BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 21 2010, pages 2767–2775
doi:10.1093/bioinformatics/btq459

Data and text mining

Advance Access publication September 1, 2010

Click-words: learning to predict document keywords from a user perspective

Rezarta Islamaj Doğan and Zhiyong Lu*

National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20894, USA

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: Recognizing words that are key to a document is important for ranking relevant scientific documents. Traditionally, important words in a document are either nominated subjectively by authors and indexers or selected objectively by some statistical measures. As an alternative, we propose to use documents' words popularity in user queries to identify *click-words*, a set of prominent words from the users' perspective. Although they often overlap, click-words differ significantly from other document keywords.

Results: We developed a machine learning approach to learn the unique characteristics of click-words. Each word was represented by a set of features that included different types of information, such as semantic type, part of speech tag, term frequency–inverse document frequency (TF–IDF) weight and location in the abstract. We identified the most important features and evaluated our model using 6 months of PubMed click-through logs. Our results suggest that, in addition to carrying high TF–IDF weight, click-words tend to be biomedical entities, to exist in article titles, and to occur repeatedly in article abstracts. Given the abstract and title of a document, we are able to accurately predict the words likely to appear in user queries that lead to document clicks.

Contact: luzh@ncbi.nlm.nih.gov

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on April 12, 2010; revised on July 27, 2010; accepted on August 8, 2010

1 INTRODUCTION

Finding relevant publications is critical for individual researchers to keep pace with the state of the art in their fields. Most scholars in the biomedical domain use PubMed®, a free web service provided by the US National Center for Biotechnology Information (NCBI), to access over 20 million biomedical citations. Improving the retrieval effectiveness of PubMed while accommodating the exponential growth of biomedical literature is a challenging and critical task for the NCBI, as well as for the researchers in the field of biomedical information retrieval (Hersh, 2003; Lu *et al.*, 2009; Tsai *et al.*, 2009; Tsuruoka *et al.*, 2008; Zhu *et al.*, 2009).

Interaction with PubMed is generally initiated by a user query and that contains three words on average. An intelligent system should be able to use this information efficiently to retrieve the citation(s), which the user is looking for. The user may click to view one or more abstracts, modify the original query, issue a different query or leave the system (Islamaj Doğan *et al.*, 2009).

This sequence of interactions with PubMed is similar to what users experience with general web search engines. However, searching the biomedical literature in PubMed is also unique from at least two perspectives: (i) PubMed search is built as a Boolean system—by default, only the documents matching all the words in the query are retrieved; and (ii) PubMed results are listed in reverse chronological order. As a result, the top returned documents are the newly indexed citations but not necessarily the most relevant ones. On the other hand, 80% of retrievals are for the top 20 returned results in PubMed (Islamaj Doğan *et al.*, 2009). The search in PubMed, therefore, is much more sensitive to the user selection of query terms than in search systems that weigh and rank results by relevance. Although a PubMed article can be retrieved by various user queries, only certain queries lead to user clicks (retrievals) for that article—a strong indication of relevance between the query and the clicked document. In this work, we identify document keywords from a list of PubMed queries that resulted in user clicks, and we name such keywords as *click-words*.

In document retrieval, a keyword refers to a term that captures the essence of the topic of a document. They are integral to the document management both for indexing and for retrieval. One type of keywords used in MEDLINE citations is known as Medical Subject Heading® (MeSH) indexing terms, which are assigned by professional indexers. Another common type of keywords is author keywords, provided by authors when submitting an article. A third type of keywords may be computed using statistics, instead of relying on human annotation. For example, the classic term frequency–inverse document frequency (TF–IDF) weighting schema can be used to identify highly weighted words that stand out in an article when compared to the other articles in the collection (Salton and Buckley, 1988).

Comparing click-words with other document keywords we found that, although there was overlap, user click-words were quite different from other types of important keywords (see Fig. 1 for an illustration). Document keywords are all meant to capture the important contributions of a document, but they rely on different weighting mechanisms, which may be the reason for their difference. Click-words are the product of click-through logs and they represent the 'wisdom of the crowds' as to what terms in an article may be important from the users' perspective. Top weighted TF–IDF words capture the importance of words with respect to other articles in a collection. In contrast, PubMed relies on indexers to assign

*To whom correspondence should be addressed.

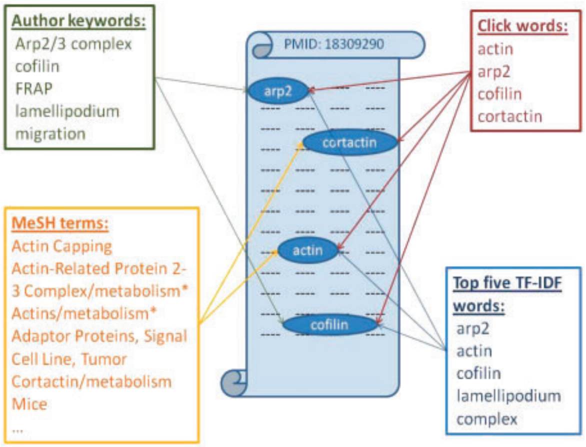

Fig. 1. An example of click-words, top-scoring TF–IDF words, author keywords and MeSH indexing terms for a PubMed article. User click-words are listed by the frequency in which they appear in user queries for this article. TF–IDF words are listed in decreasing order of their TF–IDF weight. MeSH terms follow the order in which they are listed in PubMed. Author keywords are listed as they appear in the article.

the appropriate MeSH indexing terms to PubMed articles. As a result, these words are not immediately available for new articles. Moreover, they are not necessarily found in the title and abstract of the article. Author keywords, on the other hand, are not included in the MEDLINE citation. In addition, they are not easily procured—we found that they are available for only 13% of the articles in the PubMed Central full text database.

