BAYESIAN LEARNING OF N-GRAM STATISTICAL LANGUAGE MODELING

Shuanhu Bai and Haizhou Li

Institute for Infocomm Research, Republic of Singapore
{sbai,hli}@i2r.a-star.edu.sg

ABSTRACT

The *n*-gram language model adaptation is typically formulated using *deleted interpolation* under the maximum likelihood estimation framework. This paper proposes a Bayesian learning framework for *n*-gram statistical language model training and adaptation. By introducing a Dirichlet conjugate prior to the *n*-gram parameters, we formulate the *deleted interpolation* under maximum *a posterior* criterion with a Bayesian learning procedure. We study the Bayesian learning formulation for *n*-gram and continuous *n*-gram language models. The experiments on North American News Text corpus have validated the effectiveness of the proposed algorithms.

1. INTRODUCTION

A typical large vocabulary continuous speech recognition (LVCSR) system consists of two components. An acoustic component matches the input sound wave into words in a vocabulary. The second component, which incorporates a statistical language model (LM), estimates the probability of a word hypothesis given the word history. The most popular statistical LM is the *n*-gram model, which estimates the probability of each word depending on the *n*-1 words that precede it from a large training corpus.

Assuming ample training data, the *n*-gram language models are still far from optimal. Studies show that they are extremely sensitive to changes in the style, topic or genre. The mismatch between training and test domains can lead to drastic performance degradation [1]. Different approaches have been studied to improve robustness of LM. Among many others, class-based *n*-gram shares parameters within a word-class to alleviate the data sparseness problem [2]; LM adaptation aims at bridging the mismatch between the models and the test domain [3].

It is generally the case that there are much less domain specific data than general data. The key of LM adaptation is to make good use of the domain specific data to effectively bring the baseline model towards the test domain. One typical adaptation technique is called *deleted interpolation* which combines the flat, reliable general model (baseline model) with the sharp, volatile domain specific model. Other techniques attempt to inject human knowledge of language into the model. Rosenfeld [1] indicates that

> *"One of the perils of using human knowledge is that it is often overstated and sometimes wrong. Thus a better solution might be to encode such knowledge as a prior in a Bayesian updating scheme…"*

A typical *n*-gram LM is trained under maximum likelihood estimation (MLE) criterion. In this paper, we will study the Bayesian learning formulation for *n*-gram LM adaptation. By introducing a Dirichlet conjugate prior distribution to each of the *n*-gram parameters, we can formulate the *n*-gram learning under the maximum *a posterior* (MAP) criterion [3,4]. The LM adaptation becomes a natural extension of *n*-gram modeling process. Under the Bayesian learning framework, an *incremental adaptation* procedure is also proposed for dynamically updating of cache-based *n*-gram.

This paper is organized as follows. In Section 2, we formulate the traditional *n*-gram LM and its cache-based adaptation under the Bayesian learning; In Section 3, we extend the formulation towards the continuous *n*-gram LM; In Section 4, we report experiment results on a LDC database; Finally, we conclude in Section 5.

2. N-GRAM MODEL

An *n*-gram LM is usually used in the context of a Bayes classifier, where it can play the role of the prior in speech recognition. Given an acoustic signal O, the goal is to find the sentence W' that is most likely to have been spoken

$$W' = \arg\max_{W} p(W/O) = \arg\max_{W} p(O/W)p(W) \tag{1}$$

where *n*-gram LM $p(W)$ represents the prior of the language. Let $W' = \{w_1, w_2, ...w_T\}$ denote one of the possible word strings, each of the word is drawn from a vocabulary of I words $w_t \in \{1,...,I\}$. $p(W)$ can be written as

$$p(W) = \prod_{t=1}^{T} p(w_t \mid \{w_{t-n+1},...,w_{t-1}\}) = \prod_{i,h} p(i/h)^{C_{ih}} \tag{2}$$

where $p(i/h) = p(w_t = i/\{w_{t-n+1},...,w_{t-1}\} = h)$ and C_{ih} is the count of occurrence of the string $\{w_t = i, \{w_{t-n+1},...,w_{t-1}\} = h\}$.

