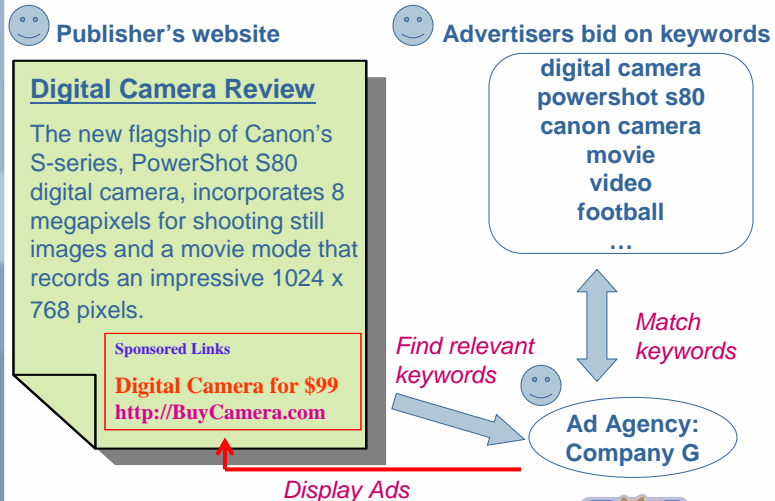


Finding Advertising Keywords on Web Pages

Scott Wen-tau Yih Joshua Goodman
Microsoft Research

Vitor R. Carvalho
Carnegie Mellon University

Contextual Ads 101



- Google's AdSense program
– More than 40% of its revenues



Keyword Extraction is the Key!!

- Company M wants to copy this business model...

Publisher's website

Chinese Restaurant Review

...

Yen Ching's menu is of daunting length and enormous breadth. For example, a lot of vegetarians like their Braised Fungus and Winter Bamboo Shoots, while others love the special Stewed Duck and Iron Plate Beef.

...

Sponsored Links

Eliminate Nail Fungus

<http://pharmacy.com/nail-fungus>

- Keywords extracted are more relevant
 - More useful and interesting to readers
 - Higher click-through rate, more revenue

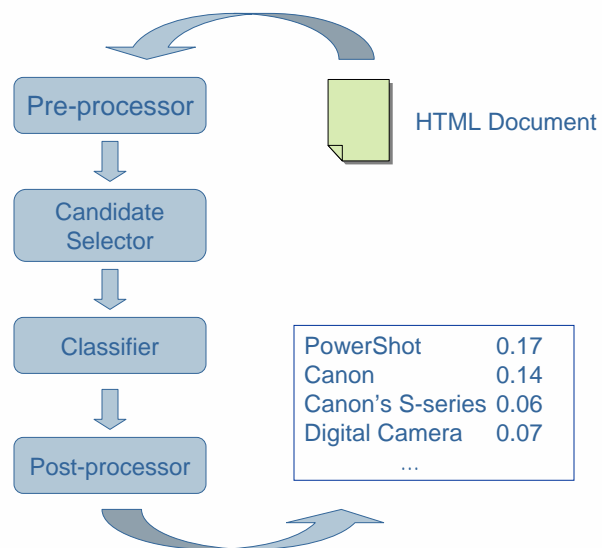
Introduction

- A machine learning based system
 - Significantly better than simple TF×IDF baseline
 - Better than an existing system, KEA
- Explore different frameworks of choosing keyword candidates
 - Phrases vs. Words
 - Looking at whole phrases monolithically is better
 - Combined vs. Separate
 - Will show that looking at all instances of a phrase together (combined) is better
- Extensive feature study
 - TF and DF
 - Instead of TF×IDF, use them as separate features
 - Search Query Log
 - Keywords that people use to query are good features to find keywords people like

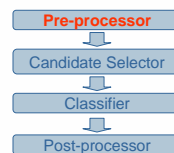
Outline

- System Architecture
 - Preprocessor
 - Candidate selector
 - Classifier
 - Postprocessor
- Experiments
 - Data preparation
 - Performance measures
 - Results
- Related Work

System Architecture



Pre-processor



- Facilitate keyword candidate selection and feature extraction
- Transform HTML documents into sentence-split plain-text documents
 - No sophisticated parsing
 - No block detection
 - Preserve/Augment some information
 - Some HTML tags
 - Linguistic analysis: POS tagging

Candidate Selector Monolithic (1/2)



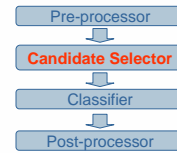
- Consider consecutive words up to length 5 as candidates
- Candidates do not cross sentence boundaries

Digital Camera Review

The new flagship of Canon's S-series, PowerShot S80 digital camera, incorporates 8 megapixels for shooting still images and a movie mode that records an impressive 1024 x 768 pixels.

