NEIGHBOURHOOD PRESERVING QUANTISATION FOR LSH

School of informatics

SEAN MORAN†, VICTOR LAVRENKO, MILES OSBORNE

† SEAN.MORAN@ED.AC.UK

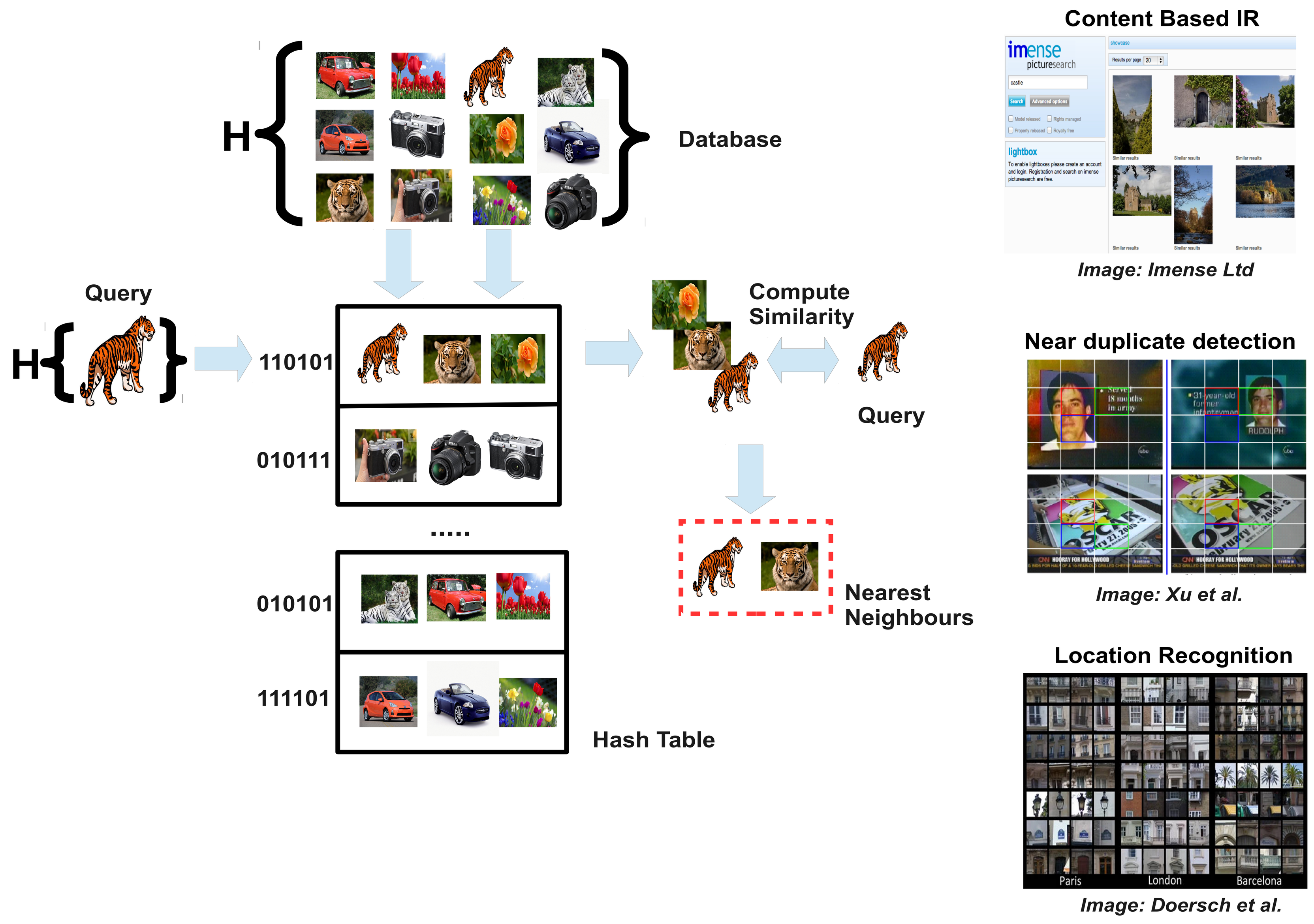
THE UNIVERSITY OF EDINBURGH

RESEARCH QUESTION

- LSH uses 1 bit per hyperplane. Can we do better with multiple bits?

