following decay function for a word w_i and its position p in the cache.

$$f_d(w_i, p) = e^{\left(\frac{-0,5(p-\mu)}{\sigma}\right)^2} \qquad (5)$$

$\sigma = \mu/3$ if $p < \mu$ and $\sigma = l/3$ if $p \geq \mu$. The function returns 0 if w_i is not in the cache, and it is 1 if $p = \mu$. A typical graph for (5) can be seen in figure (2).

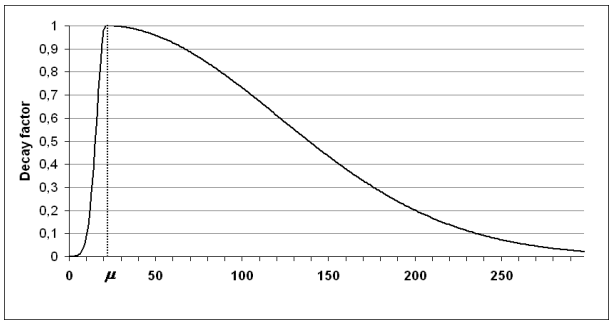

Figure 2: Decay function with μ=20 and l=300.

We extend this model by calculating for each element having occurred in the context its m nearest LSA neighbors ($NN_m(\vec{w}_{occ}, \theta)$, using cosine similarity), if their cosine lies above a threshold θ, and add them to the cache as well, right after the word that has occurred in the text ("*Bring your friends*"-strategy). The size of the cache is adapted accordingly (for μ, σ and l), depending on the number of neighbors added. This results in the following cache function:

$$P_{cache}(w_i) = \sum_1^l \beta \cdot f_{\cos}(w^i_{occ}, w_i) \cdot f_d(w_i, p) \quad (6)$$

with l = size of the cache. β is a constant controlling the influence of the component (usually $\beta \approx 0.1/l$); w^i_{occ} is a word that has already recently occurred in the context and is therefore added as a standard cache element, whereas w_i is a nearest neighbor to w^i_{occ}. $f_{cos}(w^i_{occ}, w_i)$ returns the cosine similarity between $\vec{w}^i_{occ}$ and $\vec{w}_i$, with $cos(\vec{w}^i_{occ}, \vec{w}_i) > \theta$ (Rem: w_i with $cos(\vec{w}^i_{occ}, \vec{w}_i) \leq \theta$ have not been added to the cache). Since $cos(\vec{w}_i, \vec{w}_i)$=1, terms having actually occurred before will be given full weight, whereas all w_i being only nearest LSA neighbors to w^i_{occ} will receive a weight corresponding to their cosine similarity with w^i_{occ}, which is less than 1 (but larger than θ).

$f_d(w_i,p)$ is the decay factor for the current position p of w_i in the cache, calculated as shown in equation (5).

3.2 Partial reranking

The underlying idea of partial reranking is to regard only the best n candidates from the basic language model for the semantic model in order to prevent the LSA model from making totally implausible (i.e. improbable) predictions. Words being improbable for a given context will be disregarded as well as words that do not occur in the semantic model (e.g. function words), because LSA is not able to give correct estimates for this group of words (here the base probability remains unchanged).

For the best n candidates their semantic probability is calculated and each of these words is assigned an additional value, after a fraction of its base probability has been subtracted (*jackpot* strategy).

For a given context h we calculate the ordered set

$\text{BEST}_n(h) = \langle w_1, \ldots, w_n \rangle$, so that $P(w_1|h) \geq P(w_2|h) \geq \ldots \geq P(w_n|h)$

For each w_i in $\text{BEST}_n(h)$ we then calculate its reranking probability as follows:

$$P_{RR}(w_i) = \beta \cdot \cos(\vec{w}_i, \vec{h}) \cdot D(w_i) \cdot I(Best_n(h), w_i) \quad (7)$$

β is a weighting constant controlling the overall influence of the reranking process, $cos(\vec{w}_i, \vec{w}_i)$ returns the cosine of the word's vector and the current context vector, $D(w_i)$ gives the confidence measure of w_i and I is an indicator function being 1, iff $w_i \in \text{BEST}(h)$, and 0 otherwise.

3.3 Standard interpolation

Interpolation is the standard way to integrate information from heterogeneous resources. While for a linear combination we simply add the weighted probabilities of two (or more) models, geometric interpolation multiplies the probabilities, which are weighted by an exponential coefficient ($0 \leq \lambda_1 \leq 1$):

Linear Interpolation (LI):

$$P'(w_i) = \lambda_1 \cdot P_b(w_i) + (1 - \lambda_1) \cdot P_s(w_i) \quad (8)$$