As an illustrative example, we randomly selected an article from MEDLINE (PubMed ID: 18309290) and identified the document keywords. Figure 1 illustrates the relationship between the click-words and other keyword types for this article. It also shows that the top five TF–IDF words list has the largest overlap with the click-words list: three of the four click-words of this article: *arp2, actin* and *cofilin*, were also ranked in the top five TF–IDF words of this article. In contrast, the other two words of the top five TF–IDF list: *lamellipodium* and *complex*, were not popular in user queries.

Our objective is to learn what characteristics make document words important from a collective user perspective—words used as click-words. For each word, we consider several characteristics including the word itself, its part of speech (POS), its position in the title/abstract, etc. By learning the importance of each characteristic, we aim to build a learning system that will be able to predict which words are likely to become click-words for a given article. Thus, no search history would be required to automatically predict click-words in any documents, including those that lack user search history (e.g. new PubMed citations). The predicted click-words can not only serve as an alternative type of document keywords for indexing, but they can also provide assistance in curating MeSH terms or suggesting author keywords. Moreover, as briefly discussed in Section 6, predicted click-words could play important roles in many PubMed-related applications such as ranking relevant results and finding similar documents.

Since top-weighted TF–IDF words provided the largest overlap with the user click-words in the documents in our data, we built a machine learning model with novel features to recognize click-words from a pool of candidate words with high TF–IDF weights. For example, take the five TF–IDF words in Figure 1. In our proposed approach, we aim to identify prediction scores such that the three click-words (*arp2, cortactin* and *actin)* are ranked higher than the other two words *(lamellipodium* and *complex)*.

The specific contributions of this article are the following:

- We introduce click-words as representative document keywords. We observed that click-words (obtained from click-through logs) can represent the meaning of scientific text from a collective user perspective. Therefore, they are complementary to the current types of document keywords such as MeSH, and they may serve as an alternative type of keywords for document management.
- We developed and evaluated a machine learning method with novel features for identifying the unique characteristics of click-words. The proposed approach is capable of automatically finding click-words in any document, regardless of its search history.
- We performed empirical studies on large-scale real-world datasets (over 50 000 frequently accessed MEDLINE articles over the course of 6 months). Evaluation results show that the classifier is able to successfully perform automatic assignment of high-quality click-words.

2 RELATED WORK

Automatically identifying click-words from a pool of candidates (e.g. top-scoring TF–IDF words) is closely related to the problem of searching for words that are key to the meaning of a document (also known as keyword extraction). Much research has been devoted to this problem in the field of Natural Language Processing because of its importance in categorizing and summarizing documents, and making efficient retrievals (Manning *et al.*, 2008). In the web context, keyword identification is important for content-targeted advertizing (Lacerda *et al.*, 2006; Yih *et al.*, 2006), sponsored search (Ciaramita *et al.*, 2008; Fuxman *et al.*, 2008) and search retrieval performance (Hawking *et al.*, 2006; Ji *et al.*, 2009).

A classic algorithm for weighing words in an article is the TF–IDF measure. Other unsupervised techniques with comparable performance to this measure include Matsuo and Ishizuka (2003) and Liu *et al.* (2009). The former proposed to build and use co-occurrence distributional characteristic of a word to evaluate its importance. The latter demonstrated that additional information from POS and sentence salience score can improve the classic TF–IDF approach.

Alternatively, supervised methods have been used to classify keywords versus non-keywords. Commonly used features include those that are able to characterize context such as term frequency, term position and TF–IDF score. In addition, better classification results were achieved when advanced features such as linguistic knowledge (Hulth, 2003) or graph-based features (Litvak and Last, 2008) were added. A similar set of features was also used by a learning-to-rank method in a more recent study (Jiang *et al.*, 2009). On the other hand, Yih *et al.* (2006) observed that keyword frequency in query log files substantially helped the linguistic features and information retrieval-based features in identifying content-based keywords of web pages. In their study, they relied on human annotations of advertizing keywords. Thus, they had only a small number of annotated documents (30) with modest annotator agreement. In comparison, we use document-specific keyword frequency to label positive and negative instances of click-words. As a result, not only do we use this important information implicitly, but we are also able to obtain a sufficient number of training and test data for building and evaluating our classification algorithm.

Click-through logs seem the perfect source of information when deciding which documents to show in response to a query. It can be thought of as the result of users voting in favor of the document that they found interesting. This source has been exploited for purposes of gaining insight into users browsing behavior (Dupret and Piwowarski, 2008; Islamaj Doğan *et al.*, 2009), of improving the ranking of search results (Ciaramita *et al.*, 2008; Ji *et al.*, 2009), of studying user queries to mine query relationships (Shen *et al.*, 2007) or of identifying queries related to an advertizing campaign in order to display ads alongside search results (Fuxman *et al.*, 2008).

Studies mentioned above mostly dealt with processing web pages and news articles, perhaps due to the lack of large-scale datasets with scientific publications in the general domain. Regarding applications in the biomedical domain, Andrade and Valencia (1998) automatically extracted keywords from biomedical literature by their relative accumulation in comparison with a domain-specific background distribution. Liu *et al.* (2004a, b) reported studies using extracted keywords from MEDLINE citations to describe the most prominent functions of the genes and the resulting weights of the keywords as feature vectors for gene clustering. In their work, they compared two keyword-weighing schemes: normalized z-score and TF–IDF. Tudor *et al.* (2008) proposed another ranking algorithm for identifying keywords relating to a gene. In their work, the produced keywords were not limited to the gene functional terms.

In our approach, we used machine learning to identify document keywords, which would likely be used frequently in user queries and become click-words for the given document. In addition to the previously examined features—details of feature contributions in Section 3.3—we included novel features such as 'named entity', which have not been explored before in this context. We used MetaMap (Aronson, 2001), as the biomedical concept recognizer to identify biomedical entities in scientific text and included the recognized semantic types as classification features. As reported in Section 4, this feature showed a relatively strong discriminative power in our experiments.