The quality of a given *n*-gram LM Θ on a corpus D of size T is commonly assessed by the log-likelihood probability

$$LL(D \mid \Theta) = \frac{1}{T}\sum_{i,h} C_{ih} \log p(i/h) \tag{3}$$

which is described as an empirical estimate of the *cross-entropy* [4] of the true data distribution C_{ih} with regard to the model $H(D \mid \Theta) = -LL(D \mid \Theta)$. The performance of a LM is often reported in terms of perplexity [5]

$$PP(D \mid \Theta) = 2^{H(D \mid \Theta)} \tag{4}$$

The perplexity can be interpreted as the average branching factor of the language according to the model. The *cross-entropy* measures the match between two distributions. The perplexity is a function of both the model and the language. As a function of model, it measures how good the model matches the test data. As a function of the language, it estimates the entropy or complexity of that language.

1-4244-0469-X/06/$20.00 ©2006 IEEE

2.1. Model Smoothing and Adaptation

To facilitate the discussion, let's define some notions. We denote $p(w_t = i)$ as θ_i for unigram and $p(w_t = i, w_{t-1} = j)$ and $p(w_t = i \mid w_{t-1} = j)$ as θ_{ij}, and $\theta_{i|j}$ for bigram. As unigram θ_i is the marginal probability of bigram θ_{ij}, we have $\theta_i = \sum_j \theta_{ij}$, and $C_i = \sum_j C_{ij}$, the unigram count is the sum over all the respective bigram counts. During model training, Eq.(3) serves as the optimization criterion. For unigram and bigram, it can be rewritten in the form of a multinomial model of the parameters $\theta \in \Theta$.

$$p_{unigram}(D \mid \Theta_{unigram}) = \prod_{i=1}^{I} \theta_i^{C_i} \propto \sum_{i=1}^{I} C_i \log\theta_i \tag{5}$$

$$p_{bigram}(D \mid \Theta_{bigram}) = \prod_{i=1}^{I}\prod_{j=1}^{I} \theta_{i|j}^{C_{ij}} \propto \sum_{i=1}^{I}\sum_{j=1}^{I} C_{ij} \log\theta_{i|j} \tag{6}$$

In the formula, we have the so-called naïve assumption that *n*-gram are independent of each other. Given a corpus generated by the same model, the parameters Θ can be estimated with MLE, subject to the constraints of $\sum_i \theta_i = 1$ and $\sum_i \theta_{i|j} = 1$, as follows

$$\Theta_{ML} = \arg\max_{\Theta} p(D \mid \Theta) \tag{7}$$

$$\theta_i = C_i / \sum_i C_i \text{ and } \theta_{i|j} = C_{ij} / \sum_j C_{ij} \tag{8}$$

MLE assigns zero probability to unseen *n*-gram. To address data sparseness, higher order *n*-gram is usually "smoothed" by lower order estimate [5]. The smoothing technique combines knowledge sources at different levels to improve robustness in *n*-gram prediction. For example, a bigram is backed-off by unigram $\hat{\theta}_{i|j} = \lambda\theta_i + (1-\lambda)\theta_{i|j}$, or

$$\hat{\Theta} = \lambda\Theta_{unigram} + (1-\lambda)\Theta_{bigram} \tag{9}$$

In adaptive language modeling, we first build domain independent, static *n*-gram, then we adapt the static *n*-gram by interpolating the static model with a dynamic cache model that is derived from the current, domain specific topical documents [1].

$$\hat{\Theta} = \lambda\Theta_{static} + (1-\lambda)\Theta_{cache} \tag{10}$$

As given in Eq.(9) and Eq.(10), both backoff smoothing and cache LM are motivated by the idea to combine a flat and reliable model with a sharp and volatile model. The weighting parameters λ are typically optimized on held-out data using cross-validation procedure called *deleted interpolation*, which can be formulated under a maximum *a posteriori* (MAP) adaptation strategy [3], also called Bayesian learning.

