"Without the Clutter of Unimportant Words": Modeling How People Choose Descriptive Keyphrases

Jason Chuang, Jeffrey Heer, Christopher D. Manning
Stanford University
{jcchuang, jheer, manning}@cs.stanford.edu

ABSTRACT

Keyphrases aid exploration of text collections by communicating salient aspects of documents and are often critical for creating effective visualizations of text. In this paper, we investigate the statistical and linguistic properties of keyphrases chosen by human judges and propose an improved method for automatic keyphrase extraction. Based on 5,611 responses from 69 graduate students describing a corpus of dissertation abstracts, we identify characteristics of human-generated keyphrases, including phrase length, commonness, position, and part of speech. Next, we systematically assess the contribution of each feature within a statistical model of keyphrase quality. We evaluate our resulting keyphrase extraction algorithm through crowdsourced ratings of keyphrase quality and a comparative analysis of automatically and manually selected phrases. We find that our technique generates keyphrases that human judges prefer to other automatic techniques, and whose precision and recall can match that of manually selected terms.

Author Keywords

Automatic keyphrase extraction, text collections, human judges, visualization, linguistic properties[1]

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: UI;
I.2.7 Artificial Intelligence: Natural Language Processing

INTRODUCTION

Document collections, from academic publications and legal decisions to blog feeds, provide rich sources of information. People explore these collections to better understand their contents, uncover patterns, and find documents matching an information need. Keywords (or *keyphrases*) aid exploration by providing summary information and quickly communicating salient aspects of one or more documents. Keyphrase selection is also critical to the effectiveness of text visualization, as when choosing salient terms for tag clouds [5, 28, 29] or labeling documents, clusters, or themes [11, 12]. While terms hand-selected by people are considered the gold standard, manually assigning keyphrases to thousands of documents simply does not scale.

To aid exploration of text collections, keyphrase extraction algorithms automatically generate descriptive phrases from text. Keyphrases are often extracted using bag-of-words frequency statistics [16, 21, 22, 23, 24]. However, such measures may not be suitable for summarizing short texts [2] and typically return unigrams (single words), rather than longer phrases [27]. While others have proposed methods for extracting longer phrases [1, 6, 7, 8, 13, 19], we lack a deeper understanding of how people choose keyphrases and how to leverage that understanding—in concert with computational tools—to better extract effective descriptive keyphrases.

In this paper, we contribute a **characterization of the statistical and linguistic properties of human-generated keyphrases**. Our analysis—based on 5,611 responses from 69 graduate students describing Ph.D. thesis abstracts—reveals a number of applicable insights. For example, we find that: longer keyphrases outnumber unigrams three to one; increasing the size or topical diversity of a collection reduces the length and specificity of selected terms; and choices of expert and non-expert readers show few differences.

Leveraging these findings, we propose a **two-stage method for automatic keyphrase extraction**. First, a regression model trained using data from our user study identifies and scores candidate keyphrases. We find that a simplified model combining document term counts and language-wide term commonness performs as well as more complex statistical measures, and can be further improved by incorporating positional and grammatical features. A configurable second phase then organizes candidate keyphrases to group related terms and reduce redundancy. The resulting organization enables users to vary the level of specificity of displayed terms, and allows applications to dynamically select terms based on the available screen space or current context of interaction. For example, a keyphrase label in a visualization might grow longer and more specific through semantic zooming.

We assess our algorithm through crowdsourced ratings and comparisons of both automatically and manually selected phrases. Human judges rated the quality of tag clouds using phrases selected by our technique and unigrams selected using G^2 [7, 22]. We find that raters prefer the tag clouds generated by our method. Moreover, the precision and recall of candidate keyphrases selected by our regression model can match that of phrases hand-selected by human readers.

[1] All author keywords appear in the top 10 chosen by our algorithm.

Submitted for review to CHI 2011.

Much prior work leverages large text corpora and machine learning to select salient descriptive terms for text. In this paper, we take a different approach—by observing how people read and summarize text. We present a model for identifying high-quality keyphrases to improve text analysis and visualization. Using simple linguistic features, our model performs well on short texts that cause existing approaches to degrade, in domains where a large reference corpus is unavailable, and for applications that have limited computational resources or require interactive response rates.

RELATED WORK

The most common means of selecting descriptive terms is via bag-of-words frequency statistics of single words (unigrams). In their simplest form, frequency statistics count the number of occurrences of single words within a text. Researchers have developed various techniques to improve frequency statistics, including removal of frequent "stop words," weighting by inverse document frequency as in tf.idf [24] and BM25 [23], heuristics such as WordScore [16], or more sophisticated probabilistic measures such as G^2 [7, 22] or the variance-weighted log-odds ratio [21]. While unigram frequency statistics have proven popular in practice (e.g., [5, 28, 29]), several factors limit their usefulness.

For decades, researchers have anecdotally noted that the best descriptive terms are often neither the most frequent nor infrequent terms, but rather medium frequency terms [17]. Existing frequency statistics designed for document retrieval may be good at weighting terms for maximizing search effectiveness, but it is unclear whether the same terms provide good summaries for document understanding [2]. In addition, frequency statistics often require long documents and a large reference corpus, and may not work well for short texts [2]. Moreover, it is unclear which existing frequency statistics (if any) are best suited for keyphrase extraction.

Unigrams are also unlikely to provide the best descriptions. In a survey of journals, Turney [27] found that unigrams account for only a small fraction of human-assigned index terms. To allow for longer phrases, Dunning proposed modeling words as binomial distributions to identify domain-specific bigrams (two-word phrases) [7]. Systems such as KEA++ or Maui use pseudo-phrases ("phrases" that remove stop words and ignore the ordering of words) for extracting longer phrases [19]. Hulth considered all trigrams (phrases up to length three) in her algorithm [13]. While the inclusion of longer phrases potentially allows for more expressive keyphrases, systems that permit longer phrases can suffer from poor precision and meaningless phrases. The inclusion of longer phrases may also result in redundant terms at different levels of specificity [8], such as "visualization," "data visualization," and "interactive data visualization."

Researchers have taken several approaches to ensure that longer keyphrases are meaningful and that phrases of the appropriate specificity are chosen. Many approaches [1, 6, 8, 13] filter candidate keyphrases by identifying noun phrases using a part-of-speech tagger, shallow parser, or full statistical parser. Of note is the use of so-called *technical terms* [14] that match regular expression patterns over part-of-speech tags. To reduce redundancy, Barker [1] chooses the most specific keyphrase by eliminating any keyphrases that are a subphrase of another. Medelyan's KEA++ system [19] trains a Naïve Bayes classifier to match the specificity of keyphrases produced by professional indexers.

