“Without the Clutter of Unimportant Words”: Descriptive Keyphrases for Text Visualization

JASON CHUANG, CHRISTOPHER D. MANNING, and JEFFREY HEER, Stanford University

Keyphrases aid the exploration of text collections by communicating salient aspects of documents and are often used to create effective visualizations of text. While prior work in HCI and visualization has proposed a variety of ways of presenting keyphrases, less attention has been paid to selecting the best descriptive terms. In this article, we investigate the statistical and linguistic properties of keyphrases chosen by human judges and determine which features are most predictive of high-quality descriptive phrases. Based on 5,611 responses from 69 graduate students describing a corpus of dissertation abstracts, we analyze characteristics of human-generated keyphrases, including phrase length, commonness, position, and part of speech. Next, we systematically assess the contribution of each feature within statistical models of keyphrase quality. We then introduce a method for grouping similar terms and varying the specificity of displayed phrases so that applications can select phrases dynamically based on the available screen space and current context of interaction. Precision-recall measures find that our technique generates keyphrases that match those selected by human judges. Crowdsourced ratings of tag cloud visualizations rank our approach above other automatic techniques. Finally, we discuss the role of HCI methods in developing new algorithmic techniques suitable for user-facing applications.

Categories and Subject Descriptors: H.1.2 [**Models and Principles**]: User/Machine Systems

General Terms: Human Factors

Additional Key Words and Phrases: Keyphrases, visualization, interaction, text summarization

ACM Reference Format:
Chuang, J., Manning, C. D., and Heer, J. 2012. “Without the clutter of unimportant words”: Descriptive keyphrases for text visualization. ACM Trans. Comput.-Hum. Interact. 19, 3, Article 19 (October 2012), 29 pages.
DOI = 10.1145/2362364.2362367 http://doi.acm.org/10.1145/2362364.2362367

1. INTRODUCTION

Document collections, from academic publications to blog posts, provide rich sources of information. People explore these collections to understand their contents, uncover patterns, or find documents matching an information need. Keywords (or *keyphrases*) aid exploration by providing summary information intended to communicate salient aspects of one or more documents. Keyphrase selection is critical to effective visualization and interaction, including automatically labeling documents, clusters, or themes [Havre et al. 2000; Hearst 2009]; choosing salient terms for tag clouds or other text visualization techniques [Collins et al. 2009; Viégas et al. 2006, 2009]; or summarizing text to support small display devices [Yang and Wang 2003; Buyukkokten et al. 2000,

This work is part of the Mimir Project conducted at Stanford University by Daniel McFarland, Dan Jurafsky, Christopher Manning, and Walter Powell. This project is supported by the Office of the President at Stanford University, the National Science Foundation under Grant No. 0835614, and the Boeing Company.

Authors’ addresses: J. Chuang, C. D. Manning, and J. Heer, 353 Serra Mall, Stanford, CA 94305; emails: {jcchuang, manning, jheer}@cs.stanford.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2012 ACM 1073-0516/2012/10-ART19 $15.00

DOI 10.1145/2362364.2362367 http://doi.acm.org/10.1145/2362364.2362367

2002]. While terms hand-selected by people are considered the gold standard, manually assigning keyphrases to thousands of documents simply does not scale.

To aid document understanding, keyphrase extraction algorithms select descriptive phrases from text. A common method is bag-of-words frequency statistics [Laver et al. 2003; Monroe et al. 2008; Rayson and Garside 2000; Robertson et al. 1981; Salton and Buckley 1988]. However, such measures may not be suitable for short texts [Boguraev and Kennedy 1999] and typically return single words, rather than more meaningful longer phrases [Turney 2000]. While others have proposed methods for extracting longer phrases [Barker and Cornacchia 2000; Dunning 1993; Evans et al. 2000; Hulth 2003; Kim et al. 2010; Medelyan and Witten 2006], researchers have yet to systematically evaluate the contribution of individual features predictive of keyphrase quality and often rely on assumptions—such as the presence of a reference corpus or knowledge of document structure—that are not universally applicable.

In this article, we characterize the statistical and linguistic properties of human-generated keyphrases. Our analysis is based on 5,611 responses from 69 students describing Ph.D. dissertation abstracts. We use our results to develop a two-stage method for automatic keyphrase extraction. We first apply a regression model to score candidate keyphrases independently; we then group similar terms to reduce redundancy and control the specificity of selected phrases. Through this research, we investigate the following concerns.

Reference Corpora. HCI researchers work with text from various sources, including data whose domain is unspecified or in which a domain-specific reference corpus is unavailable. We examine several frequency statistics and assess the trade-offs of selecting keyphrases with and without a reference corpus. While models trained on a specific domain can generate higher-quality phrases, models incorporating language-level statistics in lieu of a domain-specific reference corpus produce competitive results.

Document Diversity. Interactive systems may need to show keyphrases for a collection of documents. We compare descriptions of single documents and of multiple documents with varying levels of topical diversity. We find that increasing the size or diversity of a collection reduces the length and specificity of selected phrases.

Feature Complexity. Many existing tools select keyphrases solely using raw term counts or tf.idf scores [Salton and Buckley 1988], while recent work [Collins et al. 2009; Monroe et al. 2008] advocates more advanced measures, such as G^2 statistics [Dunning 1993; Rayson and Garside 2000]. We find that raw counts or tf.idf alone provide poor summaries but that a simple combination of raw counts and a term's language-level commonness matches the improved accuracy of more sophisticated statistics. We also examine the impact of features such as grammar and position information; for example, we find that part-of-speech tagging provides significant benefits over which more costly statistical parsing provides little improvement.

Term Similarity and Specificity. Multiword phrases identified by an extraction algorithm may contain overlapping terms or reference the same entity (person, place, etc). We present a method for grouping related terms and reducing redundancy. The resulting organization enables users to vary the specificity of displayed terms and allows applications to dynamically select terms in response to available screen space. For example, a keyphrase label might grow longer and more specific through semantic zooming.

We assess our resulting extraction approach by comparing automatically and manually selected phrases and via crowdsourced ratings. We find that the precision and recall of candidate keyphrases chosen by our model can match that of phrases hand-selected by human readers. We also apply our approach to tag clouds as an example of real-world presentation of keyphrases. We asked human judges to rate the quality of tag clouds using phrases selected by our technique and unigrams selected using G^2. We find that raters prefer the tag clouds generated by our method and identify other factors such as layout and prominent errors that affect judgments of keyphrase quality. Finally, we conclude the article by discussing the implications of our research for human-computer interaction, information visualization, and natural language processing.

2. RELATED WORK

Our research is informed by prior work in two surprisingly disjoint domains: (1) text visualization and interaction and (2) automatic keyphrase extraction.

2.1. Text Visualization and Interaction

Many text visualization systems use descriptive keyphrases to summarize text or label abstract representations of documents [Cao et al. 2010; Collins et al. 2009; Cui et al. 2010; Havre et al. 2000; Hearst 2009; Shi et al. 2010; Viégas et al. 2006, 2009]. One popular way of representing a document is as a tag cloud, that is, a list of descriptive words typically sized by raw term frequency. Various interaction techniques summarize documents as descriptive headers for efficient browsing on mobile devices [Buyukkokten et al. 2000, 2002; Yang and Wang 2003]. While HCI researchers have developed methods to improve the layout of terms [Cui et al. 2010; Viégas et al. 2009], they have paid less attention to methods for selecting the best descriptive terms.

Visualizations including Themail [Viégas et al. 2006] and TIARA [Shi et al. 2010] display terms selected using variants of tf.idf (term frequency by inverse document frequency [Salton and Buckley 1988])—a weighting scheme for information retrieval. Rarely are more sophisticated methods from computational linguistics used. One exception is Parallel Tag Clouds [Collins et al. 2009], which weight terms using G^2 [Dunning 1993], a probabilistic measure of the significance of a document term with respect to a reference corpus.

Other systems, including Jigsaw [Stasko et al. 2008] and FacetAtlas [Cao et al. 2010], identify salient terms by extracting named entities, such as people, places, and dates [Finkel et al. 2005]. These systems extract specific types of structured data but may miss other descriptive phrases. In this article, we first score phrases independent of their status as entities but later apply entity recognition to group similar terms and reduce redundancy.

2.2. Automatic Keyphrase Extraction

As previously indicated, the most common means of selecting descriptive terms is via bag-of-words frequency statistics of single words (unigrams). Researchers in natural language processing have developed various techniques to improve upon raw term counts, including removal of frequent “stop words,” weighting by inverse document frequency as in tf.idf [Salton and Buckley 1988] and BM25 [Robertson et al. 1981], heuristics such as WordScore [Laver et al. 2003], or probabilistic measures [Kit and Liu 2008; Rayson and Garside 2000] and the variance-weighted log-odds ratio [Monroe et al. 2008]. While unigram statistics are popular in practice, there are two causes for concern.

First, statistics designed for document retrieval weight terms in a manner that improves search effectiveness, and it is unclear whether the same terms provide good summaries for document understanding [Boguraev and Kennedy 1999; Collins et al. 2009]. For decades, researchers have anecdotally noted that the best descriptive terms are often neither the most frequent nor infrequent terms, but rather mid-frequency terms [Luhn 1958]. In addition, frequency statistics often require a large reference corpus and may not work well for short texts [Boguraev and Kennedy 1999]. As a result, it is unclear which existing frequency statistics are best suited for keyphrase extraction.

