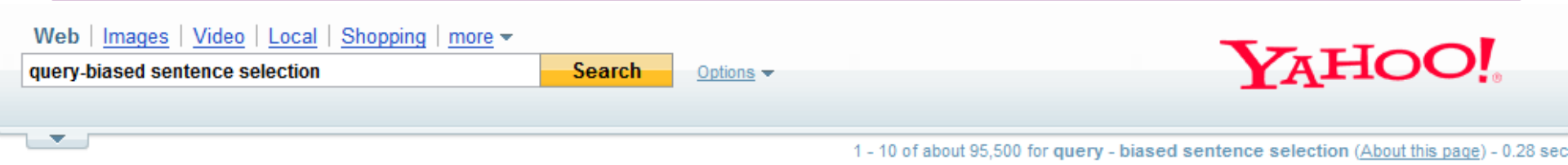
Machine Learned Sentence Selection Strategies for Query-Biased Summarization

Donald Metzler and Tapas Kanungo
Yahoo! Labs

July 24, 2008
SIGIR Learning to Rank Workshop

Query-Biased Summarization



1 - 10 of about 95,500 for **query - biased sentence selection** (About this page) - 0.28 sec

Did you mean: *query-based* sentence selection

Learning **Query-Biased** Web Page Summarization (PDF)
Query-biased Web page summarization is the summarization of a **...** problem is solved within the typical **sentence selection** framework. **...**
mail2.ustc.edu.cn/~wchtiger/index/published papers/cikm076-wang.pdf - 125k - View as html

Part II (PDF)
... select these **sentence** fragments and present **query** terms in the **...** Top-Ranking **sentence selection** architecture. **...** Through combining **query-biased** methods and **...**
research.microsoft.com/.../thesis-files/RyenWhiteThesis-PARTTWO.pdf - 832k - View as html

Advantages of **Query Biased** Summaries in Information Retrieval (PDF)
so called **query biased** (or user directed) summaries: **...** the use of **query biased** summaries **...** condensation of electronic publications by **sentence selection**. **...**
ciir.cs.umass.edu/pubfiles/ir-130.pdf - 154k - View as html

Summarizing Relevant Information for Question-Answering
The problem is addressed as a **query-biased sentence** retrieval task. **... sentence selection** to generate a summary of. approximate 250 words to reflect the information **...**
jenyuan.yeh.googlepages.com/jyyeh-WSEAS-ISA06.pdf - 205k - Cached

Question answering, relevance feedback and summarisation: TREC-9 **...** (PDF)
... report on the effectiveness of **query-biased** summaries for a **... query** words in a **sentence**. **... selection**. Information Processing and Management. 31. 5. pp **...**
trec.nist.gov/pubs/trec9/papers/glasgow_proceedings.pdf - 42k - View as html

Each summary consists of a title, abstract, and URL.

Overview of Query-Biased Summarization

- Input: query, document pair
- Pre-processing Step
  - Segment document into sentences or passages
  - Done offline at index time
- Sentence Selection
  - Identify sentences that are most relevant to the query
  - Return ranked list of sentences + scores
- Construction
  - Compress the sentences to maximize query term coverage, novelty, readability, etc.
  - Must make sure everything fits within screen real estate

Sentence Selection

- This talk focuses entirely on the sentence selection task
- Typically framed as an information retrieval problem
  - Find the sentences within the document that “best match” the query
- Many different features important for sentence selection

Our Approach

- We cast the sentence selection problem as a machine learning problem
- Pros
  - Ability to use many different features
  - Principled parameter estimation
- Cons
  - Need training data
  - Less efficient than rule-based system

Related Work

- Sentence selection
  - Query independent summaries [Kupiec and Pederson '95]
  - Usefulness of query-biased summaries [Tombros and Sanderson '98]
  - TREC Novelty Track / DUC [2001-2004]
  - Machine learned query-biased summaries [Wang et al. '07]
- Learning to rank
  - Logistic regression, SVMs, maximum entropy, perceptrons, ranking SVMs, RankNet, LambdaRank, ordinal regression, gradient boosted decision trees, and on, and on, and on…

Learning Models

- What do we model?
  - Input: query / sentence pair
  - Output: real-valued “relevance” score
- Consider two ‘classes’ of models
  - Pairwise ranking models
  - Regression models

Pairwise Models

- Learn pairwise preferences
  - P is a set of pairwise preferences
  - (s_1, s_2) in P => sentence 1 is preferred to sentence 2
- Shown to be effective for ranking problems
  - [Joachims, KDD '02]
  - [Burges et al., ICML '05]

