What's New On Wiki? - News feed di Wikipedia

Stefano D'Arrigo X81000675

24 marzo 2020

Sommario

Giornalmente migliaia di pagine di Wikipedia vengono modificate in tutto il mondo. La natura collaborativa di tale enciclopedia online fa sì che il trend delle modifiche segua gli accadimenti più recenti a livello globale. Partendo da tale osservazione, nel presente progetto si è scelto di realizzare un servizio web che fornisca un news feed con i contenuti di Wikipedia (versione in lingua inglese), sulla scorta di servizi di aggregazione di notizie quali Google News, Microsoft News e Flipboard.
Tramite interrogazioni all'API REST fornita da MediaWiki, gestore dei contenuti di Wikipedia, è stato costruito un dataset composto da 39.897 estratti di pagine appartenenti a cinque distinte categorie: Economia, Intrattenimento, Politica, Scienza e tecnologia, Sport. Sulla base di tale dataset, il sistema classifica le pagine modificate più di recente mediante classificatore Naive Bayes complementare e le presenta all'utente tramite interfaccia web. Il servizio implementa un sistema di raccomandazione content-based, fornendo del contenuto pertinente agli interessi dell'utente, dati dall'interazione con l'interfaccia.
Dalle misurazioni di performance effettuate, il sistema mostra un indice *f1-score* dell'81,82%.

Introduzione

Wikipedia ha notevolmente cambiato la fruizione delle informazioni da parte degli individui, rendendo accessibili a tutti i contenuti non solo in lettura, ma anche in scrittura. Il contenuto di Wikipedia viene costantemente controllato e modificato da una comunità autocostituitasi che, in molti casi, partecipa in prima persona agli eventi annotati nelle corrispondenti pagine. Ne segue che il trend delle pagine modificate giornalmente dipende strettamente dagli eventi che accadono contestualmente nel mondo. Basti pensare che, al momento della stesura di tale relazione, le pagine più modificate sono relative alla pandemia di Covid-19, argomento che, nei giorni presenti, tiene banco nelle cronache mondiali. Conseguentemente, la lista delle modifiche recenti alle pagine restituisce uno spaccato sull'attualità a livello globale. In altre parole, fornisce un flusso di notizie in costante aggiornamento.
Alla luce di tali considerazioni, si è scelto di sfruttare le informazioni provenienti da Wikipedia per implementare un sistema di news feed personalizzato per ciascun utente fruitore.

Dal momento che il sistema è stato basato su un algoritmo di machine learning supervisionato, è stato necessario costruire un dataset con i contenuti di Wikipedia etichettati secondo l'argomento principale trattato: economia, politica, scienze e tecnologia, sport, intrattenimento. Per aumentare le performance del sistema e, al tempo stesso, per ridurre la complessità dell'elaborazione successiva, gli estratti delle pagine del dataset sono stati preprocessati con tecniche di Natural Language Processing e sono stati rappresentati con una rappresentazione Bag of Words. Al fine di ottenere un flusso di notizie aggiornato, è stato messo a punto un modulo del sistema per richiedere a Wikipedia e processare le pagine modificate recentemente, ottendendo una rappresentazione analoga a quella dei record del dataset.

Non avendo a disposizione informazioni riguardanti gli argomenti trattati nelle pagine recenti, è stato necessario l'uso di un algoritmo di classificazione che assegnasse ogni pagina alla classe corrispondente, sulla base del dataset costruito; nella fattispecie è stato usato il classificatore Naive Bayes complementare che, dalle misurazioni di performance effettuate, ha prestazioni migliori del 3% rispetto al classificatore Naive Bayes multinomiale.

Infine è stato implementato il sistema di raccomandazione content-based, definendo il profilo dell'utente in funzione delle cinque classi sopracitate e selezionando le pagine pertinenti attraverso l'applicazione della similarità del coseno.

Cenni di teoria

Di seguito vengono esposti brevemente gli algoritmi teorici che sono stati applicati.

Tecniche di Natural Language Processing

Tokenizzazione Per poter rappresentare un testo e processarlo con algoritmi di machine learning, è necessario scomporlo nelle sue componenti fondamentali, ossia le parole. Il processo che, dato un testo non strutturato, restituisce una lista di token (parole) è detto tokenizzazione.