However, we still lack a systematic characterization of the features most predictive of high-quality keyphrases. Prior evaluations of keyphrase extraction methods [1, 27] have focused on the accuracy of individual keyphrases in isolation, which may not be indicative of the quality of the set as a whole. Additionally, past work has examined the consistency of keyphrase specificity in comparison to terms chosen by professional indexers [19]. It is unclear to what degree the choices made by these indexers match those of readers.

USER STUDY OF HUMAN-GENERATED KEYPHRASES

To better understand how people choose descriptive keyphrases, we compiled a corpus of phrases that were manually selected by both expert and non-expert readers. We then analyzed this corpus to assess how various statistical and linguistic features may contribute to keyphrase quality.

Study Design

We asked graduate students to provide descriptive phrases for a collection of Ph.D. dissertation abstracts. We selected 144 documents from a larger corpus of 9,068 Ph.D. theses published at Stanford University from 1993 to 2008. These abstracts constitute a meaningful and diverse corpus well-suited to the interests of our study participants. To ensure a diverse set of topics, we selected 24 abstracts each from the following six departments: Computer Science, Mechanical Engineering, Chemistry, Biology, Education, and History.

We recruited graduate students from two major universities via student e-mail lists. Students came from departments matching the topic areas of selected abstracts. Subjects participated over the Internet between December 2009 and January 2010. We received 69 completed studies. Note that while we use the terminology *keyphrase* in this paper for brevity, the longer description "keywords and keyphrases" was used throughout the study to avoid biasing responses. The online study was titled and publicized as an investigation of "keyword usage."

In each session, we presented participants with a series of webpages and asked them to read and summarize text. We showed participants either one or three documents on the same webpage and instructed them to summarize the document(s) using five or more keyphrases. Subjects would sequentially describe three individual documents and then summarize the three as a whole. They would then repeat this process for two more collections. We collected a total of 5,611 free-form responses from the 69 participants.

We varied three independent factors in the user study:

Familiarity. We considered a subject familiar with a topic if they had conducted research in the same discipline (department) as the presented text. We relied on self-reports to match subjects to both familiar and unfamiliar topics.

Document count. Participants were asked to summarize the content of either a single document or three documents as a group. In the case of multiple documents, we used three dissertations supervised by the same primary advisor.

Topic diversity. We measured the similarity between two documents using the cosine distance between their tf.idf term vectors. We considered a collection topically coherent if its three constituent documents were the three most similar dissertations from the same advisor. We categorized a collection topically diverse if the documents were the three least similar dissertations from the same advisor. We considered only advisors with ten or more graduated Ph.D. students.

Statistical and Linguistic Features

To analyze and model subject responses, we compute the following features from the presented documents and subject-authored keyphrases. Note that we use "term" and "phrase" interchangeably. Term length refers to the number of words in a phrase; an *n*-gram is a phrase consisting of *n* words.

Documents are the texts we show to subjects, while *responses* are the subject-provided summary keyphrases. We tokenize text based on the Penn Treebank tokenization standard [18], and extract all terms of up to length 5. We record the position of each phrase in the document, as well as whether or not a phrase occurs in the first sentence. *Stems* are the roots of words with variational suffixes removed. We apply light stemming [20], which removes only noun and verb inflections (such as plural *s*) according to a word's part of speech. Stemming allows us to group variants of a term when counting frequencies or testing if a phrase occurs in a document.

Term frequency (*tf*) is the number of times a phrase occurs in the document (*document term frequency*), in the full dissertation corpus (*corpus term frequency*), or in all English webpages indexed by Google [3] (*web term frequency*). We define *term commonness* as the normalized term frequency relative to the most frequent n-gram, either in the dissertation corpus or on the web. For example, the commonness of a unigram equals $\log(tf)/\log(tf_{\text{the}})$ where tf_{the} is the term frequency of "the"—the most frequent unigram. When distinctions need to be made, we refer to the former as *corpus commonness* and the latter as *web commonness*.

Term position is a normalized measure of a term's location in a document; 0 corresponds to the first word and 1 to the last. The *absolute first occurrence* is the minimum position of a term (c.f., [19]). However, frequent terms are more likely to appear early in a document due to higher rates of occurrence. We introduce a new feature—the *relative first occurrence*—to factor out the correlation between position and frequency. Relative first occurrence is the probability that a term's first occurrence is lower than the first occurrence of a randomly sampled term with the same frequency. This measure makes a simplistic assumption—that term positions are uniformly distributed—but allows us to assess the effect of term position as an independent feature.

We annotate terms that are *noun phrases*, *verb phrases*, or match *technical term* patterns [14]; see Table 1. Part-of-speech information is determined using the Stanford POS Tagger [26]. We additionally determine grammatical information using the Stanford Parser [15] and annotate the corresponding words in each sentence.

Technical Term	$T = (A\vert N)^+ (N\vert C) \mid N$
Compound Technical Term	$X = (A\vert N)^* N \text{ of } T$

Table 1. Technical terms are defined by the above part-of-speech regular expressions. N is a noun, A an adjective, and C a cardinal number. We modify the definition of technical terms by permitting cardinal numbers as the trailing word, to include terms like *Windows 95*.

Characterization of Human-Generated Keyphrases

Using the above features, we then sought to characterize various properties of collected human-generated keyphrases.

For single documents, the **number of responses** varies between 5 and 16 keyphrases (see Figure 1). In the study, we required subjects to enter a minimum of five responses for each document collection. The peak at five in Figure 1 suggests that subjects might respond with fewer than five phrases without this requirement. However, it is unclear whether this reflects a lack of appropriate keyphrase choices or a desire to minimize effort. For tasks with multiple documents or diverse topics, participants assigned fewer keyphrases despite the increase in the amount of text and topics. Subject familiarity with the readings does not have a discernible effect on the number of keyphrases.

Assessing the prevalence of **words vs. phrases**, Figure 2 shows that bigrams are the most common type of response, accounting for 43% of all free-form keyphrase responses, followed by unigrams (25%) and trigrams (19%). For multiple documents or documents with diverse topics, we observe an increase in the use of unigrams and a corresponding decrease in the use of trigrams and longer terms. The proportion of bigrams stayed relatively constant across readings.

Figure 3 shows the distribution of responses as a function of **web commonness**. We observe a bell-shaped distribution centered around mid-frequency, consistent with the distribution of significant words posited by Luhn [17]. As the number of documents and topic diversity increases, the distribution shifts toward more common terms. We found similar correlations for corpus commonness.