Second, the set of good descriptive terms usually includes multiword phrases as well as single words. In a survey of journals, Turney [2000] 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 using G^2 statistics to identify domain-specific bigrams (two-word phrases) [Dunning 1993]. Systems such as KEA++ or Maui use pseudo-phrases (phrases that remove stop words and ignore word ordering) for extracting longer phrases [Medelyan and Witten 2006]. Hulth considered all trigrams (phrases up to length of three words) in her algorithm [2003]. While the inclusion of longer phrases may allow for more expressive keyphrases, systems that permit longer phrases can suffer from poor precision and meaningless terms. The inclusion of longer phrases may also result in redundant terms of varied specificity [Evans et al. 2000], 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 [Barker and Cornacchia 2000; Daille et al. 1994; Evans et al. 2000; Hulth 2003] filter candidate keyphrases by identifying noun phrases using a part-of-speech tagger or a parser. Of note is the use of so-called *technical terms* [Justeson and Katz 1995] that match regular expression patterns over part-of-speech tags. To reduce redundancy, Barker and Cornacchia [2000] choose the most specific keyphrase by eliminating any phrases that are a subphrase of another. Medelyan and Witten's KEA++ system [2006] trains a naïve Bayes classifier to match keyphrases produced by professional indexers. However, all existing methods produce a *static* list of keyphrases and do not account for task- or application-specific requirements.

Recently, the Semantic Evaluation (SemEval) workshop [Kim et al. 2010] held a contest comparing the performance of 21 keyphrase extraction algorithms over a corpus of ACM Digital Library articles. The winning entry, named HUMB [Lopez and Romary 2010], ranks terms using bagged decision trees learned from a combination of features, including frequency statistics, position in a document, and the presence of terms in ontologies (e.g., MeSH, WordNet) or in anchor text in Wikipedia. Moreover, HUMB explicitly models the structure of the document to preferentially weight the abstract, introduction, conclusion, and section titles. The system is designed for scientific articles and intended to provide keyphrases for indexing digital libraries.

The aims of our current research are different. Unlike prior work, we seek to systematically evaluate the contributions of individual features to keyphrase quality, allowing system designers to make informed decisions about the trade-offs of adding potentially costly or domain-limiting features. We have a particular interest in developing methods that are easy to implement, computationally efficient, and make minimal assumptions about input documents.

Second, our primary goal is to improve the design of text visualization and interaction techniques, not the indexing of digital libraries. This orientation has led us to develop techniques for improving the quality of extracted keyphrases as a whole, rather than just scoring terms in isolation (cf., [Barker and Cornacchia 2000; Turney 2000]). We propose methods for grouping related phrases that reduce redundancy and enable applications to dynamically tailor the specificity of keyphrases. We also evaluate our approach in the context of text visualization.

3. CHARACTERIZING HUMAN-GENERATED KEYPHRASES

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

3.1. User Study Design

We asked graduate students to provide descriptive phrases for a collection of Ph.D. dissertation abstracts. We selected 144 documents from a corpus of 9,068 Ph.D. dissertations 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 coverage over a variety of disciplines, we selected abstracts each from the following six departments: Computer Science, Mechanical Engineering, Chemistry, Biology, Education, and History. We recruited graduate students from two universities via student email lists. Students came from departments matching the topic areas of selected abstracts.

3.1.1. Study Protocol. We selected 24 dissertations (as eight groups of three documents) from each of the six departments in the following manner. We randomly selected eight faculty members from among all faculty who have graduated at least ten Ph.D. students. For four of the faculty members, we selected the three most topically diverse dissertations. For the other four members, we selected the three most topically similar dissertations.

Subjects participated in the study over the Internet. They were presented with a series of webpages and asked to read and summarize text. Subjects received three groups of documents in sequence (nine in total); they were required to complete one group of documents before moving on to the next group. For each group of documents, subjects first summarized three individual documents in a sequence of three webpages and then summarized the three as a whole on a fourth page. Participants were instructed to summarize the content using five or more keyphrases, using any vocabulary they deemed appropriate. Subject were not constrained to only words from the documents. They would then repeat this process for two more groups. The document groups were randomly selected such that they varied between familiar and unfamiliar topics.

We received 69 completed studies, comprising a total of 5,611 free-form responses: 4,399 keyphrases describing single documents and 1,212 keyphrases describing multiple documents. Note that while we use the terminology keyphrase in this article 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.”

3.1.2. Independent Factors. We varied the follwing 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 as the presented text. We relied on self-reports to determine subjects’ familiarity.

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 of the angle between tf.idf term vectors. Our experimental setup provided sets of three documents with either low or high topical similarity.

3.1.3. Dependent Statistical and Linguistic Features. To analyze responses, we computed the following features for the documents and subject-authored keyphrases. 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 showed to subjects, while *responses* are the provided summary keyphrases. We tokenize text based on the Penn Treebank standard [Marcus et al. 1993] and extract all terms of up to length five. 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 inflectional suffixes removed. We apply light stemming [Minnen et al. 2001] 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.

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 (*Web term frequency*), as indicated by the Google Web *n*-gram corpus [Brants and Franz 2006]. 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 frequency of "the"—the most frequent unigram. When distinctions are needed, 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 (cf., [Medelyan and Witten 2006]). However, frequent terms are more likely to appear earlier 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 (formally defined in Section 4.3.1) is the probability that a term's first occurrence is lower than that 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 term position as an independent feature.

We annotate terms that are *noun phrases*, *verb phrases*, or match *technical term* patterns [Justeson and Katz 1995] (see Table I). Part-of-speech information is determined using the Stanford POS Tagger [Toutanova et al. 2003]. We additionally determine grammatical information using the Stanford Parser [Klein and Manning 2003] and annotate the corresponding words in each sentence.

3.2. Exploratory Analysis of Human-Generated Phrases

Using these features, we characterized the collected human-generated keyphrases in an exploratory analysis. Our results confirm observations from prior work—the prevalence of multiword phrases [Turney 2000], preference for mid-frequency terms [Luhn 1958], and pronounced use of noun phrases [Barker and Cornacchia 2000; Daille et al. 1994; Evans et al. 2000; Hulth 2003]—and provide additional insights, including the effects of document count and diversity.

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

Assessing the prevalence of words versus phrases, Figure 2 shows that bigrams are the most common 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

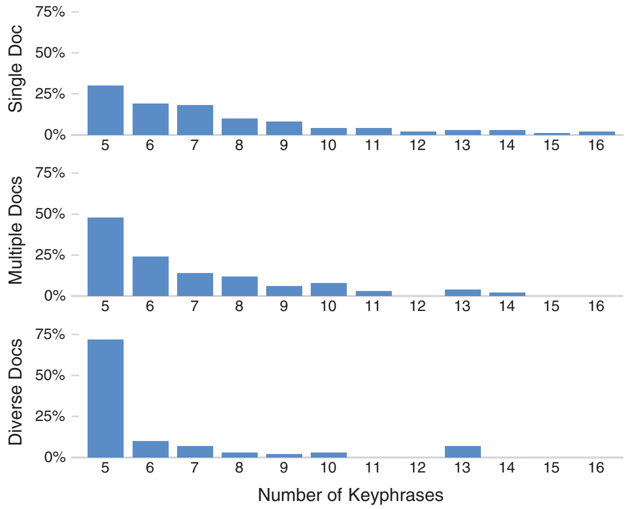


Fig. 1. How many keyphrases do people use? Participants use fewer keyphrases to describe multiple documents or documents with diverse topics, despite the increase in the amount of text and topics.

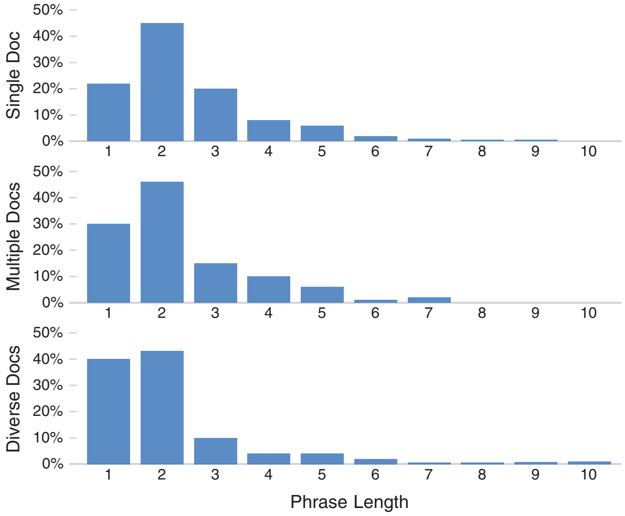


Fig. 2. Do people use words or phrases? Bigrams are the most common. For single documents, 75% of responses contain multiple words. Unigram use increases with the number and diversity of documents.

decrease in the use of trigrams and longer terms. The prevalence of bigrams confirm prior work [Turney 2000]. By permitting users to enter any response, our results provide additional data on the tail end of the distribution: there is minimal gain when assessing the quality of phrases longer than five words, which account for <5% of responses.

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 [1958]. As the number of documents and topic diversity increases, the distribution shifts toward more common terms. We found similar correlations for corpus commonness.

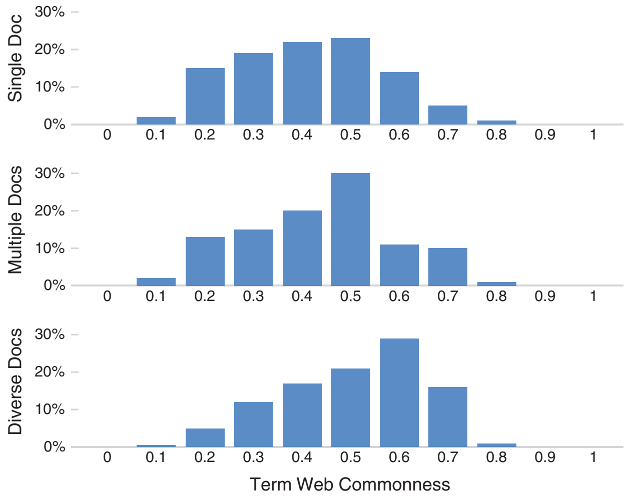


Fig. 3. Do people use generic or specific terms? Term commonness increases with the number and diversity of documents.