Rimozione delle stop words Per ridurre il rumore all'interno del testo, è utile rimuovere le parole molto frequenti ma poco informative, come articoli e preposizioni.

Lemmatizzazione In un testo non strutturato sono frequentemente presenti parole aventi uguale radice ma prefissi, infissi e suffissi diversi; basti pensare alle forme plurali dei sostantivi, ai superlativi degli aggettivi o alle forme verbali. Il processo di lemmatizzazione consente di ridurre la cardinalità dell'insieme delle parole del testo, limitando la variabilità delle parole sopraesposta.

Bag of Words

Bag of Words è un metodo per rappresentare documenti ignorando l'ordine delle parole. Un documento, dunque, viene rappresentato con un vettore in cui ogni elemento corrisponde ad una parola del dizionario definito; ogni elemento del vettore contiene la frequenza relativa della parola corrispondente all'interno del documento; in tal modo il documento risulta rappresentato con un vettore normalizzato a lunghezza fissa e può essere elaborato con algoritmi di machine learning.

Naive Bayes

Naive Bayes è una famiglia di algoritmi di classificazione. Sia C l'insieme delle classi e sia X l'insieme degli elementi da classificare. Siano $x = (x_1, ..., x_n) \in \text{X}$ e $c \in \text{C}$. La classe predetta $\bar{c}$, secondo l'algoritmo Maximum A Posteriori (MAP) da cui Naive Bayes deriva, è data da

$$\bar{c} = \arg_c \max P(c|x) = \arg_c \max P(x|c)P(c) = \arg_c \max P(x_1, ..., x_n|c)P(c)$$

ricordando il teorema di Bayes $P(c|x) = \dfrac{P(x|c)P(c)}{P(x)}$. Quindi

$$\bar{c} = \arg_c \max P(x_1|c)P(x_2|x_1, c)P(x_3|x_1, x_2, c)...P(x_n|x_1, ..., x_{n-1}, c)$$

per la regola del prodotto. Naive Bayes introduce l'ipotesi naive secondo cui per ciascun elemento di X vale la proprietà di indipendenza condizionale: $x_i \perp x_j | c, \forall i \neq j$. La classe predetta è, dunque, data da

$$\bar{c} = \arg_c \max P(x_1|c)P(x_2|c)...P(x_n|c)P(c)$$

Naive Bayes multinomiale Naive Bayes multinomiale è un'implementazione per dati distribuiti secondo una multinomiale ed è spesso usato per classificare testi. La distribuzione è parametrizzata da vettori $\theta_c = (\theta_{c1}, ..., \theta_{cn})$ per ogni classe c, dove n è il numero di features (nel caso dei testi, la cardinalità del dizionario) e θ_{ci} è la probabilità $P(x_i|c)$ che la feature x_i appaia in un elemento di classe c. I parametri θ_c sono stimati dalla funzione:

$$\hat{\theta}_{ci} = \frac{N_{ci} + \alpha}{N_c + \alpha n}$$

dove $N_{ci} = \sum_{x \in T} x_i$ è il numero di volte in cui la feature i-esima appare in un elemento di classe c del training set T e $N_c = \sum_{i=1}^{n} N_{ci}$ è il numero totale di features della classe c. Il coefficiente di smoothing $\alpha \geq 0$ tiene conto delle features non presenti negli esempi di training ed evita che ci siano probabilità nulle in computazioni successive. Per $\alpha = 1$ si ha uno smoothing laplaciano, mentre se $\alpha < 1$, si ha uno smoothing di Lidstone.

Naive Bayes complementare Naive Bayes complementare è una variante dell'algoritmo Naive Bayes multinomiale particolarmente adatta ad ad insiemi di dati non bilanciati. In particolare, tiene conto del complemento di ciascuna classe per calcolare i parametri del modello. Tendenzialmente, ha prestazioni migliori rispetto a NB multinomiale nella classificazione di testi. La procedura per calcolare i parametri è la seguente:

$$\hat{\theta}_{ci} = \frac{\alpha_i + \sum_{j:y_j \neq c} d_{ij}}{\alpha + \sum_{j:y_j \neq c} \sum_k d_{kj}}$$