INTRODUCTION

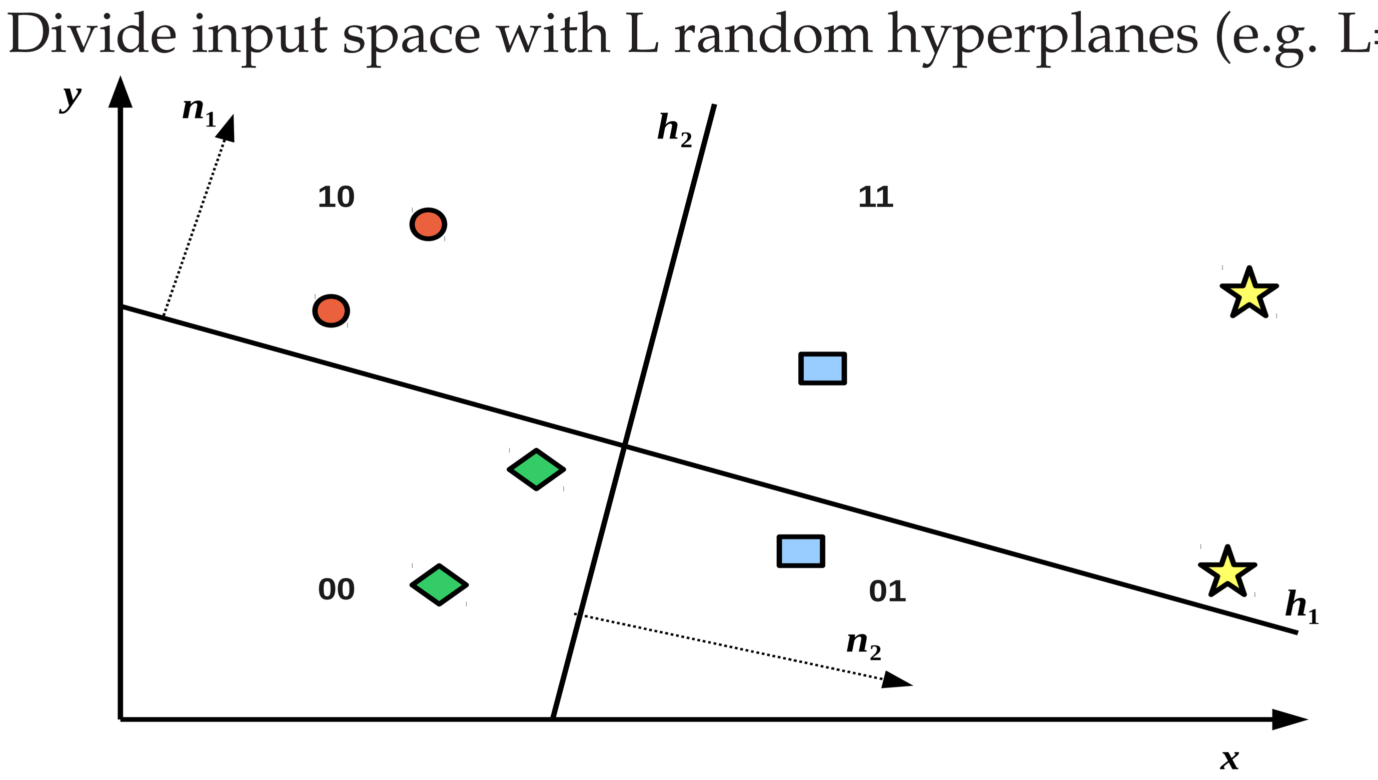
- **Problem:** Fast Nearest Neighbour (NN) search in large datasets.
- **Hashing-based approach:**
  - Generate a similarity preserving binary code (fingerprint).
  - Use fingerprint as index into the buckets of a hash table.
  - If collision occurs only compare to items in the same bucket.