Table I. Technical Terms

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

Note: Technical terms are defined by part-of-speech regular expressions. N is a noun, A an adjective, and C a cardinal number. We modify the definition of technical terms [Justeson and Katz 1995] by permitting cardinal numbers as the trailing word. Examples of technical terms include the following: *hardware*, *interactive visualization*, *performing arts*, *Windows 95*. Examples of compound technical terms include the following: *gulf of execution*, *War of 1812*.

Table II. Positional and Grammatical Statistics

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%

Note: Position and grammar features of keyphrases present in a document (65% of total). Keyphrases occur earlier in a document: two-thirds are noun phrases, over four-fifths are technical terms.

For each user-generated keyphrase, we find matching text in the reading and note that 65% of the responses are present in the document. Considering for the rest of this paragraph just the two-thirds of keyphrases present in the readings, the associated *positional* and *grammatical* properties of this subset are summarized in Table II. 22% of keyphrases occur 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 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 (i.e., continuous subsequence fully contained in the 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.

In summary, our exploratory analysis shows that subjects primarily choose multi-word phrases, prefer terms with medium 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.

4. STATISTICAL MODELING OF KEYPHRASE QUALITY

Informed by our exploratory analysis, we systematically assessed the contribution of statistical and linguistic features to keyphrase quality. Our final result is a pair of regression models (one corpus-dependent, the other independent) that incorporate term frequency, commonness, position, and grammatical features.

We modeled keyphrase quality using logistic regression. We chose this model because its results are readily interpretable: contributions from each feature can be statistically assessed, and the regression value can be used to rank candidate phrases. We initially used a mixed model [Faraway 2006], which extends generalized linear models to let one assess random effects, to include variation due to subjects and documents. We found that the random effects were not significant and so reverted to a standard logistic regression model.

We constructed the models over 2,882 responses. We excluded user-generated keyphrases longer than five words (for which we are unable to determine term commonness; our data on Web commonness contains only n-grams up to length five) or not present in the documents (for which we are unable to determine grammatical and positional information). We randomly selected another set of 28,820 phrases from the corpus as negative examples, with a weight of 0.1 (so that total weights for positive examples and negative examples are equal during model fitting). Coefficients generated by logistic regression represent the best linear combination of features that differentiate user-generated responses from the random phrases.

We examine three classes of features—frequency statistics, grammar, and position—visited in order of their predictive accuracy as determined by a preliminary analysis. Unless otherwise stated, all features are added to the regression model as independent factors without interaction terms.

We present only modeling results for keyphrases describing single documents. We did fit models for phrases describing multiple documents, and they reflect observations from the previous section, for example, weights shifted toward higher commonness scores. However, the coefficients for grammatical features exhibit large standard errors, suggesting that the smaller data set of multi-document phrases (641 phrases vs. 2,882 for single docs) is insufficient. As a result, we leave further modeling of multi-document descriptions to future work.

We evaluate features using precision-recall curves. Precision and recall measure the accuracy of an algorithm by comparing its output to a known "correct" set of phrases; in this case, the list of user-generated keyphrases up to length five. Precision measures the percentage of correct phrases in the output. Recall measures the total percentage of the correct phrases captured by the output. As more phrases are included, recall increases but precision decreases. The precision-recall curve measures the performance of an algorithm over an increasing number of output phrases. Higher precision is desirable with fewer phrases, and a larger area under the curve indicates better performance.

Table III. Frequency Statistics

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

Note: Given a document from a reference corpus with N documents, the score for a term is given by these formulas. t_{Doc} and t_{Ref} denote term frequency in the document and reference corpus; T_{Doc} and T_{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 term frequencies in each document by 0.01; terms present in the corpus but not in the document are defined as $t_{\overline{\mathrm{Doc}}} = t_{\mathrm{Ref}} - t_{\mathrm{Doc}}$ and $T_{\overline{\mathrm{Doc}}} = T_{\mathrm{Ref}} - T_{\mathrm{Doc}}$. Among the family of tf.idf measures, we selected a reference-relative form as shown. For BM25, the parameters $k_1 = 2$ and $b = 0.75$ are suggested by Manning et al. [2008]. A term is any analyzed phrase (n-gram). When frequency statistics are applied to n-grams with $n = 1$, the terms are all the individual words in the corpus. When $n = 2$, scoring is applied to all unigrams and bigrams in the corpus, and so on.

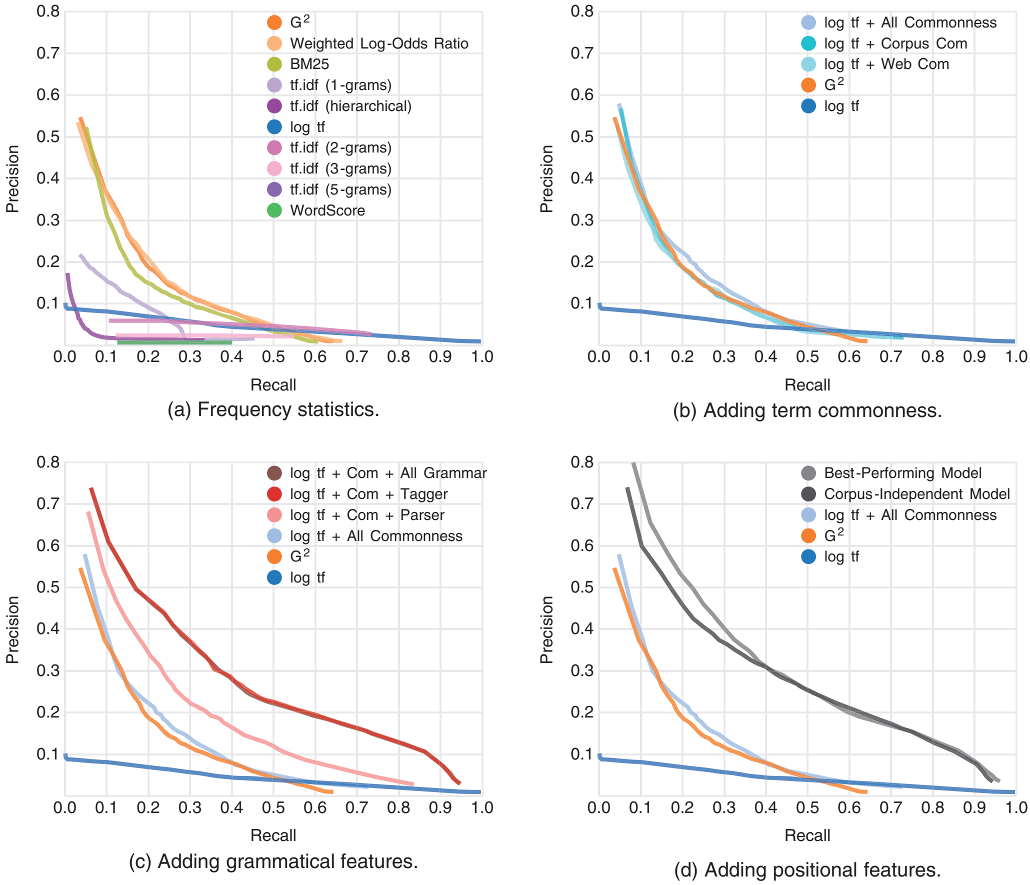
We also assessed each model using model selection criteria (i.e., AIC, BIC). As these scores coincide with the rankings from precision-recall measures, we omit them.

4.1. Frequency Statistics

We computed seven different frequency statistics. Our simplest measure was log term frequency: *log (tf)*. We also computed *tf.idf*, *BM25*, *G*$^{\mathbf{2}}$, *variance-weighted log-odds ratio*, and *WordScore*. Each requires a reference corpus, for which we use the full dissertation collection. We also created a set of *hierarchical tf.idf* scores (e.g., as used by Viégas et al. in Themail [2006]) 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 III.

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 due to an excessive number of ties (many 4- and 5-grams occur only once in the corpus). However, tf.idf on unigrams still performs much worse than probabilistic approaches.

4.1.1. Adding Term Commonness. During keyphrase characterization, we observed a bell-shaped distribution of keyphrases as a function of commonness. We quantiled 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 up to 20 bins.



(a) Frequency statistics.

(b) Adding term commonness.

(c) Adding grammatical features.

(d) Adding positional features.

Fig. 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) Grammatical features improve performance. (d) Positional features provide further gains for both a complete model and a simplified corpus-independent model.

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 also measured the effects due to bin count. Precision-recall increases as the number of bins are increased up to about five bins, and there is marginal gain between five and eight bins. Examining the regression coefficients for a large number of bins (ten bins or more) shows large random fluctuations, indicating overfitting. As expected, the coefficients for commonness peak at middle frequency (see Table V). 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; tf.idf scores have a slight negative correlation with keyphrase quality.

4.2. Grammatical Features

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

4.2.1. Parser Features. For each term extracted from the text, we tag the term as a *full noun phrase* or *full verb phrase* if it 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. We add two additional features that are associated with words at the boundary of a noun phrase. Leading words in a noun phrase are referred to as *optional leading words* if their part-of-speech is one of *cardinal number*, *determiner*, or *pre-determiner*. The last word in a noun phrase is the *head noun*. If the first word of a term is an optional leading word or if the last word of a term is a head noun, then the term is tagged accordingly. These two features occur only if the beginning or end of the term is aligned with a noun phrase boundary.