$$w_{ci} = \log \hat{\theta}_{ci}$$

$$w_{ci} = \frac{w_{ci}}{\sum_j |w_{cj}|}$$

dove le sommatorie sono definite per tutti i documenti j non appartenenti alla classe c, d_{ij} è, alternativamente, la frequenza assoluta o il valore tf-idf del termine i-esimo del documento j, α_i è un iperparametro di smoothing in NB multinomiale e $\alpha = \sum_i \alpha_i$. La seconda normalizzazione risolve l'influenza che documenti di elevata lunghezza hanno sulla stima del parametro in NB multinomiale. La regola di classificazione è:

$$\hat{c} = \arg\min_c \sum_i t_i w_{ci}$$

Sistema di raccomandazione content-based

L'obiettivo di un sistema di raccomandazione content-based è raccomandare all'utente x_i contenuti simili a quelli che costui ha gradito. Vengono dunque definiti un vettore delle feature $f(s_j)$ per rappresentare il contenuto s_j e un profilo $profile(x_i)$ che rappresenti l'utente x_i. Il profilo dell'utente è uguale a:

$$profile(x_i) = \frac{1}{\sum_j U(x_i, s_j)} \sum_{j=1}^{n} U(x_i, s_j) f(s_j) \tag{1}$$

dove $\frac{1}{\sum_j U(x_i, s_j)}$ è il numero di contenuti valutati dall'utente. Per stabilire se un contenuto è pertinente per l'utente, esistono diversi metodi, tra cui il metodo di similarità del coseno:

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}} \tag{2}$$

Fissata una soglia $t \in [0, 1]$, un contenuto s_j è raccomandato all'utente x_i se $u(x_i, s_j) = \cos(profile(x_i), f(s_j)) > t$.

Implementazione

Il sistema è suddiviso in due moduli distinti. Il primo modulo, denominato `recommendation_engine`, è sviluppato in Python e ha la funzione di richiedere i dati, processarli e organizzarli; è stato fatto uso di alcune librerie del linguaggio: per eseguire le richieste all'API di Wikipedia sono state usate le librerie wrapper `wikipedia` e `mwclient`, le cui funzionalità si sono dimostrate, in diverse occasioni, complementari; per le elaborazioni dei dati sono stati usati, principalmente, gli strumenti delle librerie `spacy` e `sklearn`.
Il secondo modulo è invece sviluppato in PHP e HTML attraverso il framework Laravel e si occupa della presentazione dei dati e dell'interazione con l'utente. I dati sono memorizzati in un database SQL a cui accedono entrambi i moduli.