2.2. Bayesian Learning

In Eq.(5) and Eq.(6), we note that the bigram can be decomposed into I independent unigram-equivalents. For simplicity, we hereafter use unigram formulation. In MAP, a practical candidate for the prior distribution of the unigram is the Dirichlet density $m_i \in M$ [3], also known as hyperparameter [6], over each of the parameter $\theta_i \in \Theta$.

$$p(\Theta) \propto \prod_{i=1}^{I} \theta_i^{\alpha m_i - 1} \tag{11}$$

where m_i is a normalized measure over the I component, subject to $\sum_{i=1}^{I} m_i = 1$, and α is a positive scalar. We have as many hyperparameters $m_i \in M$ as the parameters $\theta_i \in \Theta$. The set of hyperparameters also includes α . The probability of generating a text corpus is obtained by integrating over the parameter space:

$$p(D) = \int p(D \mid \Theta) p(\Theta) d\Theta \tag{12}$$

This integration can be easily written down in a closed form due to the conjugacy between Dirichlet as in Eq.(11) and multinomial distribution as in Eq.(6). Instead of finding Θ that maximizes $p(D \mid \Theta)$ with MLE, we maximize *a posterior* (MAP) probability as follows:

$$\begin{aligned} \Theta_{MAP} &= \arg\max_{\Theta} p(\Theta \mid D) = \arg\max_{\Theta} p(D \mid \Theta) p(\Theta) / p(D) \\ &= \arg\max_{\Theta} p(D \mid \Theta) p(\Theta) \end{aligned} \tag{13}$$

The MAP solution to Eq.(13) is different from the MLE one in that the former uses a distribution to model the uncertainty of the parameter Θ, while the latter gives a point estimation [5][6]. We rewrite Eq.(13) as Eq.(14) using Eq.(5) and Eq.(11).

$$\Theta_{MAP} = \arg\max_{\Theta} \prod_i \theta_i^{C_i + \alpha m_i - 1} \tag{14}$$

With the conjugacy between Dirichlet and multinomial distributions, Eq.(14) can be seen as a Dirichlet function of Θ given M , or a multinomial function of M given Θ. With given priors M , The MAP estimation is therefore similar to the MLE problem which is to find the mode of the kernel density in Eq.(14).

$$\theta_i = \lambda m_i + (1-\lambda) f_i \tag{15}$$

where $f_i = C_i / C$, $\lambda = \alpha / (C + \alpha)$ and $C = \sum_i C_i$. Instead of fixing λ as a constant through *deleted interpolation*, λ is here estimated through an estimate with prior knowledge and the given statistics C . α serves as a weighting factor between the prior and the current observations. In different modeling scenarios, there could be different prior knowledge. For example, in the case of bigram smoothing, the prior knowledge is motivated to model the unigram as in Eq.(9); in the case of cache-based LM, the static LM is taken as the prior as in Eq.(10). Eq.(15) offers a Bayesian learning solution to the problem of *deleted interpolation*.

2.3. QB Estimation for Incremental Learning

The Bayesian learning procedure in Section 2.2 is another interpretation of the MAP strategy in [3]. The formulation of Eq.(15) combines two knowledge sources using one as the prior and the other as the current observation, assuming that the prior is known and static while the current observation is available all at once. However, this is not the case in on-line application where the observation comes in sequence. The idea of cache adaptation is to benefit from the continuously developing history to update the static model towards the intended topic, or even evolving topics. In an on-line system, it is of practical use to devise such an incremental learning mechanism that adapts both parameters and the prior knowledge over time. The *quasi-Bayes* (QB) method offers a solution to it [4,7]. In general, suppose that we have a sequence of sub-corpus $D^n = \{D_1, D_2 ..., D_n\}$, The QB method approximates the posterior density $p(\Theta \mid D^{n-1})$ by the closest tractable prior density $p(\Theta \mid M^{(n-1)})$ with $M^{(n-1)}$ evolved from historical corpus D^{n-1}.

$$\begin{aligned} \Theta_{QB}^{(n)} &= \underset{\Theta}{\arg\max}\, p(\Theta \mid D^n) \approx \underset{\Theta}{\arg\max}\, p(D_n \mid \Theta) p(\Theta \mid D^{n-1}) \\ &= \underset{\Theta}{\arg\max} \prod_{i=1}^{I} \theta_i^{C_i + \alpha m_i^{(n-1)} - 1} \end{aligned} \qquad (16)$$

