range is low). For this reason a contrasting (or temperature) factor γ is normally applied (cf. Coccaro and Jurafsky, 1998), which raises the cosine to some power (γ is normally between 3 and 8). After normalization we obtain a probability distribution which can be used for prediction purposes. It is calculated as follows:

$$P_{LSA}(w_i|h) = \frac{\left(\cos(\vec{w}_i,\vec{h}) - \cos_{\min}(\vec{h})\right)^{\gamma}}{\sum_k \left(\cos(\vec{w}_k,\vec{h}) - \cos_{\min}(\vec{h})\right)^{\gamma}} \qquad (3)$$

w_i is a word in the vocabulary, h is the current context (history) $\vec{w}_i$ and $\vec{h}$ are their corresponding vectors in the LSA space; $\cos_{\min}(\vec{h})$ returns the lowest cosine value measured for $\vec{h}$). The denominator then normalizes each similarity value to ensure that $\sum_k^n P_{LSA}(w_k, h) = 1$.

Let us illustrate the capacities of this model by giving a short example from the French version of our own LSA predictor:

Context: *"Mon père était professeur en mathématiques et je pense que "*
("My dad has been a professor in mathematics and I think that ")

Rank	Word	P
1.	*professeur* ('professor')	0.0117
2.	*mathématiques* ("mathematics")	0.0109
3.	*enseigné* (participle of 'taught')	0.0083
4.	*enseignait* ('taught')	0.0053
5.	*mathematicien* ('mathematician')	0.0049
6.	*père* ('father')	0.0046
7.	*mathématique* ('mathematics')	0.0045
8.	*grand-père* ('grand-father')	0.0043
9.	*sciences* ('sciences')	0.0036
10.	*enseignant* ('teacher')	0.0032

Example 1: Most probable words returned by the LSA model for the given context.

As can be seen in example 1, all ten predicted words are semantically related to the context, they should therefore be given a high probability of occurrence. However, this example also shows the drawbacks of the LSA model: it totally neglects the presence of function words as well as the syntactic structure of the current phrase. We therefore need to find an appropriate way to integrate the information coming from a standard n-gram model and the LSA approach.

2.4 Density as a confidence measure

Measuring relation quality in an LSA space, Wandmacher (2005) pointed out that the reliability of LSA relations varies strongly between terms. He also showed that the entropy of a term does not correlate with relation quality (i.e. number of semantically related terms in an LSA-generated term cluster), but he found a medium correlation (*Pearson* coeff. = 0.56) between the number of semantically related terms and the average cosine similarity of the *m* nearest neighbors (density). The closer the nearest neighbors of a term vector are, the more probable it is to find semantically related terms for the given word. In turn, terms having a high density are more likely to be semantically related to a given context (i.e. their specificity is higher).

We define the density of a term w_i as follows:

$$D_m(w_i) = \frac{1}{m}\cdot\sum_{j=1}^{m}\cos(\vec{w}_i, NN_j(\vec{w}_i)) \qquad (4)$$

In the following we will use this measure (with *m*=100) as a confidence metric to estimate the reliability of a word being predicted by the LSA component, since it showed to give slightly better results in our experiments than the entropy measure.

3 Integrating semantic information

In the following we present several different methods to integrate semantic information as it is provided by an LSA model into a standard LM.

3.1 Semantic cache model

Cache (or recency promotion) models have shown to bring slight but constant gains in language modeling (Kuhn and De Mori, 1990). The underlying idea is that words that have already occurred in a text are more likely to occur another time. Therefore their probability is raised by a constant or exponentially decaying factor, depending on the position of the element in the cache. The idea of a decaying cache function is that the probability of reoccurrence depends on the cosine similarity of the word in the cache and the word to be predicted. The highest probability of reoccurrence is usually after 15 to 20 words.

Similar to Clarkson and Robinson (1997), we implemented an exponentially decaying cache of length *l* (usually between 100 and 1000), using the