For each user-generated keyphrase, we find matching text in the reading, and note that 65% of the responses are **present in the document**[2]. Considering for the rest of this paragraph just the two thirds of keyphrases in the reading, the associated *positional* and *grammatical* properties for this subset are summarized in Table 2. 22% of these keyphrases are present in the first sentence, even though first sentences contain only 9% of all terms. Comparing the first occurrence of keyphrases with that of randomly-sampled phrases of the

[2] Most part-of-speech taggers do not perform as well on sentence fragments such as keyphrases. This can negatively impact our light stemming approach. To address this, if a word is given any noun or verb tag, we try stemming it with all noun or verb tags, respectively, and match words if any stem matches. Without this looser matching, only 61.79% of the responses are present in the document.

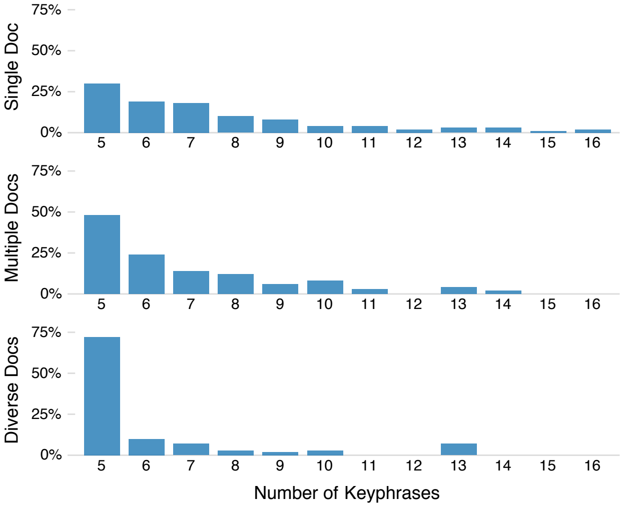


Figure 1. ***How many keyphrases do people use?*** **Participants use fewer keyphrases to describe multiple documents or documents with diverse topics, despite the increase in the amount of text and topics.**

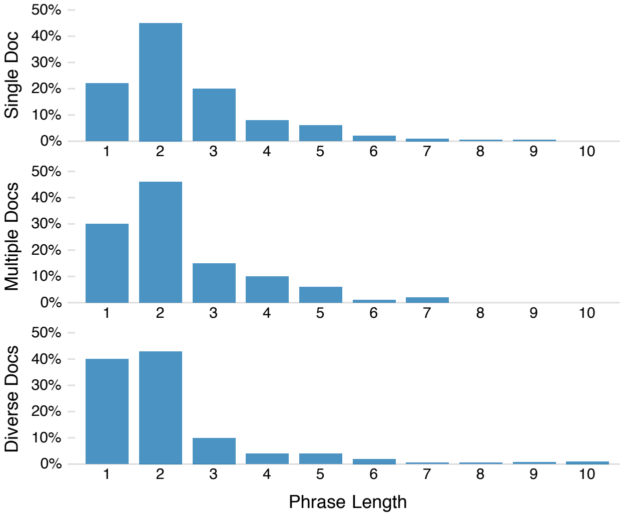


Figure 2. ***Do people use words or phrases?*** **Bigrams are the most common. For single documents, 75% of responses contain multiple words. Unigram use increases with the number and diversity of documents.**

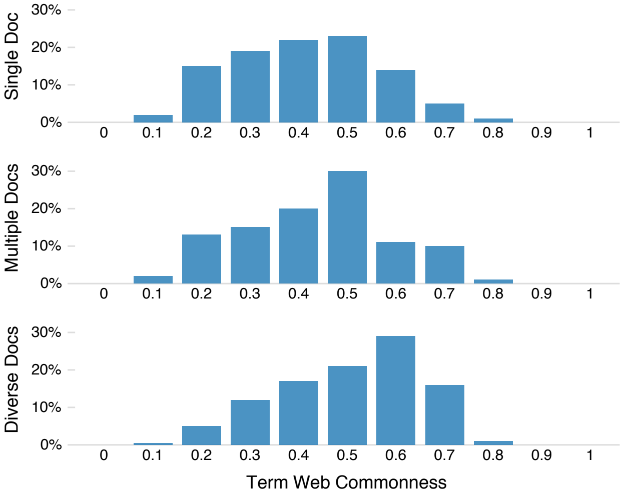


Figure 3. ***Do people use generic or specific terms?*** **Term commonness increases with the number and diversity of documents.**

Feature	% of Keyphrases	% of All Phrases
First sentence	22.09%	8.68%
Relative first occurrence	56.28%	50.02%
Noun phrase	64.95%	13.19%
Verb phrase	7.02%	3.08%
Technical term	82.33%	8.16%
Compound tech term	85.18%	9.04%

Table 2. ***Positional and grammatical features*** **of the 65.12% human-generated keyphrases present in the documents. Keyphrases occur earlier in the document; two-thirds are noun phrases and over four-fifths match a technical term pattern.**

same frequency, we find that keyphrases occur earlier 56% of the time, a statistically significant result ($\chi^2(1) = 88$, $p < 0.001$). Nearly two-thirds of keyphrases found in the document are part of a noun phrase. Only 7% are part of a verb phrase, though this is still statistically significant ($\chi^2(1) = 147{,}000$, $p < 0.001$). Most strikingly, over 80% of the keyphrases are part of a technical term.

Summary

Our study quantifies how people select keyphrases. Subjects primarily choose multi-word phrases, prefer terms with median commonness, and largely use phrases already present in a document. Moreover, these features shift as the number and diversity of documents increases. Keyphrase selection also correlates with term position, suggesting we should treat documents as more than just "bags of words." Finally, human-selected keyphrases show recurring grammatical patterns, indicating the utility of linguistic features.

A STATISTICAL MODEL OF KEYPHRASE QUALITY

We now present a regression model for predicting the quality of keyphrases describing a single document, constructed using our user study results. We systematically assess the contribution of various statistical and linguistic features, and summarize significant findings. Our final result is a pair of models of keyphrase quality (one corpus-dependent, the other independent) that incorporate term frequency, commonness, position, and grammatical features. Due to space constraints, we limit our attention to keyphrases describing a single document. However, the same modeling approach is applicable to keyphrase selection for multiple documents.

Modeling Methodology

We modeled keyphrase quality using logistic regression. We chose this model because its results are readily interpretable and contributions from each feature can be easily assessed. As we demonstrate later, our regression models perform comparably with human subjects, indicating that more advanced machine learning algorithms may not be necessary. To assess the impact of individual variation, we initially used a generalized linear mixed model [9], in which we modeled subjects and documents as random effects.[3] We found that the random effects were not significant, and so reverted to a standard logistic regression model.