4.2.2. Tagger features. Phrases that match technical term patterns (Table I) are tagged as either a *technical term* or *compound technical term*. Phrases that match a substring in a technical term are tagged as *partial* or *partial compound technical terms*.

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 improves precision and recall more than parser-related features. Combining both POS and parser features yields only a marginal improvement. Head nouns (cf., [Barker and Cornacchia 2000]) did not have a measurable effect on keyphrase quality. The results indicate that statistical parsing may be avoided in favor of POS tagging.

4.3. Positional Features and Final Models

Finally, we introduce *relative first occurrence* and *presence in first sentence* as positional features; both predictors are statistically significant.

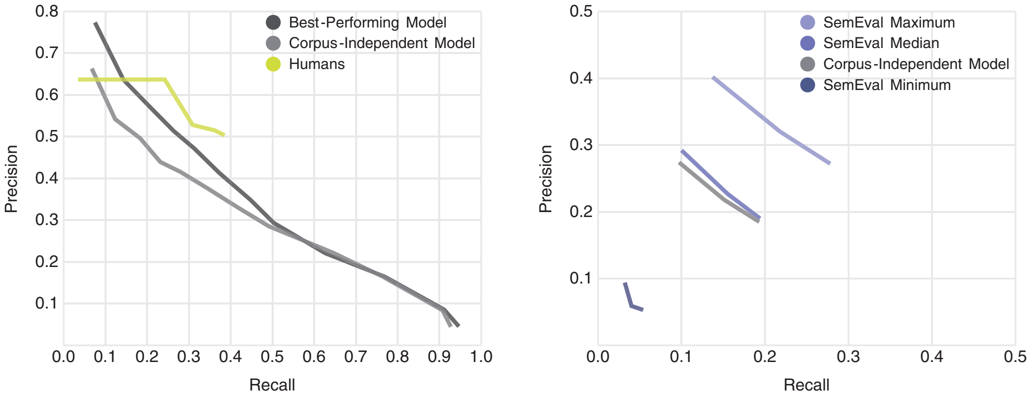
4.3.1. First Occurrence. The *absolute first occurrence* of a term is the earliest position in the document at which a term appears, normalized between 0 and 1. If a term is the first word of a document, its absolute first occurrence is 0. If the only appearance of a term is as the last word of a document, its absolute first occurrence is 1. The absolute first occurrences of frequent terms tend to be earlier in a document, due to their larger number of appearances.

We introduce *relative first appearance* to have a measure of early occurrence of a word independent of its frequency. Relative first occurrence measures how likely a term is to initially appear earlier than a randomly sampled phrase of the same frequency. Let $P(W)$ denote the the expected position of words W in the document. As a null hypothesis, we assume that words are uniformly distributed $P(W) \sim \text{Uniform}[0, 1]$. The expected absolute first occurrence of a randomly selected term that appears k times in the document is the minimum of the k instantiations of the term $P(w_1), \ldots, P(w_k)$ and is given by the following probability distribution.

$$\min_{i=1}^{k} P(w_i) = \eta(1 - x)^{k-1},$$

for position $x \in [0, 1]$ and some normalization constant η. Suppose a term w' occurs k times in the document and its first occurrence is observed to be at position $a \in [0, 1]$. Its relative first occurrence is the cumulative probability distribution from a to 1.

$$\text{Relative first occurrence of } w' = \int_a^1 \min_{i=1}^{k} P(w_i) = \int_a^1 \eta\,(1 - x)^{k-1}\,dx = (1 - a)^k .$$



(a) Comparison with human-selected phrases.

(b) Comparison with SemEval 2010.

Fig. 5. Precision-recall curves for keyphrase regression models. Legends are sorted by decreasing initial precision. (a) Comparison with human-selected keyphrases; our models provide higher precision at low recall values. (b) Comparison with SemEval 2010 [Kim et al. 2010] results for 5, 10, and 15 phrases; our corpus-independent model closely matches the median scores.

Combining log(tf), commonness (five bins), grammatical, and positional features we built two final models for predicting keyphrase quality. Our full model is based on all significant features using our dissertation corpus as reference. In our simplified model (Table V), we excluded corpus commonness and statistical parsing to eliminate corpus dependencies and improve runtime. Omitting the more costly features incurs a slight decrease in precision, as shown in Figure 4(d).

4.4. Model Evaluation

We evaluated our models in two ways. First, we compared the performance of our models with that of our human judges. Second, we compared our techniques with results from the Semantic Evaluation (SemEval) contest of automatic keyphrase extraction methods [Kim et al. 2010].

4.4.1. Comparison with Human-Selected Keyphrases. We compared the precision-recall of keyphrases extracted using our methods to human-generated keyphrases. 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, phrases selected by one participant can only be matched against responses from the $K - 1$ other remaining 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 the participant’s responses with our model’s output, ensuring that both 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 5(a) 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.

4.4.2. Comparison with SemEval 2010 Contest Task #5. Next we compared the precision-recall performance of our corpus-independent model to the results of the SemEval

Table IV. Regression Coefficients for the Full (Corpus-Dependent) Model Based on the Ph.D. Dissertations

Model Feature	Regression Coefficients
(intercept)	−2.88114***
log(tf)	0.74095***
WC ∈ (0%, 20%]	0.08894
WC ∈ (20%, 40%]	0.04390
WC ∈ (40%, 60%]	−0.19786
WC ∈ (60%, 80%]	−0.46664*
WC ∈ (80%, 100%]	−1.26714***
CC ∈ (0%, 20%]	0.20554
CC ∈ (20%, 40%]	0.39789**
CC ∈ (40%, 60%]	0.24929
CC ∈ (60%, 80%]	−0.34932
CC ∈ (80%, 100%]	−0.97702**
relative first occurrence	0.52950***
first sentence	0.83637**
partial noun phrase	0.14117
noun phrase	0.29818*
head noun	−0.16509
optional leading word	0.46481*
partial verb phrase	0.15639
verb phrase	1.12310*
full technical term	−0.58959
partial technical term	1.37875*
full compound technical term	1.09713
partial compound technical term	1.10565*

Note: WC = Web commonness, *CC* = corpus commonness; statistical significance = *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$.

Table V. Regression Coefficients for Corpus-Independent Model

	Regression Coefficients	
Model Feature	*Dissertations*	*SemEval*
(intercept)	−2.83499***	−5.4624**
log(tf)	0.93894***	2.8029*
WC ∈ (0%, 20%]	0.17704	0.8561
WC ∈ (20%, 40%]	0.23044*	0.7246
WC ∈ (40%, 60%]	0.01575	0.4153
WC ∈ (60%, 80%]	−0.62049***	−0.5151
WC ∈ (80%, 100%]	−1.90814***	−2.2775
relative first occurrence	0.48002**	−0.2456
first sentence	0.93862***	0.9173
full tech. term	−0.50152	1.1439
partial tech. term	1.44609**	3.4539***
full compound tech. term	1.13730	1.0920
partial compound tech. term	1.18057*	2.0134

Note: WC = web commonness; statistical significance = *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$.

2010 contest. Semantic Evaluation (SemEval) is a series of workshops focused on evaluating methods for specific text analysis problems. Task 5 of SemEval 2010 [Kim et al. 2010] compared 21 keyphrase extraction algorithms for scientific articles. A total of 244 articles from four different subdisciplines were chosen from the ACM Digital Library. Contestants received 144 articles for training; the submitted techniques were then tested on the remaining 100 articles. Three classes of keyphrases were evaluated: author-assigned, reader-assigned, and the combination of both. Reader-assigned phrases were provided by volunteers who were given five papers and instructed to spend 10–15 minutes per paper generating keyphrases. For each class, precision and recall were computed for the top 5, 10, and 15 keyphrases.

We used this same data to evaluate the performance of our corpus-independent modeling approach trained on the SemEval corpus. The coefficients of our SemEval model differ slightly from those of our Stanford dissertations model (Table V), but the relative feature weightings remain similar, including a preference for mid-commonness terms, a strong negative weight for high commonness, and strong weights for technical term patterns.

Figure 5(b) compares our precision-recall scores against the distribution of SemEval results for the combined author- and reader-assigned keyphrases. Our corpus-independent model closely matches the median scores. Though intentionally simplified, our approach matches or outperforms half of the contest entries. This outcome is perhaps surprising, as competing techniques include more assumptions and complex features (e.g., leveraging document structure and external ontologies) and more sophisticated learning algorithms (e.g., bagged decision trees vs. logistic regression). We believe these results argue in favor of our identified features.

4.4.3. Lexical Variation and Relaxed Matching. While we are encouraged by the results of our precision-recall analysis, some skepticism is warranted. Up to this point, our analysis has concerned only exact matches of stemmed terms. In practice, it is reasonable to expect that both people and algorithms will select keyphrases that do not match exactly but are lexically and/or conceptually similar (e.g., “analysis” vs. “data analysis”). How might the results change if we permit a more relaxed matching?

To gain a better sense of lexical variation among keyphrases, we analyzed the impact of a relaxed matching scheme. We experimented with a number of matching approaches by permitting insertion or removal of terms in phrases or re-arrangement of terms in genitive phrases. For brevity, we report on just one simple but effective strategy: we consider two phrases “matching” if they either match exactly or if one can induce an exact match by adding a single word to either the beginning or the end of the shorter phrase.