- Some candidates
 - “The”, “The new”, “The new flagship”, “The new flagship of”, “The new flagship of Canon”, “new”, “new flagship”, ...

Candidate Selector Monolithic (2/2)

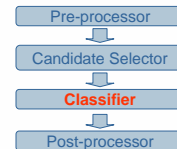


- Combined vs. Separate
 - Information extraction community usually looks at candidate phrases separately, while previous work in this area has combined all instances together

Digital Camera Review

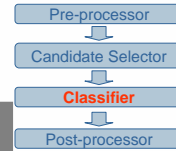
The new flagship of Canon's S-series, PowerShot S80 **digital camera**, incorporates 8 megapixels for shooting still images and a movie mode that records an impressive 1024 x 768 pixels.

Classifier



- Once we have candidates, must determine which ones are the best
- Two steps:
 - For each phrase, extract its “features”
 - Indications of whether a candidate phrase is relevant to the document
 - Use both binary and real-valued features
 - From features, determine score of the phrase
 - Learn the weights of features

Important Features

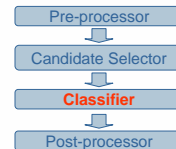


Digital Camera Review

The new flagship of Canon's S-series, PowerShot S80 digital camera, incorporates 8 megapixels for shooting still images and a movie mode that records an impressive 1024 x 768 pixels.

- Term Frequency & Document Frequency (IR features)
- Search Query Log
 - Most frequent 7.5 million query terms from MSN search
 - Whether the phrase is in the query log, as well as the frequency
- Whether the phrase appears in <TITLE>
- Sentence Length (where the phrase is in)
- Capitalization (whether the phrase is capitalized)
- Location (relative to the whole document and sentence)
- Linguistics (noun or proper noun)
- MetaSec (keywords, description, etc)

Logistic Regression

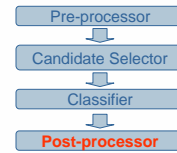


- Need to combine the features to get a score for each phrase
- For each feature, compute a weight
 - For a given phrase, find weighted sum of features, add them up
- Need to find the weights
 - Use training data (more later) with list of “correct” keyphrases for each document
 - Use “logistic regression” to find best weights

$$p(y | \bar{x}) = \frac{\exp(\bar{x} \cdot \bar{w}_i)}{1 + \exp(\bar{x} \cdot \bar{w}_i)}$$

- y is 1 if word/phrase is relevant
- x is the features of the word/phrase (a vector of numbers)
- Learning: find weights that match the labeled training data

Post-processor



- Monolithic Combined
 - (Consider identical phrases as one candidate)
 - Direct output what classifier predicts
- Monolithic Separate
 - Output the largest probability estimation of identical candidates

Experiments

- How do we collect data to train and evaluate our system?
- How *good* is our system?
 - How to measure performance
 - Which framework is the best?
 - Compare it with other systems
- Feature contribution

Data Annotation

- Raw data: 828 web pages
 - Have content-targeted advertising
 - Remove advertisements
- 5 annotators pick keywords
 - Asked them to choose only words/phrases that occurred in the documents
 - Asked them to label phrases about “things they might want to buy when reading this page”
- 10-fold cross validation for experiments

Performance Measures

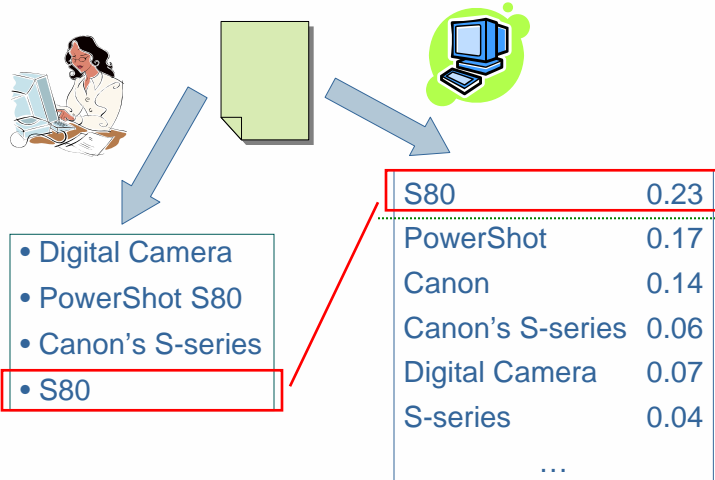
- Accuracy or Recall is not very meaningful
 - Hard to define/pick a complete set of keywords
 - Rank of keywords is also important
- Top- n scores
 - We return our top n phrases
 - Get 1 point for each correct phrase we return
 - (Annotator listed that keyphrase)
 - Divide by maximum points any system could possibly get
 - Score is between 0 and 1 (1 is best)