Encoding Pairwise Preferences

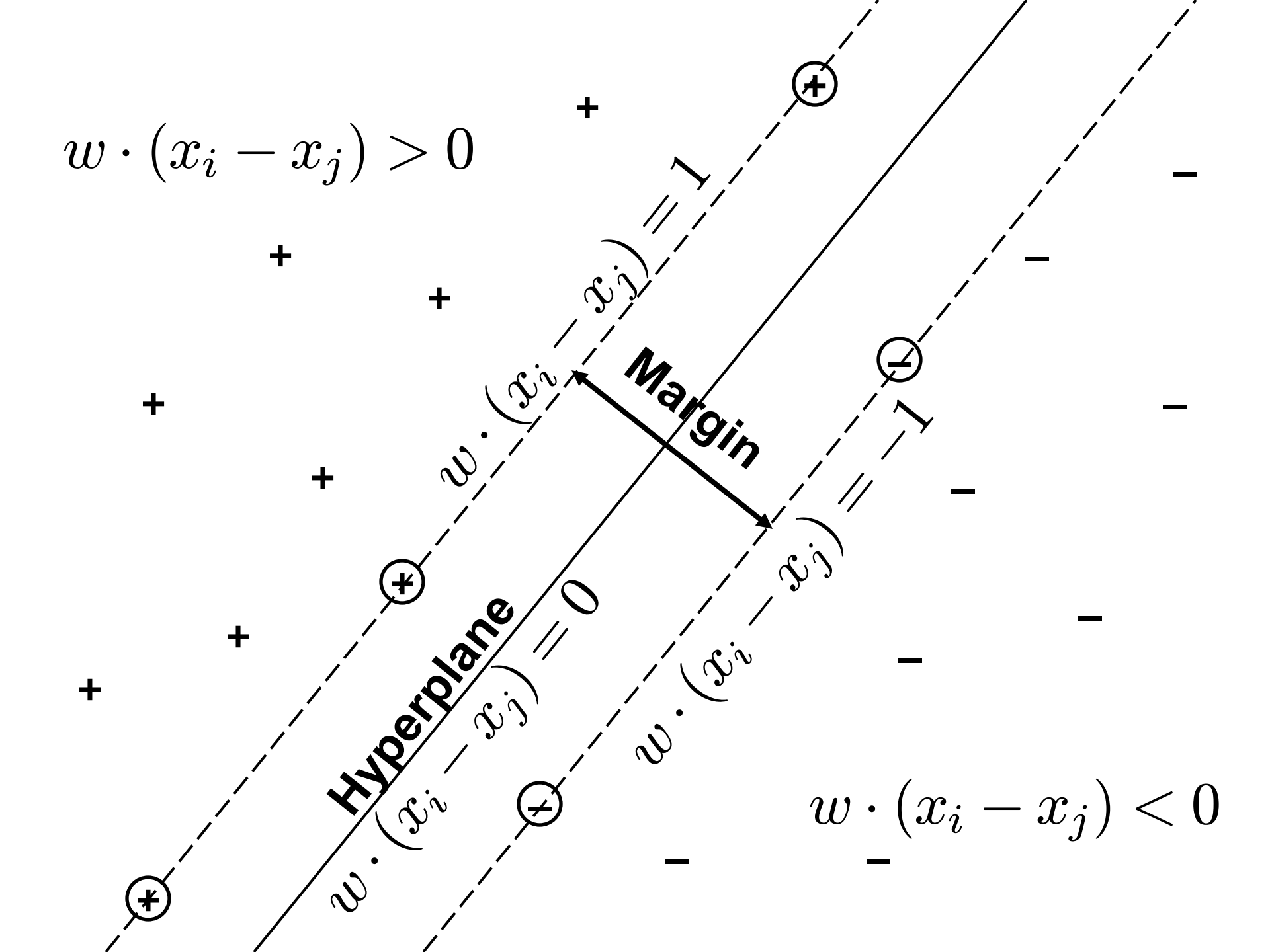
- Given a query / document pair, suppose our training data was the following:
  - Sentence 1 is relevant
  - Sentence 2 is non-relevant
  - Sentence 3 is relevant
  - Sentence 4 is non-relevant
- Encode these judgments as pairwise preferences as follows:
  - $P = \{ (s_1, s_2), (s_1, s_4), (s_3, s_2), (s_3, s_4) \}$
- There are (# relevant)·(# non-relevant) total elements in *P* for each query.

