Statistical language model adaptation: review and perspectives

Jerome R. Bellegarda
Spoken Language Group, Apple Computer

Speech Communication 2004

Adaptation framework

- The general SLM adaptation framework:

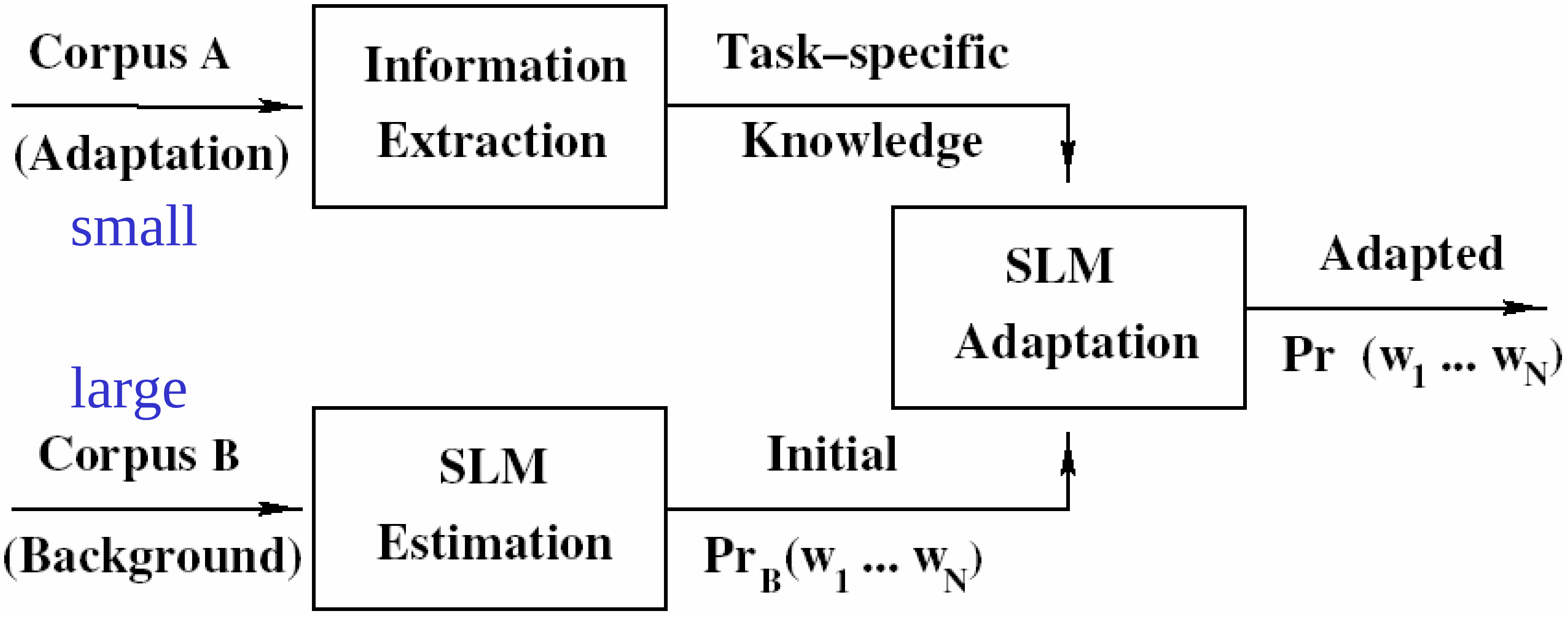


Fig. 1. General framework for SLM adaptation.

