ONLINE BOOK SALES IN E-COMMERCE WITH MULTILINGUAL SUPPORT AND MOBILE ALERT

S Monisha*, K Sangeetha¹ and S Annapoorna¹

*Corresponding Author: S.Monisha monisha.st@gmail.com

INTRODUCTION

E-commerce encompasses buying and selling of products and services through electronic systems such as internet and other computer related networks. With English being the most predominantly used language in conventional websites, it naturally limits the number of potential customers who are from other parts of the world speaking different languages. Hence, a multilingual online e-commerce store that can be able to translate the content of website into multiple languages and a function for multi-currency that lets users to place their orders and pay in their relevant currency are needed. This means that rather than restricting the sales of company to the local markets, it provides global internet users the chance to learn more about the products and purchase them from e-commerce website. With different modes of payment and easy sign-up options, online shopping saves time and reduces search costs compared to conventional shopping process.

However, there is concern over online shopping especially when customer’s personal information and financial transactions is required to facilitate transaction through internet medium. The illegal use of customer’s information by third parties discourages online consumer’s intentions from purchasing via online stores. The most important security methodologies followed to
overcome this include Privacy Enhancing Technologies (PETs) that informs about the privacy policies of organizations, privacy seal program by TRUSTe that verifies the privacy policies of web sites that are a part of their program and Public Key Infrastructure (PKI) based Secure Socket Layer (SSL) protocols.

BACKGROUND AND RELATED WORKS

The rapid growth of online shopping is vivid from the statistical report of US which shows that US retailers sales is expected to increase 13% to $262 bn this year that generated $231 bn last year, according to forecasts technology and market research. Several research works are going on to improve the security aspects of e-commerce websites and optimize search results based on Natural Language Processing (NLP), which allows the users to efficiently search for complex information in their own natural language. Some of the areas of research are discussed below.

Building a Multilingual E-commerce Website for Global Market

An Oracle Technical White Paper (2000) demonstrates how to develop a global e-commerce website using Oracle8i, JDBC (Java Data Base Connectivity), and JSP (Java Server Pages). Oracle8i UTF8 database serves as the back-end for a global online bookstore where data of different languages is loaded into the database, which supports multilingual text in a single encoding. Using UTF8 as its database character set, an Oracle8i database server stores multilingual data that is used by the application. Using the JDBC API, JSP pages, one can access the database in a portable fashion using the Structured Query Language (SQL).

A key element of handling multilingual data is the use of Unicode (UTF), a character encoding scheme developed by the Unicode Consortium for the purpose of including all characters used by all languages of the world.

Protecting Web Usage of Credit Cards Using One-time Pad Cookie Encryption

Donghua Xu et al. (2002) introduces a scheme for distributed storage of sensitive information, in specific a One-Time Pad HTTP cookie encryption protocol, to avoid gathering credit card in a central database, while providing similar user convenience at the same time. This approach encrypts the credit card information using One-Time Pad, an unconditionally secure encryption method, and stores this information as cookies on the customers’ computers. The central database of the web site only stores the one-time keys of the cookies, so even if an attacker breaks into the server, what she would obtain are the one-time keys which are just random strings to her without the corresponding cookies.

Using a Relational Data Base For an Inverted Text Index

Steve Putz (1991) illustrates the implementation of text search using Inverted indices, a data structure that maps a search key (e.g., a word) to a postings list enumerating documents containing the key that are commonly stored and updated using B-Trees. It allows efficient insertion, lookup and deletion, the file-based B-Tree data structure is useful for implementing a large dynamically updated inverted index. It also illustrates the drawbacks of B-trees which include unavailability of good B-Tree packages for most computer languages and operating systems, difficulties in implementing efficient and reliable B-Tree software, and complex insertion and
deletion algorithms for lengthy records.

**CHALLENGES OF DEVELOPMENT, PROTECTION AND RELEVANCY**

By definition, e-commerce websites have global reach. So, major websites are already transforming their monolingual content to be able to present in multiple languages. A multilingual web presence will help increase sales by making website content available to a wider audience providing consumer a matter of trust thereby increasing on-line revenue, market share, and client satisfaction. The major challenge in providing multilingual support is the translation of up-to-date content in all the languages as new words are booming daily. There are two methods that solve this purpose. In first method, the content on the website can be provided in multiple languages using Unicode encoding discussed earlier. In second method, the website can use online machine translation tools to translate the content in one language to other languages. Google Translate is one such free translation service that provides instant translations between dozens of different languages using a method of statistical machine translation.

Another aspect that drags customers concern is the protection of user’s personal and financial information. The use of security mechanisms such as digital signature and certificates are helpful in controlling the risk of fraud for online-based transactions to some extent. Currently, online transactions uses One-time Password Authentication (OTP Authentication) technique that generates highly secure one-time passwords ensuring that only properly authenticated users have authorized access to critical applications and data.

The success and popularity of any e-commerce website is decided by the fast, accurate and relevant results based on the customer’s searching. Displaying irrelevant results creates bad experience to the users. A good website search engine displays appropriate results irrespective of the different search queries used. Hence, besides the conventional method of Inverted Indexing, techniques to eliminate stop words (common words) are implemented.