QB estimation offers a recursive learning mechanism, starting with a hyperparameter set $M^{(0)}$ and a cache sub-corpus D_1, we estimate $M^{(1)}$ and $\Theta_{QB}^{(1)}$, then $M^{(2)}$ and $\Theta_{QB}^{(2)}$ and so on until $M^{(n)}$ and $\Theta_{QB}^{(n)}$ as observations arrive in sequence. The updating of parameters can be iterated between the reproducible prior and posterior estimates as in Eq.(17) and Eq.(18), called *Algorithm 1.*

i) Reproduce prior parameters $M^{(n-1)} \rightarrow M^{(n)}$:

$$\hat{m}_i^{(n)} = m_i^{(n-1)} + C_i^{(n)} / \alpha \qquad (17)$$

ii) Re-estimate parameters as in Eq.(8) $M^{(n)} \rightarrow \Theta_{QB}^{(n)}$:

$$\theta_i^{(n)} = \hat{m}_i^{(n)} / \sum_i m_i^{(n)} \qquad (18)$$

The scalar factor α can be seen as a forgetting parameter. When α is big, the updating of hyperparameters favors the prior. Otherwise, when α small, current observation is given higher attention.

3. CONTINUOUS N-GRAM MODEL

Consider the *n*-gram LM as discrete model at integer order *n*, the continuous *n*-gram model, also called *aggregate Markov model* [8], are intermediate between different order of *n*-gram in terms of size and accuracy. With continuous *n*-gram, we introduce *Z* hidden variables as the "soft" word classes. An *n*-gram probability is given as a mixture of probability $p(i/h) = \sum_{z=1}^{Z} p(i/z) p(z/h)$ where h represents the history of context, $p(z/h)$ denotes the probability that history *h* is mapped to class *z*, $p(i/z)$ denotes the probability that words in class *z* are followed by the word *i*. For bigram, we have $p(i/h) = p(i/j)$. Note that when $Z = 1$, a continuous bigram is reduced to unigram, when $Z = I$, the continuous bigram becomes a full bigram. By adjusting *Z*, continuous bigram is scalable between unigram and bigram. For simplicity, let $p_{i|z}$ denote $p(i/z)$, $p_{z|j}$ denote $p(z/j)$ hereafter.

The continuous *n*-gram model has two obviously advantages over the discrete n-gram: i) Assuming vocabulary size *I*, it reduces the parameter space from potentially $I \times I$ to $I \times Z \times 2$; ii) With the hidden variables representing the word classes, one is able to apply EM algorithm to estimate the parameters and to find word clusters at the same time under the MLE criterion. In this section, we will introduce the Bayesian learning to the continuous *n*-gram training. Let's rewrite Eq.(6) using soft word classes,

$$\log p(D \mid \Theta) = \sum_{i=1}^{I} \sum_{j=1}^{I} C_{ij} \log \sum_{z=1}^{Z} p_{i|z} p_{z|j} \qquad (19)$$

The parameter set $\Theta = \{p_{i|z}, p_{z|j}\}$ together with the word classes *z* can be estimated by the EM algorithm. In the E-step, the *posterior* probability $p_{z|ij} = p(z/i, j)$ is estimated given the current parameters $\Theta = \{p_{i|z}, p_{z|j}\}$

$$p_{z|ij} = p_{i|z} p_{z|j} / \sum_z p_{i|z} p_{z|j} \qquad (20)$$

In the M-step, we maximize $Q(\hat{\Theta} \mid \Theta)$ [4] with respect to $\hat{\Theta}$ and derive the new ML estimate by

$$\hat{p}_{i|z} = \sum_{j=1}^{I} C_{ij} p_{z|ij} / \sum_{i'=1}^{I} \sum_{j=1}^{I} C_{i'j} p_{z|i'j} \qquad (21)$$

$$\hat{p}_{z|j} = \sum_{i=1}^{I} C_{ij} p_{z|ij} / \sum_{z=1}^{Z} \sum_{i=1}^{I} C_{ij} p_{z|ij} \qquad (22)$$

3.1. Bayesian Learning

Similar to the formulation in Section 2.2, we introduce a Dirichlet conjugate prior as the hyperparameters $\Omega = \{\alpha, \eta_{iz}, \mu_{zj}\}$ for the parameters set $\Theta = \{p_{i|z}, p_{z|j}\}$, with $\sum_{i=1}^{I} \eta_{iz} = \sum_{z=1}^{Z} \mu_{zj} = 1$. Assuming the variables $p_{i|z}$ and $p_{z|j}$ are independent, similar to Eq.(11) and Eq.(14), we have the prior density function as

$$p(\Theta) \propto \prod_{z=1}^{Z} \left[\prod_{i=1}^{I} p_{i|z}^{\alpha \eta_{iz} - 1} \prod_{j=1}^{I} p_{z|j}^{\alpha \mu_{jz} - 1} \right] \qquad (23)$$