Adaptation problem

- Given a sequence of N words, the language model probability is

$$\Pr(w_1,\ldots,w_N)=\prod_{q=1}^{N}\Pr(w_q\mid h_q)$$

h_q represents the history

- For an n-gram model, the Markovian assumption implies

$$h_q=w_{q-n+1},\ldots,w_{q-1}$$

Adaptation problem

- The estimation of $\Pr(w_1,\ldots,w_N)$ leverages two distinct knowledge sources:
  - (i) the well-trained, but possibly mismatched, background SLM, which yields an initial estimate $\Pr_B(w_1,\ldots,w_N)$
  - (ii) the adaptation data, which is used to extract some specific information relevant to the current task

Adaptation problem

- The general idea of language model adaptation is to dynamically modify the background SLM estimate on the basis of what can be extracted from *A*

Adaptation approach

- Model interpolation
- Constraint specification
- Topic information
- Semantic knowledge
- Syntactic infrastructure
- Multiple sources

Model interpolation

- In interpolation-based approaches, the corpus *A* is used to derive a task-specific (*dynamic*) SLM, which is then combined with the background (*static*) SLM

Model merging

- Linear interpolation:

$$\Pr(w_q \mid h_q) = (1-\lambda)\Pr_A(w_q \mid h_q) + \lambda \Pr_B(w_q \mid h_q)$$

where $0 \le \lambda \le 1$ serves as the interpolation coefficient

- Back-off model (fill-up technique):

$$\Pr(w_q \mid h_q) = \begin{cases} \Pr_A(w_q \mid h_q) & \text{if } C_A(h_q w_q) \ge T \\ \beta \Pr_B(w_q \mid h_q) & \text{otherwise} \end{cases}$$

where T is an empirical threshold, and the back - off coefficient β is calculated to ensure that $\Pr(w_q \mid h_q)$ is a true probability

Dynamic Cache models

- A special case of linear interpolation, widely used for within-domain adaptation
- Cache models exploit self-triggering words inside the corpus *A* to capture short-term (dynamic) shifts in word-use frequencies which cannot be captured by the background model
- In other words, they correspond to the unigram case (n=1) of the general model merging strategy just discussed

Dynamic Cache models

- Propagate the power to higher order cases (class):

$$\Pr(w_q \mid h_q) = \sum_{\{c_q\}} \Pr(w_q \mid c_q) \Pr(c_q \mid h_q)$$

where $\{c_q\}$ is a set of possible classes for word w_q, given the current history h_q

The language model probability thus comprises a class *n*-gram component –$\Pr(c_q|h_q)$ – and a class assignment component –$\Pr(w_q|h_q)$

Dynamic Cache models

- The class *n*-gram component is assumed to be task independent, and is therefore taken from the background SLM

$$\Pr(c_q \mid h_q) = \Pr_B(c_q \mid h_q)$$

- The class assignment component is subject to dynamic cache adaptation

$$\Pr(w_q \mid c_q) = (1-\lambda)\Pr_A(w_q \mid c_q) + \lambda \Pr_B(w_q \mid c_q)$$

where λ is estimated in the same manner as before

MAP adaptation

- More recently, it has been argued that the combination should be done at the frequency count level rather than the model level
  - Count merging
- In this approach, the MAP-optimal model M^* is computed as

$$M^* = \arg\max_{M} \Pr(A \mid M)\Pr(M)$$

where $\Pr(M)$ is a prior distribution over all models in a particular family of interest

MAP adaptation

- Both count merging and model interpolation can both be viewed as a *maximum a posteriori* (MAP) adaptation strategy with a different parameterization of the prior distribution
- The model parameters θ are assumed to be a random vector in the space Θ, and x is a given observation sample
- The MAP estimate is the posterior distribution of θ

$$\theta_{\mathrm{MAP}} = \arg\max_{\theta} \boxed{g(\theta \mid \mathrm{x})} = \arg\max_{\theta} \underbrace{f(\mathrm{x} \mid \theta)}_{\text{multinomial}} \underbrace{g(\theta)}_{\text{Dirichlet}}$$

MAP adaptation

- The case of LM adaptation is very similar to MAP estimation of the mixture weights of a mixture distribution
- The prior distribution of the weights $\omega_1, \omega_2, \ldots, \omega_K$ is Dirichlet density

$$g(\omega_1, \omega_2, \ldots, \omega_K \mid v_1, v_2, \ldots, v_K) \propto \prod_{i=1}^{K} \omega_i^{v_i - 1}$$

where $v_i > 0$ are the parameters of the Dirichlet distribution

MAP adaptation

- c_i : expected counts for the *i*-th component

$$f(\mathrm{x} \mid \theta) = f(x_1,\ldots,x_T \mid \omega_1,\ldots,\omega_K) \propto \prod_{i=1}^{K} \omega_i^{c_i}$$