**PROPOSED SYSTEM**

Online Book Sales System is equipped with global, faster, secure and efficient access and an android application for mobile users. The global reach is provided with the Google translate API which transforms the web content to different languages for better understanding. A system is secure when the user’s information are protected. Encryption of data is a wise choice and the award winning Keccak algorithm takes care of it. The transactions are carried out using trusted gateways such that transfer of money is preceded only if the customer enters randomly generated One Time Password correctly which is sent to their mobile phones. The processing of search queries is made faster and appropriate using Inverted indexing with stemming algorithm that helps to fetch the keywords irrespective of user’s irrelevant sentences.

**Implementation of Multilingual Support Using Google Translate**

Google Translate is a free translation service that provides instant translations of words, sentences and web pages between any combinations of the 80 languages supported by it. It uses the method of Statistical Machine Translation (SMT) which treats the translation of natural language as a
machine learning problem. By examining many samples or patterns of human-produced translation, SMT algorithms automatically learn how to translate. Google Translate can make intelligent guesses as to what an appropriate translation should be. Statistical machine translation starts with a very large data set of good translations, that is, a corpus of texts (e.g., a Nations documents as referred in Figure 1) which have already been translated into multiple languages, and then uses those texts to automatically infer a statistical model of translation. That statistical model is then applied to new texts to make a guess as to a reasonable translation (Figure 1).

The translation model attempts to match the strings (i.e., words or phrases) of the source language to strings of the target language. Toward that end, the model looks at each pair of strings and assigns a probability value to the pair, \( p(f|e) \). This value is a conditional probability, and in this case is the probability of one string in the source language (labeled “f” for French) given the occurrence of another string in the target language (labeled “e” for English). The values of \( p(f|e) \) are determined based on preexisting human translations of one language to the other, called “parallel corpus”. The value of \( p(f|e) \) is calculated as follows,

\[
p(f|e) = \sum_a \left( p(f, a|e) \right)
\]

The higher the value of \( p(f|e) \), the more likely that the machine translation will actually look like a translation; and the lower the value of \( p(f|e) \), the less likely that the machine translation will look like a translation. The language model determines the probability of the string of the target language actually occurring in that language, \( p(e) \). Unlike the translation model, parallel corpus is not needed and text in only one language is required. There are number of ways to determining this value. One example is the trigram model, perhaps the simplest model. In this model, the probability that a sentence of length \( N \) will occur in the language is the product of probability of each \( k \)-th word given the occurrence of the prior two words, \( k-1 \) and \( k-2 \). Finally, after having calculated the product of the translation model and the language model, the decoder algorithm selects the string of the target language with the highest probability.

**Password Encryption Using Keccak Or SHA-3 Algorithm**

Keccak or SHA - 3 (Secure Hash Algorithm)
algorithm is a family of sponge functions in which message blocks are XORed into the initial bits of the state, which is then invertibly permuted. In the version used in SHA-3, the state consists of a 5×5 array of 64-bit words, 1600 bits total. The sponge function is a generalization of the concept of cryptographic hash function with infinite output and can perform quasi all symmetric cryptographic functions, from hashing to pseudo-random number generation to authenticated encryption. Instead of storing the passwords of the users directly into the database, this algorithm encrypts it and stores the encrypted value protecting the passwords from attacks. SHA-3 has three phases: Initialization for padding, absorbing phase and squeezing phase.

It is a sponge function with members KECCAK[r, c]. The parameters r = bit rate, c = capacity determine width of Keccak-f permutation, and also applied to sponge construction. The width values are restricted to 25, 50, 100, 200, 400, 800, 1600 right now. Thus the width can be thought as 25*2l where the values for l are (0, 1, 2, 3, 4, 5, 6). It also does not require masking in case the output is smaller since you can generate arbitrarily long outputs.

The sponge construction consists of building a function that takes arbitrary length input and outputs (Figure 2). It does some fixed length of transformations on fixed number of bits say ‘b’ called width. This width can be expressed as ‘b = r + c’ where ‘r’ is the bit rate and ‘c’ is the capacity. The input message is divided into pieces of length r bits each.

The illustration of construction of sponge function is shown in Figure 2 given a message M the following procedure takes place.

**Initialization and Padding**: The state is first initialized to zero. || operator below stands for concatenation.

S[x, y] = 0, for all (x, y) in (04, 04)

P = M || 0x01 || 0x00 || || 0x00

P = P XOR (0x00 || || 0x00 || 0x80)

**Absorption Phase**: The sponge state initially consists of all zeros. The first input block of length ‘r’ is XORed with ‘r’ bits of the state and transform

---

**Figure 2: Illustration of Sponge Function**

![Figure 2: Illustration of Sponge Function](image)

This article can be downloaded from http://www.ijerst.com/currentissue.php
functions are applied on the state. Next input block is then XORED with this state like the previous one and transformed. This continues till all the input is consumed.