[3] These models let you assess dimensions of random effects, such as variation due to document and subject, in one coherent model that extends generalized linear models such as logistic regression.

We constructed models by fitting to the features of 2,707 responses from our user study. We used keyphrases up to length 5 that described single documents and occurred within the text. (Our data on web commonness contains only phrases up to length 5.) We examine four classes of features, visited in order of their predictive power, as determined by a preliminary analysis. Due to space limitations, we only present features found to be statistically significant. Unless otherwise stated, all features are added to the regression model as an independent factor without interaction terms.

First, we examine *frequency statistics* by comparing the performance of seven measures on phrases up to length 5. We find there is little gain in accuracy when multiple frequency statistics are added to a single model. Second, we investigate the effect of introducing *term commonness* features to assess whether language-wide term frequency can be utilized in predicting keyphrase quality. Third, we add *grammatical features* and compare the difference in features derived from part-of-speech tagging (*tagger features*) versus more costly statistical parsing (*parser features*). Finally, we assess the contribution of *positional* features.

We evaluate the contribution of features using precision-recall curves and statistical significance determined by a likelihood ratio test. Precision-recall curves measure the accuracy of an algorithm by comparing its output to a known, "correct" set of keyphrases; in this case, the list of keyphrases present in the documents. Precision measures the percentage of correct keyphrases in the output. Recall measures the percentage of the correct keyphrases captured by the output. As a model produces more keyphrases, recall increases (i.e., when all 5-grams from the entire document are given as output, recall is 100%) but precision decreases. The precision-recall curve measures the performance of an algorithm over an increasing number of output keyphrases. Higher precision is desirable with fewer phrases, and a higher area under the curve generally indicates better performance. We also assessed each model using model selection criteria (i.e., AIC, BIC). As these scores coincide with the rankings derived from precision-recall measures, we omit them presently.

Frequency Statistics

We computed seven different frequency statistics. Our simplest measure was log term frequency: **log(*tf*)**. We also computed **tf.idf**, **BM25**, **G^2**, **variance-weighted log-odds ratio**, and **WordScore** using the dissertation corpus as reference. We created a set of **hierarchical tf.idf** scores (c.f., [28]) by computing tf.idf with five nested reference corpora: all terms on the web, all dissertations in the Stanford dissertation corpus, dissertations from the same school, dissertations in the same department, and dissertations supervised by the same advisor. Due to its poor performance on 5-grams, we assessed four variants of standard tf.idf scores: tf.idf on unigrams, and all phrases up to bigrams, trigrams, and 5-grams. Formulas for frequency measures are shown in Table 3.

Figure 4(a) shows the performance of these frequency statistics. Probabilistic measures, namely G^2, BM25 and weighted log-odds ratio, perform better than count-based approaches (e.g., tf.idf) and heuristics such as WordScore. Count-based approaches suffer with longer phrases because of an excessive number of ties due to many 4- and 5-grams occurring only once in the corpus. However, tf.idf on unigrams still performs worse than probabilistic approaches.

Statistic	Definition
log(tf)	$\log\left(t_{\text{Doc}}\right)$
tf.idf	$\left(t_{\text{Doc}}/t_{\text{Ref}}\right)\cdot\log\left(N/D\right)$
G^2	$2\left(t_{\text{Doc}}\log\frac{t_{\text{Doc}}\cdot T_{\text{Ref}}}{T_{\text{Doc}}\cdot T_{\text{Doc}}}+t_{\overline{\text{Doc}}}\log\frac{t_{\overline{\text{Doc}}}\cdot T_{\text{Ref}}}{T_{\overline{\text{Doc}}}\cdot T_{\text{Doc}}}\right)$
BM25	$3\cdot t_{\text{Doc}}/\left(t_{\text{Doc}}+2\left(0.25+0.75\cdot T_{\text{Doc}}\cdot r\right)\right)$
WordScore	$\left(t_{\text{Doc}}-t_{\text{Ref}}\right)/\left(T_{\overline{\text{Doc}}}-T_{\overline{\text{Ref}}}\right)$
log-odds ratio (weighted)	$\left(\log\frac{t'_{\text{Doc}}}{t'_{\overline{\text{Doc}}}}-\log\frac{T'_{\text{Doc}}}{T'_{\overline{\text{Doc}}}}\right)/\sqrt{\frac{1}{t'_{\text{Doc}}}+\frac{1}{t'_{\overline{\text{Doc}}}}}$

Table 3. Frequency Statistics. Given a document from a reference corpus with N documents, the score for a term is given by the above formulas where $t_{\textbf{Doc}}$ and $t_{\textbf{Ref}}$ denote term frequency in the document and reference corpus; $T_{\textbf{Doc}}$ and $T_{\textbf{Ref}}$ are the number of words in the document and reference corpus; D is the number of documents in which the term appears; r is the average word count per document; t' and T' indicate measures for which we increment each term's frequency in each document by 0.01; $t_{\overline{\textbf{Doc}}}=t_{\textbf{Ref}}-t_{\textbf{Doc}}$ and $T_{\overline{\textbf{Doc}}}=T_{\textbf{Ref}}-T_{\textbf{Doc}}$.

