SEMANTIC CLUSTERING FOR ADAPTIVE LANGUAGE MODELING

Reinhard Kneser, Jochen Peters

Philips GmbH Forschungslaboratorien, Weisshausstr. 2, D-52066 Aachen, Germany
Email: {kneser|peters}@pfa.research.philips.com

ABSTRACT

In this paper we present efficient clustering algorithms for two novel class-based approaches to adaptive language modeling. In contrast to bigram and trigram class models, the proposed classes are related to the distribution and co-occurrence of words within complete text units and are thus mostly of a semantic nature. We introduce adaptation techniques such as the adaptive linear interpolation and an approximation to the minimum discriminant estimation and show how to use the automatically derived semantic structure in order to allow a fast adaptation to some special topic or style. In experiments performed on the Wall-Street-Journal corpus, intuitively convincing semantic classes were obtained. The resulting adaptive language models were significantly better than a standard cache model. Compared to a static model a reduction in perplexity of up to 31% could be achieved.

1. INTRODUCTION

The general task of a stochastic language model is to provide estimates for the conditional probability $P(w|h)$ of some word w given its history $h = w_1 \dots w_i$. N-gram models which consider histories ending in the same last few words to be equivalent describe the local structure of the language quite well but are not able to capture longer ranging dependences, such as e.g. style or topic of a text. The idea of adaptive language models is to compensate for this by introducing a small number of additional adaptive parameters which can be used to modify the model. A small amount of adaptation data which is assumed to be typical for the expected kind of language is then sufficient to optimize the adaptive parameters and thus adapt the total language model. This adaptation data might be the text immediately preceding the current word as in our experiments but also other sources such as e.g. the recognizer output from a first path over a total text might be used.

Several approaches to adaptive language modeling have been proposed including cache models[1], trigger models[2] and domain specific models[3]. Commonly used adaptive methods are based on maximum entropy[2][4] and on linear interpolation[3][5]. We propose to take advantage of the co-occurrence of words in a text unit by clustering semantically related words. In the following, two different class models are presented having either complete texts or single words as basic units. Both approaches try to model the variation within the training data assuming that the language model is homogeneous within one article but may change across articles. This makes it necessary to have the training data divided into a set of separate articles $\mathcal{A}$. In contrast to the mainly syntactic structure of clusters coming from bigram statistics[6] we now obtain classes of a more semantic nature. The proposed models can be directly used in an adaptive way once the semantic classes are found but also combinations with other adaptation techniques are possible.

2. SEMANTIC WORD CLASSES

A text often relates to several different topics. An article about *Volkswagen* for example may have aspects of both *cars* and *Germany*. In an attempt to model the varying proportions of different topics within an article we assume a set $\mathcal{S}$ of topics or *semantic classes* such that each word w in our vocabulary $\mathcal{V}$ belongs to exactly one semantic class $S(w) \in \mathcal{S}$. Furthermore we assume the model

$$P_{adap}(w) = P_{adap}(S(w)) P_{static}(w|S(w)) \qquad (1)$$

where the proportions of the topics within an article adaptively change while the probability of a word given its semantic class is taken to be fixed.

This work was partially funded by the German Federal Ministry for Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil Project under Grant 01 IV 102 B. The responsibility for the contents of this study lies with the author.

Copyright 1997 IEEE

Since semantic classes are usually not available, some way to find them automatically from training data is desirable. In [7] a clustering algorithm based on latent semantic analysis is presented. We instead propose to find semantic classes by maximizing the log-likelihood of the training data according to the class model of eq. (1). Taking the maximum-likelihood estimates for the probabilities we obtain the log-likelihood of the training data as

$$LL = \sum_{A \in \mathcal{A}} \sum_{w \in A} N(w,A) \log \frac{N(S(w),A)}{N(A)} \frac{N(w)}{N(S(w))} \quad (2)$$

where $N(.)$ stands for the occurrence count of the appropriate event within the training data.