Ranking SVMs

- Ranking SVMs
  - Equivalent to ‘classical’ SVMs learned over pairwise preferences
  - Uses hinge loss
- Formulation:

$$\begin{aligned} \min \quad & \tfrac{1}{2}||w||^2 + C\sum_{i,j}\xi_{i,j} & \\ \text{s.t.} \quad & (w\cdot x_i - w\cdot x_j) \geq 1-\xi_{i,j} & \forall (i,j)\in\mathcal{P} \\ & \xi_{i,j}\geq 0 & \forall (i,j)\in\mathcal{P} \end{aligned}$$

[Joachims ‘02]


$w \cdot (x_i - x_j) > 0$
$w \cdot (x_i - x_j) = 1$
$w \cdot (x_i - x_j) = 0$
$w \cdot (x_i - x_j) = -1$
$w \cdot (x_i - x_j) < 0$
Margin
Hyperplane

Regression Models

- Directly models the response (human judgment)
- Have recently been shown to be highly effective for learning to rank
  - [Li et al., NIPS '07]
  - [Zheng et al., NIPS '07]
- Consider two regression models here
  - Support vector regression
  - Gradient boosted decision trees

Support Vector Regression

- ε-SVM regression
  - ε-sensitive hinge loss over residuals
  - Use different costs for $y = 1$, $y = -1$
  - Constraints: $|y - f(x)| \leq \varepsilon$
- Formulation:

$$\begin{aligned} \min \quad & \tfrac{1}{2}||w||^2 + C_+ \textstyle\sum_{i:y_i=1}(\xi_i + \xi_i^*) + C_- \sum_{i:y_i=-1}(\xi_i + \xi_i^*) \\ \text{s.t.} \quad & y_i - w \cdot x_i - b \leq \epsilon + \xi_i \\ & w \cdot x_i + b - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned}$$

Support Vector Regression

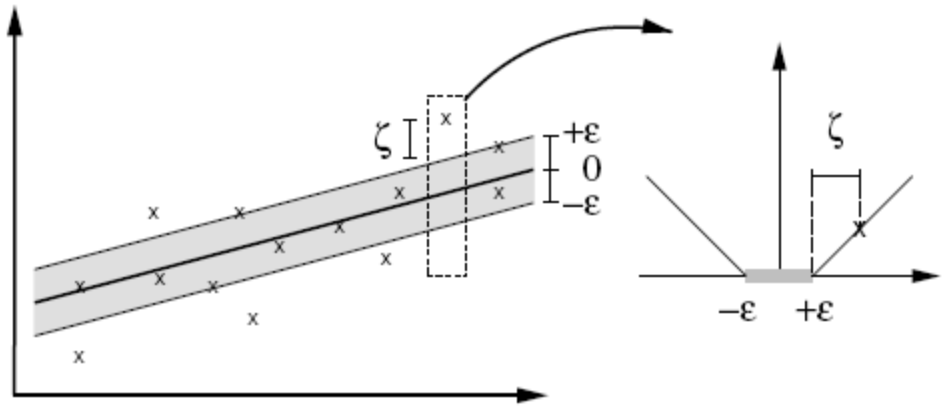


Illustration of ε-SVM regression and loss function from Smola and Schölkopf ’03.

Gradient Boosted Decision Trees

- Additive model of the form:

$$f_M(x) = \sum_{m=1}^{M} T(x; \Theta_m)$$

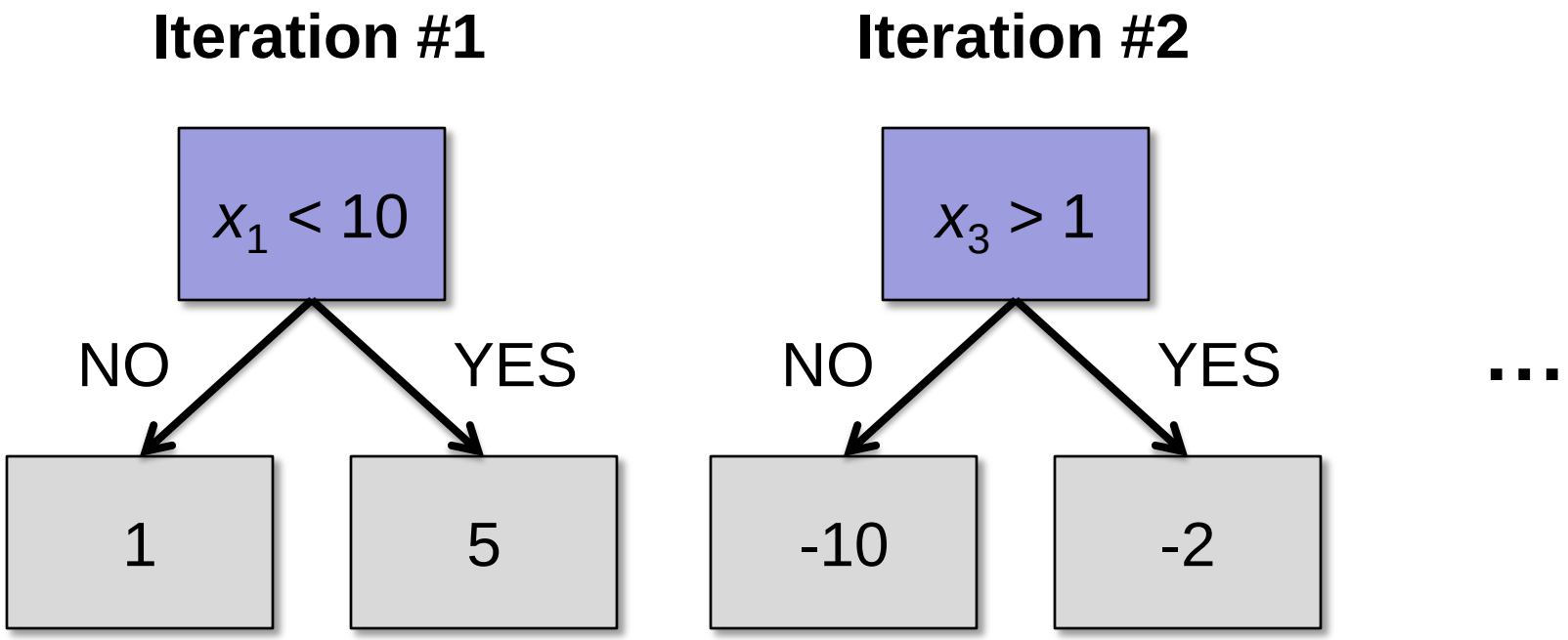
- Where $T(x;\theta)$ is a regression tree with parameters θ
- Adds a new tree to the model during each iteration
- The new tree is fit to the residuals of the loss from the (m-1)th stage

