# Methods of Reducing Verbose Queries

Martha Enderby

University of Minnesota, Morris

November 15, 2011

# What Are Verbose Queries?



"Explain some methods of reducing verbose queries into keyword-focused queries"

- Long natural language search queries
- "Wh-" queries: "What are some methods of reducing verbose queries?"
- "terms" are single words ("reduce") or a small group of connected words ("University of Minnesota")

# Why Is Reduction Important?



- Many words in verbose queries are not useful

# Why Is Reduction Important?



- Many words in verbose queries are not useful
- Perfect reduction can improve search performance by 30% [2]

# Why Is Reduction Important?



- Many words in verbose queries are not useful
- Perfect reduction can improve search performance by 30% [2]
- Around 10% of search queries are verbose [1]

# Reduction Methods

## Weighting

# Reduction Methods



**Weighting**

Explain some methods of reducing verbose queries into keyword-focused queries

# Reduction Methods



**Weighting**

Explain some methods of reducing verbose queries into keyword-focused queries

**Elimination**

# Reduction Methods



**Weighting**

Explain some methods of reducing verbose queries into keyword-focused queries

**Elimination**

~~Explain some~~ methods ~~of~~ reducing verbose queries ~~into~~ ~~keyword-focused queries~~

# Collections and Training



Text REtrieval Conference (TREC): An ongoing series of workshops about information retrieval.

# Collections and Training



Text REtrieval Conference (TREC): An ongoing series of workshops about information retrieval.
TREC documents consist of a title, summary, and document

# Collections and Training



Text REtrieval Conference (TREC): An ongoing series of workshops about information retrieval.
TREC documents consist of a title, summary, and document

- Wt10g - web archive, 1.7M documents

# Collections and Training



Text REtrieval Conference (TREC): An ongoing series of workshops about information retrieval.
TREC documents consist of a title, summary, and document

- Wt10g - web archive, 1.7M documents
- Robust2004 - workshop, 500K documents

# Collections and Training



Text REtrieval Conference (TREC): An ongoing series of workshops about information retrieval.
TREC documents consist of a title, summary, and document

- Wt10g - web archive, 1.7M documents
- Robust2004 - workshop, 500K documents
- Gov2 - web archive, 25.2M documents

# Collections and Training



Text REtrieval Conference (TREC): An ongoing series of workshops about information retrieval.
TREC documents consist of a title, summary, and document

- Wt10g - web archive, 1.7M documents
- Robust2004 - workshop, 500K documents
- Gov2 - web archive, 25.2M documents
- TREC123 - TREC proceedings, 150 documents

# Collections and Training



Text REtrieval Conference (TREC): An ongoing series of workshops about information retrieval.
TREC documents consist of a title, summary, and document

- Wt10g - web archive, 1.7M documents
- Robust2004 - workshop, 500K documents
- Gov2 - web archive, 25.2M documents
- TREC123 - TREC proceedings, 150 documents

Training: teaching a program to evolve based on data, in this case the search performance a sub-query
All methods discussed here were trained with RankSVM, a pairwise learning-to-rank algorithm.

# Dependency Parsing (2010)



- Weighting

# Dependency Parsing (2010)



- Weighting
- Developed by Jae-Hyun Park and W. Bruce Croft from the Center for Intelligent Information Retrieval

# Dependency Parsing (2010)



- Weighting
- Developed by Jae-Hyun Park and W. Bruce Croft from the Center for Intelligent Information Retrieval
- Based on dependencies between words

# Dependency Parsing (2010)



- Weighting
- Developed by Jae-Hyun Park and W. Bruce Croft from the Center for Intelligent Information Retrieval
- Based on dependencies between words
- Utilize dependency parsing trees