$$\therefore f(\mathrm{x} \mid \theta) g(\theta)$$

$$= f(x_1,\ldots,x_T \mid \omega_1,\ldots,\omega_K) \cdot g(\omega_1,\ldots,\omega_K \mid v_1,\ldots,v_K)$$

$$= \prod_{i=1}^{K} \omega_i^{v_i - 1 + c_i}$$

取log, ,並帶入Largrange multiplier :

$$\sum_{i=1}^{K} \log \omega_i^{v_i - 1 + c_i} = \sum_{i=1}^{K} (v_i - 1 + c_i) \log \omega_i + l(\sum_{i=1}^{K} \omega_i - 1)$$

MAP adaptation

對ω_i微分,微分=0有極值：

$$(v_i - 1 + c_i)\frac{1}{\omega_i} + l = 0 \quad \Rightarrow \quad l = -\frac{v_i - 1 + c_i}{\omega_i} \;\ldots\ldots(1) \;,\omega_i = -\frac{v_i - 1 + c_i}{l}$$

$$\sum_{i=1}^{K}\omega_i = -\sum_{i=1}^{K}\frac{v_i - 1 + c_i}{l} = 1 \quad \therefore \; l = -\sum_{i=1}^{K}(v_i - 1 + c_i) \quad \text{代入}(1)$$

得 $$\omega_i = \frac{v_i - 1 + c_i}{\sum_{k=1}^{K}(v_k - 1 + c_k)}$$

MAP adaptation
count mixing

- Mixing parameters α and β

$$v_i = \tilde{c}(h)\frac{\alpha}{\beta}\tilde{P}(w_i \mid h) + 1$$

$$\hat{P}(w_i \mid h) = \frac{\tilde{c}(h)\frac{\alpha}{\beta}\tilde{P}(w_i \mid h) + \overline{c}_d(hw_i)}{\sum_{k=1}^{K}\left[\tilde{c}(h)\frac{\alpha}{\beta}\tilde{P}(w_k \mid h)\right] + \overline{c}(h)}$$

$$= \frac{\alpha\tilde{c}_d(hw_i) + \beta\overline{c}_d(hw_i)}{\alpha\tilde{c}(h) + \beta\overline{c}(h)}$$

MAP adaptation
model interpolation

$$v_i = \overline{c}(h)\frac{\lambda}{1-\lambda}\widetilde{P}(w_i \mid h) + 1$$

$$\hat{P}(w_i \mid h) = \frac{\overline{c}(h)\frac{\lambda}{1-\lambda}\widetilde{P}(w_i \mid h) + \overline{c}_d(hw_i)}{\sum_{k=1}^{K}\left[\overline{c}(h)\frac{\lambda}{1-\lambda}\widetilde{P}(w_k \mid h)\right] + \overline{c}(h)}$$

$$= \frac{\frac{\lambda}{1-\lambda}\widetilde{P}(w_i \mid h) + \overline{P}(w_i \mid h)}{\frac{\lambda}{1-\lambda} + 1}$$

$$= \lambda\widetilde{P}(w_i \mid h) + (1-\lambda)\overline{P}(w_i \mid h)$$

Adaptation approach

- Model interpolation
- Constraint specification
- Topic information
- Semantic knowledge
- Syntactic infrastructure
- Multiple sources

Constraint specification

- In approaches based on constraint specification, the corpus A is used to extract features that the adapted SLM is constrained to satisfy

Exponential models

- Historically, constraint-based methods have been associated with exponential models trained using the maximum entropy (ME) criterion. This leads to minimum discrimination information (MDI) estimation
- Typically, features extracted from the training corpus are considered to be constraints set on single events of the joint probability distribution (such as, for example, a word and a history), in such a way that the constraint functions obey the marginal probabilities observed in the data