3 METHODS

In this section, we describe how we compute the click-words in the context of highly accessed articles, we describe how we create our training and evaluation datasets from the information in the query logs data. Next, we describe how we build the click-word model, the features used for characterizing the click-words, the learning algorithm and the evaluation measures.

3.1 Computation of click-words and top-scoring TF–IDF words

Click-words were produced in two steps. First, we identified highly accessed articles from a large pool of query logs: articles whose citations have been clicked on, on average, at least once per day by different users for a given period of time (e.g. 2 months). When we collected data for our experiments, a user query followed by a click constituted an association from the query to the clicked article, regardless of the article's rank in the results page (because PubMed does not return results by relevance). Next, we computed click-words of a given article: popular words that appeared in at least 10% of the user queries that produced clicks for that article during the same period of time. The salience of query terms was not a factor for consideration because each term is treated equally in Boolean searches. Note that our definition of frequently accessed articles and click-words are empirically determined. However, based on our analysis, varying the selection criteria (e.g. relaxing from 10% to 5% in click-words selection) does not affect our overall observations. Because we are interested in words that represent a document's content rather than its bibliographic information, we discarded any query words that did not appear in the title or abstract (e.g. author names). Using this procedure, some documents did not produce any click-words. These documents were subsequently discarded. This procedure resulted in 4.5 click-words on average per article.

We computed the TF–IDF weight for each of the words in the title and abstract of an article and ranked them according to their weight. (See Supplementary Material for the TF–IDF definition). Next, for each document in our dataset, we chose the top five TF–IDF-weighted words. We decided to pick the top five operational TF–IDF-weighted words for each document because this is consistent with the number of click-words per document (4.5).

Note that in order to accurately capture the differences in words of various morphological classes, we preserved their original forms in the document and did not perform any stemming. In fact, PubMed does not employ any stemming algorithm (e.g. porter stemming algorithm). Instead, it relies on a newly introduced feature, which adds related terms (not necessarily always word stems) through a sophisticated mapping process (http://www.nlm.nih.gov/pubs/techbull/mj08/mj08_pubmed_atm_cite_sensor.html). As a result, word stems are very *inconsistently* recruited into original PubMed search, which makes it practically difficult to align results with any existing stemming algorithms.

3.2 Datasets

We used two separate datasets for the purposes of this study. The first dataset consisted of 47 609 PubMed articles, and was used in a 5-fold cross-validation setting to train the click-word model. The second dataset consisted of 11 237 articles that do not overlap with the articles in the training dataset, and was used for evaluating the click-word model.

Training dataset: for the training dataset, we collected 2 months of PubMed log data (March 2008 and February 2009), consisting of more than 100 million user queries and 130 million abstract clicks in 51 million user sessions. We identified the highly accessed articles (i.e. accessed more than 60 times by different users during the 2 months) and computed their user click-words. Our training dataset consisted of 47 609 articles.

From the articles in our training dataset, we identified a total of 237 155 top five TF–IDF words, of which 101 377 were click-words (42.7%). Of the top five TF–IDF words per article, we found 2.2 click-words on average.

Evaluation dataset: for the evaluation dataset, we collected 6 months of PubMed log data (February, April, May, June, July and August, 2009), consisting of more than 333 million user queries, 329 million abstract views in 144 million user sessions. From these click-through logs, we identified a total of 38 852 highly accessed MEDLINE citations (i.e. accessed over 180 times by different users during the 6 months). We separated the articles which were different from the documents in the training dataset and extracted the click-words. Our evaluation dataset consisted of 11 237 highly accessed articles.

From the articles in our evaluation dataset, we identified an average of 4.5 click-words per article. This number is in agreement with the average number of click-words computed for the 2-month dataset (Section 3.1). When we selected the top five TF–IDF words of the evaluation dataset articles, we identified a total of 62 310 words, of which 22 663 were click-words (36.4%). These were distributed with an average of 2.0 click-words per article.

3.3 The click-word learning method

Our aim is to build a learning system that, given the title and abstract of a MEDLINE article, will be able to predict which words are likely to be used frequently in user queries and become click-words for this article. To build such a learning system, we needed to specify positive and negative instances of click-words in articles. Thus, for each article in our datasets (both training and evaluation), first, we selected the five top-scoring TF–IDF words, and then we labeled the pre-identified click-words positive and the rest negative. Accordingly, for the sample article in Figure 1, we labeled the words: *arp2, cofilin* and *actin* as positive, and the words *lamellipodium* and *complex* as negative.

Once we labeled the top five TF–IDF word of each article, we computed the values for each of the designed click-word features. We applied a classification algorithm in a 5-fold cross-validation setting to learn a click-word classification model. This schema was fitted into an iterative feature selection algorithm to improve the classification results and identify the most important features for the final click-word model. We discuss these steps in detail below.

3.3.1 Click-word features To deduce context and be able to use supervised machine learning methods to predict click-words, we represented each word instance by a set of features, determined by both our own studies and prior observations in the literature. These binary features are:

- TF–IDF rank (TF–IDF): TF–IDF rank of word *w* denotes the rank position of a word *w* according to its weighted TF–IDF value. Because we only consider five words per article, we have five different TF–IDF rank features.
- Part of speech (POS): The POS tag of word *w* denotes whether word *w* is a noun, verb, adjective or a different POS. The POS tags were computed using MedPost (Smith *et al.* 2004) with the default settings. A total of 37 POS tags were obtained in our data and each was subsequently used as a binary feature in the classification task.
- Semantic type (SEM): The semantic type of word *w* denotes whether word *w* corresponds to a particular named entity in biomedicine. According to our previous study (Islamaj Doğan *et al.*, 2009), the majority of the PubMed queries do not include any bibliographic information. Instead, they contain named entities such as gene and protein names. The semantic type feature is designed to capture this aspect of the click-words. Specifically, if a word can be mapped to a biological concept, its semantic type (category) information is used. We applied MetaMap (Aronson, 2001) to our data and obtained 134 semantic types (e.g. molecular function—mofc), each of which is used as a binary feature.
- Word frequency rank (WFR): The frequency rank of word *w* denotes the rank position of word *w* according to its frequency value. This feature is similar to the local term frequency in the TF–IDF definition. However, instead of using raw term frequency, we chose to use rankings based on word frequency because word frequency can vary significantly in different articles for the most frequently occurring words. Specifically, after removing the stop words, we ranked each word by its frequency in the article. We assigned the same rank to words with equal number of occurrences. The resulting rank of the words in our training dataset ranged from 1 to 46, each of which was then used as a binary feature.
- Word location (LOC): the word location of word *w* denotes some specifics regarding whether word *w* appears in the title and/or abstract of the article. We created binary features to capture a given word's importance using four different locations: title, the first sentence, the last sentence or the middle of the abstract. When a word appeared in multiple locations, each individual position was marked, creating a combined location feature. We had 15 different binary features denoting the various combinations of word locations.
- Abbreviation (ABBR): the abbreviation feature of word *w* denotes whether word *w* is an abbreviated form of a known concept. It has been shown that important biological concepts like gene names are often abbreviated in user queries (Federiuk, 1999; Islamaj Doğan *et al.*, 2009). In fact, our previous analysis revealed that 13% of PubMed user queries contained at least one abbreviated term. Previous studies on PubMed describe highly precise methods that identify abbreviated terms in PubMed abstracts (Sohn *et al.*, 2008). We used their list of abbreviated terms in PubMed to build our ABBR feature. The value of this feature is 'is-an-abbreviation' if the word instance in our dataset is matched exactly to a term from that abbreviation list or 'is-not-an-abbreviation' otherwise.
- Phrase (PHR): the phrase feature of word *w* denotes whether word *w* is part of a common MEDLINE phrase. As shown in a previous study (Yeganova *et al.*, 2009), words in PubMed queries tend to associate in phrases. Hence, we hypothesized that given a PubMed abstract, a word having the tendency of appearing as a phrase with its neighboring words may have a higher probability of catching the readers' attention and being a click-word. For this reason, we formed MEDLINE phrase candidates of our dataset by combining the word of interest with one, two or three of its neighboring words. If such a phrase candidate was repeated somewhere else in the same article that it originated from, and was also found in at least one other article in our dataset, we considered it a valid phrase. After identifying all the valid phrases in our training dataset, we could readily determine whether or not a word is part of a phrase. Again, we considered both the confirmation 'is-part-of-phrase' and the negation 'is-not-part-of-phrase' features. For example, the document PMID:17850624 contains the word '*migration'*, which is part of '*neuronal migration'*, a valid phrase with multiple occurrences in 17850624 and other PubMed citations (e.g. PMID: 18075253). Thus, for the word '*migration'*, the value of its PHR feature 'is-part-of-phrase' is TRUE.
- Neighboring words (NBR): the neighbor of word *w* denotes the word(s) that *w* has as neighbors and its (their) relative position(s) to *w*. We considered up to the three words on each side of the word of interest, when available, as its neighbors. This was the most sparse feature type in our data. It resulted in more than 261 974 different individual binary features (i.e. unique words found in any of the considered six positions).
- Word (WRD): the final feature used was the word instance itself. There were 41 152 unique words in our training dataset, which served as binary features. Of these, 17 662 (43%) words were repeated in at least two different documents. Of the repeated words, 27% were labeled as click-words for some articles but not for other articles even though they were listed in their top five TF–IDF weight lists. This suggested that we could not rely on only the word instances themselves as features.

In addition, using only word instances would have limited our ability to predict click-words that were not seen in our dataset. However, if we represent a click-word in the context of all the features that we defined above, our proposed model can make a prediction based on the learned click-word characteristics (e.g. a word's semantic type) in addition to the word itself, making it much more robust when handling new/unseen words.

3.3.2 Classification algorithm For our experiments, we considered a range of classifiers. We selected a wide margin classifier similar to a linear support vector machine as our principal learner, which was based on the modified Huber loss function (Zhang, 2004). The Huber loss function is smooth, hence differentiable; therefore, it allows the application of a gradient search method for optimization. These properties, as well as the speed of optimization, make this algorithm suitable for large datasets, such as the one in this work. In addition, in the preliminary experiments, we also used the multivariate Bernoulli Naïve Bayesian classifier for the purpose of comparing different learning algorithms. The results presented in this article were obtained using the Huber algorithm, which produced better results when compared with the Naïve Bayes classifier.

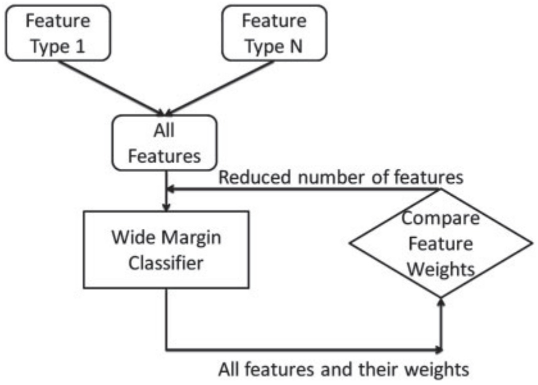

Fig. 2. The iterative feature selection model.

The set of all PubMed articles in our training dataset was randomly divided into five subsets and a 5-fold cross-validation was performed. At each round of cross-validation, the classifier was trained with four-fifths of the data, and tested on the one-fifth of the held-out data. The number of articles and the number of positive and negative class instances for each fold was balanced (see Supplementary Material).

3.3.3 Iterative feature selection method Feature selection is usually employed to filter out redundant and/or irrelevant features, while maintaining or improving the accuracy of prediction, thus resulting in a better and simpler model. For this study, we used an iterative feature selection scheme to tackle the 290 000 features that we constructed to describe the click-words (Fig. 2).