We apply the EM algorithm to iteratively calculate the *posterior* expectation function $R(\hat{\Theta} \mid \Theta)$ (E-step) as in Eq.(20), and maximize it with respect to $\hat{\Theta}$ so as to find new MAP estimate (M-step), which can be given in a closed-form solution.

$$\hat{p}_{i|z} = \sum_{j=1}^{I} C_{ij} p_{z|ij} + \alpha \eta_{iz} / \sum_{i'=1}^{I} [\sum_{j=1}^{I} C_{i'j} p_{z|i'j} + \alpha \eta_{i'z}] \qquad (24)$$

$$\hat{p}_{z|j} = \sum_{i=1}^{I} C_{ij} p_{z|ij} + \alpha \mu_{zj} / (C_j + \alpha) \qquad (25)$$

Given the priors $\Omega = \{\alpha, \eta_{iz}, \mu_{zj}\}$, Eq.(24) and Eq.(25) are interpreted as a smoothing between the known priors and the current observations, or cache corpus.

3.2. QB Estimation for Incremental Learning

Following the *quasi-Bayes* assumption as in Eq.(16), a reproducible prior/posterior pair for QB estimation of continuous *n*-gram can be formulated as follows, called *Algorithm 2*.

i) Reproduce prior parameters $\Omega^{(n-1)} \rightarrow \Omega^{(n)}$:

$$\eta_{iz}^{(n)} = \eta_{iz}^{(n-1)} + \sum_{j=1}^{I} C_{ij} p_{z|ij}^{(n)} / \alpha \qquad (26)$$

$$\mu_{zj}^{(n)} = \mu_{zj}^{(n-1)} + \sum_{i=1}^{I} C_{ij} p_{z|ij}^{(n)} / \alpha \qquad (27)$$

ii) Re-estimate parameters $\Omega^{(n)} \rightarrow \Theta_{QB}^{(n)}$:

$$\hat{p}_{i|z} = \eta_{iz}^{(n)} / \sum_{z=1}^{Z} \eta_{iz}^{(n)} \text{ and } \hat{p}_{z|j} = \mu_{zj}^{(n)} / \sum_{j=1}^{I} \mu_{zj}^{(n)} \qquad (28)$$

where η_{iz} and μ_{zj} as the expected counts of the respective parameters, in analogy to the integer counts in Eq.(8).

4. EXPERIMENTS

The objectives of the experiments are to evaluate the performance of proposed methods using a publicly available corpus. We design four datasets from North American News Text Supplement corpus (LDC98T30). Corpus A (CA) contains text of 60 million words extracted from LDC98T30 corpus of finance and business topics. Corpus B (CB) contains text of 20 million words in the domains of sports and fashion. Corpus B is divided into 5 adaptation blocks for incremental training purpose. Corpus C (CC) is a combination of Corpus A and B. Corpus D (CD) contains 20 million words in the same domains as that of CA and CB. CD serves as the open test dataset. We construct a vocabulary for the top 50K high frequency words derived from CA and CB, adding three tokens representing the sentence begin, the sentence end and the unknown words.

4.1. Batch Adaptation

The proposed Bayesian framework applies to both LM smoothing and cache-based LM adaptation, without loss of generality, we only report experiments of LM adaptation here in Table 1. We first train the bigram LM. We also train a continuous bigram LM with 32 latent variables, amounting to a model of $3,200,000 = 32 \times 50,000 \times 2$ free parameters $\Theta = \{p_{i|z}, p_{z|j}\}$. Both baseline LMs are trained on CA (Case B1). We also adapt the LMs using CC in Case B2. As CC includes domain text in CD, perplexity in open test of Case B2 is improved over Case B1 as expected. In Case B3, we examine how Bayesian learning helps accelerate the domain adaptation using α as a *forgetting factor*.