[Friedman '01]

Gradient Boosted Decision Trees

- For regression, the loss is mean squared error
  - Could also use more complex losses
  - Even those that are non-differentiable!
- Trees learned during each iteration are typically ‘stumps’ (trees of depth 1)
- Automatic feature selection built in

Gradient Boosted Decision Trees



$$f(x_1 = 5, x_2 = 3, x_3 = 3) = 1 + (-2) + \ldots$$

$$f(x_1 = 20, x_2 = 0, x_3 = 5) = 5 + (-2) + \ldots$$

Loss Functions

- Loss functions for our learning models
  - RankSVMs (hinge loss over pairs of scores)
  - SVR (hinge loss over residuals)
  - GBDTs (MSE)
- Our ultimate goal is to maximize some information retrieval metric, such as F1 or R-Precision
  - However, such metrics are non-differentiable and difficult to optimize directly
  - None of the learning methods use actually optimize these metrics
- We use a heuristic training procedure in order to implicitly maximize the metrics of interest

Training / Evaluation Algorithm

Algorithm 1 Evaluation Algorithm

for $i = 1$ to 5 **do**
  $(TRAIN, VALIDATE) \leftarrow split(TRAIN_i, p)$
  $utility_{max} \leftarrow -\infty$
  for $\theta \in \Theta$ **do**
    $model \leftarrow train(TRAIN; \theta)$
    $utility \leftarrow eval(model, VALIDATE)$
    if $utility > utility_{max}$ **then**
      $model_{max} \leftarrow model$
    end if
  end for
  output $rank(TEST_i, model_{max})$
end for