The task of the clustering algorithm is to find that mapping $S(w)$ of words to semantic classes that maximizes LL. Note that the maximal number of semantic classes must be given beforehand since the likelihood increases with this number. Due to the huge number of possible mappings a complete search is not feasible but the following greedy search strategy analogous to that presented in [6] can be used to find at least a locally optimal solution:

```
Start with some initial classification $S(w)$
Iterate until some convergence criterion is met
    Loop over all words $w \in \mathcal{V}$
        Loop over all classes $S \in \mathcal{S}$
            Check how $LL$ changes if $w$ was moved from its current class to a different class $S$
        Move word $w$ to that class $S$ which results in the highest increase of $LL$
```

3. TEXT CLUSTERING

It can be observed that individual articles of a corpus usually belong to a certain domain or subdomain. In the case of the Wall-Street-Journal task for example, articles on politics, stock market, financial news etc. can be found. This observation suggests the use of separate domain specific language models. Since usually no domain information is given in the training data, text clustering algorithms are used to divide the training data automatically [5][8][9].

We assume that each article $A \in \mathcal{A}$ in the training data belongs to exactly one domain $D(A)$ of a certain set of domains $\mathcal{D}$. Applying these assumptions to a unigram model we get similar to eq. (2) the log-likelihood

$$LL = \sum_{A \in \mathcal{A}} \sum_{w \in A} N(w,A) \log \frac{N(w,D(A))}{N(D(A))}. \quad (3)$$

Here we search for that mapping $D(A)$ from articles to domains that maximizes the log-likelihood in eq. (3). A similar exchange algorithm to the one described in the previous section can be used to perform this search. More details on the algorithm and an information theoretic derivation of eq. (3) can be found in [9].

4. ADAPTIVE MODELING TECHNIQUES

In the following we present three basic techniques for adapting a language model according to a given small amount of adaptation data.

Relative Frequencies: As long as an event has been observed often enough in the adaptation data its probability can be estimated by maximum likelihood. The cache model[1] for example usually has a component where a unigram is estimated by relative frequencies within the adaptation data, which in this case consists of the cached history.

Adaptive Linear Interpolation: Consider the case where a number of different language models $P_i(w|h)$ is given, for example domain specific models. A convenient way to get a new model is by a linear combination

$$P(w|h) = \sum_i \lambda_i P_i(w|h) \quad (4)$$

with the interpolation parameters λ_i summing to 1. This interpolation can be made adaptive by taking λ_i as adaptive parameters and by choosing them to maximize the likelihood of the adaptation data[3]. The likelihood maximization can be achieved with a few iterations of the EM-algorithm.

Minimum Discriminant Estimation: The principle idea of adaptation using minimum discriminant estimation[4] is to take some adaptively estimated marginal distributions as constraints and to find a distribution respecting these constraints and being as near as possible to the background model in terms of the Kullback-Leibler distance. In the spirit of this idea we estimate an adaptive unigram $P_{adap}(w)$ and get an adaptive conditional probability by introducing a corrective factor $\alpha(w) = P_{adap}(w)/P_{static}(w)$. After renormalization we then obtain

$$P_{adap}(w|h) = \frac{\alpha(w) P_{static}(w|h)}{\sum_v \alpha(v) P_{static}(v|h)}. \quad (5)$$

This equation can also be interpreted as the first iteration of the Generalized-Iterative-Scaling algorithm[10] which is usually applied to find a solution to the minimum discriminant estimation problem.

Copyright 1997 IEEE

Table 1: Most frequent words of some sample clusters

Cluster 1: AUTO CARS CAR MAKERS FORD MOTOR CHRYSLER MODEL MICHIGAN MOTORS MODELS VEHICLES TRUCK ASSEMBLY TRUCKS VEHICLE DETROIT INCENTIVES AUTOMOTIVE LUXURY RECALL FORD'S TOYOTA HIGHWAY RENTAL HONDA NISSAN VOLKSWAGEN ...

Cluster 2: POLITICAL PARTY CAMPAIGN VOTE BUSH DEMOCRATIC LEADERS ELECTION LEADER GEORGE PRESIDENTIAL DEMOCRATS OPPOSITION DUKAKIS MAJORITY WIN GOVERNOR CONSERVATIVE REPUBLICAN DEBATE CANDIDATES CANDIDATE VOTERS POLITICS SPEECH ...