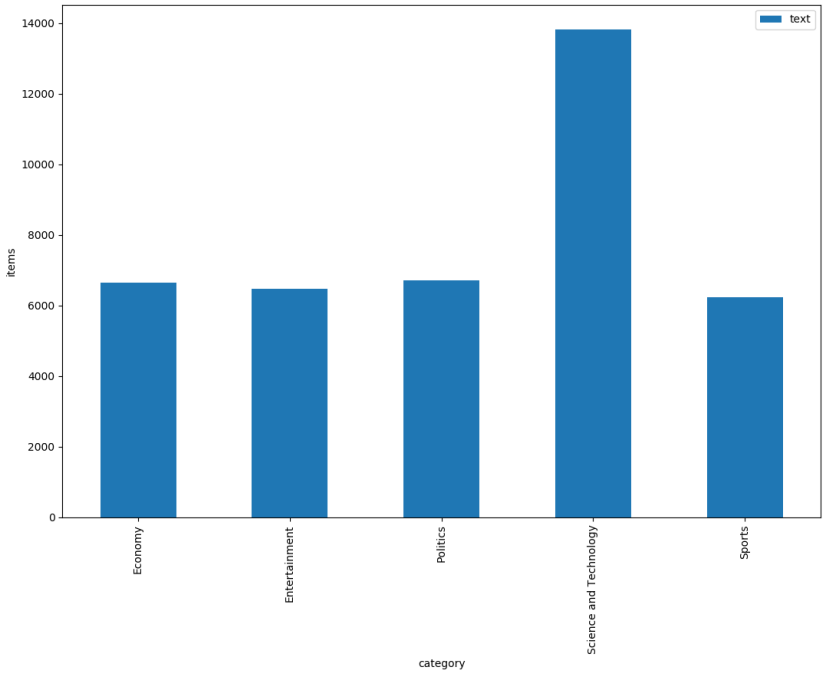
Dati e rappresentazione

Per costruire il dataset, è stato sviluppato un sotto-modulo che esegue iterativamente delle interrogazioni all'API REST di Wikipedia e ottiene il numero desiderato di pagine, dividendole nelle cinque categorie. Lo pseudocodice della funzione per richiedere le pagine di una categoria è il seguente:

```
def function(category, pages_number):
    pages_set:=[]
    while category.not_empty
            and len(pages_set)<=pages_number:
        query_result:=make_request(category.pop())
        for element in query_result:
            if len(pages_set)>=pages_number:
                break
            elif 'Category:' in element['title']:
                category.append(element['title'])
            else:
                summary:=get_summary(element['title'])
                pages_set.append(summary)
    return {'text':pages_set, 'category':category}
```

A causa dell'organizzazione a grafo delle categorie di Wikipedia, è stato necessario eliminare i duplicati degli estratti delle pagine; da un dataset richiesto di 45.000 elementi è stato pertanto ottenuto un dataset definitivo di 39.897 elementi. Tali estratti sono stati infine processati con un lemmatizzatore e uno stop words remover. Il dataset risultante ha la seguente distribuzione, come mostrato in figura 1:

Figura 1: Numero di documenti del dataset divisi per categoria



- Economy: 6.636 elementi - 16,6%
- Entertainment: 6.482 elementi - 16,3%
- Politics: 6.722 elementi - 16,8%
- Science and Technology: 13.826 elementi - 34,7%
- Sports: 6.231 elementi - 15,6%

Per poter applicare l'algoritmo di classificazione, è stata elaborata una rappresentazione vettorizzata dei dati, effettuando una normalizzazione delle frequenze dei termini all'interno del corpus di documenti. Inoltre, è stato riscontrato un miglioramento dello 0,1-0,5% nelle prestazioni applicando la normalizzazione *inverse document frequency (idf)*. In questo modo è stata fornita al modulo classificatore una rappresentazione Bag of Words dei dati. Con una frequenza fissata, il sistema interroga il server Wikipedia per ottenere la lista delle pagine modificate nell'intervallo di tempo intercorso dall'ultima richiesta. Di ciascuna pagina viene richiesto il sommario, ossia la sezione introduttiva che ne sintetizza, per sommi capi, il contenuto. Il sistema processa l'estratto della pagina come precedentemente descritto per il dataset.

Classificazione

Per classificare le pagine in base all'estratto, è stato scelto un classificatore Naive Bayes complementare (NBC), risultato, in media, migliore dell'1,5% nei test effettuati rispetto al classificatore Naive Bayes multinomiale (NBM). Il dataset è stato suddiviso casualmente in training set TR e test set TS con una proporzione di 3:1.
A parità di seed per la suddivisione casuale in TR e TS del dataset, per ciascun algoritmo sono stati ottenuti i seguenti valori massimi di *f1-score*:

- Naive Bayes multinomiale: 80,18%, 9000 features
- Naive Bayes complementare: 81,82%, 39000 features

Il confronto dell'indice di prestazione *f1-score* dei due algoritmi in funzione del numero di features, ossia della cardinalità del dizionario, è visibile nelle figure 2 e 3. Come riscontrabile in figura 3, l'algoritmo ha elevato *f1-score* in un intorno di raggio 3 di 19000; per una questione di efficienza, si è scelto di limitare la cardinalità del dizionario, ovvero il numero di features, a 19000. Sotto tali condizioni, la matrice di confusione relativa all'algoritmo di classificazione è presentata in figura 4.
Gli errori di classificazione risultano legati, principalmente, alle pagine i cui sommari processati hanno una lunghezza inferiore alla media, pari a $330,68$ caratteri.