Term Commonness

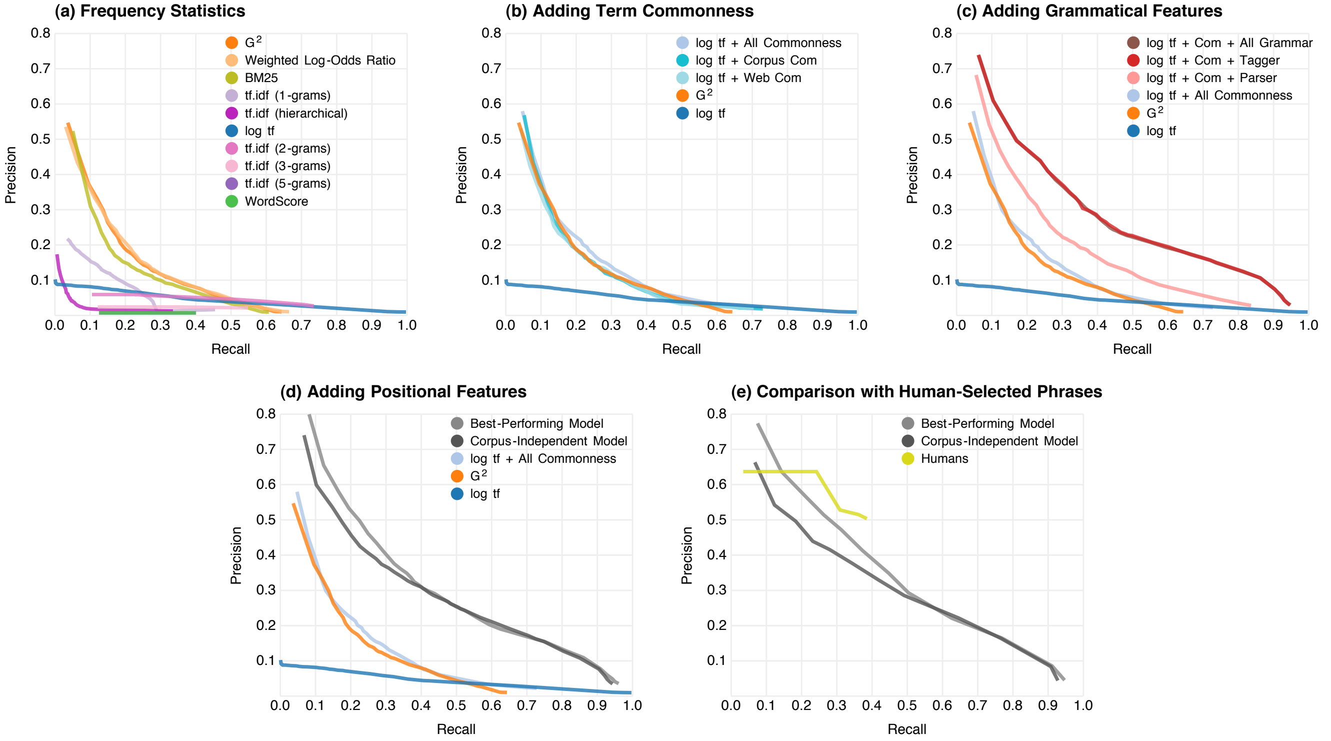
During keyphrase characterization, we observed a bell-shaped distribution of keyphrases as a function of commonness. We discretized commonness features into **web commonness** bins and **corpus commonness** bins in order to capture this nonlinear relationship. We examined the effects of different bin counts over 2, 3, 4, 5, 6, 8, 10, 15, and 20 bins.

As shown in Figure 4(b), the performance of log(*tf*) + commonness matches that of statistical methods such as G^2. As corpus and web commonness are highly correlated, the addition of both commonness features yields only a marginal improvement over the addition of either feature alone. We then measure the effects due to number of bins. Precision-recall increases as the number of bins are increased up to about 5 bins, and there is marginal gain between 5 and 8 bins. Examining the regression coefficients for a large number of bins (10 bins or more) shows great randomness in the coefficients, indicating likely overfitting. We note that the coefficients for commonness peak at middle frequency; see Table 4. Adding an interaction term between frequency statistics and commonness yields no increase in performance. Interestingly, the coefficient for tf.idf is negative when combined with web commonness, indicating that tf.idf scores have a slight negative correlation with keyphrase quality.

Grammatical Features

Computing grammatical features requires either full parsing of the text or part-of-speech tagging. Of note is the significantly higher computational cost of parsing—nearly two orders of magnitude in run time. We measure the effectiveness of these two classes of features separately to determine if the extra computational cost of parsing pays dividends.

Parser features. For each term extracted from the text, we tag the term as a **full noun phrase** or **full verb phrase** if it


(a) Frequency Statistics

(b) Adding Term Commonness

(c) Adding Grammatical Features

(d) Adding Positional Features

(e) Comparison with Human-Selected Phrases

Figure 4. Precision-Recall curves for keyphrase regression models. Legends are sorted by decreasing initial precision. (a) Frequency statistics only; G^2 and log-odds ratio perform well. (b) Adding term commonness; a simple combination of log(*tf*) and commonness performs competitively to G^2. (c) Adding grammatical features further improves performance. (d) Adding positional features provides additional gains for both a complete model and a more convenient corpus-independent model. (e) Comparison with human-selected keyphrases; our models provide higher precision at low recall values. As described in the text, this final comparison must be computed differently than the others, hence the slight shift in scores.

matches exactly a noun phrase or verb phrase identified by the parser. A term is tagged as a **partial noun phrase** or **partial verb phrase** if it matches a substring within a noun phrase or verb phrase. The last word in a noun phrase is tagged as a **head noun**. Leading words in a noun phrase are tagged as **optional leading words** if their part-of-speech is one of *cardinal number*, *determiner*, or *pre-determiner*.

Tagger features. Phrases that match the technical term patterns defined in Table 1 are tagged as either a **full technical term** or **full compound technical term**. Phrases that match a substring in a technical term are tagged as a **partial technical term** or **partial compound technical term**.

As shown in Figure 4(c), adding parser-derived grammar information yields an improvement significantly greater than the differences between leading frequency statistics. Adding technical terms matched using POS tags improved precision and recall more than parser-related features. Combining both POS and parser features yields only a marginal improvement over POS features. Head nouns (c.f., [1]) did not have a measurable effect on keyphrase quality.

Positional Features

Finally, we introduce **relative first occurrence** and **presence in first sentence** as positional features; both predictors are statistically significant. The regression model with frequency statistics, term commonness, grammatical features, and term position features achieves a precision of 79.96% for the top-ranked keyphrase (at 8.17% recall) and 49.78% for the top 5 ranked keyphrases (at 22.48% recall).

Model Feature	Coefficients
constant	-2.3550***
log(tf)	0.9390***
WC $\in (0\%, 20\%]$	0.1770
WC $\in (20\%, 40\%]$	0.2304*
WC $\in (40\%, 60\%]$	0.0158
WC $\in (60\%, 80\%]$	-0.6205***
WC $\in (80\%, 100\%]$	-1.9081***
relative first occurrence	0.4800**
first sentence	0.9386***
full tech. term	-0.5015
partial tech. term	1.4461**
full compound tech. term	1.1373
partial compound tech. term	1.1806*

Table 4. Corpus-Independent Model. WC **= web commonness bins, stat. significance:** $* = p < 0.05$, $** = p < 0.01$, $*** = p < 0.001$

Full and Simplified Model

Combining frequency statistics, term commonness, grammatical, and positional features, we build two models for predicting keyphrase quality. The full model is based on all statistically-significant features using the Stanford dissertations corpus as reference. Table 4 shows a simplified model where we excluded corpus-dependent (i.e., corpus commonness) and parser features to eliminate dependencies and improve running time. Omitting the more costly features incurs a slight decrease in precision-recall as shown in Figure 4(d).

KEYPHRASE COMPETITIVE SELECTION

In the second processing phase, we consider the overall quality of keyphrases as a set. We filter keyphrases to reduce redundancy—an issue noted in prior work, but more prominent in our approach because our model often yields a large number of reasonable phrases (around 50 for abstracts averaging 350–400 words). We also note that keyphrase quality might depend on application and user needs. We present a simple approach for filtering and selection, which is sufficient to remove a reasonable amount of redundancy and provides a means for adapting keyphrase specificity on demand.