It is found that Case B3 adapts the model faster towards intended domains with improved open test perplexity over the MLE. The same effects are observed in both bigram and continuous bigram. Reducing α accelerates further the model adaptation. The continuous bigram reduces the number of free parameters over bigram at a cost of higher perplexity.

We have studied different number of latent variables 16, 32, 64 and 128. As latent variable increases, the number of free parameters increases and lower perplexity is obtained.

	Case B1: MLE on CA	Case B2: MLE on CC	Case B3: Case B1 + Bayesian on CB ($\alpha = 0.8$)
bigram	148/189	153/167	151/166
continuous bigram	256/337	267/329	271/306

Table 1 Perplexity of bigram LM (close/open test) in 3 cases

4.2. Incremental Adaptation

We report incremental adaptation in Table 2. Note that the choice of initial hyperparameter has impact on the adaptation path. The initial hyperparameters can be estimated from training data in an empirical Bayes manner. We follow the formulation introduced by Huo [7] and Chien [4]. In bigram LM adaptation, we choose the initial hyperparameter $m_i^{(0)}$ to be the bigram count resulting from Case B1. Then we proceed with *Algorithm 1*. In continuous bigram adaptation, we choose the initial hyperparameter $\{\eta_{iz}^{(0)}, \mu_{zj}^{(0)}\}$ to be the expected counts resulting from Case B1. Then we proceed with *Algorithm 2*:

$$\eta_{iz}^{(0)} = 1 + \sum_{j=1}^{I} p_{z|ij} \text{ and } \mu_{zj}^{(0)} = 1 + \sum_{i=1}^{I} p_{z|ij} \qquad (29)$$

In addition, we train a class-based bigram model following the formulation introduced by Brown et al. in [2] with 1,800 classes that comes with $3,290,000 = 1800 \times 1800 + 50,000$ free parameters, which are comparable to those for continuous bigram.

Comparing continuous bigram and class-based bigram, we find that continuous bigram generally outperforms the class-based bigram in open tests. This can be credited to the soft-clustering strategy that continuous *n*-gram adopts as opposed to the hard clustering decision in class-based *n*-gram.

Comparing Case I and Case B2, we find that incremental adaptation can be more effective than batch adaptation with the appropriate parameter (α) settings.

We have studied the case of bigram. Note that typically, it is easier to report improvements on bigram models than trigram models. We will extend the proposed framework to study the trigram behavior in the future.

	Case B1: MLE on CA	Case I: Case B1+ Bayesian on CB ($\alpha = 0.8$)
bigram	148/189	151/173
continuous bigram	256/337	263/294
class-based bigram	225/351	261/317

Table 2. Perplexity of bigram LM (close/open test) in 3 cases

5. CONCLUSION

We have proposed a Bayesian learning approach to *n*-gram language modeling. The learning approach offers several interesting properties for language modeling: 1) an interpretation for the smoothing or adaptation of language model as a weighting between prior knowledge and current observations. 2) The Dirichlet conjugate prior not only leads to a *batch adaptation* procedure but also a *quasi-Bayes* incremental learning strategy for on-line language modeling. The Bayesian learning framework has shown to be effective in both *n*-gram and continuous *n*-gram LM adaptation.

6. REFERENCES

[1] R. Rosenfeld, “A Maximum Entropy Approach to Adaptive Statistical Language Modeling”, Computer Speech and Language, 10:187-228, 1996.

[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): 467-479, Dec 1992

[3] M. Bacchiani, B. Roark, “Unsupervised Language Model Adaptation”, ICASSP 2003

[4] J.-T. Chien, M.-S. Wu, and C.-S. Wu, “Bayesian Learning for Latent Semantic Analysis”, Interspeech, Lisbon, Sept 4-8, 2005

[5] F.Jelinek, “Self-organized language modeling for speech recognition”, Readings in speech recognition, Morgan Kaufmann, 1990

[6] D.J.C., MacKay and L. Peto “A Hierarchical Dirichlet Language Model”, Natural Language Engineering, Vol 1, No 3, pp1-19, Cambridge University Press, 1994.

[7] Q. Huo, C. Lee “On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate”, IEEE Trans. on SAP, 5(2), pp.161-172, 1997

[8] L. Saul, F. Pereira, “Aggregate and mixed-order Markov models for statistical language Processing”, EMNLP 1997