Terms and Vocabulary

- What are pros and cons of stemming? (2013) (5%)
  What are pros and cons of stemming in a web searching engine? (2010) (5%)
  Why do we need stemming? And what's the difference between it and lemmatization? (2008) (10%)
- What are Heap's law and Zipf's Law? (2013, 2012, 2010, 2009, 2008) (10%)
  And use them to explain frequency-based IR or texting mining problems are hard even with a large text corpus? (2009, 2008)
- Explain the difference between terms and tokens. (2012) (3%)
- What is normalization? (2009) (5%)
- Will stemming lower recall in a Boolean retrieval system and why? (2009) (5%)

PAT Trees

- Construct a PAT tree by inserting the first 8 sistrings of the following text. **You need to show the PAT tree after each sistring insertion**.
  Text: 000110011101110… (2012, 2009) (10%)
- What is the longest string repetition in the PAT tree? (2012, 2009) (5%)

Term Weighting and Vector Space Model

- Explain why term-document-matrix based indexing is infeasible? (2013) (5%)
- Show the formula and explain the effect of TF and IDF (2013, 2008) (10%)
- The following shows a simple scoring mechanism used to rank the documents matching a query. Explain why the scoring mechanism is biased. (2013) (5%)
$$score(q,d) = \sum_{t \in q} tfidf_{t,d}$$
- In VSM, a popular measure used ti rank documents for a query is Euclidean distance:
$$|\vec{q} - \vec{d}| = \sqrt{\sum_{k=1}^{M} (q_k - d_k)^2}$$
Show that if query q and the documents in the text collection D ($D = \{d\}$) are represented as **unit vectors**, then the ranking order produced by Euclidean distance is identical to that produced by cosine measure.
Hint: You need to prove that for any two documents d_i and d_j, if $|\vec{q} - \vec{d}_i| \leq |\vec{q} - \vec{d}_j|$, then $cosine(\vec{q}, \vec{d}_i) \geq cosine(\vec{q}, \vec{d}_j)$. (2012, 2009) (10%)
Explain why length normalization is needed for tf-idf weighting scheme to compute unbiased Euclidean distance between documents. (2009) (5%)

2010 Suppose you are developing a VSM-based information retrieval system and you adopt cosine similarity to rank documents matching a query. Show that removing the length normalization of query vector from cosine similarity would not affect document ranking. (2012) (10%)

2010 Assume the system has a fixed document collection; design an efficient algorithm for computing similarity scores of all the documents. (2012) (10%)

- Suppose that the matching score between a document d and a query q is $score(q,d) = \sum_{t \in q} tfidf_{t,d}$. Show that the base of the logarithm in *idf* is not material to document ranking. (2008) (10%)

BIM, Relevance Feedback, and Evaluation

- Explain why precision and recall generally trade off against off each other (2013) (5%)
- Given the following term-document incidence matrix, employ BIM to rank the documents for the query "3G". (Assume that $p_t = 0.5$, and $u_t = \frac{df_t}{N}$) (2013, 2012, 2010, 2009, 2008) (10%)

	d_1	d_2	d_3	d_4	d_5
cellphone	0	1	1	1	0
LCD	1	0	1	0	1
3G	0	1	1	0	0
GPS	1	0	0	0	0
PDA	1	0	0	0	1

- Assume the ground truth of relevance consists of d_2, d_3 and d_4, show the 11-points precision/recall graph. (2013, 2012, 2010, 2009, 2008) (5%)
- Re-rank the documents using the technique of pseudo-relevance feedback which assumes the top **2** documents are relevant. (**To avoid the probability of zero, you need to employ the adding 1/2 smoothing mechanism; in addition, all the terms in the pseudo relevant documents need to be included in the expanded query**)(2013, 2012, 2010, 2009, 2008) (10 points)
- Explain *k*-fold cross validation. (2012) (3%)
- Explain why the odds rations of a term *t* (i.e., c_t) can be negative. (2010) (5%)
- Explain why recall is a non-decreasing function of the number of documents retrieved? (2010) (5%)

- Below is a table showing how two human judges rated the relevance of the documents to the information need (0 = non-relevant, 1 = relevant). Calculate the kappa measure between two judges. (2009) (5%)

	d_1	d_2
d_1	0	0
d_2	1	1
d_3	1	1
d_4	1	1
d_5	1	0

- Explain the following terms (2009) (10%)
  - Bag-of-words model
  - F1 of information retrieval effectiveness
  - Validation data

Language Models

- What is Markov assumption? (2013, 2012, 2010, 2008) (5%)
- Explain why n-gram systems rarely using high order (i.e., $n > 3$) Markov models. (2013, 2012) (5%)
  Explain why it's necessary for building a language model? (2012)
- Given a corpus containing 600 uniques terms, (i.e., $|V| = 600$), how many parameters should we estimate to build a bigram language model? (2013) (5%)
- The following table presents the bigram information of the corpus. Calculate the expected frequency r^* of each bigram type using Laplace's Law. (2013, 2009) (10%)

r	N_r	r^*
1	8,000	
2	1,500	
3	500	

- Calculate the probability of an unseen bigram using Good-Turing estimation. (Good-Turing is employed only for $r \leq 2$) (2013, 2009) (10%)
- Explain the zero propagation problem of n-gram modeling. (2012, 2009) (3%)
- Decompose $P(w_1w_2w_3w_4)$ using the **1st** order Markov model. (2012) (5%)
- The following table illustrates the statistics of corpus used to train a b-gram model. Calculating the probability of an unseen bigram using Laplace's law and Good-Turing estimation, respectively. (Good-Turing estimation is employed only when $r < 3$) (2012, 2008)

r	N_r
1	10,050
2	4,050
3	2,800
4	2,050
5	1,750
	$|V|$=4,000

- Given a corpus of 12,500 words (i.e., N = 12,500), containing a vocabulary V of 600 terms (i.e., $|V|$ = 600), computing the probability of an unseen bigram using Laplace's Law. (2010) (5%)
- Assume that 2,500 (distinct) bigrams actually appear in the corpus, compute the percentage of probability space that will be given to unseen bigrams. (2010) (5%)
- Given a corpus containing 600 unique terms (i.e., $|V|$ = 600), how many parameters should we estimate to build a bigram language model? (2009) (5%)
- What's the problem of MLE when estimating n-gram probabilities? (2008) (5%)