a lower bound on WER. The baseline trigram language model trained with modified Knesser-Ney smoothing [8] obtained a WER of 19.9%. In the upper part of the table, the performances of Model M with various automatic classes are shown. This range of classes sample the typical number of classes used in numerous Model M training across various tasks [6]. Unlike what was observed in large vocabulary domain independent speech recognition tasks [6], the WER fluctuates quite a bit as a function of the chosen number of classes. The best performance with Model (19.5%) is obtained when the class size is set to 500. In the bottom half of the table a number of alternative word classing schemes are evaluated. All of the classing techniques, *Merge1* through *Merge5* demonstrate different ways of classing the remaining words that are not covered in the predefined classes. *Merge1* refers to classing with 39 predefined classes in combination with automatic classing. The automatic class tags are used for the words that are not covered (about 2.6K words) in the predefined classes. These words tend to be frequent words. All of the *Merge1* classing results are better than the corresponding automatic classing given in the upper part of the table. The lowest WER of (18.8%) is obtained when the uncovered words are assigned the automatic tags with 200 classes along with the predefined classes. *Merge2* refers to the classing where uncovered words are classed according to their POS tags, whereas in *Merge3* all the uncovered

Method	Air Travel
N-best oracle	8.8
Trigram	19.9
Model M (c=500)	19.5
Model M (c=300)	20.3
Model M (c=200)	19.8
Model M (c=100)	19.9
Model M (c=50)	19.8
Model M (c=40)	20.0
Model M (Merge1 c=500)	19.3
Model M (Merge1 c=300)	19.2
Model M (Merge1 c=200)	18.8
Model M (Merge1 c=100)	19.3
Model M (Merge1 c=50)	19.2
Model M (Merge1 c=40)	19.7
Model M (Merge2)	19.1
Model M (Merge3)	19.2
Model M (Merge4)	19.1
Model M (Merge5)	19.0
Model M (POS classes)	19.9
Model M (Soft POS classes)	19.4
Model M (Soft POS + NLU classes)	18.9
Model M (Soft POS + Pre-defined classes)	18.9
ClassLM	19.0

Table 1: Word error rates obtained using various language modeling methods for Air Travel Reservation Task.

words are mapped to distinct individual classes, creating an additional 2.6K classes. In the *Merge4* all the uncovered words are mapped to a single classes, resulting into 40 (39 predefined + 1) classes. Finally, *Merge5* uses semantic classer tags for the uncovered words (79 classes).

We also used the POS based classing for words but did not observe any improvements over the baseline trigram model. However, soft classing using POS tags, where each word can assume different tags depending on the context in which it is used, does show an improvement (19.4%). When using POS tags as word classes, soft tagging is a more natural choice than hard tagging where each word can assume only a single tag independent of where and how it is used. In hard classing with POS tags we assign a tag to each word based on the maximum observed counts in the training data. Combining POS Tags with semantic classes reduced the WER to 18.9%. We also experimented with combining dialog state information with POS classes by using a set of composite classes which combined utterance dialog state and word POS tags. This did not lead to any improvement over using just POS tags (19.4%).

In Table 2, we present the results for the military/medical domain task. The oracle WER of the N-best list is 9.6%. The baseline language model achieved a WER of 15.7%. The best Model M performance (14.8%) with automatic classing is obtained when 300 classes are used. Model M with predefined word classes reduced the WER to 13.9%. Interestingly, using POS tags as classes provided the best result (12.7%). We speculate that the POS tags may have provided a more consistent classing than manual classing. A traditional class based language model [2] with predefined classes, has a WER of 17.3%, which is 1.6% worse than the baseline word trigram model. We believe this result is another piece of evidence about the nature of this task where there is not enough data and structure to ob-

Method	Military/Medical
N-best oracle	9.6
Trigram	15.7
Model M (c=500)	15.6
Model M (c=300)	14.8
Model M (c=200)	16.0
Model M (c=100)	16.0
Model M (c=50)	15.7
Model M (c=40)	15.9
Model M (Predefined classes)	13.9
Model M (POS classes)	12.7
ClassLM	17.3

Table 2: Word error rates obtained using various language modeling methods for Military/Medical Domain.

tain a consistent manual classing of words.