Figura 2: F1-score dell'algoritmo Naive Bayes multinomiale in funzione del numero di features

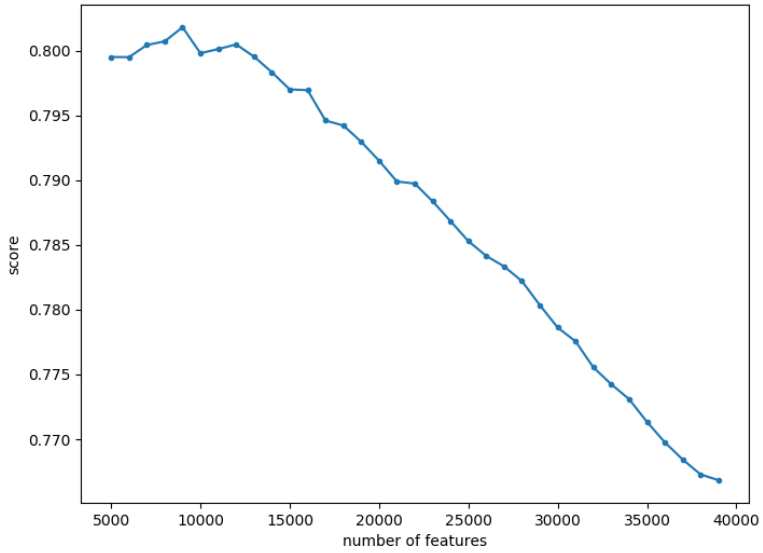


Figura 3: F1-score dell'algoritmo Naive Bayes complementare in funzione del numero di features

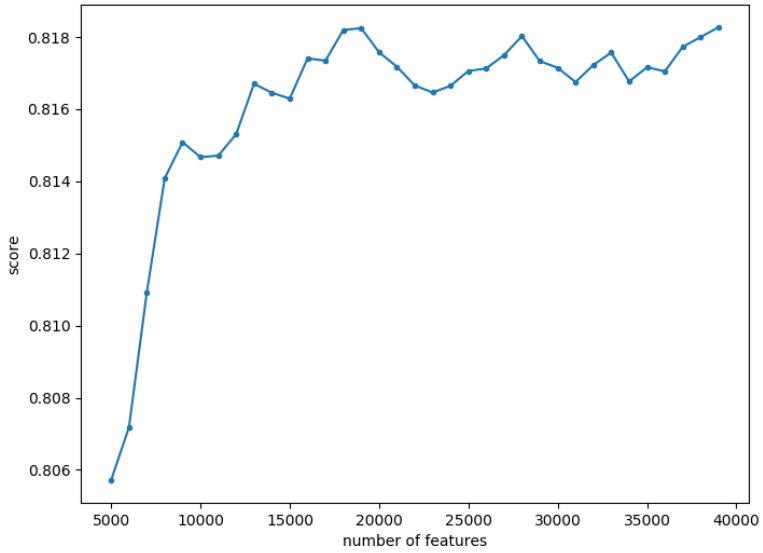
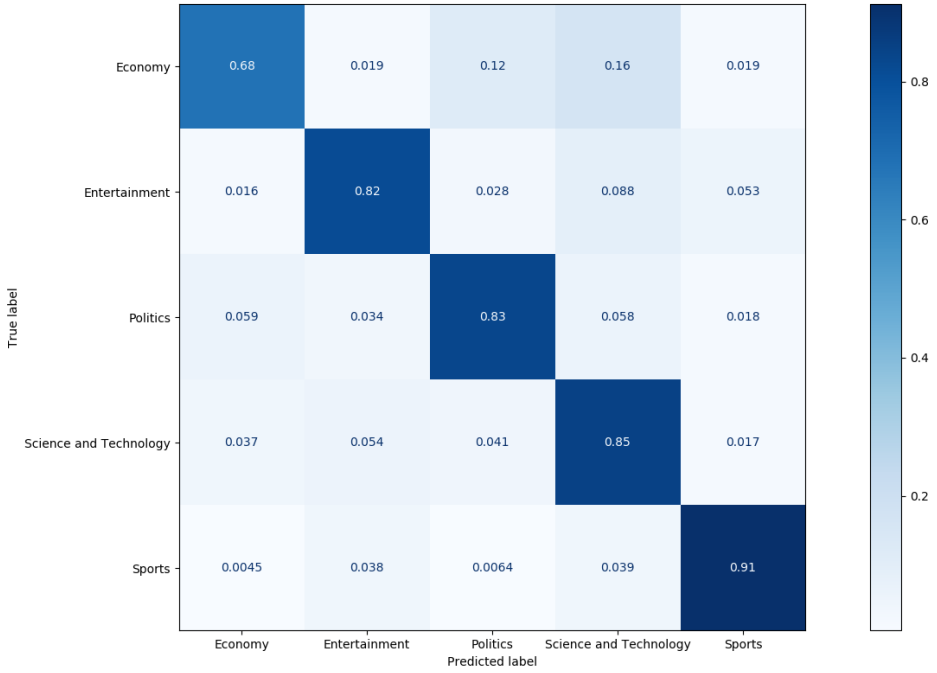