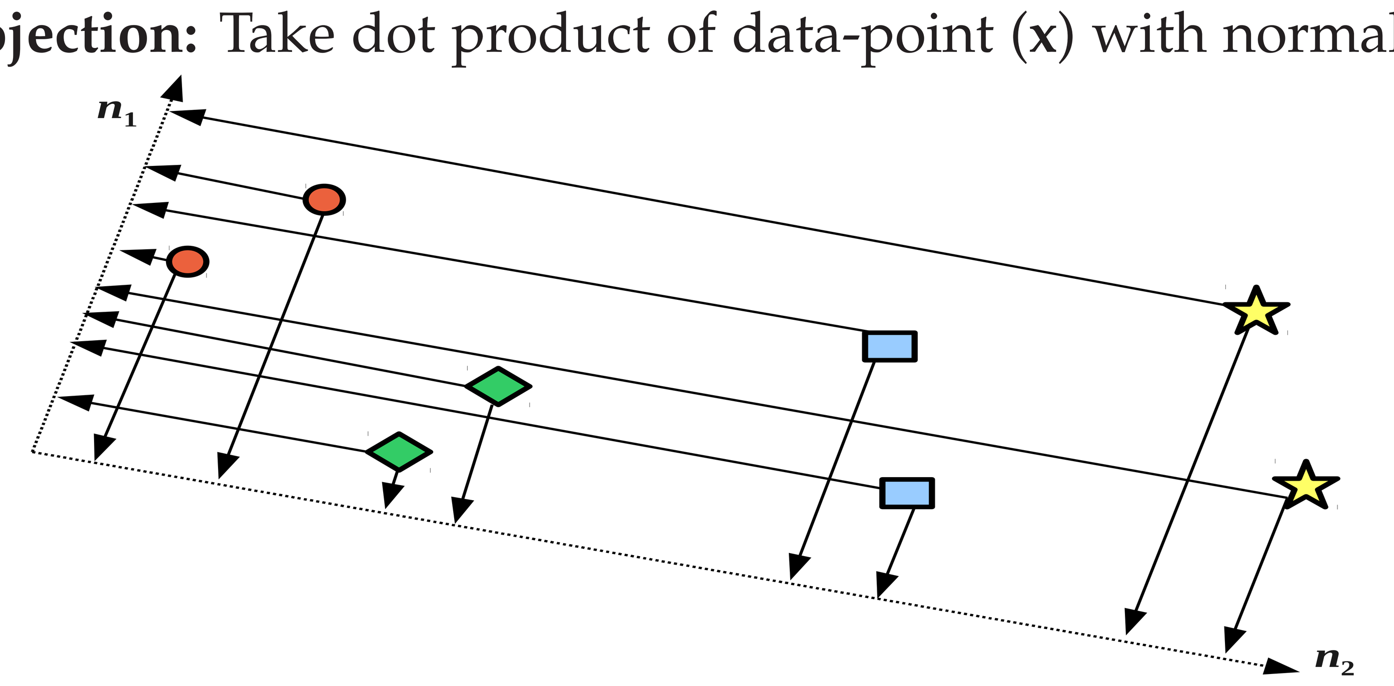
- **Advantages:**
  - Constant query time (with respect to the database size).
  - Compact binary codes are extremely storage efficient.

LOCALITY SENSITIVE HASHING (LSH) (INDYK AND MOTWANI, '98)

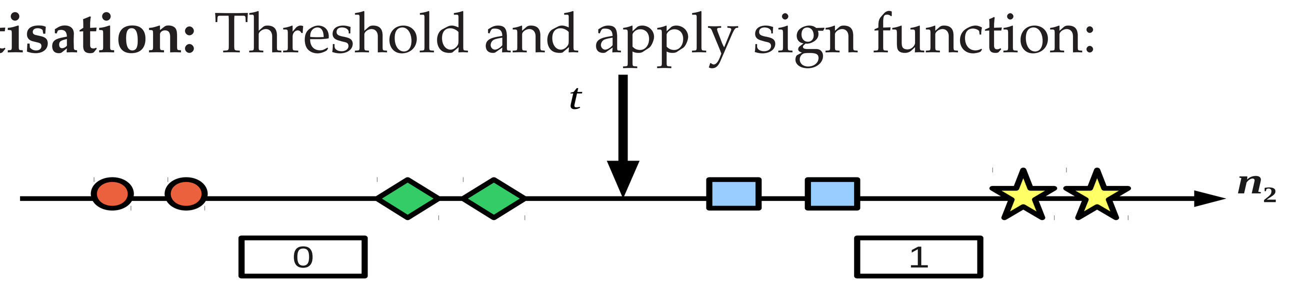
- Randomized algorithm for approximate nearest neighbour (ANN) search using binary codes.
- Probabilistic guarantee on retrieval accuracy versus search time.
- LSH for inner product similarity:
  - Divide input space with L random hyperplanes (e.g. L=2):



  - **Projection:** Take dot product of data-point (**x**) with normal (**n**.**x**):