As illustrated in Figure 2, this iterative feature selection method first used the wide margin classifier (Huber algorithm) to learn a classification model for the training dataset, in a 5-fold cross-validation setting. Next, for each individual feature it examined the average weight assigned by the five wide-margin classifiers to judge whether the current feature is valuable for this problem. Usually, a non-contributing feature receives a weight close to zero. In contrast, a contributing feature is assigned a relatively high weight. We applied an aggressive feature selection procedure. After examining the features' weights, it eliminated 1000 lowest-weight features. Then, in an iterative fashion, we retrained the click-word model on the refined set of features using the Huber algorithm. We performed feature selection iterations until we eliminated all features. To select the best click-word model, we examined the performance levels of all feature subsets and selected the most significant one.

3.4 Evaluation measures

Precision, or the positive predictive value, is calculated as the ratio of the number of correct answers to the number of all the answers given by the classification algorithm. Recall, or sensitivity, is calculated as the ratio of number of correct answers to the total number of possible correct answers (Hersh, 2003).

The identification of click-words, as mentioned earlier, is article dependent: a given word may act as a click-word for one article but not for another. For this reason, we evaluated our ability of identifying click-words per each article. After applying the learning model, we scored each word instance and ranked all of the word instances of each article in descending order of their scores. Instead of relying on a global score threshold for the whole set of articles, we computed the break-even point between precision and recall for each article in the test set (during the cross-validation procedure or the evaluation stage). The break-even point is calculated as the value where precision equals recall.

We also computed the mean average precision, the area under the ROC curve (ROC score) and the precision at the first retrieved level (precision for the highest-scoring word of each article which received the highest confidence of being a click-word for that article). In the event that for a given article, two or more word instances received the same click-word score, we considered all of the possible rankings and computed the break-even point of precision–recall and mean average precision averaging over all enumerations. Finally, the value of the corresponding measure was averaged over all articles in the test set. For the training dataset experiments, we report this value averaged over the five folds of cross-validation.

Next, using the evaluation dataset, we performed two different evaluations. First, we applied the model to the top five TF–IDF words list of each article in the evaluation dataset. Second, in order to test the performance of our model in a more realistic setting, we applied the model to every single word in the title and abstract of each article in the evaluation dataset.

In addition, as a comparison, we ranked the words of each article according to two different baselines. First, in the random selection baseline we assigned every word of an article the same score. Next, we computed the evaluation measures by considering all possible word rankings, and averaged the scores. Second, in the TF–IDF weight baseline we ranked the words of each article according to their TF–IDF weight values, and computed the evaluation measures.

4 RESULTS

We conducted a wide range of experiments to identify user click-words from PubMed abstracts, and here we present a summary of them. We begin the presentation of results by describing the click-word prediction results for each individual feature type. This analysis allowed us to understand each of our feature types, and their individual contributions. Following this analysis, we combined all features and learned a click-word model consisting of a mix of all features. Next, we performed our iterative feature selection method. We show how we reduced the number of features and the corresponding break-even precision recall point average for each feature selection step. Finally, we show the results of our selected click-word prediction model when applied to evaluation dataset.

4.1 Click-word prediction results using single feature types

In the following experiments, we describe the click-word prediction performance for each individual feature type. Table 1 summarizes the break-even results of precision–recall values averaged over five folds of cross-validation when using each feature type individually. In addition, it lists the number of contributing features for each individual feature type. As shown in Table 1, the click-word model trained on the *Word* features gave the best result, while the click-word model trained on the *Abbreviation* features gave the worst result. For comparison, we use two baseline methods: the random selection baseline and the TF–IDF weight baseline. The random selection baseline uniformly picks at random words from the article. The TF–IDF weight baseline assigns the TF–IDF weight as the score for each word in the article, and then ranks them accordingly. As shown in Table 1, three feature types: *word location*, *neighboring words* and *word*, gave (statistically significant) better classification results than the TF–IDF baseline. Each of the click-word models trained on individual feature types gave significantly better results than the random selection baseline. Finally, in the last row of the Table 1 we list the result of the click-word model when we have combined all features. This result is statistically significant when compared to each of the individual models.

Table 1. The break-even precision recall point for each individual feature type and the corresponding number of contributing features (non-zero Huber weights) for each feature type, when learning to differentiate articles' click-words from the top five TF–IDF-weighted words

Model	Precision–recall break-even point	Number of features
Word	0.748	36489
Word location	0.663	15
Neighbor words	0.661	253 340
TF–IDF rank	0.613	5
WFR	0.609	46
MetaMap semantic types	0.594	134
POS tag	0.524	37
Part of phrase	0.510	2
Abbreviation	0.448	2
Random selection	0.429	–
TF–IDF weight	0.613	–
ALL features	0.781	290 069

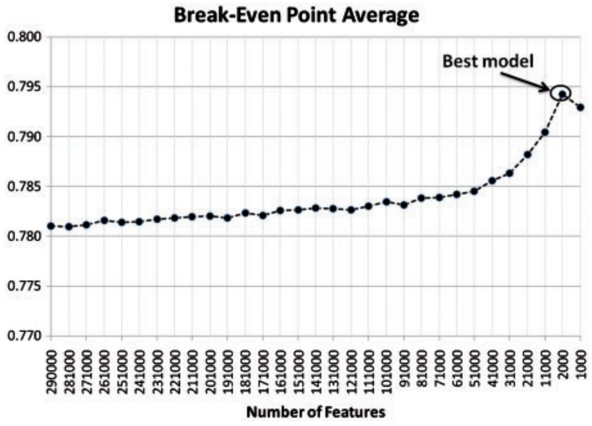

Fig. 3. Results of break-even point of precision and recall averaged among the 5-folds of cross-validation test sets, through the progression of iterative feature selection method.