Exponential models

- Assume further that this joint distribution is constrained by *K* linearly independent constraints, written as

$$\sum_{\{(h,w)\}} I_k(h,w)\Pr(h,w) = \alpha(\hat{h}_k\hat{w}_k), \quad 1 \le k \le K \tag{10}$$

where I_k is the indicator function of an appropriate subset of the sample space (selecting the appropriate feature $\hat{h}_k\hat{w}_k$), and $\alpha(\hat{h}_k\hat{w}_k)$ denotes the relevant empirical marginal probability

Exponential models

- It can be shown (GIS) that the joint distribution Pr(*h*,*w*) satisfying the constraints belongs to the exponential family. It has the parametric form:

$$\Pr(h, w) = \frac{1}{Z(h, w)} \prod_{k=1}^{K} \exp\left\{\lambda_k I_k(h, w)\right\}$$

where λ_k is the MDI parameter associated with the *k*th linear constraint in (10), and *Z*(*h*,*w*) is a suitable normalization factor. The λ parameters are typically trained using the generalized iterative scaling (GIS) algorithm

ME adaptation

w_q	h ends in w_1	h ends in w_2		
	S_1= $\{(w_q,h_{1,1}), (w_q,h_{1,2}), \ldots, (w_q,h_{1,\lvert S1\rvert}),\}$	S_1= $\{(w_q,h_{2,1}), (w_q,h_{2,2}), \ldots, (w_q,h_{2,\lvert S2\rvert}),\}$		

$$\text{marginal distribution} \Rightarrow \sum_{i=1}^{|S_1|} P(h_{1,i}, w_q) = \tilde{P}(h\ ends\ in\ w_1, w_q)$$

w_q	
t∈h	$S_3=\{(w_q,h_{3,1}), (w_q,h_{3,2}),\ldots, (w_q,h_{3,\lvert S3\rvert}),\}$
t∉h	$S_4=\{(w_q,h_{4,1}), (w_q,h_{4,2}),\ldots, (w_q,h_{4,\lvert S4\rvert}),\}$

ME adaptation
Information source ➔ constraint

- Bigram:

$$\sum_{i=1}^{|S_1|} P(h_{1,i}, w_q) = \widetilde{P}(h\ ends\ in\ w_1, w_q)$$

$$\text{constraint} \Rightarrow \sum_{w,h} P(h,w) f_1(h,w) = K_1$$

$$\text{slelctor function}: f_1(h,w) = \begin{cases} 1 & h\ ends\ in\ w_1, w = w_q \\ 0 & \text{otherwise} \end{cases}$$