# Parse Trees



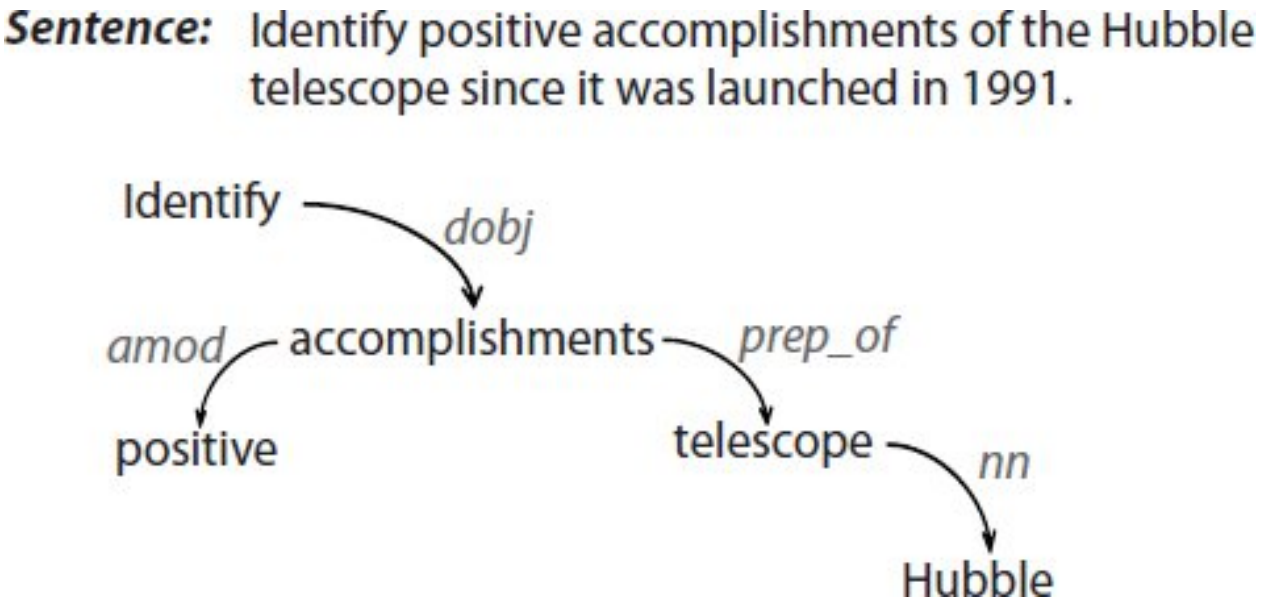


Figure: from [3]

dobj = direct object
amod = modifying adjective
prep_of = the preposition "of"
nn = noun

# Ranking Terms



Data comes from parse trees, term ranking yields labels.
Ranking equation for a term $t$:

$$E(t) = \frac{1}{N_m} \cdot \sum_{c \epsilon C_m} (\varphi(c,t) - \varphi(c))$$

$m$ = number of terms in a query, excluding $t$
$C_m$ = all possible combinations of $m$ terms
$c$ = a combination in $C_m$
$N_m$ = number of terms in $C_m$
$\varphi(c)$ = search performance of $c$
$\varphi(c,t)$ = search performance of $c$ and $t$ together

# Query Quality Predictors (2009)



- Elimination

# Query Quality Predictors (2009)



- Elimination
- Developed by Giridhar Kumaran and Vitor R. Carvalho from Microsoft

# Query Quality Predictors (2009)



- Elimination
- Developed by Giridhar Kumaran and Vitor R. Carvalho from Microsoft
- Depend on the collection of documents

# Query Quality Predictors (2009)



- Elimination
- Developed by Giridhar Kumaran and Vitor R. Carvalho from Microsoft
- Depend on the collection of documents
- Attempts to find the single best subquery

# Query Quality Predictors (2009)



- Elimination
- Developed by Giridhar Kumaran and Vitor R. Carvalho from Microsoft
- Depend on the collection of documents
- Attempts to find the single best subquery
- Query quality predictors (QQPs) are measurable heuristic properties of a query

# Query Quality Predictors (2009)



- Elimination
- Developed by Giridhar Kumaran and Vitor R. Carvalho from Microsoft
- Depend on the collection of documents
- Attempts to find the single best subquery
- Query quality predictors (QQPs) are measurable heuristic properties of a query
- QQPs can be pre-retrieval or post-retrieval

# Query Quality Predictors (2009)



- Elimination
- Developed by Giridhar Kumaran and Vitor R. Carvalho from Microsoft
- Depend on the collection of documents
- Attempts to find the single best subquery
- Query quality predictors (QQPs) are measurable heuristic properties of a query
- QQPs can be pre-retrieval or post-retrieval
- QQPs are also called “features”

# Some Query Quality Predictors



**Query Quality Predictors**

| QQP Name | Description |
|---|---|
| Mutual Information | Dependency between terms. High MI indicates closely-related terms. |
| Sub-Query Length | Number of terms in a sub-query. Optimally between 3 and 6. |
| Inverse Document Frequency | Relative rarity of a term within a collection. High IDF indicates a term is rare enough to be worth searching for. |

# More Query Quality Predictors



**Query Quality Predictors**

| QQP Name | Description |
|---|---|
| Query Clarity | Post-retrieval divergence between returned documents and the collection as a whole. High QC indicates specificity. |
| Simplified Clarity Score | Less-expensive version of query clarity. |
| Similarity Collection/Query | Similarity of query to collection. High SCQ indicates high similarity. |

# Query Quality Predictor Comparisons



**Most Important QQPs by Collection**

| Rank | TREC123 | Robust2004 |
|---|---|---|
| 1 | Clarity | Clarity |
| 2 | $IDF_{max}/IDF_{min}$ | MI |
| 3 | Total IDF | SCQ |

- Query Clarity and Simplified Clarity Score were the most useful QQPs
- Other QQPs varied in usefulness

# Subset Distribution (2010)



- Elimination

# Subset Distribution (2010)



- Elimination
- Developed by Xiaobing Xue, Samuel Huston and W. Bruce Croft from the Center for Intelligent Information Retrieval

# Subset Distribution (2010)



- Elimination
- Developed by Xiaobing Xue, Samuel Huston and W. Bruce Croft from the Center for Intelligent Information Retrieval
- Average performance of all sub-queries between 3-6 terms