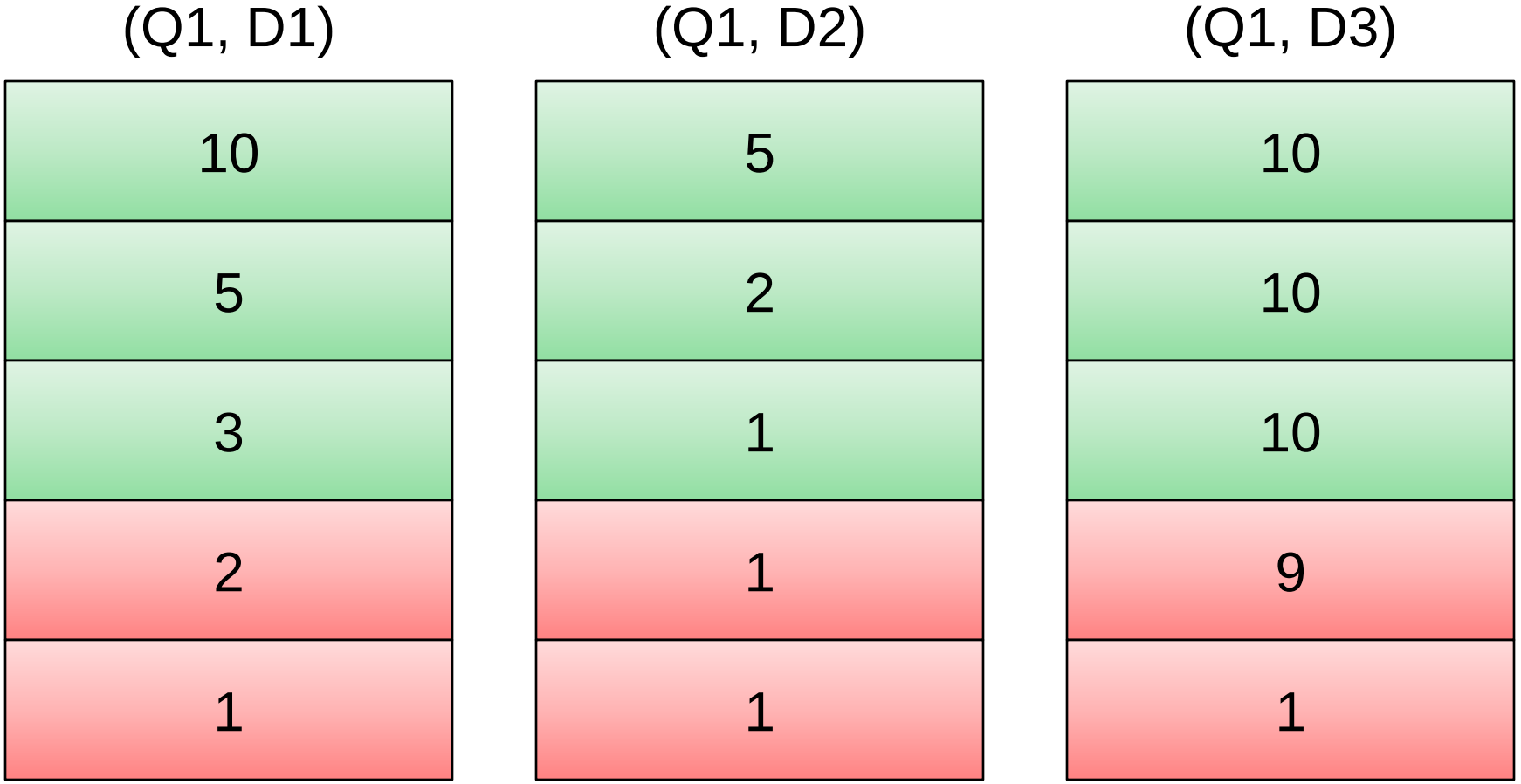
Features

- Query dependent
  - Exact match
  - Overlap
  - Overlap w/ synonyms
  - Language modeling score
- Query independent
  - Sentence length
  - Sentence location

Sentence Filtering

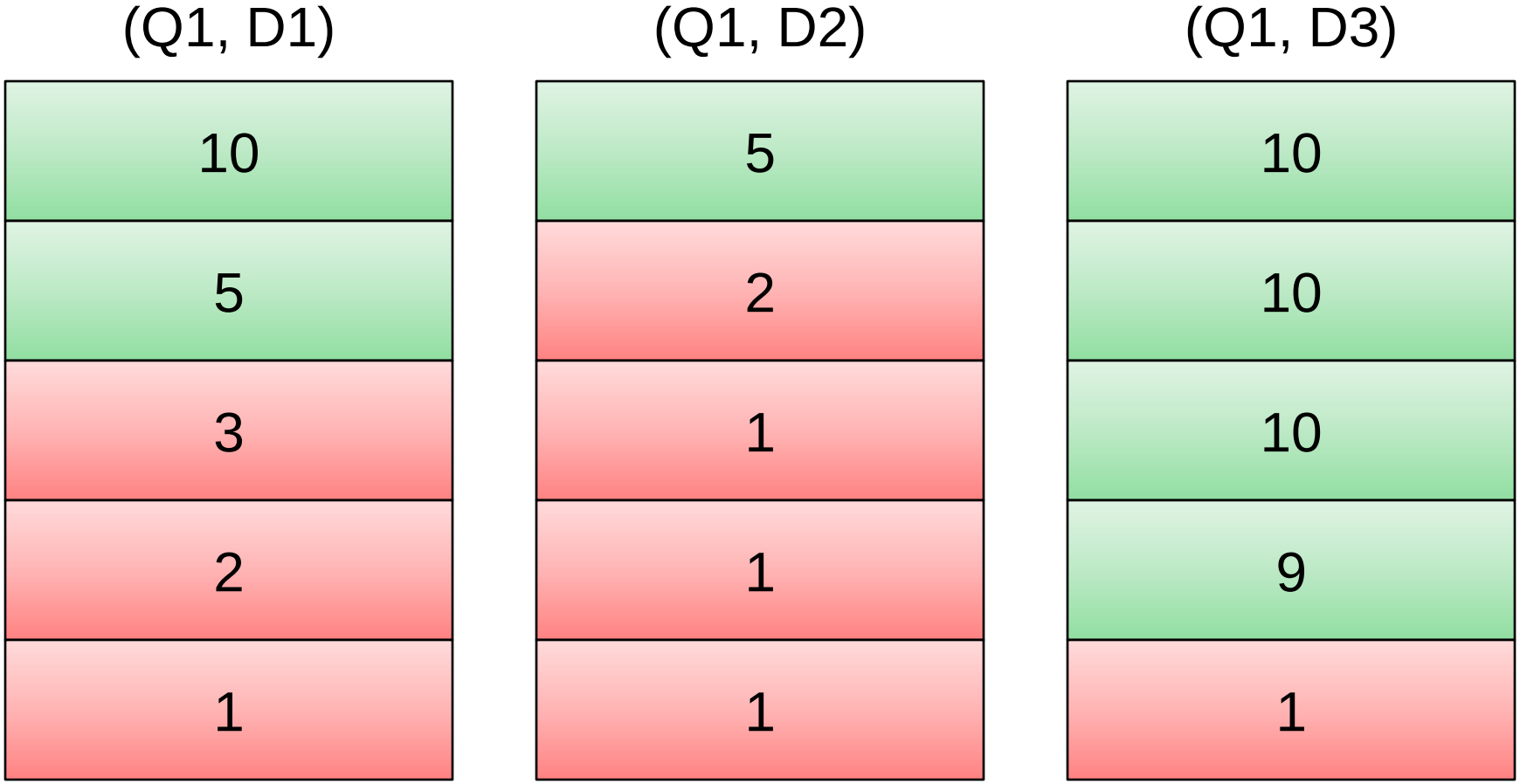
- Scores produced by ML models can be used to rank sentences
- In practice, want to filter result set to include only the most relevant documents
- How can we choose the number of sentences to return to the abstract composer?
- Fixed depth
  - Choose the same number of sentences per document
- Global score threshold
  - Only choose those sentences with a score greater than some global threshold