4.2 Feature selection method

In the following set of experiments, we show the results of our iterative feature selection algorithm. After we combined all features together, we had a total of more than 290 000 features and, at that stage, the click-word model had a break-even point performance of 0.781. This is shown in the last row of Table 1. As illustrated in Figure 3, we used aggressive feature selection steps, removing 1000 features with each iteration. For each refined feature set, we retrained the Huber algorithm, and computed the break-even point of precision and recall for the resulting click-word model. Figure 3 summarizes each break-even value, averaged over the five folds of the cross-validation test sets. We observe this value ultimately increases, despite slight fluctuations.

We selected the click-word model that consisted of just 2000 features as our final model, because at that point, we reached the best performance of break-even point average, 0.794. This constituted a significantly simpler model when compared with the initial model of more than 290 000 features. Moreover, it exhibited a significantly better performance when compared with the initial model of 0.781. Its performance was also better when compared with the next model of 1000 features, represented as the last dot in the graph in Figure 3. Compared with the random baseline performance of 0.429, and with the TF–IDF weight performance of 0.613, the performance of the best click-word model represents an 85% and 30% increase, respectively. More details regarding the performance of the best click-word model are listed in Table 4.

4.3 Feature analysis of the final click-word prediction model

In the following analysis, we considered the feature composition of the final click-word model, summarized in Table 2, and the relative predictive strength of each feature according to the weight assigned by the Huber algorithm, as illustrated in Table 3. We observe that, even though *Abbreviation* was the least predictive feature type among the other individual models (Table 1), the 'is-an-abbreviation' and 'is-not-an-abbreviation' features were part of the final click-word model. The same is true for the 'in-phrase' and 'not-in-phrase' features. This means that even though they did not have enough predictive strength on their own, they worked well in combination with the rest of the features to have an impact in the final performance. In fact, the feature 'not-in-phrase' is listed in Table 3 as one of the features having the highest negative weight. In Table 3, we also observe that the feature with the most positive weight was the feature that describes the location of the word both in the title and in the first sentence of the abstract. In contrast, the feature that described the word location in the middle of the abstract was assigned a negative weight.

4.4 Click-word prediction results using the evaluation dataset

To further evaluate the performance of our click-word prediction model, we applied this model to the documents in our evaluation dataset. To apply the click-word model to new articles, we followed this sequence of steps: First, we extracted its title and abstract. Second, we tokenized these into single words and removed the stop words. Third, for each single word we computed the values for each of the 2000 features in the final click-word model. For each article, each word was scored according to these values. Next, we ranked the words of each article according to the score of the click-word model. As a comparison, we also ranked the words of each article according to the random selection baseline and the TF–IDF weight baseline. Based on these rankings, we computed the mean average precision, break-even point of precision and recall, ROC score and precision at the first recall value. These results are summarized in Tables 4 and 5.

The training and evaluation datasets results are contrasted in Table 4. For these results, we considered only the top five weighted TF–IDF words for each article in the evaluation dataset, as this is the equivalent to the results of the training dataset.

Although our model was trained on the top five TF–IDF terms, we also tested it for *every* word present in the titles and abstracts of the articles in the evaluation dataset. These results are presented in Table 5. As shown in the Table 5, when we used the random selection baseline, the mean average precision of picking the user click-words was 11%. When we used the TF–IDF weight as a baseline, the mean average precision of picking the user click-words was 51%, and

Table 2. Total number of features: 2000, best break-even point avg: 0.794

Feature	Number
Neighbors	1322
Word	556
Semantic types	83
Word location	15
POS tag	12
TF–IDF	5
WFR	3
Phrase	2
Abbreviation	2

Table 3. Top features of the best model

Huber weight	Positive features	Huber weight	Negative features
0.455	LOC: title + first sentence	−0.409	LOC: middle abstract only
0.368	WRD: mirna	−0.342	LOC: middle + last sentence
0.343	WRD: cancer	−0.312	LOC: first + middle sentence
0.337	SEM: disease or syndrome	−0.312	LOC: first + middle + last sentence
0.328	WRD: il	−0.270	POS: plural noun
0.317	LOC: title + first sentence + middle abstract	−0.263	PHR: not in phrase
0.293	SEM: bacterium	−0.251	SEM: functional concept

when we used our click-word model to score the words, the mean average precision of picking the user click-words increased to 61%.

In addition, in Table 5 we list *t*-test results of comparing the click-word model with the TF–IDF baseline, and the corresponding *P*-values. Each evaluation measure is highly statistically significant. We conclude that our click-word model is very accurate at identifying the user click-words for any given article. Moreover, our click-word model is significantly more accurate at recognizing the user interest in a given article when compared with the TF–IDF weighting model, evident from every evaluation measure.

5 DISCUSSION

In this work, we proposed click-words as an alternative representation of document keywords. We built a click-word model able to identify the words of a document that conveyed the readers' interest.

Click-word characteristics A click-word model based only on words would not be useful for several reasons. First, new words in new articles would be a problem. Second, the words selected as click-words for certain articles but not for other articles would create confusion. Finally, it would not be possible to apply a model based solely on the words to rank the articles matching a given query. Such a model needs context. In this work, the click-word context was provided by the rich set of features that we designed and implemented.