Cluster 3: DRUG TEST DOCTOR AIDS HUMAN DRUGS PATIENTS TREATMENT HEART TESTING CANCER TESTS BLOOD DOCTORS DISEASE RESEARCHERS SCIENTISTS SCIENCE SCIENTIFIC PHARMACEUTICAL PATENT VIRUS ...

Table 2: Unigram (Adaptive Linear Interpolation)

Components	MW1	MW5	MW38
Static Uni	983.5	968.5	959.3
+ Cache	726.4	719.4	715.4
+ Semantic Cache	673.5	650.0	640.8
+ Domain Models	672.6	641.3	628.8
+ Sem. C. + Dom. M.	651.0	619.9	607.6

Table 3: Trigram (Adaptive Linear Interpolation)

Components	MW1	MW5	MW38
Static Tri	219.3	147.3	95.4
+ Cache	166.7	118.8	81.9
+ Semantic Cache	165.4	117.9	81.7
+ Domain Models	160.1	111.1	74.2

5. EXPERIMENTAL RESULTS

Experiments were carried out on the Wall-Street-Journal corpus using the same conditions as in [2]. Training was performed on three training sets of different sizes comprising approximately 1 (MW1), 5 (MW5) and 38 million words (MW38), respectively. A separate set of about 300,000 words was used for testing. Both the training and the test data are partitioned into separate articles having an average size of about 450 words. In the experiments article boundaries were known and all words in the article preceding the word to predict were used as adaptation data. The official ARPA 20,000 word lexicon was used and all words outside this lexicon were mapped to a special unknown-word symbol.

For each of the three training sets we created 100 semantic word classes using the clustering algorithm described in Section 2. As can be seen in the examples listed in Table 1 the clusters are indeed of semantic nature. Words within most of the clusters are semantically related or belong to a certain topic and together with some word usually also its inflections are present in the same clusters (e.g. *test, tests, testing, tested* in Cluster 3). In addition to these topic classes a few special classes were created by the clustering algorithm containing mostly function words. Based on these semantic classes we built an adaptive unigram component by estimating $P_{adap}(S)$ in eq. (1) by relative frequencies of the adaptation data. We shall call this component a *semantic cache*.

We applied the text clustering algorithm of Section 3 and divided the training data into 10 text clusters, again separately for each of the three training sets. A manual analysis showed that the resulting text clusters were intuitively satisfying and that it was easy to characterize them by labels such as *financial news*, *prominent people*, *health care* etc. By restricting the training data to the respective text class, domain-specific trigram and unigram models were trained. In addition static trigram and unigram models were trained on the complete training data. Our best interpolation technique with marginal-constraint backing off distributions[11] was used for all these models. Finally we have a unigram word cache[1] and a bigram word cache[2] as further components.

In first experiments we calculated the perplexity of different adaptive unigram models (Table 2). All considered models were obtained as adaptive linear interpolations (ALI) of different unigram components. A large improvement of about 25% was already achieved by combining the static unigram with the unigram cache component. A further improvement could be observed by adding either the semantic cache or the 10 domain specific unigram models to these two components. The best perplexity values, being up to 37% smaller compared to the static unigram, were achieved by combining all the available 13 components.

In a second series of experiments we took trigram models instead of unigram models for the static background model and for the 10 domain specific models and added a bigram cache component whenever a cache was used. Only the semantic cache remained unchanged since due to its unigram nature it is not easily extended to a longer context. Similar to the unigram experiments we observe a large gain by just adding a cache component. Only minor improvements were

Copyright 1997 IEEE

Table 4: MDE-trigram interpolated with bigram cache

Marginal Unigram	MW1	MW5	MW38
Static Uni + Cache	161.5	114.2	78.5
+ Semantic Cache	156.4	109.6	75.8
+ Domain Models	155.6	108.0	74.2
+ Sem. C. + Dom. M.	153.5	106.7	73.6
Combined Model (see text)	151.9	103.9	69.5

achieved by further adding a semantic cache but the introduction of domain specific trigram models gave a perplexity reduction of up to 9% as compared to the cache model. Probably due to the better trained specific models this effect was largest for the 38 million word training corpus.

