Geometric Interpolation (GI):

$$P'(w_i) = \frac{P_b(w_i)^{\lambda_1} \cdot P_s(w_i)^{(1-\lambda_1)}}{\sum_{j=1}^{n} P_b(w_j)^{\lambda_1} \cdot P_s(w_j)^{(1-\lambda_1)}} \quad (9)$$

The main difference between the two methods is that the latter takes the agreement of two models into account. Only if each of the single models assigns a high probability to a given event will the combined probability be assigned a high value. If one of the models assigns a high probability and the other does not the resulting probability will be lower.

3.4 Confidence-weighted interpolation

Whereas in standard settings the coefficients are stable for all probabilities, some approaches use confidence-weighted coefficients that are adapted for each probability. In order to integrate n-gram and LSA probabilities, Coccaro and Jurafsky (1998) proposed an entropy-related confidence measure for the LSA component, based on the observation that words that occur in many different contexts (i.e. have a high entropy), cannot well be predicted by LSA. We use here a density-based measure (cf. section 2.2), because we found it more reliable than entropy in preliminary tests. For interpolation purposes we calculate the coefficient of the LSA component as follows:

$$\lambda_i = \beta \cdot D(w_i), \text{ iff } D(w_i) > 0; \text{ 0 otherwise} \quad (10)$$

with β being a weighting constant to control the influence of the LSA predictor. For all experiments, we set β to 0.4 (i.e. $0 \leq \lambda_i \leq 0.4$), which proved to be optimal in pre-tests.

4 Results

We calculated our baseline n-gram model on a 44 million word corpus from the French daily *Le Monde* (1998-1999). Using the *SRI* toolkit (Stolcke, 2002)[1] we computed a 4-gram LM over a controlled 141,000 word vocabulary, using *modified Kneser-Ney* discounting (Goodman, 2001), and we applied *Stolcke* pruning (Stolcke, 1998) to reduce the model to a manageable size ($\theta = 10^{-7}$).

The LSA space was calculated on a 100 million word corpus from *Le Monde* (1996 – 2002). Using the *Infomap* toolkit[2], we generated a term × term co-occurrence matrix for an 80,000 word vocabulary (matrix size = 80,000 × 3,000), stopwords were excluded. After several pre-tests, we set the size of the co-occurrence window to ±100. The matrix was then reduced by singular value decomposition to 150 columns, so that each word in the vocabulary was represented by a vector of 150 dimensions, which was normalized to speed up similarity calculations (the scalar product of two normalized vectors equals the cosine of their angle).

Our test corpus consisted of 8 sections from the French newspaper *Humanité*, (January 1999, from 5,378 to 8,750 words each), summing up to 58,457 words. We then calculated for each test set the keystroke saving rate based on a 5-word list (ksr_5) and perplexity for the following settings[3]:

1. 4-gram LM only (baseline)
2. 4-gram + decaying cache (l = 400)
3. 4-gram + LSA using linear interpolation with λ_{LSA} = 0.11 (LI).
4. 4-gram + LSA using geometric interpolation, with λ_{LSA} = 0.07 (GI).
5. 4-gram + LSA using linear interpolation and (density-based) confidence weighting (CWLI).
6. 4-gram + LSA using geometric interpolation and (density-based) confidence weighting (CWGI).
7. 4-gram + partial reranking (n = 1000, β = 0.001)
8. 4-gram + decaying semantic cache (l = 4000; m = 10; θ = 0.4, β = 0.0001)

Figures 3 and 4 display the overall results in terms of *ksr* and perplexity.

[1] SRI Toolkit: www.speech.sri.com.

[2] Infomap Project: http://infomap-nlp.sourceforge.net/

[3] All parameter settings presented here are based on results of extended empirical pre-tests. We used held-out development data sets that have randomly been chosen from the *Humanité* corpus.(8k to 10k words each). The parameters being presented here were optimal for our test sets. For reasons of simplicity we did not use automatic optimization techniques such as the EM algorithm (cf. Jelinek, 1990).