Permitting relaxed matching significantly raises the proportion of automatically extracted keyphrases that match human-selected terms. Considering just the top-ranked term produced by our model for each document in the SemEval contest, 30.0% are exact matches, while 75.0% are relaxed matches. Looking at the top five terms per document, 27.4% exactly match a human-selected term, permitting a relaxed match increases this number to 64.2%. These results indicate that human-selected terms regularly differ from our automatically extracted terms by a single leading or trailing word. This observation suggests that (a) precision-recall analysis may not reveal the whole picture and (b) related keyphrases might vary in length but still provide useful descriptions. We now build upon this insight to provide means for parameterizing keyphrase selection.

5. KEYPHRASE GROUPING AND SELECTION

The previous section describes a method for scoring keyphrases in isolation. However, candidate keyphrases may overlap (e.g., “visualization”, “interactive visualization”) or reference the same entity (e.g., “Barack Obama”, “President Obama”). Keyphrase selection might be further improved by identifying related terms. An intelligent grouping can also provide a means to interactively parameterizing the display of keyphrases. Users might request shorter/longer—or more general/more specific—terms. Alternatively, a user interface might automatically vary term length or specificity to optimize the use of the available screen space. Once we have extracted a set of candidate keyphrases, we can next optimize the overall quality of that set. Here we present a simple

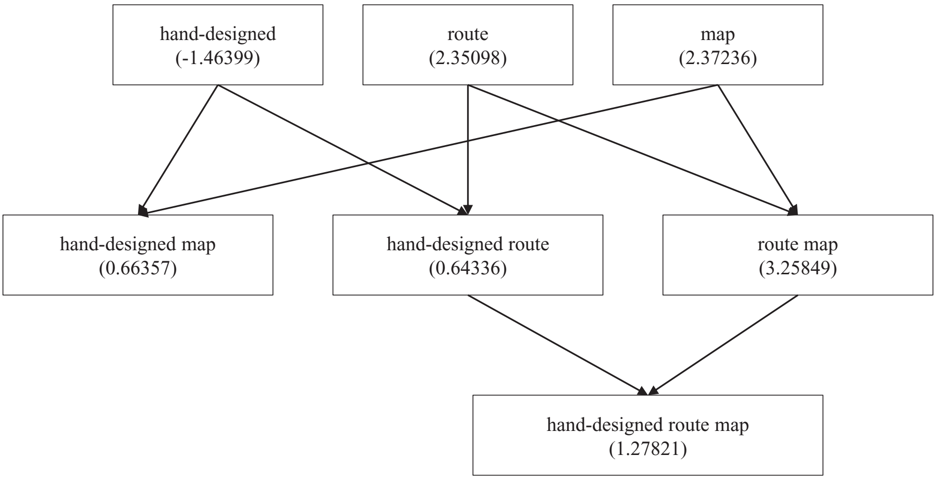


Fig. 6. Term grouping. The graph shows a subset of unigrams, bigrams, and trigrams considered to be conceptually similar by our algorithm. Connected terms differ by exactly one word at the start or the end of the longer phrase. Values in parentheses are the scores from our simplified model for the dissertation "Visualizing Route Maps." By default, our algorithm displays the keyphrase "*route map*" and suppresses "*route*", "*map*", and "*hand-designed route maps*". Users may choose to display a shorter word ("*map*") or longer phrase ("*hand-designed route map*") to describe this document.

approach for filtering and selecting keyphrases—sufficient for removing a reasonable amount of redundancy and adapting keyphrase specificity on demand.

5.1. 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 instead of lexical variations. For example, the following list of keyphrases differ lexically but are 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.

We use string similarity to approximate 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 top-ranked keyphrases, this assumption is true. Figure 6 shows an example of terms considered conceptually similar by our algorithm.

We also account for the special case of names. We apply named entity recognition [Finkel et al. 2005] to identify persons, locations, and organizations. To resolve entities, we consider two people identical if the trailing substring of one matches the trailing substring of the other. For example, "*Obama*", "*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 to be identical, for example, "*Intel*" and "*Intel Corporation*." We also apply acronym recognition [Schwartz and Hearst 2003] 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 will likely be needed for robust entity and acronym resolution. Figure 7 shows additional typed edges connecting terms that our algorithm considers as referring to the same entity.

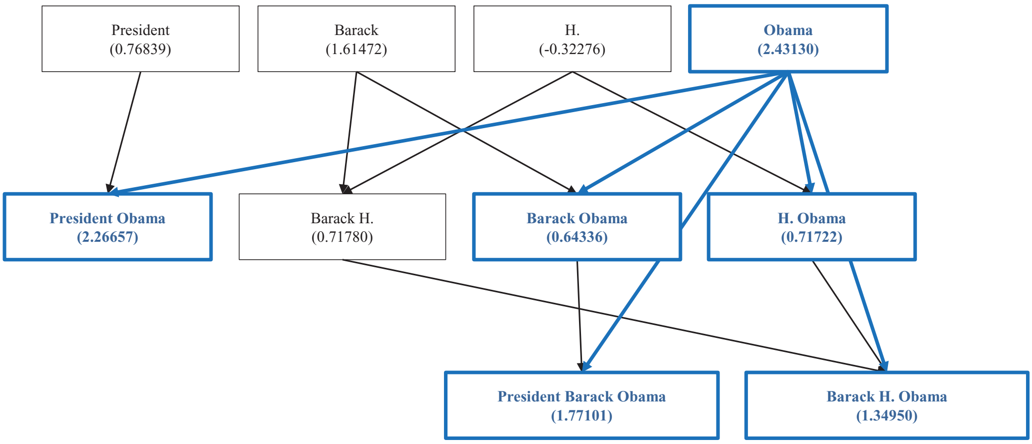


Fig. 7. Term grouping for named entities and acronyms. The graph shows typed edges that embed additional relationships between terms in a document about President Obama. Black edges represent basic term grouping based on string similarity. Bold blue edges represent people: terms that share a common trailing substring and are tagged as “person” by a named entity recognition algorithm. By default, our algorithm displays “*Obama*” to summarize the text. Users may choose to show a longer phrase “*President Obama*” or display a longer and more specific description “*President Barack Obama*” by shifting the scores along the typed edges. Users may also apply type-specific operations, such as showing the longest name without honorifics, “*Barack H. Obama*.”

5.2. Length and Specificity Adjustment

Once similar terms have been grouped, we must select which term to present. To parameterize final keyphrase selection, we allow users to optionally choose longer/shorter and more generic or specific terms. We use two simple features to determine which form of similar phrases to display: term length and term commonness. 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. The operation is equivalent to shifting the weights along edges in Figures 6 and 7.

Other adjustments can be specified directly by users. For recognized people, users can choose to expand all names to full names or contract to last names. For locations and organizations, users can elect to use the full-length or shortened form. For identified acronyms, users may choose to expand or contract the terminology. In other words, for each subgraph of terms connected by named entity typed edges, the user may choose to assign the maximum node weight to any other nodes in the subgraph. In doing so, the chosen term is displayed suppressing all other alternative forms.

6. QUALITATIVE INSPECTION OF SELECTED KEYPHRASES

As an initial evaluation of our two-stage extraction approach, we compared the top 50 keyphrases produced by our 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. Terms from the 9,068 Ph.D. dissertations are used as the reference corpus for all methods except our simplified model, which is corpus independent. We applied redundancy reduction to the output of each extraction method.

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 start producing unhelpful terms after the top ten results. The difference is exacerbated for short texts. For example, in a 59-word article about San Francisco's Mission District, our algorithm returns noun phrases such as "*colorful Latino roots*" and "*gritty bohemian subculture*", while the other methods produce only one to three usable phrases: "*Mission*", "*the District*", or "*district*." In these cases, our method benefits from grammatical information.

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). Earlier in the exploratory analysis, we found that the inclusion of optional leading words degrades the quality of descriptive phrases. However, many phrases tend to be preceded by the same determiner and pre-determiner. Without a sufficiently large reference corpus, statistics alone often cannot separate meaningful phrases from common leading words. By applying technical term matching patterns, our model naturally excludes most types of non-descriptive leading words and produces more grammatically appropriate phrases, such as "*long exposure*" (our models) versus "*a long exposure*" (G^2, BM25, weighted log-odds ratio). Even though term commonness favors mid-frequency phrases, our model can still select salient words from all commonness levels. For example, from an article about the technologies in Google versus Bing, our models choose "*search*" (common word), "*navigation tools*" (mid-frequency phrase), and "*colorful background*" (low-frequency phrase), while all other methods output only "*search*".

We observe few differences between our full and simplified models. Discernible differences are typically due to POS tagging errors. 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 omit useful verb phrases, such as "*civilians killed*" in a news article about the NATO forces in Afghanistan. Our algorithm chooses "*civilian casualties*" 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 [Justeson and Katz 1995]. 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, we often observe redundant keyphrases similar in term length, concept, or identity. For example, "*Mission*", "*Mission District*", and "*Mission Street*" in an article about San Francisco. Our heuristics based on string similarity, named entity recognition, and acronym recognition improve the returned keyphrases (see Tables VI and VII). As we currently consider single-term differences only, some redundancy is still present.

7. CROWDSOURCED RATINGS OF TAG CLOUDS

We evaluated our extracted keyphrases in a visual form and asked human judges to rate the relative quality of tag cloud visualizations with terms selected using both our technique (i.e., simplified model) and G^2 scores of unigrams (cf., [Collins et al. 2009; Dunning 1993; Rayson and Garside 2000]). We chose to compare tag cloud visualizations for multiple reasons. First, keyphrases are often displayed as part of a webpage

Table VI. Top Keyphrases

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

Note: Top 25 keyphrases for an open letter from Adobe about Flash technologies. We apply redundancy reduction to both lists.