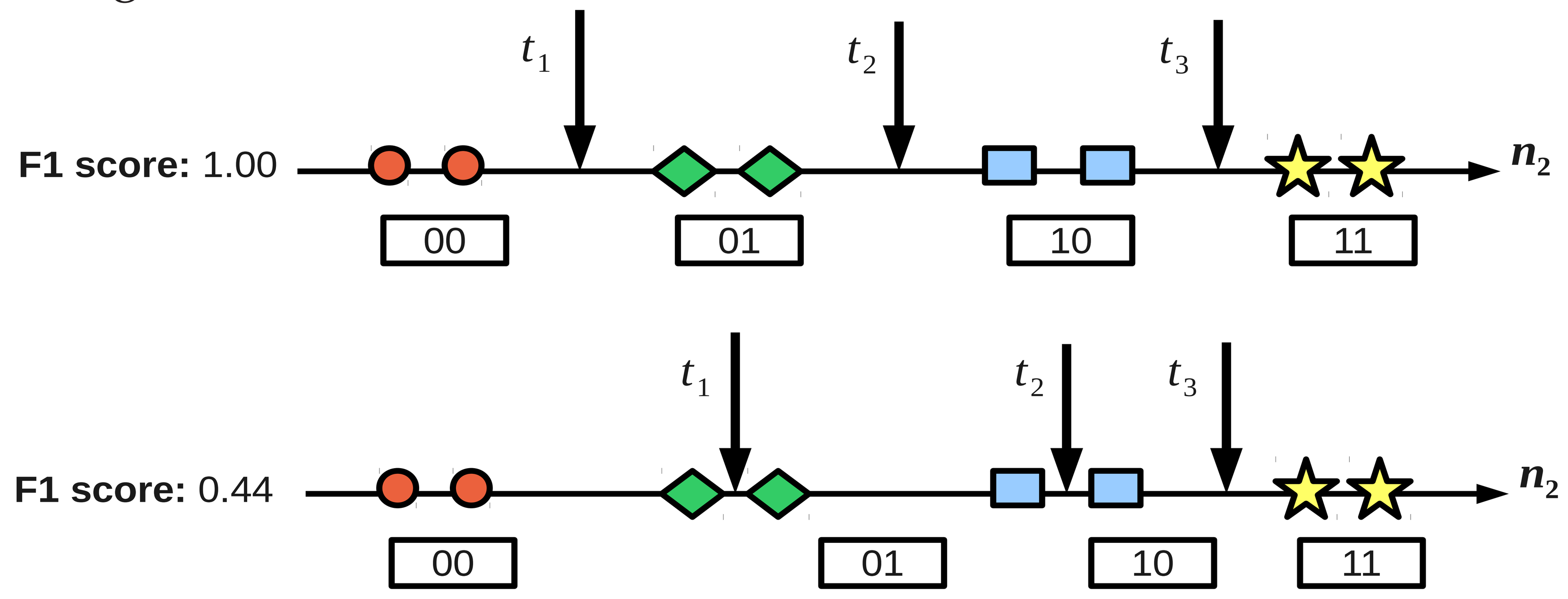
  - **Quantisation:** Threshold and apply sign function:



- **Vanilla LSH**: each hyperplane gives 1 bit of the binary code.

NEIGHBOURHOOD PRESERVING QUANTISATION (NPQ)

- Assigns multiple bits per hyperplane using multiple thresholds.
- F_1 optimisation using pairwise constraints matrix S: if $S_{ij} = 1$ then points $\mathbf{x}_i, \mathbf{x}_j$ with projections $\mathbf{y}_i, \mathbf{y}_j$ are true nearest neighbours.
- TP: # $S_{ij} = 1$ pairs in same region. FP: # $S_{ij} = 0$ pairs in same region. FN: # $S_{ij} = 1$ pairs in different regions. Combine TP, FP, FN using F_1:



- Interpolate F_1 with a *regularisation term* $\Omega(T_{1:u})$:

$$Z_{npq} = \alpha F_1 + (1-\alpha)(1-\Omega(T_{1:u})) \text{ with: } \Omega(T_{1:u}) = \frac{1}{\sigma}\sum_{a=0}^{u}\sum_{i:y_i \in r_a}\{y_i - \mu_{r_a}\}^2$$

where: $\sigma = \sum_{i=1}^{n}\{y_i - \mu_d\}^2$, μ_d is dimension mean, μ_{r_a} is mean of region r_a

- Random restarts used to optimise Z_{npq}. Time complexity $\sim O(N^2)$, where N is # data points in training dataset.