To understand the contributions of each feature type, we examined the feature weights learned with the Huber algorithm for each individual feature of the click-word model after feature selection[1]. We found that a word was more likely to be a click-word if:

- It appeared in the title of the article. All 15 word location features were retained in the final best prediction model. They were divided into two distinct groups: 8 positively weighted features described words that appeared in the title, among other positions, while the 7 negatively weighted features described words that appeared in locations other than the title. This may also be related to the fact that in the PubMed search results pages, no parts of the abstracts are displayed to the users.
- It was repeated several times in the abstract of the article. During feature selection, most of the initial 46 binary WFR features were removed. The final three features that were retained assign positive weights to the most frequent and second most frequent words in an abstract, and assign a negative weight to the less frequent words.
- It was ranked first according to its TF–IDF weight. All five TF–IDF Rank features were retained in the final model. Interestingly, their weights were in accordance with their ranks. The feature for the top ranking in TF–IDF value (Rank 1) had the highest positive weight, whereas the one for the last rank (Rank 5) had the largest negative weight. The top 2 ranks were assigned positive weights.
- It had one of the six following POS tags: *singular noun, base form lexical verb, infinitive lexical verb, nominal gerund, past tense or proper noun singular*. In comparison, words with the following six POS tags were less likely to be click-words: *plural noun, third person singular, past participle of a verb, pronominal past participle, number or numeric*, and *coordinating conjunction*. These remaining features suggest that click-words generally tend to be singular noun (e.g. *cancer*) rather than plural noun (e.g. *cancers*), to be base form verb (e.g. *migrate*) rather than third person singular (e.g. *migrates*), and to be nominal gerund (e.g. *misfolding*) rather than pronominal past participle (e.g. *misfolded*).
- It was found as part of a phrase. In addition, when a word was not found as part of a phrase, this increased its probability of not being a click-word (see Table 3).
- It had certain word neighbors such as: *background, syndrome, diagnosis, receptor, infection* or *cells*. Neighboring words account for the largest number of final features in the final feature set. Although their number is significantly reduced, there are 745 positively and 577 negatively weighted neighboring word features in the final model. The positively weighted neighboring words tend to be general words providing context for specific content words. For instance, the word preceding *syndrome* is likely to be a click-word because it is typically a specific disorder name.
- It belonged to one of the following semantic types: *virus, neoplastic process* or *disease or syndrome*, rather than *research*

[1] The entire set of 2000 final features and their respective weights can be found in Supplementary Material.

Table 4. Performance evaluation of the click-word model when compared with the TF–IDF weighting and random selection, for the top five weighted TF–IDF words for each article in the evaluation dataset

	Classification model	Mean average precision	Break-even precision–recall	ROC	Precision@1
Training dataset results of 5-fold cross-validation	Random selection	0.612	0.428	0.498	0.428
	TF–IDF weight	0.757	0.611	0.691	0.671
	Click-word model	0.888	0.794	0.868	0.863
Evaluation dataset results for top 5 TF–IDF words	Random selection	0.596	0.405	0.495	0.405
	TF–IDF weight	0.737	0.581	0.681	0.631
	Click-word model	0.855	0.743	0.832	0.810

Table 5. Performance evaluation results for the baseline random selection model, TF–IDF weighting of the words model and the click-word prediction model, for all the words in the title and abstract of the articles in the evaluation dataset

	Classification model	Mean average precision	Break-even precision–recall	ROC	Precision@1
Evaluation dataset results for all words	Random selection	0.112	0.065	0.499	0.065
	TF–IDF weight	0.513	0.454	0.861	0.631
	Click-word model	0.627	0.547	0.904	0.806
Statistical analysis	*t*-test	45.195	31.824	32.425	32.877
	P-value	0.0002	0.0005	0.0005	0.0005

activity (e.g. *trials*), *population group* (e.g. *women*), *idea* or *concept* (e.g. *recommendation*). After feature selection, the number of semantic type features decreased substantially from the initial 135 to the final 83, which still would cover almost all types of common information needs in biomedicine. For instance, diseases and disorders—frequently sought information in PubMed—are covered by the following semantic types: *Disease or Syndrome, Neoplastic Process, Mental or Behavioral Dysfunction, Injury or Poisoning, Sign or Symptom*.

- It was a word such as *cancer, stem, mirna, diabetes*, rather than *proteins, genes, diseases, therapies* and *trials*. This set of 556 words includes 308 words with positive weighs and 248 with negative weights. This is the second largest subset among the 2000 remaining features. We found that, the aforementioned click-word characteristics were largely applicable to those positively weighted Word features. For instance, it is shown that they are more likely to be singular nouns (e.g. *mirna*) than plural nouns (e.g. *mirnas*), and that they tend to be specific content words (e.g. *diabetes*) rather than the context words (e.g. *diseases)*. Our further analysis shows that among the top 100 most frequent PubMed search words (e.g. *cancer*), 37 of them are positively weighted Word features. Interestingly, we also find 13 negatively weighted click-word features in the top 100 most frequent user search terms. Despite their frequent occurrences, those 13 words tend to represent general concepts such as: *gene, infection* and *treatment*.

Note that the designed feature types are not fully independent of one another. For instance, the WFR feature is implicated in computing both the word location and TF–IDF features. Yet, we found that all the designed feature types contributed features to the final click-word model. Although specific words on their own still carried some weight towards being recognized as click-words, all the other features that we designed provided strong characteristics that collectively identified the click-words. As a result, such a click-word model could also make it possible to rank and index the articles according to the perceived users' interest.

6 CONCLUSIONS AND FUTURE WORK

In this work, we proposed click-words as document keywords as they refer to words that readers find important. Next, we showed that click-words overlap significantly with top-scoring TF–IDF words. We implemented a supervised method to learn what characteristics, in addition to TF–IDF weights, make click-words important for PubMed users. Our results show that a word's *semantic type, location, POS*, *neighboring words* and *phrase* information together could best determine if a word will be a click-word. In addition, we have detailed the individual contribution of each of the click-word features. For example, a word's location was found to have the strongest power in classification. Specifically, click-words were more likely to appear in the title of the document, rather than only in the middle of the abstract. Click-words tended to be names of recognized biological entities, and they could be identified by their neighboring words and their positions in a sentence. Click-words tended to be abbreviated terms, and appeared in phrases. Although trained to identify click-words from the top-weighted TF–IDF words, our click-word model showed significant robustness when applied to identify click-words from all the words in the titles and abstracts of more than 11 000 PubMed articles.