Table VII. Term-Length Adjustment

Flash	←	**Flash Player**	→	Flash Player 10.1
devices	←	**mobile devices**	→	Apple mobile devices
happiness	←	national happiness	←	**Gross national happiness**
emotion	←	**emotion words**	→	use of emotion words
networks	←	**social networks**	→	online social networks
Obama	←	President Obama	←	**Barack H. Obama**
Bush	←	President Bush	←	**George H.W. Bush**
		WoW	→	World of Warcraft

Note: Examples of adjusting keyphrase length. Terms in boldface are selected by our corpus-independent model. Adjacent terms show the results of dynamically requesting shorter (←) or longer (→) terms.

or text visualization; we hypothesize that visual features such as layout, sizing, term proximity, and other aesthetics are likely to affect the perceived utility of and preferences for keyphrases in real-world applications. Tag clouds are a popular form used by a diverse set of people [Viégas et al. 2009]. Presenting selected terms in a simple list would fail to reveal the impact of these effects. Second, keyphrases are often displayed in aggregate; we hypothesize that the perceived quality of a collective set of keyphrases differs from that of evaluating each term independently. Tag clouds encourage readers to assess the quality of keyphrases as a whole.

Parallel Tag Clouds [Collins et al. 2009] use unigrams weighted by G^2 for text analytics, making G^2 statistics an interesting and ecologically valid comparison point. We hypothesized that tag clouds created using our technique would be preferred due to more descriptive terms and complete phrases. We also considered variable-length G^2 that includes phrases up to 5-grams. Upon inspection, many of the bigrams (e.g., "*more about*", "*anyone can*") and the majority of trigrams and longer phrases selected by G^2 statistics are irrelevant to the document content. We excluded the results from the study, as they were trivially uncompetitive. Including only unigrams results in shorter terms, which may lead to a more densely-packed layout (this is another reason that we chose to compare to G^2 unigrams).

7.1. Method

We asked subjects to read a short text passage and write a 1–2 sentence summary. Subjects then viewed two tag clouds and were asked to rate which they preferred on a 5-point scale (with 3 indicating a tie) and 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 corpus-independent model with redundancy reduction and with the default preferred length. We weighted our terms by their regression score: the linear combination of features used as input to the logistic function. 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 [Viégas et al. 2009] using the same layout and color parameters for each. We randomized the presentation order of the tag clouds.

We included tag clouds of 24 text documents. To sample a variety of genres, we used documents in four categories: CHI 2010 paper abstracts, short biographies (three U.S. presidents, three musicians), blog posts (two each from opinion, travel, and photography blogs), and news articles. Figure 8 shows tag clouds from a biography of the singer Lady Gaga; Figures 9 and 10 show two other clouds used in our study.

We conducted our study using Amazon's Mechanical Turk (cf., [Heer and Bostock 2010]). Each trial was posted as a task with a US$0.10 reward. We requested 24 assignments per task, resulting in 576 ratings. Upon completion, we tallied the ratings for each tag cloud and coded free-text responses with the criteria invoked by raters' rationales.

7.2. 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.").

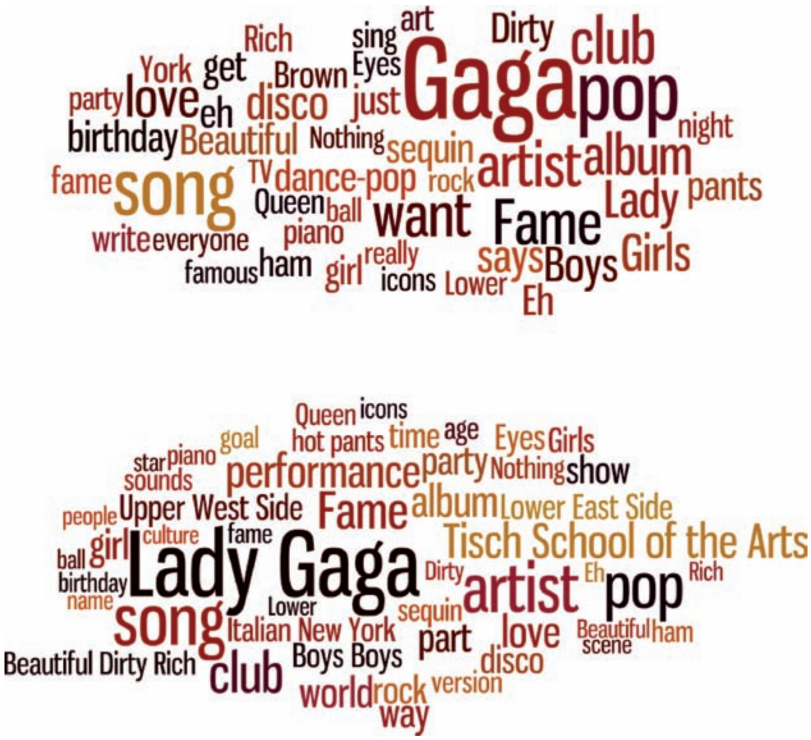


Fig. 8. Tag cloud visualizations of an online biography of the pop singer Lady Gaga. (top) Single-word phrases (unigrams) weighted using G^2. (bottom) Multiword phrases, including significant places and song titles, selected using our corpus-independent model.

Tag clouds in both conditions were sometimes preferred due to visual features, such as layout, 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. Confirming our hypothesis, layout affects tag cloud ratings.

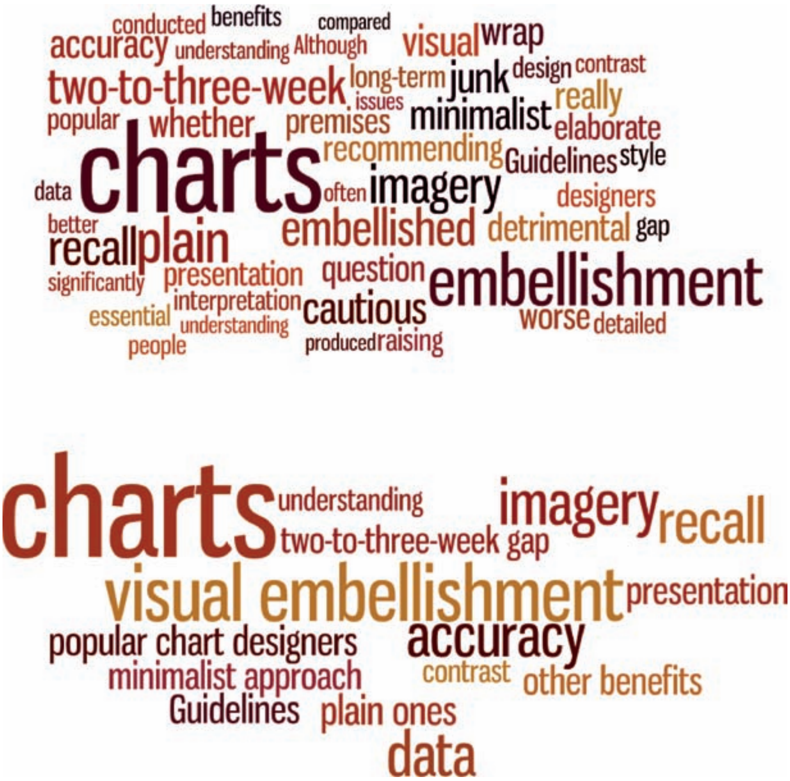


Fig. 9. Tag clouds for a research paper on chart perception. (top) Unigrams weighted using G^2. (bottom) Multiword phrases selected by our method.

The ability to dynamically adjust keyphrase length, however, can produce alternative terms and may allow users to generate tag clouds with better spatial properties.

8. IMPLICATIONS FOR HCI, VISUALIZATION, AND NLP

In this section, we highlight our contributions to the fields of human-computer interaction, information visualization, and natural language processing. First, we summarize our experiences and distill them in a set of design guidelines. Second, we demonstrate how our work can enable novel interactive visualizations. Finally, our keyphrase extraction algorithm is the cumulative result of applying HCI methods to collect data, analyze, develop, and evaluate text summarization techniques. We review the process through which we arrived at our model and emphasize how HCI concepts and approaches can help advance research in natural language processing and other fields.

8.1. Guidelines for Human-Centered Design

We summarize the key lessons from our study and evaluations and distill them in the following set of guidelines on designing text visualizations and model feature selection.

Multiword phrases. Our results find that multiword phrases—particularly bigrams—often provide better descriptions than unigrams alone. In the case of multiple documents, this decision may need to be traded off against the better aggregation afforded by unigrams. Designers may wish to give users the option to

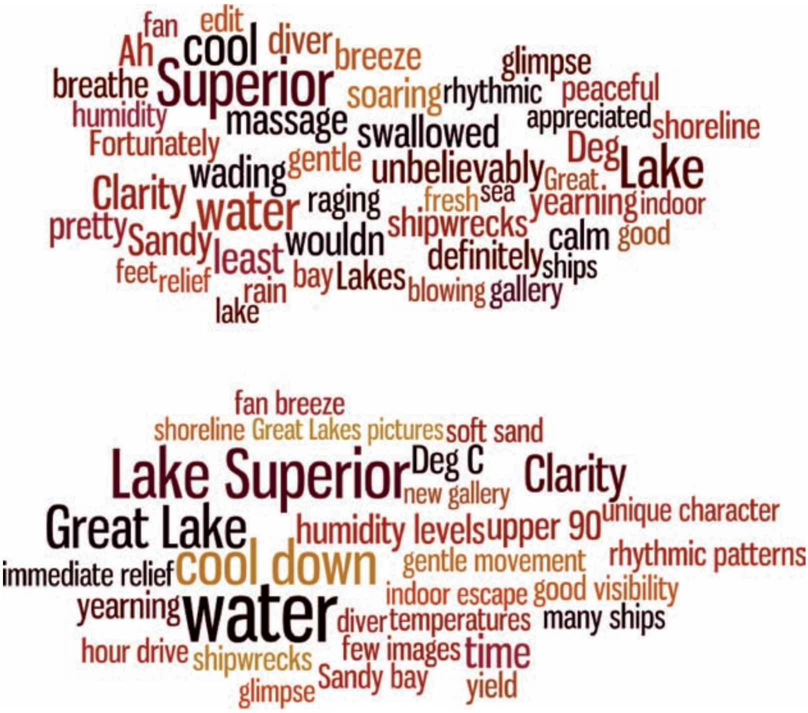


Fig. 10. Tag clouds for a travel article. (top) Unigrams weighted using G^2. (bottom) Multiword phrases selected by our method.

parameterize phrase length. Our grouping approach (§5) provides a means of parameterizing selection while preserving descriptive quality.

