	Formula	Description
Basic LM	$P_{basic}[q\|x] = \prod_i P[q_i\|x];$	A unigram/bigram LM described by the probability of generation of a query q from a document x. The weight of the i^{th} word in q is given by $P[q_i\|x]$. The product over all words of the query ensures a conjunctive query.
	$P[q_i\|x] = \frac{\alpha_x}{\|j\|} \sum_j sim(q_i, xx_j) P[xx_j\|x] + \frac{\alpha_v}{\|l\|} \sum_l sim(q_i, xv_l) P[xv_l\|x];$	A word in the query may match with the textual or visual features of a document weighted by α_x and α_v, and normalised with number of matches $\|j\|$ and $\|l\|$ respectively.
Smoothed LM	$P_{smoothed}[q\|x] = \alpha P_{basic}[q\|x] + (1-\alpha) P[q\|B];$ $P[q\|B] = \prod_i P[q_i\|B]$	The Basic LM after smoothing on background corpus B. The relative frequency of q_i in B ($P[q_i\|B]$) is used for smoothing the LM.
Commonsense-aware LM	$P_{CS}[q\|x] = \prod_i \left[\frac{\sum_k P[q_i\|y_k] P[y_k\|x]}{\|k\|} \right];$	A translation LM describing the probability of generation of a query from the k commonsense knowledge triples y_k. The summation over k includes all triples bridging the gap between the query vocabulary and the document vocabulary; it is normalized by the total number of such triples.
	$P[q_i\|y_k] = \sum_j sim(q_i, y_{kj})$	The probability that the query word q_i has been generated from the CSK triple y_k is the sum of similarity scores between the two words/phrases, normalised by the number of words/phrases ($\|j\|$) in the CSK triples.
Mixture LM	$P[q\|x] = \beta_{CS} P_{CS}[q\|x] + (1-\beta) P_{smoothed}[q\|x]$	Combination of the weighted Commonsense-aware LM and Smoothed LM for ranking a document x for a query q.

Table 1: Mathematical formulations of Language Models for Ranking

	Formula	Description
Textual feature weight	$P[xx_j\|x] = \frac{idf(xx_j)}{\sum_\nu idf(xx_\nu)}$	The informativeness or weight of a word/phrase xx_j in a document is captured by calculating it's idf in a large background corpus ν.
Visual feature weight	$P[xv_j\|x] = \frac{conf(xv_j)}{\sum_\nu conf(xv_\nu)} \times \frac{idf(xv_j)}{\sum_\nu idf(xv_\nu)}$	The weight of a object class xv_j in a document is calculated by the product of it's confidence (from LSDA) and it's informativeness.
CSK feature weight	$P[y_k\|x] = \frac{\sum_i \sum_j sim(y_{kj}, x_i) sal(y_{kj}) inf(y_{kj})}{\|i\|\|j\|}$	The relevance of a commonsense triple y to a document is decided by the similarity of its words/phrases y_k to the features of the document, the salience (or importance) of the match, and the informativeness of the word/phrase.

Table 2: Mathematical formulations of Feature Weights

Hyper-parameter	Description
α	Weight of the basic document features; $(1-\alpha)$ being the weight for smoothing.
α_x	Weight associated with the textual features of a document.
α_v	Weight associated with the visual features of a document.
β_{CS}	Weight pertaining to the commonsense knowledge features of an expanded document.

Table 3: Definition of Hyper-parameters

	Function	Description
Confidence	$conf(w)$	A score output by the LSDA to depict the confidence of detection of an object class. The hypernyms of the detected visual object classes are assigned the same confidence score.
Informative-ness	$inf(w) = idf_B(w)$	We measure informative-ness of a word by its idf value in a larger corpus, such that common terms are penalised.
Similarity	$sim(w_1, w_2) = \frac{\lvert substring(w_1, w_2)\rvert}{max(length(w_1), length(w_2))}$	This function calculates the amount to string overlap between w_1 and w_2.
Salience	$sal(w) = \lambda_s$ if $w \in subject$ $= \lambda_p$ if $w \in predicate$ $= \lambda_o$ if $w \in object$ where $t_{csk} = \langle subject, predicate, object \rangle$	The importance of the string match position in a commonsense knowledge triple t_{csk} is captured by this function. Intuitively, the textual features in the subject and the object are more important that those in the predicate. Therefor we assign $\lambda_s = \lambda_o > \lambda_p$ and $\lambda_s + \lambda_p + \lambda_o = 1$

Table 4: Function definitions