Redundancy Reduction

Redundancy reduction suppresses phrases similar in concept. The goal is to ensure that each successive output keyphrase provides a useful marginal information gain [4] instead of lexical variations. For example, the following list of keyphrases differ lexically but can be similar, if not identical, in concept: "*Flash Player 10.1*", "*Flash Player*", "*Flash*." We propose that an ideal redundancy reduction algorithm should group phrases that are similar in concept (e.g., perhaps similar to synsets in WordNet), choose the most prominent lexical form of a concept, and suppress other redundant phrases. Making this decision crucially involves consideration of a set of candidate keyphrases at once rather than scoring each independently (as in our logistic regression model).

We use string similarity as a very rough approximation of conceptual similarity between phrases. We consider two phrases A and B to be similar if A can be constructed from B by prepending or appending a word. For example, "*Flash Player 10.1*" and "*Flash Player*" are considered similar. For many of the top-ranked keyphrases, this assumption is true. We use two simple features to determine which form of similar phrases to display: term length and term commonness.

We also account for the special case of names. We apply named entity recognition [10] to identify persons, locations, and organizations. As an approximation, if the trailing substring of a person matches the trailing substring of another person, we consider the two people to be identical. For example, "*President Obama*" and "*Barack Obama*" are considered the same person. If the name of a location or organization is a substring of another, we consider the two entities to be identical, e.g., "*Intel*" and "*Intel Corporation*." We apply acronym recognition [25] to identify the long and short forms of the same concept such as "*World of Warcraft*" and "*WoW*." For most short texts our assumptions hold; however, in general a more principled approach is required for robust entity and acronym resolution.

Keyphrase Selection and Specificity

Once similar terms have been grouped, we must select which term to present. For this selection process, we contend that *keyphrase specificity* should be application and user dependent. For example, the choice of keyphrase may depend on available screen space or on the diversity of topics in the text.

To parameterize final keyphrase selection, we allow users to optionally choose longer/shorter and more generic or specific terms. When two terms are deemed similar, we can bias for longer keyphrases by subtracting the ranking score from the shorter of the two terms and adding that to the score of the longer term, in proportion to the difference in term length. Similarly, we can bias for more generic or specific terms by shifting the ranking score between similar terms in proportion to the difference in term commonness. For recognized people, users may choose to expand all names to full names or contract to last names. For locations and organizations, users may elect to use the full-length form of the entity. For identified acronyms, users may choose to expand or contract the terminology to its long or short forms.

EVALUATION: HUMAN-SELECTED KEYPHRASES

To assess the effectiveness of our keyphrase extraction algorithm, we first compare the precision-recall of keyphrases from our algorithm to human-generated keyphrases. We describe a necessary modification to our precision-recall calculation and then demonstrate that our full model performs comparably to human selection and that our corpus independent model performs well for low-recall values.

In our previous comparisons of model performance, a candidate phrase was considered "correct" if it matched a term selected by any of the K human subjects who read a document. When evaluating human performance, however, a selected phrase can only be matched against responses from the $K-1$ other participants. A naïve comparison would thus unfairly favor our algorithm, as human performance would suffer due the smaller set of "correct" phrases. To ensure a meaningful comparison, we randomly sample a subset of K participants for each document. When evaluating human precision, a participant's response is considered accurate if it matches any phrase selected by another subject. We then *replace* that participant's responses with our model's output, ensuring that both lists are compared to the same $K-1$ subjects. We chose $K = 6$, as on average each document in our study was read by 5.75 subjects.

Figure 4(e) shows the performance of our two models versus human performance. At low recall (i.e., for the top keyphrase), our full model achieves higher precision than human responses, while our simplified model performs competitively. The full model's precision closely matches that of human accuracy until mid-recall values. We note, however, that while algorithms are required to generate phrases until they produce the entire set of terms (i.e., 100% recall), participants in our study are not required to provide responses beyond a minimum of five keyphrases. Human participants can stop answering at anytime—a potential explanation of why human precision remains higher at mid-recall values.

EVALUATION: INSPECTION OF TOP KEYPHRASES

We next qualitatively evaluate the output of our algorithm, both the initial ranked keyphrases and the effectiveness of the filtering and selection. We compared the top 50 keyphrases produced by our algorithm (both models) with outputs from G^2, BM25, and variance-weighted log-odds ratio. We examined both dissertation abstracts from our user study and additional documents described in the next section.

As mentioned earlier, our regression models often choose up to 50 or more reasonable keyphrases. In contrast, we find that G^2, BM25, and variance-weighted log-odds ratio typically select a few reasonable phrases, but starts producing unhelpful terms after the top 10 results. The difference is exacerbated for short texts. For example, in a 59-word article about San Francisco's Mission District, our algorithm returns appropriate noun phrases as keyphrases such as "*colorful Latino roots*" and "*gritty bohemian subculture*" while the other three methods produce only one to three usable phrases: "*Mission*", "*the District*", or "*district.*"

Our algorithm regularly extracts salient longer phrases such as "*open-source digital photography software platform*" (not chosen by other algorithms), "*hardware-accelerated video playback*" (also selected by G^2, but not others), and "*cross platform development tool*" (not chosen by others). The effect of including optional leading words is evident in the selection of more grammatically appropriate phrases such as "*long exposure*" (our models) vs. "*a long exposure*" (G^2, BM25, weighted log-odds ratio) in an article on photography. Even though term commonness favors mid-frequency phrases, our model can still select salient words from all commonness levels. For example, on an article about the technologies in Google vs. Bing, our models choose "*search*" (common word), "*navigation tools*" (med-frequency phrase), and "*colorful background*" (low-frequency phrase), while all other methods output only "*search*".

We observe few qualitative differences between our full and simplified models. One discernible difference is due to mistakes in part-of-speech tagging. In one case, the full model returns the noun phrase "*interactive visualization*", but the simplified model returns "*interactive visualization leverage*", as the POS tagger mislabels "*leverage*" as a noun.