Fixed Depth Filtering (Depth = 3)



Green = Retrieved, **Red** = Not Retrieved

Global Score Threshold Filtering (Threshold = 5)



Green = Retrieved, **Red** = Not Retrieved

Evaluation

- Data
  - TREC Novelty track data from 2002-2004
- Human judgments
  - For every query / document pair, all of the sentences in the document are judged to be relevant or non-relevant to the query
- Data sets have differing characteristics

	N2002	N2003	N2004
Query / Doc. Pairs	597	1187	1214
Avg. Sentences per Pair	52.1	31.9	30.5
Avg. Relevant Sentences per Pair	2.3	13.1	6.9

Sentence Selection Evaluation

	N2002	N2003	N2004
LM	.2602	.5566	.3944
Ranking SVM	$.3792^{\alpha}$	$.6904^{\alpha}$	$.4771^{\alpha}$
SVR	$.3587^{\alpha}$	$.7005^{\alpha\beta}$	$.4757^{\alpha}$
GBDT	$.4047^{\alpha\beta\delta}$	$.7060^{\alpha\beta}$	$.4806^{\alpha}$

R-Precision for each data set and sentence selection approach. The α, β, and δ subscripts indicate a statistically significant improvement over language modeling, ranking SVMs, and SVR, respectively, according to a one-tailed pair *t*-test with $p < 0.05$.

Summary:

Machine learned techniques always significantly better than language modeling. GBDTs significantly outperform ranking SVMs on two out of three data sets, and SVR on one data set.

Sentence Filtering Evaluation

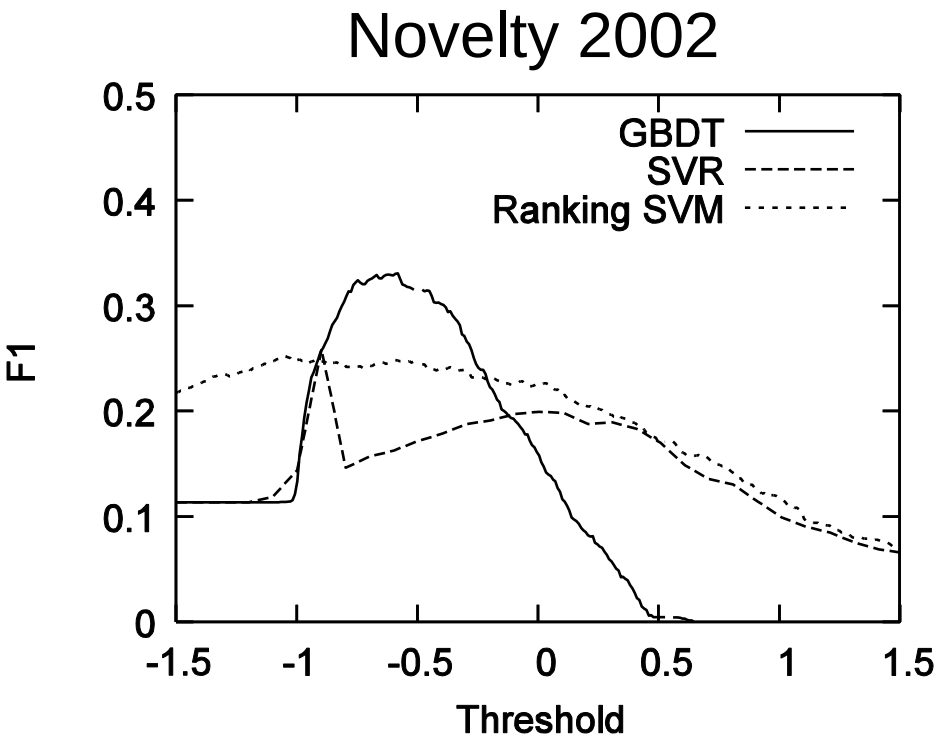
	N2002		N2003		N2004	
	Depth	Thresh.	Depth	Thresh.	Depth	Thresh.
Ranking SVM	.3411	.2474	.5794	.6330	.4416	.4736
SVR	.3350	.2880	.5791	.6503	.4407	.4637
GBDT	.3576	.3302	.5771	.6691	.4389	.4745