Figura 4: Matrice di confusione relativa a Naive Bayes complementare con dizionario di cardinalità pari a 19000



Sistema di raccomandazione

Il vettore delle features delle pagine e il profilo dell'utente sono stati fissati come vettori di lunghezza pari al numero delle classi, vale a dire pari a 5. Per quanto riguarda le pagine, come valori del vettore delle feaures sono state considerate le probabilità calcolate da Naive Bayes per ciascuna classe; in tal maniera, si è cercato di tener conto di eventuali argomenti secondari trattati in ciascuna pagina (argomenti estranei alla pagina hanno invece probabilità approssimabile a zero).
Il profilo dell'utente è stato computato in base all'interazione con i contenuti; come manifestazione di interesse per un contenuto da parte dell'utente è stata scelta l'apertura del link alla pagina. Nel calcolare il profilo, per ragioni di implementazione, è stato preferito binarizzare il vettore delle features f della pagina selezionata s_j:

$$f_i(s_j) = \begin{cases} 1 & \text{se } s_j \in c_i, \text{ con } c_i \in \text{C} \\ 0 & \text{altrimenti} \end{cases}$$

dove C è l'insieme delle classi. L'aggiornamento del profilo è stato ottenuto applicando la formula:

$$profile(x_i) = \begin{cases} 0 & \text{se n=0} \\ \frac{(profile(x_i) \cdot n) + f(s_j)}{n+1} & \text{altrimenti} \end{cases}$$

dove n è il numero di pagine gradite dall'utente. Una pagina viene raccomandata sulla base della similarità del coseno (eq.2) con soglia $t = 0,8$, in modo da evitare l'eccessiva specializzazione del sistema di raccomandazione. Infine, in fase di cold start il profilo dell'utente è stato posto a zero, come specificato nella formula sopra riportata.

Conclusioni

In questo progetto è stato sviluppato un sistema di news feed basato sui contenuti di Wikipedia, implementando un sistema di classificazione e un sistema di raccomandazione content-based. Grazie ad un dataset di 39.897 estratti di pagine di Wikipedia, è stata compiuta una classificazione in tempo reale delle pagine più recenti. Dai risultati mostrati, il sistema dimostra buone prestazioni, ma, chiaramente, c'è ancora un ampio margine di miglioramento. Una possibile miglioria potrebbe consistere in un maggior controllo dei dati in input, per individuare e trattare conseguentemente le pagine con un contenuto esiguo o incompleto.

Bibliografia

[1] *Bag-of-words model*, [*From Wikipedia, the free encyclopedia*], https://en.wikipedia.org/wiki/Bag-of-words_model

[2] Michel Kana, Ph.D. *Representing text in natural language processing*, Jul 15, 2019, https://towardsdatascience.com/representing-text-in-natural-language-processing-1eead30e57d8

[3] *Naive Bayes*, https://scikit-learn.org/stable/modules/naive_bayes.html

[4] *Naive Bayes classifier*, [*From Wikipedia, the free encyclopedia*], https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[5] *Text Classification*, https://developers.google.com/machine-learning/guides/text-classification

[6] Susan Li. *Multi-Class Text Classification with Scikit-Learn*, Feb 19, 2018, https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f

[7] *Text Classification and Naïve Bayes - The Task of Text Classification*, https://web.stanford.edu/class/cs124/lec/naivebayes.pdf

[8] *Recommendation*, https://developers.google.com/machine-learning/recommendation

[9] *Content-based Filtering*, https://developers.google.com/machine-learning/recommendation/content-based/basics?hl=en

[10] Jiahui Liu, Peter Dolan, Elin Rønby Pedersen. *Personalized News Recommendation Based on Click Behavior*, https://static.googleusercontent.com/media/research.google.com/it//pubs/archive/35599.pdf