There are several directions for improving the current work and extending this research. In terms of document indexing, because click-words are readily available and significantly less costly to procure, they can be used to complement MeSH terms and author keywords that both require human annotation. In terms of document retrieval, the click-word model can produce a score for identifying word importance, similar to the classic TF–IDF measure. Thus, a natural extension of this work is to compare the retrieval effectiveness using such scores versus traditional TF–IDF scores. For example, the 'related articles' in PubMed are computed by adding up the TF–IDF weights of all the terms in common between two documents. Replacing the TF–IDF weights with the click-words weights might yield better results in that we assign higher scores to terms that represent the subject of an article from readers' perspective. Another possible direction is to expand the click-words to click phrases, as the *Phrase* feature was a prominent feature in the final click-word model. In this work, we used only exact query terms but not their translation(s) (due to PubMed's Automatic Term Mapping feature) in obtaining instances of click-words. A further direction is to expand the definition of click-words to all frequent query terms that result in document clicks, regardless of their appearance in the title or abstract. Finally, rather than treating query words as independent, we could explore the relationships *between* query words, which also may lead to a better identification of relevant articles.

ACKNOWLEDGEMENTS

The authors are grateful for the help from J. Wilbur, A. Névéol, L. Yeganova, G.C. Murray, D. Comeau, and L. Smith.

Funding: Intramural Research Program of the NIH, National Library of Medicine.

Conflict of Interest: none declared.

REFERENCES

Andrade,M.A. and Valencia,A. (1998) Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. *Bioinformatics*, **14**, 600–607.

Aronson,A.R. (2001) Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proc. AMIA Symp.*, 17–21.

Ciaramita,M. *et al.* (2008) Online learning from click data for sponsored search. In *WWW '08: Proceeding of the 17th International Conference on World Wide Web*, ACM, New York, NY, USA, pp. 227–236.

Dupret,G.E. and Piwowarski,B. (2008) A user browsing model to predict search engine click data from past observations. In *SIGIR'08: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, NY, USA, pp. 331–338.

Federiuk,C. (1999) The effect of abbreviations on MEDLINE searching. *Acad. Emerg. Med.*, **6**, 292–296.

Fuxman,A. *et al.* (2008) Using the wisdom of the crowds for keyword generation. In *International Conference on World Wide Web (WWW)*, ACM, New York, NY, USA, pp. 61–70.

Hawking,D. *et al.* (2006) Improving rankings in small-scale Web search using click-implied descriptions. *Aust. J. Intell. Inf. Process. Syst.*, 17–24.

Hersh,W.R. (2003) *Information Retrieval: A Health and Biomedical Perspective*. Springer, New York.

Hulth,A. (2003) Improved automatic keyword extraction given more linguistic knowledge. In M.Collins and M.Steedman, (eds) *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Sapporo, Japan*, pp. 216–223.

Islamaj Doğan,R. *et al.* (2009) Understanding PubMed(R) user search behavior through log analysis. *Database* [Epub ahead of print, doi:10.1093/database/bap018].

Ji,S. *et al.* (2009) Global ranking by exploiting user clicks. In *SIGIR'09: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, NY, USA, pp. 35–42.

Jiang,X. *et al.* (2009) A ranking approach to keyphrase extraction. In *SIGIR'09: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, NY, USA, pp. 756–757.

Lacerda,A. *et al.* (2006) Learning to advertise. In *SIGIR'06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, NY, USA, pp. 549–556.

Litvak,M. and Last,M. (2008) Graph-based keyword extraction for single-document summarization. In *MMIES'08: Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 17–24.

Liu,F. *et al.* (2009) Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *NAACL'09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 620–628.

Liu,Y. *et al.* (2004a) Comparison of two Schemes for automatic keyword extraction from MEDLINE for functional gene clustering. In *CSB'04: Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, IEEE Computer Society, Los Alamitos, CA, USA, pp. 394–404.

Liu,Y. *et al.* (2004b) Text mining functional keywords associated with genes. *Stud. Health Technol. Inform.*, **107**, 292-296.

Lu,Z. *et al.* (2009) Evaluating relevance ranking strategies for MEDLINE retrieval. *J. Am. Med. Inform. Assoc.*, **16**, 32–36.

Manning,C.D. *et al.* (2008) *Introduction to Information Retrieval*. Cambridge University Press.

Matsuo,Y. and Ishizuka,M. (2003) Keyword extraction from a single document using word co-occurrence statistical information. *Int. J. Artif. Intell. Tools*, **13**, 157–169.

Salton,G. and Buckley,C. (1988) Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.*, **24**, 513–523.

Shen,D. *et al.* (2007) Mining web query hierarchies from clickthrough data. In *AAAI'07: Proceedings of the 22nd National Conference on Artificial Intelligence*, AAAI Press, pp. 341–346.

Sohn,S. *et al.* (2008) Abbreviation definition identification based on automatic precision estimates. *BMC Bioinformatics*, **9**, 402.

Smith,L. *et al.* (2004) MedPost: a part-of-speech tagger for bioMedical text. *Bioinformatics*, **20**, 2320–2321.

Tudor,C.O. *et al.* (2008) Mining the biomedical literature for genic information. In *BioNLP'08: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 28–29.

Tsai,R.T. *et al.* (2009) PubMed-EX: a web browser extension to enhance PubMed search with text mining features. *Bioinformatics*, **25**, 3031

Tsuruoka,Y. *et al.* (2008) FACTA: a text search engine for finding associated biomedical concepts. *Bioinformatics*, **24**, 2559–2560

Yeganova,L. *et al.* (2009) How to interpret PubMed queries and why it matters. *JASIST*, **60**, 264–274.

Yih,W. *et al.* (2006) Finding advertising keywords on web pages. In *WWW'06: Proceedings of the 15th international conference on World Wide Web*, ACM, New York, NY, USA, pp. 213–222.

Zhang,T. (2004) Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Twenty-first International Conference on Machine Learning*, ACM, New York, NY, USA, pp. 918–922.

Zhu,S. *et al.* (2009) Enhancing MEDLINE document clustering by incorporating MeSH semantic similarity. *Bioinformatics.* 2009 Aug 1;25(15):1944–51