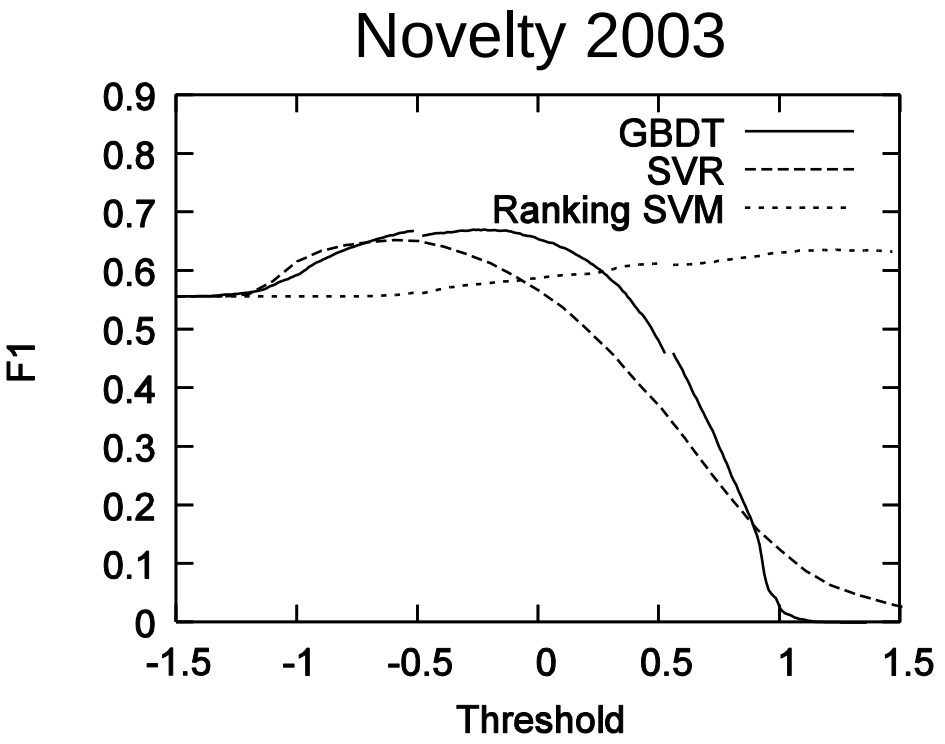
Comparison of result set filtering methods. For each data set, the optimal F1 measure for each technique is reported.

Summary:

Fixed depth thresholding better when there are very few relevant sentences per document and global score thresholding better when there are many relevant sentences per document.