On the other hand, the emphasis on noun phrases can cause our algorithm to miss important keyphrases that are in verb form, such as "*civilians killed*" in a news article about the NATO forces in Afghanistan. Our algorithm chooses "*civilian causalities*" but places it significantly lower down the list. We return several phrases with unsuitable prefixes such as "*such scenarios*" and "*such systems*" because the word "*such*" is tagged as an adjective in the Penn Treebank tag set, and thus the entirety of the phrase is marked as a technical term. Changes to the POS tagger, parser or adding conditions to the technical term patterns could ameliorate this issue. We also note that numbers are not handled by the original technical term patterns [14]. We modified the definition to include trailing cardinal numbers to allow for phrases such as "*H. 264*" and "*Windows 95*", dates such as "*June 1991*", and events such as "*Rebellion of 1798.*"

Prior to redundancy reduction and specificity selection, we often observe redundant keyphrases similar in term length, concept, or identity. For example, "*Mission*", "*Mission District*", and "*Mission Street*" in a travel article about San Francisco. Our heuristics for filtering based on string similarity, named entity recognition, and acronym recognition appear to improve the returned keyphrases (e.g. Tables 5 and 6).

Our Corpus-Independent Model	**G^2**
Adobe	Flash
Flash Player	Player
technologies	Adobe
H. 264	video
touch-based devices	Flash Player is
runtime	264
surge	touch
fair amount	open source
incorrect information	10.1
hardware-accelerated video playback	Flash Player 10.1
Player 10.1	SWF
touch	the Flash Player
SWF	more about
misperceptions	content
mouse input	H.
mouse events	battery life
Seventy-five percent	codecs
codecs	browser
many claims	desktop
content protection	FLV/F4V
desktop environments	Flash Player team
Adobe Flash Platform	Player 10.1 will
CPU-intensive task	actively maintained
appropriate APIs	Anyone can
battery life	both open and proprietary
further optimizations	ecosystem of both
Video Technology Center	ecosystem of both open and
memory use	for the Flash
Interactive content	hardware-accelerated
Adobe Flash Player runtime	hardware-accelerated video playback
static HTML documents	include support
rich interactive media	multitouch
tablets	of both open
new content	on touch-based
complete set	open source and is
vulnerabilities	play back
gesture APIs	Read more
modern software	tablets
netbooks	that Flash Player
complete multimedia runtime	The Adobe Flash
browser vendors	touch-based devices
high-definition video	©
advanced video distribution	technologies
adaptive bitrate delivery	Mac
rich ecosystem	APIs
core engine	video on the
extensive steps	runtime
FLV/F4V	and proprietary
Player team	on the web
CPU usage	multimedia

Table 5. Top 40 keyphrases for an open letter from Adobe about Flash technologies. We applied our lexical redundancy reduction to *both* lists.

More Generic	**More Specific**
Flash Player	H. 264
Adobe	Flash Player 10.1
technologies	hardware-accelerated video playback
H. 264	Flash Player team
runtime	touch-based devices
SWF	Adobe
touch-based devices	technologies
misperceptions	Symantec Global Internet Threat Report
codecs	Adobe Flash Player runtime
Player 10.1	Seventy-five percent

Table 6. Top 10 keyphrases generated by our corpus-independent model for an open letter from Adobe, biased for (left) shorter, more generic terms; (right) longer, more specific terms.

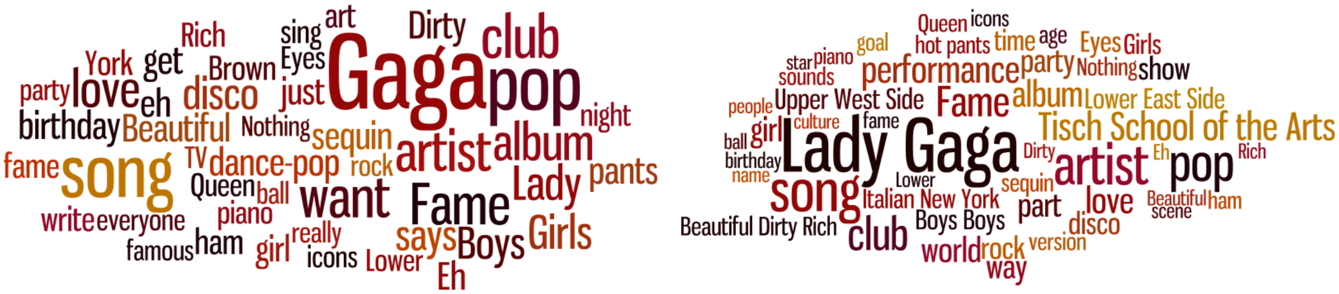


Figure 5. Tag clouds visualizing an online biography of the singer Lady Gaga. (a) Terms selected using G^2. (b) Terms selected using our algorithm.

EVALUATION: CROWDSOURCED RATINGS

To evaluate our keyphrase extraction method, we asked human judges to rate the relative quality of tag cloud visualizations with terms selected using both our technique and G^2 scores of unigrams (c.f., [5, 7, 22]). Collins [5] compellingly uses unigrams weighted by G^2 for visual text analytics, making them an interesting and ecologically valid comparison point. We also considered bigrams, but upon inspection the results were significantly less relevant than the unigrams. We hypothesized that the tag clouds created using our technique would be preferred due to better choices of descriptive terms and inclusion of complete phrases.

We chose to compare tag cloud visualizations for multiple reasons. First, tag clouds are a popular form of text visualization used by a diverse set of people [29]. Second, visual features such as sizing, layout, term proximity, and other aesthetics are likely to affect the perceived utility of, and preferences for, the visualizations. For instance, including only unigrams results in shorter terms, leading to a more densely-packed layout. Presenting selected terms in a simple list would fail to reveal the impact of these effects.

Method

Participants first read a short text passage and wrote a 1–2 sentence summary. They then viewed two tag clouds and were asked to rate which they preferred on a 5 point scale (with '3' indicating a tie) and to provide a brief rationale for their choice. We asked raters to "consider to what degree the tag clouds use appropriate words, avoid unhelpful or unnecessary terms, and communicate the gist of the text." One tag cloud consisted of unigrams with term weights calculated using G^2; the other contained keyphrases selected using our technique (corpus-independent model) and weighted by the regression score. Each tag cloud contained the top 50 terms, with font sizes proportional to the square root of the term weight. Occasionally our method selected less than 50 terms with positive weights; we omitted negatively-weighted terms. Tag cloud images were generated by Wordle [29] using the same layout and color parameters for each. We randomized the presentation order of tag clouds across trials.

We included tag clouds of 24 text documents. To sample a variety of genres, we used documents in 4 categories: abstracts of CHI 2010 papers, short biographies (3 of U.S. presidents, 3 of musicians), blog posts (two each from opinion, travel, and photography blogs), and news articles. Figure 5 shows tag clouds for a biography of the singer Lady Gaga.

We conducted our study using Amazon's Mechanical Turk. Each trial was posted as a task with a reward of $0.10 USD. We requested 24 assignments for each task, resulting in 576 ratings. Once the experiment was complete, we tallied the ratings for each tag cloud and coded free-text responses with the criteria invoked by raters' rationales.