# Subset Distribution (2010)



- Elimination
- Developed by Xiaobing Xue, Samuel Huston and W. Bruce Croft from the Center for Intelligent Information Retrieval
- Average performance of all sub-queries between 3-6 terms
- Also uses heuristic features

# Subset Distribution (2010)



- Elimination
- Developed by Xiaobing Xue, Samuel Huston and W. Bruce Croft from the Center for Intelligent Information Retrieval
- Average performance of all sub-queries between 3-6 terms
- Also uses heuristic features
- Uses retrieval models, which predict what a user will find relevant

# Features

**Independency Features**: look at a single word.

# Features

**Independency Features**: look at a single word.
Example: single-word frequency

# Features

**Independency Features**: look at a single word.
Example: single-word frequency

**Local Dependency Features**: look at the relationships between query words

# Features

**Independency Features**: look at a single word.
Example: single-word frequency

**Local Dependency Features**: look at the relationships between query words
Example: An arc in a dependency parsing tree

# Features

**Independency Features**: look at a single word.
Example: single-word frequency

**Local Dependency Features**: look at the relationships between query words
Example: An arc in a dependency parsing tree

**Global Dependency Features**: look at all the words in a sub-query

# Features

**Independency Features**: look at a single word.
Example: single-word frequency

**Local Dependency Features**: look at the relationships between query words
Example: An arc in a dependency parsing tree

**Global Dependency Features**: look at all the words in a sub-query
Example: Query length

# Retrieval Models

**Query Likelihood Model (QL)**: The probability that a document contains a given query.

**Sequential Dependency Model (DM)**: The probability that two adjacent terms in a query are related.

This method was trained using these models on both the original verbose query and on generated sub-queries. "Sub-" indicates that the model was used on sub-queries.
Models used: QL, DM, SubQL, SubDM, QL+SubQL, DM+SubQL

# CRF-perf

A conditional random field (CRF) labels and segments data.

# CRF-perf

A conditional random field (CRF) labels and segments data. Used to generate $P(\mathbf{y}|\mathbf{x})$ where $\mathbf{x}$ is a sequence of words and $\mathbf{y}$ a sequence of labels. Here, $y$ can be 0 or 1.

# CRF-perf

A conditional random field (CRF) labels and segments data. Used to generate $P(\mathbf{y}|\mathbf{x})$ where $\mathbf{x}$ is a sequence of words and $\mathbf{y}$ a sequence of labels. Here, $y$ can be 0 or 1.
CRF-perf is a type of CRF intended to optimize performance. It can be used without knowledge of the "gold standard" sub-query.

# CRF-perf Equation



$$P_M(y|x) = \frac{\exp(\sum_{k=1}^{K} \lambda_k f_k(\mathbf{x}, \mathbf{y}))m(Q_s, M)}{Z_m(\mathbf{x})}$$

$$Z_M(x) = \sum_{y} \exp(\sum_{k=1}^{K} \lambda_k f_k(\mathbf{x}, \mathbf{y}))m(Q_s, M)$$

$Q_s$ = a sub-query
$\mathbf{x}$ = set of words in $Q_s$
$\mathbf{y}$ = set of labels for $\mathbf{x}$
$M$ = a retrieval method such as subQL
$m(Q_s, M)$ = the search performance of $Q_s$ using $M$
$K$ = the number of features $Q_s$ has
$f_k$ = a specific feature
$\lambda_k$ = the weight of $f_k$

# Retrieval Method Comparisons



**Most Useful Retrieval Methods by Collection**

| Rank | Robust2004 | Wt10g | Gov2 |
|---|---|---|---|
| 1 | DM+SubQL | DM+SubQL | DM+SubQL |
| 2 | SubDM | DM | SubDM |
| 3 | DM | SubDM | DM |

- Combining DM on the original query with QL on the subquery works best
- The Sequential Dependency Model is extremely useful for improving query quality

# Results Summary



**Improvement Over Unreduced Verbose Queries**

| Method | **Robust2004** | **Wt10g** | **Gov2** | **TREC123** |
|---|---|---|---|---|
| Dependency Parsing | 8.9% | 9.3% | | |
| Query Quality Predictors | 10.0% | | | 6.8% |
| Subset Distribution | 11.7% | 19.1% | 13.6% | |

# Conclusions



- Subset Distribution is the strongest of the three
- None of these methods yield perfect reductions

# Acknowledgments

Thank you to Elena Machkasova who was my advisor for this project, and to Elijah Mayfield for his proofreading feedback.

# References

## References

1. Bendersky, Michael and Croft, W. Bruce: Discovering Key Concepts in Verbose Queries, 2008
2. Kumaran, Giridhar and Vitor R. Carvalho: Reducing Long Queries Using Query Quality Predictors, 2009.
3. Park, Jae-Hyun and Croft, W. Bruce: Query Term Ranking based on Dependency Parsing of Verbose Queries, 2010.
4. Xue, Xiaobing; Huston, Samuel; and Croft, W. Bruce: Improving Verbose Queries Using Subset Distribution, 2010.