5. Conclusions

We studied the effect of word classing on the Model M performance on two spoken dialog tasks; air travel reservation task (human-machine dialog) and speech-to-speech translation task (human-human dialog). While Model M has produced sizeable gains over word n -gram models on large training sets, we show here that gains are greatly reduced on smaller data sets when using automatic word clustering, most likely due to data sparsity. In these situations, we demonstrate that even partially human-generated classes can substantially improve the performance of Model M, achieving gains of at least 1% absolute WER over a word n -gram baseline on two different tasks. With part-of-speech tags, we achieve a gain of 3% absolute in our speech-to-speech translation task. While there has been much previous work with using part-of-speech tags as word classes, e.g., [13, 14, 12, 15], this is the largest WER gain of this type that we are aware of. In addition, we found that using soft classing can significantly help over hard classing, and lets us effectively combine semantic and syntactic tags within a single model.

6. References

[1] M. Karahan, D. Hakkani-Tur, G. Riccardi, G. Tur, "Combining Classifiers for Natural Language Understanding", *IEEE ASRU-2003*, US Virgin Islands, Dec. 2003.

[2] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai and R.L. Mercer, "Class-based N-gram Models of Natural Language", *Computational Linguistics*, 18(4), pp: 467-479, 1992.

[3] H. Schwenk, and J.L. Gauvain. "Using Continuous Space Language Models for Conversational Telephony Speech Recognition", *IEEE Workshop on Spontaneous Speech Processing and Recognition*, Tokyo, Japan, 2003.

[4] S. F. Chen, "Performance Prediction for Exponential Language Models", *Proc. of HLT/NAACL*, Boulder, CO, 2009.

[5] S. F. Chen, "Shrinking Exponential Language Models", *Proc. of HLT/NAACL*, Boulder, CO, 2009.

[6] S. F. Chen, L. Mangu, B. Ramabhadran, R. Sarikaya and A. Sethy, "Scaling Shrinkage Based Language Models", *Proc. of IEEE ASRU*, Merano, Italy, 2009.

[6] S. F. Chen, "Designing a Non-Finite-State Weighted Transducer Toolkit", *Tech. Report, IBM Research Division*, RC 24829, 2009.

[7] J. Gimenez and L. Marquez, "SVMTool: A General POS-tagger Generator Based on Support Vector Machines", *In Proc. LREC-04*, 2004.

[7] Y. Gao, L. Gu, B. Zhou, R. Sarikaya, H.-K. Kuo, A.-V.I. Rosti, M. Afify, W. Zhu, "IBM MASTOR: Multilingual Automatic Speech-to-Speech Translator", *Proc. of ICASSP*, Toulouse, France, 2006.

[8] S. Chen, J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling", *ACL*, Santa Cruz, CA, 1996.

[9] C. Chelba and F. Jelinek, "Structured language modeling", *Computer Speech and Language*, 14(4), 283-332, 2000.

[10] A. Emami and P. Xu and F. Jelinek, "Using a Connectionist Model in a Syntactical based Language Model", *Proc. of IEEE ICASSP*, v. 1, 372-375, Hong Kong, 2003.

[11] H. Erdogan, R. Sarikaya, S.F. Chen, Y. Gao and M. Picheny, "Using Semantic Analysis to Improve Speech Recognition Performance", *Computer Speech & Language Journal*, vol. 19(3), pp: 321-343, 2005.

[12] C. Samuelsson and W. Reichl, "A class-based language model for large-vocabulary speech recognition extracted from part-of-speech statistics", *In Proc. of ICASSP*, pp: 537-540, 1999.

[13] P. A. Heeman, "POS tags versus classes in language modeling", *In Proc. of Sixth Workshop on Very Large Corpora*, 1998.

[14] L. Galescu and E. Ringger, "Augmenting Words with Linguistic Information for N-Gram Language Models", *In Proc. of Eurospeech*, pp: 537-540, 1999.

[15] J. Cui, Y. Su, K. Hall, and F. Jelinek, "Investigating linguistic knowledge in a maximum entropy token-based language model", *In Proc. of ASRU*, pp: 171-176, 2007.