– K_i : set of top n keywords chosen by the system for page i

– A_i : keywords selected by the annotators for page i

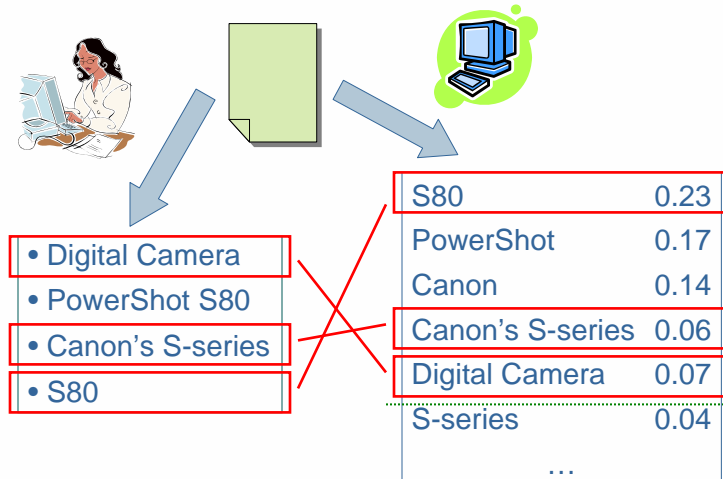
– Score =
$$\frac{\sum_i |K_i \cap A_i|}{\sum_i \min(|A_i|, n)} \times 100\%$$