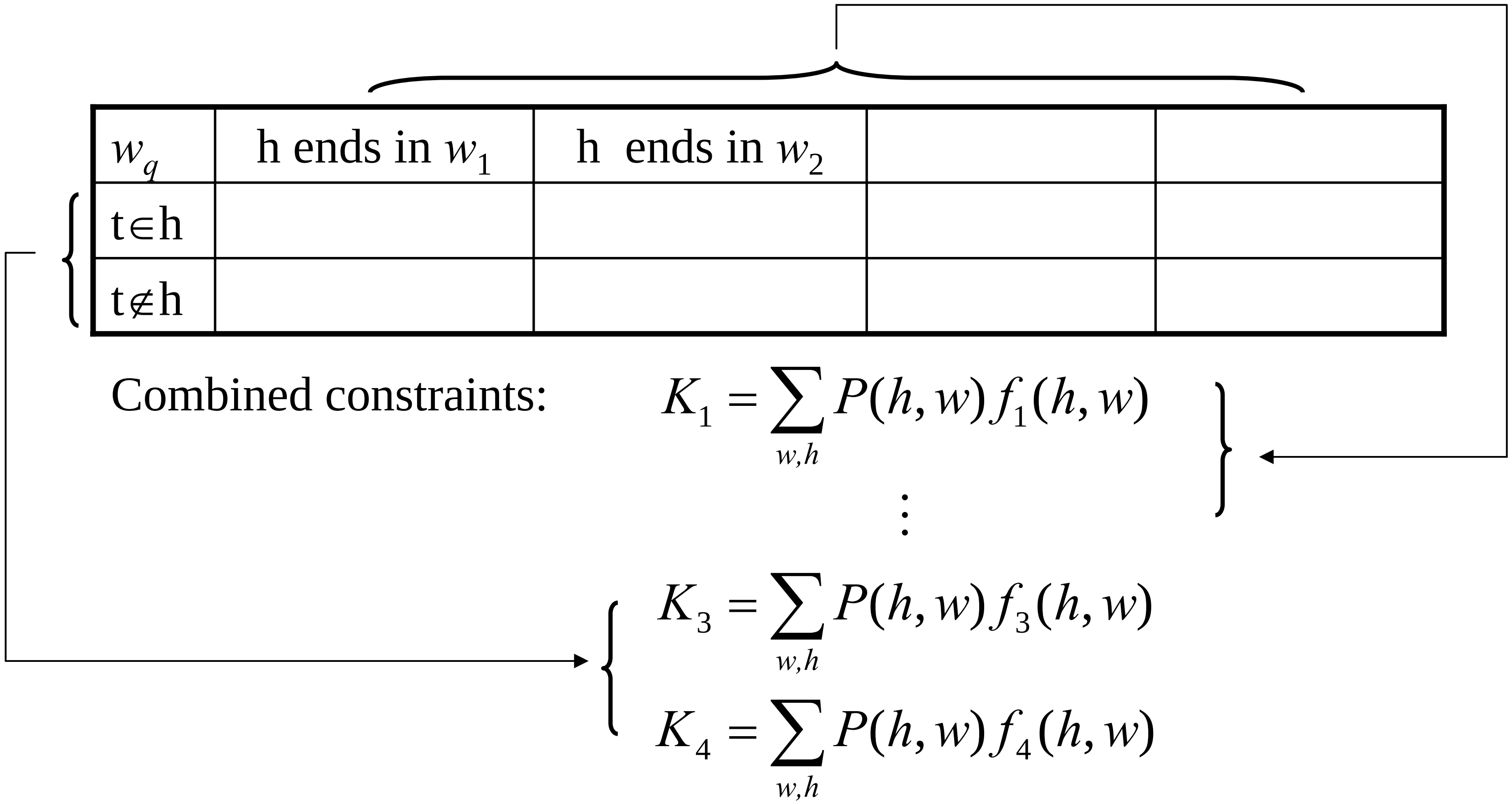
ME adaptation
Information source

- Trigger pair:

$$\text{constraint} \Rightarrow \sum_{w,h} P(h,w) f_3(h,w) = K_3$$

$$\text{slelctor function}: f_1(h,w) = \begin{cases} 1 & w = w_q, t \in h \\ 0 & \text{otherwise} \end{cases}$$

ME adaptation
Combine Information source



w_q	h ends in w_1	h ends in w_2		
t∈h				
t∉h				

Combined constraints:

$$K_1 = \sum_{w,h} P(h,w) f_1(h,w)$$

$$\vdots$$

$$K_3 = \sum_{w,h} P(h,w) f_3(h,w)$$

$$K_4 = \sum_{w,h} P(h,w) f_4(h,w)$$

$$\text{Entropy}: H(P) = -\sum_{w,h} P(h,w) \log P(h,w)$$

Each constraint can be written as $K(f_i) = K_i$

$$\Lambda(P,\lambda) \equiv H(P) + \sum_i \lambda_i \left[K(f_i) - K_i\right]$$

$$P^*(h,w) = \arg\max_P \Lambda(P,\lambda)$$

$$-\sum_{w,h} P_A(h,w) \log P_A(h,w) + \sum_i \lambda_i \left[\sum_{h,w} P_A(h,w) f_i(h,w) - K_i\right]$$

$$\frac{\partial}{\partial P_A(h,w)} = -\log P_A(h,w) - 1 + \sum_i \lambda_i f_i(h,w) = 0$$

$$\Rightarrow \log P_A(h,w) = \sum_i \lambda_i f_i(h,w) - 1$$

$$\Rightarrow P_A(h,w) = \frac{1}{e} \exp\left(\sum_i \lambda_i f_i(h,w)\right)$$

$$\boxed{\begin{aligned} P_A(h,w) &= \frac{1}{Z(h,w)} \exp\left(\sum_i \lambda_i f_i(h,w)\right) \\ &= \frac{1}{Z(h,w)} \prod_i (\exp \lambda_i)^{f_i(h,w)} \end{aligned}}$$