Global Score Thresholding

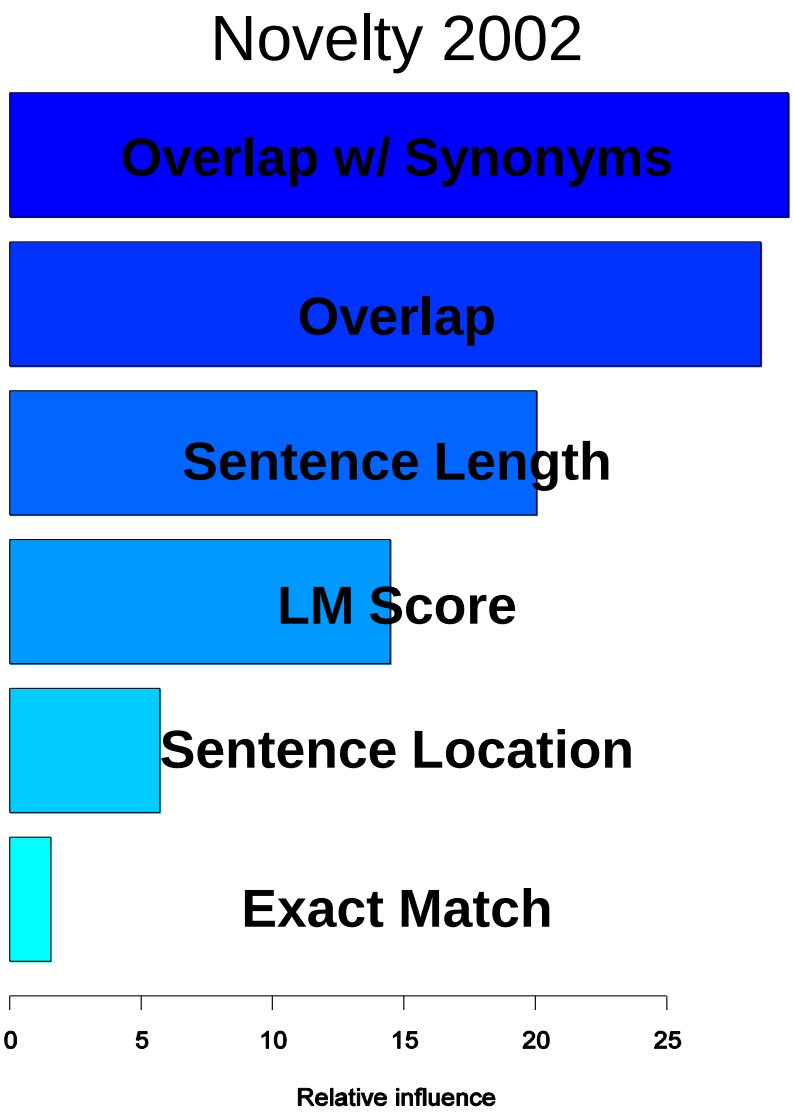


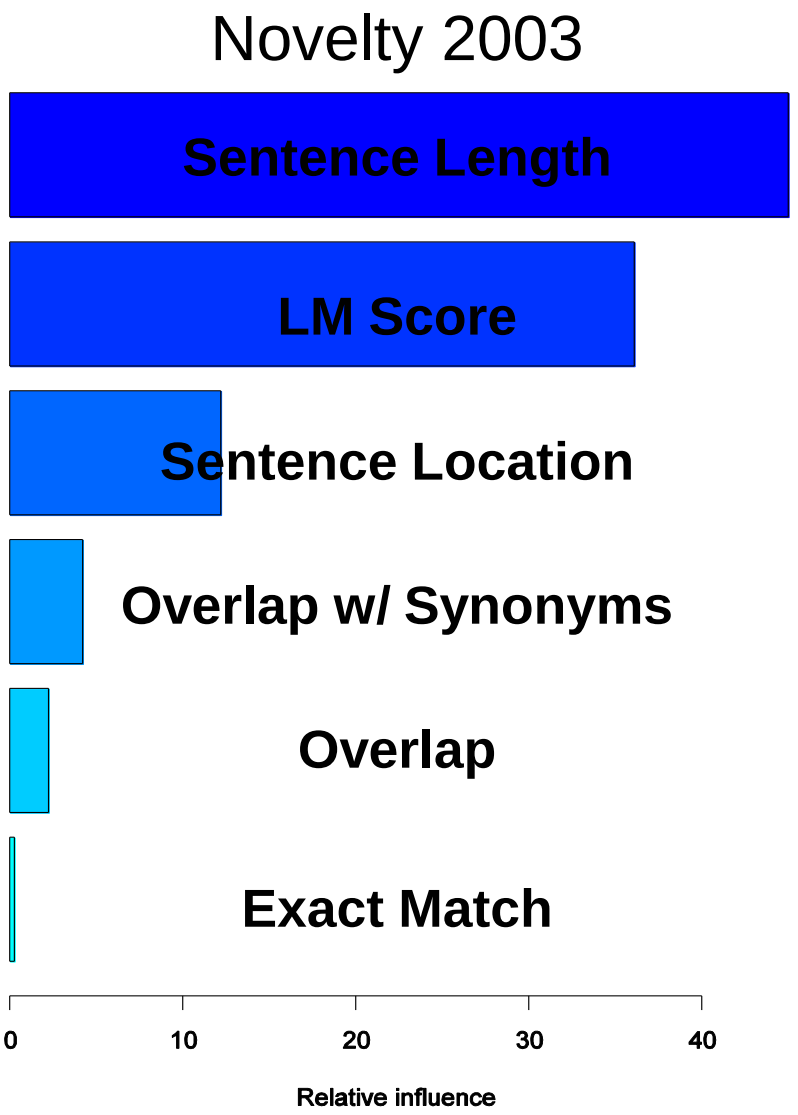


Summary:

Global score thresholding with GBDTs is more stable across data sets than ranking SVMs and SVR. Setting the GBDT threshold to -0.55 results in an F1 that is within 2% of the optimal F1 on all three data sets.

Feature Importance Analysis





Conclusions

- Machine learned sentence selection is not only feasible, but very effective
- Regression-based model, and gradient boosted decision trees, in particular, are a robust model choice
- Different result set filtering techniques are appropriate for different types of data sets