Choice of frequency statistics. In our studies, probabilistic measures such as G^2 significantly outperformed common techniques, such as raw term frequency and tf.idf. Moreover, a simple linear combination of log term frequency and Web commonness matches the performance of G^2 without the need of a domain-specific reference corpus. We advocate using these higher-performing frequency statistics when identifying descriptive keyphrases.

Grammar and position. At the cost of additional implementation effort, our results show that keyphrase quality can be further improved through the addition of grammatical annotations (specifically, technical term pattern matching using part-of-speech tags) and positional information. The inclusion of these additional features can improve the choice of keyphrases. More computationally costly statistical parsing provides little additional benefit.

Keyphrase selection. When viewed as a set, keyphrases may overlap or reference the same entity. Our results show how text visualizations might make better use of screen space by identifying related terms (including named entities and acronyms) and reducing redundancy. Interactive systems might leverage these groupings to enable dynamic keyphrase selection based on term length or specificity.

Potential effects of layout and collective accuracy. Our study comparing tag cloud designs provides examples suggesting that layout decisions (e.g., central placement of the largest term) and collective accuracy (e.g., prominent errors) impact user judgments of keyphrase quality. Our results do not provide definitive insights but suggest that further studies on the spatial organization of terms may yield insights for more effective layout and that keyphrase quality should not be assessed in isolation.

8.2. Applications to Interactive Visualization

In this section, we illustrate how our keyphrase extraction methods can enable novel interactions with text. We present two example applications: phrase-level text summarization and dynamic adjustment of keyphrase specificity.

We apply our keyphrase extraction algorithm to Lewis Carroll's *Alice's Adventures in Wonderland* and compare the text in each chapter using a Parallel Tag Cloud in Figure 11. Each column contains the top 50 keyphrases (without redundancy reduction) from a chapter of the book. By extracting longer phrases, our technique enables the display of entities, such as "*Cheshire Cat*" and "*Lobster Quadrille*", that might be more salient to a reader than a display of unigrams alone. Our term grouping approach can enable novel interactions. For example, when a user mouses over a term, the visualization highlights all terms that are considered conceptually similar. As shown in Figure 11, when the user selects the word "*tone*", the visualization shows the similar but changing tones in Alice's adventures from "*melancholy tone*" to "*solemn tone*" and from "*encouraging tone*" to "*hopeful tone*" as the story develops.

Our algorithm can enable text visualizations that respond to different audiences. The tag clouds in Figure 12 show the top keyphrases of an article discussing a new subway map by the New York City Metropolitan Transportation Authority. By adjusting the model output to show more specific or more general terms, the tool can adapt the text for readers with varying familiarity with the city's subway system. For example, a user might interactively drag a slider to explore different levels of term specificity. The top tag cloud provides a general gist of the article and of the redesigned map. By increasing term specificity, the middle tag cloud progressively reveals additional terms, including neighborhoods such as "*TriBeCa*", "*NoHo*", and "*Yorkville*", that may be of interest to local residents. The bottom tag cloud provides additional details, such as historical subway maps with the "*Massimo Vignellis abstract design*."

8.3. Applications of HCI Methods to Natural Language Processing

In addition to contributing a keyphrase extraction algorithm, we would like to emphasize the process through which the algorithm was developed. We highlight the various steps at which we applied HCI methods and point out how HCI concepts helped guide the development. We hope that our experiences can serve as an example for creating algorithms that are responsive to users' tasks and needs.

Our model arose through the cumulative application of HCI methods to collecting data, and analyzing, developing, and evaluating text summarization techniques. First, we collected human-generated keyphrases via a formal experiment. The data enabled us to examine the relationships between the descriptors and the corresponding text in a systematic manner and to determine the effects of three controlled factors. Second, an exploratory analysis yielded insights for designing more effective algorithms. We assessed the quality of various linguistic and grammatical features (e.g., accuracy of existing frequency statistics, computational cost of tagging vs. parsing) and characterized the properties of high-quality descriptors. The characterizations enabled identification of appropriate natural language processing techniques (e.g., technical terms for approximating noun phrases). In turn, the choice of features led to a simple regression model that is competitive with outputs generated by more advanced statistical models. Third, we designed ecologically valid evaluations. In addition to standard

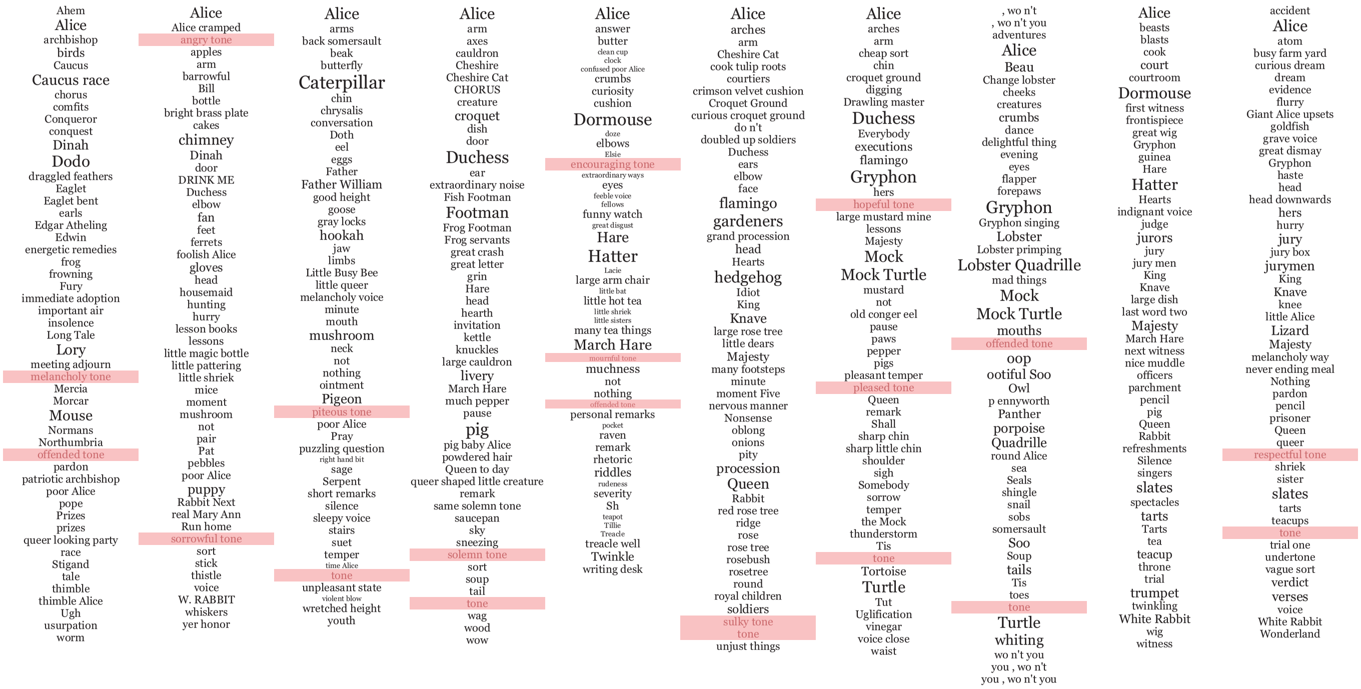


Fig. 11. Parallel tag cloud using our keyphrase extraction algorithm as the underlying text processing step. The columns contain the top 50 keyphrases (without redundancy reduction) in chapters 3 through 12 of Lewis Carroll's *Alice's Adventures in Wonderland*. Longer phrases enable display of entities, such as "*Cheshire Cat*" and "*Lobster Quadrille*", that might be more salient to a reader than unigrams alone. Term grouping can enable novel interaction techniques, such as brushing-and-linking conceptually similar terms. When a user selects the word "*tone*", the visualization shows the similar but changing tones in Alice's adventures from "*melancholy tone*" to "*solemn tone*" and from "*encouraging tone*" to "*hopeful tone*" as the story develops.