$Normalization:$

$$P_A(h,w) = \frac{1}{Z(h,w)} \exp\left(\sum_i \lambda_i f_i(h,w)\right), \quad Z(h,w) = \sum_{w.h} \exp\left(\sum_i \lambda_i f_i(h,w)\right)$$

$$P_A(w \mid h) = \frac{P(h,w)}{\sum_w P(h,w)} = \frac{1}{Z(h)} \exp\left(\sum_i \lambda_i f_i(h,w)\right), \quad Z(h) = \sum_w \exp\left(\sum_i \lambda_i f_i(h,w)\right)$$

MDI adaptation

- In MDI adaptation, the features extracted from *A* are considered as important properties of the adaptation data, that the joint *n*-gram distribution Pr(*h*,*w*) is requested to match, in the same manner as before
- But, in addition, the solution has to be close to the joint background distribution $\Pr_B(h,w)$. This is achieved by minimizing the KL distance from the joint background distribution:

$$\min_{Q(h,w)} \sum_{\{(h,w)\}} Q(h,w) \log \frac{Q(h,w)}{\Pr_B(h,w)}$$

MDI adaptation

- While simultaneously satisfying the linear constraints:

$$\sum_{\{(h,w)\}} I_k(h,w)Q(h,w) = \alpha_A(\hat{h}_k \hat{w}_k), \quad 1 \le k \le K$$

where the notation α_A emphasizes the fact that the relevant empirical marginal probabilities are now obtained from the adaptation corpus A

$$-\left(\sum_{w,h} P_A(h,w)\log\frac{P_A(h,w)}{\boxed{P_B(h,w)}}\right)+\sum_i\lambda_i\left[\sum_{h,w}P_A(h,w)f_i(h,w)-K_i\right]$$

$$=-\left(\sum_{w,h}P_A(h,w)\log P_A(h,w)+\boxed{\sum_{w,h}P_A(h,w)\log\frac{1}{P_B(h,w)}}\right)+\sum_i\lambda_i\left[\sum_{h,w}P_A(h,w)f_i(h,w)-K_i\right]$$

$$\frac{\partial}{\partial P_A(h,w)}=-\log P_A(h,w)-1\boxed{-\log\frac{1}{P_B(h,w)}}+\sum_i\lambda_i f_i(h,w)=0$$

$$\Rightarrow \log P_A(h,w)=\sum_i\lambda_i f_i(h,w)-1\boxed{+P_B(h,w)}$$

$$\Rightarrow P_A(h,w)=\frac{\boxed{P_B(h,w)}}{e}\exp\left(\sum_i\lambda_i f_i(h,w)\right)$$

$$\boxed{\begin{aligned}P_A(h,w)&=\frac{P_B(h,w)}{Z(h,w)}\exp\left(\sum_i\lambda_i f_i(h,w)\right)\\&=\frac{P_B(h,w)}{Z(h,w)}\prod_i(\exp\lambda_i)^{f_i(h,w)}\end{aligned}}$$

Normalization :

$$P_A(h,w)=\frac{P_B(h,w)}{Z(h,w)}\exp\left(\sum_i\lambda_i f_i(h,w)\right),\quad Z(h,w)=\sum_{w.h}\exp\left(\sum_i\lambda_i f_i(h,w)\right)$$

$$P_A(w\,|\,h)=\frac{P_A(h,w)}{\sum_w P_A(h,w)}=\frac{1}{Z(h)}\exp\left(\sum_i\lambda_i f_i(h,w)\right),\quad Z(h)=\sum_w\exp\left(\sum_i\lambda_i f_i(h,w)\right)$$

Unigram constraints

- MDI adaptation with unigram constraints is an important special case. Given the typically small amount of adaptation data available, it is often the case that only unigram features can be reliably estimated on the adaptation corpus A
- In this case, constraints become

$$\sum_{\{(h,w)\}} I_k(h,w)Q(h,w) = \alpha_A(\hat{w}_k), \quad 1 \le k \le K$$

where $\alpha_A(\hat{w}_k)$ now represents the empirical unigram probability obtained from A for the feature $\hat{w}_k$