In a last series of experiments we applied the approximation to the minimum discriminant estimation (MDE) as given by eq. (5) to the trigram model using the adaptive unigrams from Table 2 as constraining marginal distributions. In addition the resulting adaptive trigram models were interpolated linearly with the bigram cache component. Comparing Tables 4 and 3 we see that the MDE technique performs better than ALI. Already the MDE cache model alone is on average 4% better than the standard cache model using ALI. Improvements in the adaptive marginal unigram model always carry over to the trigram model with the MDE method. Finally we combined the two presented methods by taking the best MDE trigram model (adapted according to the semantic cache and the domain models), the bigram cache and the 10 domain specific trigrams as components of an adaptive linear interpolation. This resulted in the lowest perplexity values, giving a total improvement of 27% - 31% as compared to the static model and 9% - 15% compared to the standard cache model. This model appears to be better than the best model reported in [2] (with perplexities MW1: 163, MW5: 108 and MW38: 71) although a direct comparison is problematic due to different treatment of unknown words in the perplexity calculation.

6. SUMMARY

In two different approaches words and texts were automatically clustered according to semantic aspects. The derived semantic structure was used to build adaptive models giving improvements of up to 31% in terms of perplexity when compared to a static trigram model and of up to 15% when compared to a standard cache model. An approximation of the minimum discriminant estimation using adaptive unigram models as constraints showed to be a good technique to create adaptive models of longer contexts and proved to be superior to the adaptive linear interpolation.

7. REFERENCES

[1] R. Kuhn, R. de Mori: "A Cache-Based Natural Language Model for Speech Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 570-583, June 1990.

[2] R. Rosenfeld: "Adaptive Statistical Language Modeling: A Maximum Entropy Approach", School of Computer Science, Carnegie Mellon University, Ph. D. Thesis, Pittsburgh, PA, CMU-CS-94-138, 1994.

[3] R. Kneser, V. Steinbiss: "On the Dynamic Adaptation of Stochastic Language Models", *Proc. ICASSP*, Minneapolis, USA, Vol II, pp. 585-589, April 1993.

[4] S. Della Pietra, V. Della Pietra, R. L. Mercer, S. Roukos: "Adaptive Language Modeling Using Minimum Discriminant Estimation", *Proc. ICASSP*, San Francisco, CA, pp. I633-I636, March 1992.

[5] R. Iyer: "Language Modeling with Sentence-Level Mixtures", Boston University, Master Thesis, 1994.

[6] R. Kneser, H. Ney: "Improved Clustering Techniques for Class-Based Statistical Language Modelling", *Proc. Eurospeech*, Berlin, Germany, pp. 973-976, Sept. 1993.

[7] J. R. Bellegarda, J. W. Butzberger, Y.-L. Chow, N. B. Coccaro, D. Naik: "A Novel Word Clustering Algorithm Based on Latent Semantic Analysis", *Proc. ICASSP*, Atlanta, GA, Vol I, pp. 172-175, May 1996.

[8] D. Carter: "Improving Language Models by Clustering Training Sentences", *Proc. ACL Conference on Applied Natural Language Processing*, Stuttgart, Germany, pp. 59-64, Oct. 1994.

[9] J. Peters: "Document Clustering for the Improvement of Language Models", *Proc. ITG-Fachtagung für Sprachkommunikation*, pp. 59-63, Sept. 1996.

[10] J. Darroch, D. Ratcliff: "Generalized Iterative Scaling for Log-Linear Models", *The Annals of Math. Statistics*, Vol 43, pp. 1470-1480, 1972.

[11] R. Kneser, H. Ney: "Improved Smoothing for M-gram Language Modeling", *Proc. ICASSP*, Detroit, MI, pp. 181-184, May 1995.

Copyright 1997 IEEE