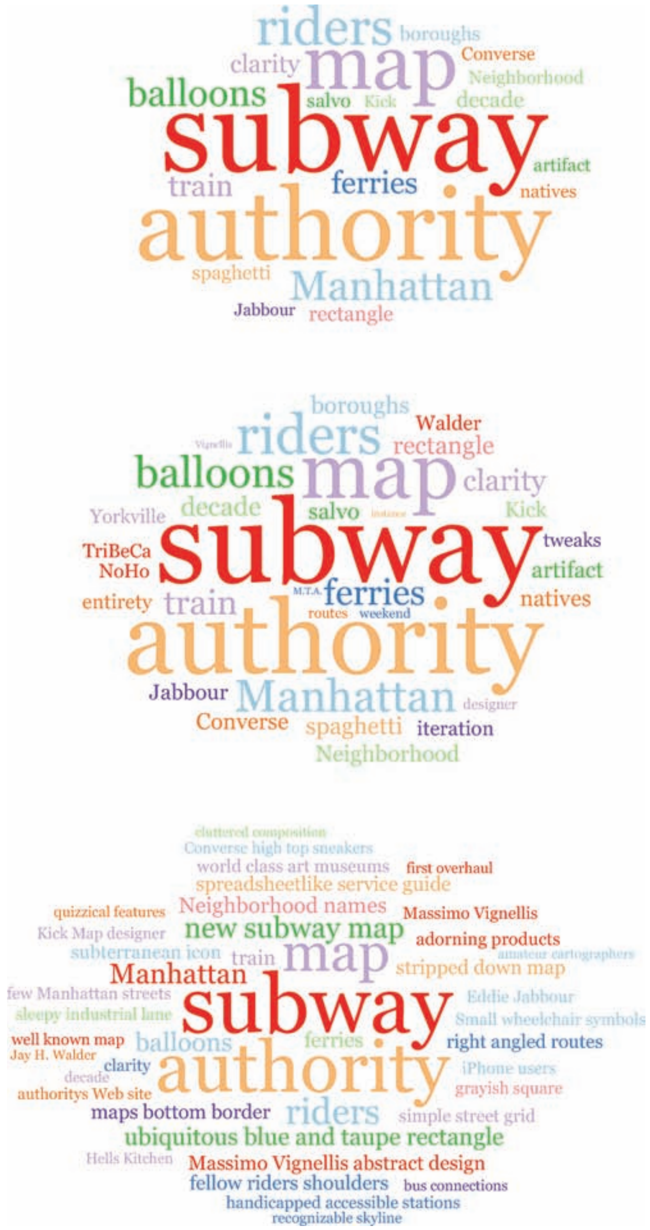


Fig. 12. Adaptive tag cloud summarizing an article about the new subway map by the New York City Metropolitan Transportation Authority. By adjusting the model output to show more specific or more general terms, a visualization can adapt the text for readers with varying familiarity with the city's subway system. For example, a user might interactively drag a slider to explore different levels of term specificity. The top tag cloud provides a general gist of the article and of the redesigned map. By increasing term specificity, the middle tag cloud progressively reveals additional terms, including neighborhoods such as "*TriBeCa*", "*NoHo*", and "*Yorkville*", that may be of interest to local residents. The bottom tag cloud provides additional details, such as historical subway maps with the "*Massimo Vignellis abstract design*."

quantitative measures (e.g., precision recall on exact matches), we evaluated the extracted keyphrases in situations closer to the actual context of use. An analysis using relaxed matching yielded insights on the shortcomings of the standard equality-based precision-recall scores and provided the basis for our redundancy reduction algorithm. Evaluating keyphrase use in tag clouds revealed effects due to visual features as well as the impact of prominent mistakes.

While many of these preceding concepts may be familiar to HCI practitioners, their uses in natural language processing are not widely adopted. Incorporating HCI methods, however, may benefit various active areas of NLP research.

For example, topic models are tools for analyzing the content of large text corpora; they can automatically produce *latent topics* that capture coherent and significant themes in the text. While topic models have the potential to enable large-scale text analysis, their deployment in the real world has been limited. Studies with domain experts might better characterize human-defined textual topics and inform better models of textual organization. HCI design methods may lead to visualizations and interfaces that better address domain-specific tasks and increase model adoption. HCI evaluations may also enable more meaningful assessment of model performance in the context of real-world tasks.

9. CONCLUSION

In this article, we characterize the statistical and grammatical features of human-generated keyphrases and present a model for identifying highly descriptive terms in a 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 preprocessed reference corpus, external taxonomies, or genre-specific document structure while supporting interactive applications. Evaluations reveal that our model is preferred by human judges, can match human extraction performance, and performs well even on short texts.

Finally, the process through which we arrived at our algorithm—identifying human strategies via a formal experiment and exploratory analysis, designing our algorithm based on these identified strategies, and evaluating its performance in ecologically-valid settings—demonstrates how HCI methods can be applied to aid the design and development of effective algorithms in other domains.

REFERENCES

Barker, K. and Cornacchia, N. 2000. Using noun phrase heads to extract document keyphrases. In *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence.* 40–52.

Boguraev, B. and Kennedy, C. 1999. Applications of term identification technology: Domain description and content characterisation. *Nat. Lang. Process. 5,* 1, 17–44.

Brants, T. and Franz, A. 2006. Web 1T 5-gram Version 1, Linguistic Data Consortium, Philadelphia.

Buyukkokten, O., Garcia-Molina, H., Paepcke, A., and Winograd, T. 2000. Power browser: Efficient Web browsing for PDAs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Buyukkokten, O., Kaljuvee, O., Garcia-Molina, H., Paepcke, A., and Winograd, T. 2002. Efficient Web browsing on handheld devices using page and form summarization. *ACM Trans. Inf. Syst. 20*, 82–115.

Cao, N., Sun, J., Lin, Y.-R., Gotz, D., Liu, S., and Qu, H. 2010. FacetAtlas: Multifaceted visualization for rich text corpora. *IEEE Trans. Visual Comput. Graphics 16*, 1172–1181.

Collins, C., Viégas, F. B., and Wattenberg, M. 2009. Parallel tag clouds to explore and analyze faceted text corpora. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology.* 91–98.

Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M. X., and Qu, H. 2010. Context-preserving, dynamic word cloud visualization. In *Proceedings of the IEEE PacificVis Symposium.* 42–53.

Daille, B., Gaussier, E., and Langé, J.-M. 1994. Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the Conference on Computational Linguistics*. 515–521.

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Comput. Ling. 19,* 1, 61–74.

Evans, D. K., Klavans, J. L., and Wacholder, N. 2000. Document processing with LinkIT. In *Recherche d'Informations Assistee par Ordinateur*.

Faraway, J. J. 2006. *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. Chapman & Hall/CRC.

Finkel, J. R., Grenager, T., and Manning, C. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*. 363–370.

Havre, S., Hetzler, B., and Nowell, L. 2000. ThemeRiver: Visualizing theme changes over time. In *Proceedings of the IEEE Symposium on Information Visualization.* 115.

Hearst, M. 2009. *Search User Interfaces*. Cambridge Press, Cambridge, U.K.

Heer, J. and Bostock, M. 2010. Crowdsourcing graphical perception: Using Mechanical Turk to assess visualization design. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems.* 203–212.

Hulth, A. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 216–223.

Justeson, J. S. and Katz, S. M. 1995. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Nat. Lang. Engi. 1,* 1, 9–27.

Kim, S. N., Medelyan, O., Kan, M.-Y., and Baldwin, T. 2010. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation.*

Kit, C. and Liu, X. 2008. Measuring mono-word termhood by rank difference via corpus comparison. *Terminol. 14,* 2, 204–229.

Klein, D. and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*. 423–430.

Laver, M., Benoit, K., and College, T. 2003. Extracting policy positions from political texts using words as data. *Am. Political Sci. Rev.* 311–331.

Lopez, P. and Romary, L. 2010. HUMB: Automatic key term extraction from scientific articles in GROBID. In *Proceedings of the International Workshop on Semantic Evaluation.*

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM J. Res. Develop. 2,* 2, 159–165.

Manning, C. D., Raghavan, P., and Schtze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. 1993. Building a large annotated corpus of English: The Penn Treebank. *Comput. Ling. 19,* 2, 313–330.

Medelyan, O. and Witten, I. H. 2006. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*. 296–297.

Minnen, G., Carroll, J., and Pearce, D. 2001. Applied morphological processing of English. *Nat. Lang. Engi. 7,* 3, 207–223.

Monroe, B., Colaresi, M., and Quinn, K. 2008. Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Anal. 16,* 4, 372–403.

Rayson, P. and Garside, R. 2000. Comparing corpora using frequency profiling. In *Proceedings of the Workshop on Comparing Corpora*. 1–6.

Robertson, S. E., van Rijsbergen, C. J., and Porter, M. F. 1981. Probabilistic models of indexing and searching. In *Research and Development in Information Retrieval*, R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, and P. W. Williams, Eds. 35–56.

Salton, G. and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Inform. Proc. Manage.* 513–523.

Schwartz, A. S. and Hearst, M. A. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Proceedings of the Pacific Symposium on Biocomputing*.

Shi, L., Wei, F., Liu, S., Tan, L., Lian, X., and Zhou, M. X. 2010. Understanding text corpora with multiple facets. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology.* 99–106.

Stasko, J., Görg, C., and Liu, Z. 2008. Jigsaw: Supporting investigative analysis through interactive visualization. *Inform. Visual. 7*, 118–132.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies (HLT-NAACL)*. 252–259.

Turney, P. D. 2000. Learning algorithms for keyphrase extraction. *Inform. Retrie. 2,* 4, 303–336.

Viégas, F. B., Golder, S., and Donath, J. 2006. Visualizing email content: Portraying relationships from conversational histories. In *Proceedings of the International Conference on Human Factors in Computing Systems (CHI)*. 979–988.

Viégas, F. B., Wattenberg, M., and Feinberg, J. 2009. Participatory visualization with Wordle. *IEEE Trans. Visual Comput. Graphics 15,* 6, 1137–1144.

Yang, C. C. and Wang, F. L. 2003. Fractal summarization for mobile devices to access large documents on the web. In *Proceedings of the 12th International Conference on World Wide Web*. 215–224.

Received July 2011; revised February 2012; accepted April 2012