Unigram constraints

- GIS: $P_A^{(0)}(h,w) = P_B(h,w)$

$$P_A^{(n+1)} = P_A^{(n)}(h,w) \cdot \prod_{i=1}^{K} \left(\frac{\alpha_A(\hat{w})}{P_A^{(n)}(h,w)} \right)^{\frac{f_i(h,w)}{m}}$$

$$\text{where } m = \sum_i f_i(h,w)$$

- Here each event (h,w) just satisfies one constraint, so m=1 and

$$P_A(h,w) = P_B(h,w)\alpha(w)$$

$$\text{where } \alpha(w) = \frac{\alpha_A(\hat{w})}{\Pr_B(w)} = \frac{\Pr_A(w)}{\Pr_B(w)}$$

Adaptation approach

- Model interpolation
- Constraint specification
- Topic information
- Semantic knowledge
- Syntactic infrastructure
- Multiple sources

Topic information

- In approaches exploiting the general topic of the discourse, the corpus *A* is used to extract information about the underlying subject matter.
- This information is then used in various ways to improve upon the background model based on semantic classification

Mixture models

- The simplest approach is based on a generalization of linear interpolation to include several pre-defined domains
- Consider a set of topics $\{t_k\}$, usually from a hand-labeled hierarchy, which covers the relevant semantic space of the background corpus B. Assume further that the background n-gram model is composed of a collection of K sub-models, each trained on a separate topic

Mixture models

- Mixture SLMs linearly interpolate these *K* *n*-grams in such a way that the resulting mixture best matches the adaptation data *A*

$$\Pr(w_q \mid h_q) = \sum_{k=1}^{K} \lambda_{A,k} \Pr_{B,k}(w_q \mid h_q)$$

  where $\Pr_{B,k}$ refers to the *k*th pre-defined topic sub-model, and the notation $\lambda_{A,k}$ for the interpolation coefficients reflects the fact that they are estimated on *A*

Practical considerations

- It turns out that, in actual usage, the mixture SLM is less practical than a single SLM, in part because it complicates smoothing
- To address this issue, it is possible to simply merge the *n*-gram counts from the mixture model and train a single SLM on these counts
- When some pre-defined topics are more appropriate than others for the recognition task at hand, the *n*-gram counts can be empirically weighted using some held-out data

Practical considerations

- Another approach is to merge the different SLM components of the SLM mixture to create a single SLM
  - There are as many *n*-grams in the resulting SLM as there are distinct *n*-grams in the individual topic SLMs trained on the separate portions of *B*
- Single merged SLM is amenable to proper optimization and smoothing

Practical considerations

- Experimental results did not show any difference between the original SLM mixture implementation and the SLM mixture merging alternative
- The biggest drawback of the adaptive mixture approach is the inherent fragmentation of the training data which occurs when partitioning the corpus *B* into different topics

Explicit topic models

- Consider the language model probability:

$$\Pr(w_q \mid h_q) = \sum_{k=1}^{K} \Pr(w_q \mid t_k) \Pr(t_k \mid h_q)$$

  where t_k is one of the K topics above.
- This approach is less restrictive than topic mixtures, since there is no assumption that each history belongs to exactly one topic cluster

Explicit topic models

- The language model probability now comprises two components:
  - a topic *n*-gram –$\Pr(t_k|h_q)$ this is assumed to remain unaffected by new material, and is therefore taken from the background SLM

    $$\Pr(t_k \mid h_q) = \Pr_B(t_k \mid h_q)$$

  - a topic assignment –$\Pr(w_q|t_k)$

    $$\Pr(w_q \mid t_k) = (1-\lambda)\Pr_A(w_q \mid t_k) + \lambda \Pr_B(w_q \mid t_k)$$

Adaptation approach

- Model interpolation
- Constraint specification
- Topic information
- Semantic knowledge
- Syntactic infrastructure
- Multiple sources