EVALUATION PROTOCOL

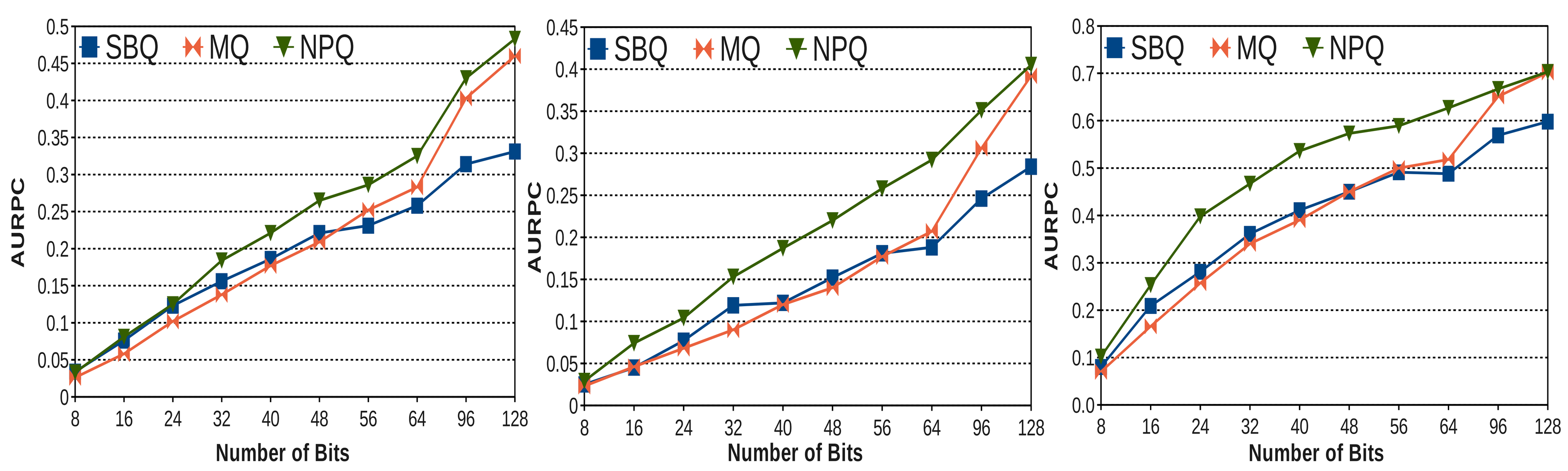
- **Task:** Image retrieval on three image datasets: *22k LabelMe*, *CIFAR-10* and *100k TinyImages*. Images encoded with GIST descriptors.
- **Projections:** LSH, Shift-invariant kernel hashing (SIKH), Iterative Quantisation (ITQ), Spectral Hashing (SH) and PCA-Hashing (PCAH).
- **Baselines:** Single Bit Quantisation (SBQ), Manhattan Hashing (MQ) (Kong et al., '12), Double-Bit quantisation (DBQ) (Kong and Li, '12).
- **Hamming Ranking:** how well do we retrieve $\epsilon-$NN of queries? Quantify using area under the precision-recall curve (AUPRC).

RESULTS

- AUPRC across different projection methods at 32 bits:

Dataset	LabelMe				CIFAR				TinyImages			
	SBQ	MQ	DBQ	**NPQ**	SBQ	MQ	DBQ	**NPQ**	SBQ	MQ	DBQ	**NPQ**
ITQ	0.277	0.354	0.308	**0.408**	0.272	0.235	0.222	**0.407**	0.494	0.428	0.410	**0.660**
SIKH	0.049	0.072	0.077	**0.107**	0.042	0.063	0.047	**0.090**	0.135	0.221	0.182	**0.365**
LSH	0.156	0.138	0.123	**0.184**	0.119	0.093	0.066	**0.153**	0.361	0.340	0.285	**0.464**
SH	0.080	0.221	0.182	**0.250**	0.051	0.135	0.111	**0.167**	0.117	0.237	0.136	**0.356**
PCAH	0.050	0.191	0.156	**0.220**	0.036	0.137	0.107	**0.153**	0.046	0.257	0.295	**0.312**

- NPQ can quantise a wide range of projection functions.
- NPQ + cheap projection (e.g. LSH) can outperform SBQ + expensive projection (e.g. PCA). NPQ is faster for $N <$ data dimensionality.
- AUPRC vs. Number of bits for LabelMe, CIFAR and TinyImages:



(a) LabelMe **(b) CIFAR-10** **(c) TinyImages**

- NPQ is an effective quantisation strategy across a wide bit range.

FUTURE WORK

- Variable bits per hyperplane: *refer to our recent ACL'13 paper*.
- Evaluation of NPQ in a hash lookup based retrieval scenario.