Top-*n* Score for 1 Document



Top-1 score? $1/1 = 1.0$

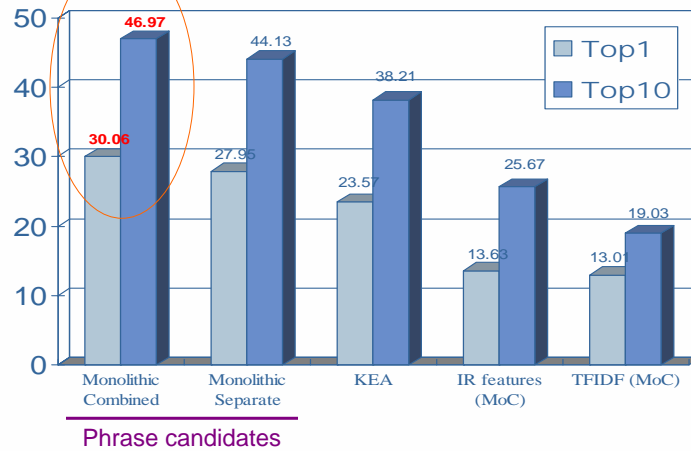
Top-*n* Score for 1 Document



Top-5 score? $3/4 = 0.75$

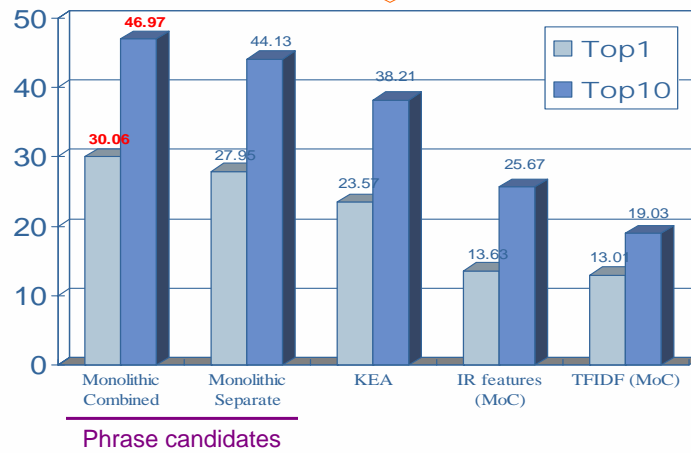
Performance Comparison

Combining identical phrases as candidates is the best framework



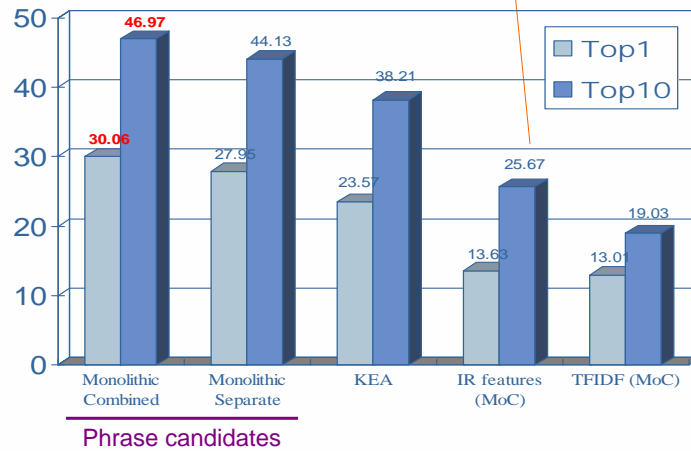
Performance Comparison

Better than KEA

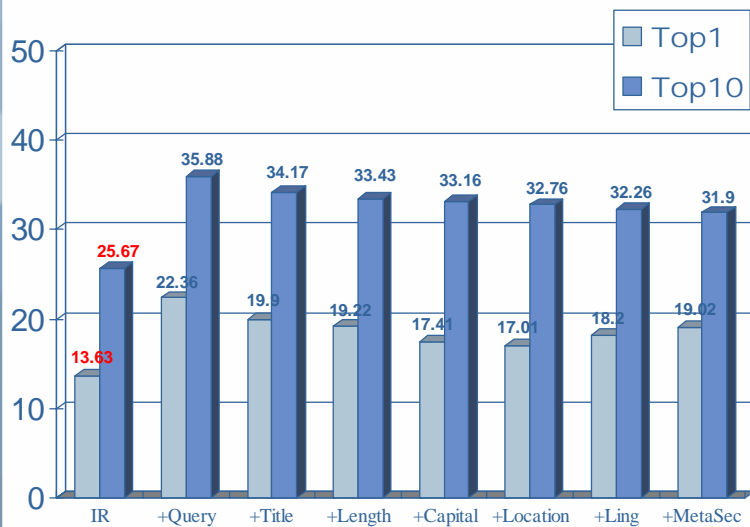


Performance Comparison

Learning weights for TF and DF separately is better than TF×IDF



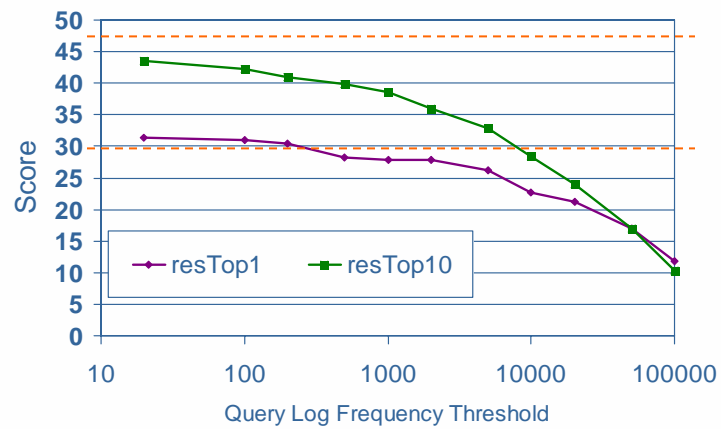
IR + One Set of Features



Search Engine Query Log

- 2nd useful feature
- Size could be too large especially for client-side applications
 - 7.5 million queries, 20 bytes per query
 - 20 languages
 - 3GB query log files
- Effects of Using a smaller query log file
- Restrict candidates by query log

Using Different Sizes of Query Log File



Related Work

- Exciting field: Researchers tend to be rich!
- Extracting keywords (from scientific papers)
 - GenEx: rules + GA [Turney IR-00]
 - KEA: Naïve Bayes using 3 features [Frank et al. IJCAI-99]
 - *Craig Nevill-Manning, Engineering Director, Google NYC*
- Query-Free News Search [Henzinger, et al. WWW-03]
 - Extract keywords from TV news caption
 - Using TF×IDF and its variations to score phrases
 - *Sergey Brin, 1 of the 2 billionaires who published in WWW*
- Impedance coupling [Ribeiro-Neto et al. SIGIR-05]
 - Match advertisements to web pages directly
 - *Berthier Ribeiro-Neto, Google Latin America R&D Center*
- Implicit Queries from Emails [Goodman&Carvalho CEAS-05]
 - *Joshua Goodman, Poor Researcher, Microsoft Research*

Conclusions

- Keyword extraction drives content-targeted advertising
 - Foundation of free web services
 - Very successful business model
- Extensive experimental study
 - TF, DF, Search Query Log are the three most useful features
 - Machine learning is important in tuning the weights
 - Monolithic combined (combine identical phrases together) is the best approach
- Our system is substantially better than KEA – the only publicly available keyword extraction system
 - Just a start; hope to see more papers