Semantic knowledge

- Approaches taking advantage of semantic knowledge purport to exploit not only topic information as above, but the entire semantic fabric of the corpus *A*

Triggers

- If a word *A* is significantly correlated with another word *B*, then (*A*→*B*) is considered a "trigger pair"
- In practice, word pairs with high mutual information are searched for inside a window of fixed duration

$$I(A:B) = P(A,B)\log\frac{P(B\mid A)}{P(B)} + P(A,\overline{B})\log\frac{P(\overline{B}\mid A)}{P(\overline{B})}$$

$$+P(\overline{A},B)\log\frac{P(B\mid \overline{A})}{P(B)} + P(\overline{A},\overline{B})\log\frac{P(\overline{B}\mid \overline{A})}{P(\overline{B})}$$

Triggers

- Trigger pair selection is a complex issue
- Different trigger pairs display different behavior, and hence should be modeled differently
- *Self triggers* (i.e. triggers of the form ($A \rightarrow A$)) are particularly powerful and robust
  - for more than two thirds of the words, the highest-MI trigger proved to be the word itself
  - For 90% of the words, the self-trigger was among the top six triggers

Latent semantic analysis

- LSA reveals meaningful associations in the language based on word-document co-occurrences, as observed in a document collection pertinent to the current task
- The resulting semantic knowledge is encapsulated in a continuous vector space (LSA space) of comparatively low dimension, where all words and documents in the training data are mapped
- This mapping is derived through a singular value decomposition (SVD) of the co-occurrence matrix between words and documents

Latent semantic analysis

- Consider a shift in subject matter. Since pre- and post-shift sub-corpora are essentially disjoint, the probabilities of almost all the words in the vocabulary are likely to change. This require an in-ordinate number of parameters
- Our intuition is that only a small fraction of all probability changes is actually relevant, so the “true” dimension of the semantic shift is probably much lower

Latent semantic analysis

- LSA framework was embedded within the conventional *n*-gram formalism, so as to combine the local constraints provided by *n*-grams with the global constraints of LSA

$$\Pr(w_q \mid h_q, \tilde{h}_q) = \frac{\Pr(w_q \mid h_q)\rho(w_q, \tilde{h}_q)}{Z(h_q, \tilde{h}_q)} \qquad (20)$$

where $\tilde{h}_q$ denotes the global ("bag-of-words") document history, $\rho(w_q, \tilde{h}_q)$ is a measure of the correlation between the current word and this global LSA history, and $Z(h_q, \tilde{h}_q)$ ensures appropriate normalization

Latent semantic analysis

- The language model (20) represents a modified *n*-gram SLM incorporating large-span semantic information derived through LSA
- Taking advantage of (20), adaptation can proceed separately for the *n*-gram and the LSA. By analogy with topic-based adaptation, the latter could conceivably be obtained as

$$\rho(w_q, \widetilde{h}_q) = (1-\lambda)\rho_A(w_q, \widetilde{h}_q) + \lambda\rho_B(w_q, \widetilde{h}_q)$$

Adaptation approach

- Model interpolation
- Constraint specification
- Topic information
- Semantic knowledge
- Syntactic infrastructure
- Multiple sources

Syntactic infrastructure

- Approaches leveraging syntactic knowledge make the implicit assumption that the background and recognition tasks share a common grammatical infrastructure
- The background SLM is used for initial syntactic modeling, and the corpus A to re-estimate the associated parameters

Structured language models

Syntactic triggers

- Two kinds of triggering events are considered:
  - Those based on the knowledge of the full parse of previous sentences
  - Those based on the knowledge of the syntactic/semantic tags to the left of and in the same sentence as the word being predicted

Adaptation approach

- Model interpolation
- Constraint specification
- Topic information
- Semantic knowledge
- Syntactic infrastructure
- Multiple sources

Multiple sources

- In approaches exploiting multiple knowledge sources, the corpus *A* is used to extract information about different aspects of the mismatch between training and recognition conditions

Combination models

- A popular way to combine knowledge from multiple knowledge sources is to use exponential models, because the underlying maximum entropy criterion offers the theoretical advantage of incorporating an arbitrary number of features

Whole sentence models