Results

On average, raters significantly preferred tag clouds generated using our keyphrase extraction approach (267 ratings vs. 208 for G^2 and 101 ties; $\chi^2(2) = 73.76$, $p < 0.0001$). Moreover, our technique garnered more strong ratings: 49% (132/267) of positive ratings were rated as "MUCH better," compared to 38% (80/208) for G^2.

Looking at raters' rationales, we find that 70% of responses in favor of our technique cite the improved saliency of descriptive terms, compared to 40% of ratings in favor of G^2. More specifically, 12% of positive responses note the presence of terms with multiple words (*"It's better to have the words 'Adobe Flash' and 'Flash Player' together"*), while 13% cite the use of fewer, unnecessary terms (*"This is how tag clouds should be presented, without the clutter of unimportant words"*). On the other hand, some (16/208, 8%) rewarded G^2 for showing more terms (*"Tag cloud 2 is better since it has more words used in the text."*). A few raters also rated tag clouds based on how the terms matched the emotional tone of the text (*"It brings up more of the emotional point of the story, such as time, rocky start, etc"*).

Tag clouds in both conditions were sometimes preferred due to visual features such as term layout, tag cloud shape, and density: 29% (60/208) for G^2 and 23% (61/267) for our technique. While visual features were often mentioned in conjunction with remarks about term saliency, G^2 led to more ratings (23% vs. 14%) that mentioned only visual features (*"one word that is way bigger than the rest will give a focal point . . . it is best if that word is short and in the center"*).

The study results also reveal limitations of our keyphrase extraction technique. While our approach was rated superior for abstracts, biographies, and blog posts, on average G^2 fared better for news articles. In one case this was due to layout issues (a majority of raters preferred the central placement of the primary term in the G^2 cloud), but others specifically cite the quality of the chosen keyphrases. In an article about racial discrimination in online purchasing, our technique disregarded the term “black” due to its commonness and adjective part-of-speech. The tendency of our technique to give higher scores to people names non-central to the text at times led raters to prefer G^2. In general, prominent “mistakes” or omissions by either technique were critically cited.

Unsurprisingly, our technique was preferred by the largest margin for research paper abstracts, the domain closest to our training data. This observation suggests that applying our modeling methodology to human-selected keyphrases from other text genres may result in better selections. Our study also suggests that we might improve our keyphrase weighting by better handling named entities, so as to avoid giving high scores to non-central actors.

CONCLUSION

In this paper, we characterized the statistical and grammatical features of human-generated keyphrases, and presented a model for identifying high-quality descriptive terms for text. The model allows for adjustment of keyphrase specificity to meet application and user needs. Based on simple linguistic features, our approach does not require a pre-processed reference corpus, and is computationally efficient to support interactive applications. Evaluations reveal that our model is preferred by human judges, can match human extraction performance, and performs well even on short texts.

The exploding amount of text as data—from digital libraries to blog feeds and social media—present both a need and an opportunity for tools that allow people to access text at a scale not possible before. We hope that our model will enable more rapid and relevant assessment of text data. Descriptive keyphrases generated by our model can help support analysis and exploration of large document collections, particularly for choosing salient terms in visualizations [5, 11, 12, 28, 29]. Our method might also be applied in other contexts, for example to aid skimming via automatic highlighting of text or to improve tasks such as tagging or search. In future work, we would like to expand our modeling approach to identify descriptive phrases for multiple documents and also to characterize when and how people choose descriptive keyphrases that do not occur within the text itself.

REFERENCES

1. K. Barker and N. Cornacchia. Using noun phrase heads to extract document keyphrases. In *Canadian Society on Computational Studies of Intelligence*, pages 40–52, 2000.
2. B. Boguraev and C. Kennedy. Applications of term identification technology: Domain description and content characterisation. *Natural Language Processing*, 5(1):17–44, 1999.
3. T. Brants and A. Franz. Web 1T 5-gram Version 1, Linguistic Data Consortium, Philadelphia, 2006.
4. J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *ACM SIGIR*, pages 335–336, 1998.
5. C. Collins, F. B. Viégas, and M. Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE Visual Analytics Science and Technology*, pages 91–98, 2009.
6. B. Daille, E. Gaussier, and J.-M. Langé. Towards automatic extraction of monolingual and bilingual terminology. In *Conference on Computational Linguistics*, pages 515–521, 1994.
7. T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.
8. D. K. Evans, J. L. Klavans, and N. Wacholder. Document processing with LinkIT. In *Recherche d’Informations Assistee par Ordinateur (RIAO)*, 2000.
9. J. J. Faraway. *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. Chapman & Hall/CRC, 2006.
10. J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Annual Meeting on Association for Computational Linguistics (ACL)*, pages 363–370, 2005.
11. S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: Visualizing theme changes over time. In *IEEE InfoVis*, page 115, 2000.
12. M. A. Hearst. *Search User Interfaces*. Cambridge Univ Press, 2009.
13. A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *EMNLP Conference*, pages 216–223, 2003.
14. J. S. Justeson and S. M. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27, 1995.
15. D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Annual Meeting on Association for Computational Linguistics (ACL)*, pages 423 – 430, 2003.
16. M. Laver, K. Benoit, and T. College. Extracting policy positions from political texts using words as data. *American Political Science Review*, pages 311–331, 2003.
17. H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.
18. M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computional Linguistics*, 19(2):313–330, 1993.
19. O. Medelyan and I. H. Witten. Thesaurus based automatic keyphrase indexing. In *ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 296–297, 2006.
20. G. Minnen, J. Carroll, and D. Pearce. Applied morphological processing of English. *Natural Lang Eng*, 7(3):207–223, 2001.
21. B. Monroe, M. Colaresi, and K. Quinn. Fightin’ words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403, 2008.
22. P. Rayson and R. Garside. Comparing corpora using frequency profiling. In *Workshop on Comparing Corpora*, pages 1–6, 2000.
23. S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In *Research and Development in Information Retrieval*, pages 35–56, 1981.
24. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, pages 513–523, 1988.
25. A. S. Schwartz and M. A. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Pacific Symposium on Biocomputing*, 2003.
26. K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Human Language Technologies (HLT-NAACL)*, pages 252 – 259, 2003.
27. P. D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.
28. F. B. Viégas, S. Golder, and J. Donath. Visualizing email content: Portraying relationships from conversational histories. In *ACM CHI*, pages 979 – 988, 2006.
29. F. B. Viégas, M. Wattenberg, and J. Feinberg. Participatory visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1137–1144, 2009.