for all block Pi in P

\[ S[x, y] = S[x, y] \oplus P_i[x+5*y], \text{XOR with the initial state for all } (x, y) \text{ such that } x+5*y < (r/w) \]

\[ S = \text{Keccak-f}[r+c] (S), \text{Calling the functions to be applied on the state.} \]

end for

**Squeezing Phase:** After all the blocks of the message have been consumed by the absorbing phase, next is to squeeze the given amount of bits required. That is the number of bits are extracted from the state at bit rate and appended to the empty string, and at each iteration till the bit rate makes up for the required number of bits the Keccak-f function is also called.

\[ Z = \text{Empty string, while output is requested} \]

\[ Z = Z \parallel S[x, y], \text{for all } (x, y) \text{ such that } x+5*y < (r/w) \]

\[ S = \text{Keccak-f}[r+c] (S) \]

The Keccak-f function called to do the transformation function does the transformation \(12+2l\) times on the provided data. Pseudo code for Keccak-f[b](A) is given by,

for all \( i \) in \( 0 \ldots n_r-1 \)

\[ A = \text{Round}[b] (A, RC[i]) \]

end for

return A

Where 'A' denotes the complete permutation state array and \( RC[i] \) are the round constants. The number of rounds \( n_r \) depends on the permutation width which is calculated by \( n_r = 12 + 2l \), where \( 2l = w \). This yield 12, 14, 16, 18, 20, 22, 24 rounds for Keccak-f[25], Keccak-f[50], Keccak-f[100], Keccak-f[200], Keccak-f[400], Keccak-f[800], Keccak-f[1600], respectively.

Since Keccak is based on sponge construction it is secure to upper limit of \( 2^{c/2} \) where 'c' is the capacity chosen. So if a higher capacity is chosen and numbers of rounds are increased, higher security can be achieved.

**Search Using Inverted Indexing**

An inverted index (also referred to as postings file or inverted file) is an index data structure storing a mapping from content, such as words or numbers, to its locations in a database file, or in a document. An inverted index contains two parts: an index of terms, (generally called lexicon, index, or term dictionary) which stores a distinct list of terms found in the collection and a posting list, a list of documents that contain the term. An inverted index is able to do many accesses in \( O(1) \) time at a price of significantly longer time to do an update, in the worst case \( O(n) \). Index construction time is longer as well, but query time is generally faster than with a b-tree. Since index construction is an off-line process, shorter query processing times at the expense of lengthier index construction times is an appropriate tradeoff.

**Index Construction:** The Index construction requires determination of indices where index is simply a list of terms. Since we rarely traverse this list, a hash table is often used as the data structure for the index. A hash table permits quick access to a term by applying a hash function on the term. If two terms have the same hash entry, a collision resolution algorithm (such as simply using a linked list of collisions) can be employed.
A posting list indicates, for a given term, which documents contain the term. Typically, a linked list is used to store the entries in a posting list. This is because in most retrieval operations, a user enters a query and all documents that contain the query are obtained. This is done by hashing on the term in the index and finding the associated posting list. Once the posting list is obtained, a simple scan of the linked list yields all the documents that contain the query.

If there are n documents to be indexed then a unique document id is set for each document from 0 to n-1. The postings list for a term is sorted based on document id. At the end of sort processing this data structure returns the top k documents in the postings list where k is the maximum returning capacity of a website search engine in a single stretch.

\[
\text{Inverted Index (docid } n, \text{ doc } d) \\
H \leftarrow \text{new HashMap} \\
\text{for all term } t \text{ in doc } d \text{ do} \\
H \{t\} \leftarrow H \{t\} + 1 \\
\text{for all term } t \text{ in } H \text{ do} \\
\]

Emit (term t, posting \{n, H \{t\}\})

Search Algorithm:

Search (HashMap M, string word)
return M[word]

In the search part of an inverted index, the word which is queried by the user is passed as input along with the hash map which has the value of ids of the entire document. Hash map takes the word as its index and returns the value stored in that index which is its location in the cache memory for smaller data or disk for larger dataset.

Figure 3 depicts the architecture of Inverted indexing using cache memory. When the user enters a keyword, the Hash Map with the indices and ids of documents are searched. The ids are obtained and the corresponding location of the file is identified. The description of products, images and other details are then fetched and displayed.

It should be noted that ignoring stop words are words like “a”, “the”, “of” etc., and transforming words it to its morphological root (e.g.,
“computer”, “computation” to compute) should be done when indexing the document collection and while searching for user queries. Porter Stemming algorithm is used for this purpose.

**Porter Stemming Algorithm Steps**

Step 1: Gets rid of plurals and -ed or -ing suffixes.
Step 2: Deals with vowel in the stem.
Step 3: Maps double suffixes to single ones: -ization, -ational, etc.
Step 4: Deals with suffixes, -full, -ness. etc.
Step 5: Takes off -ant, -ence, etc.
Step 6: Removes a final -e.

It is proved that Inverted Indexing combined with Stemming algorithm gives appropriate results according to user queries. The speed of searching and relevancy of results is improved though index construction takes considerable time.

**CONCLUSION**

E-commerce is a revolutionary tool with the potential to transform the present business transaction. Every month, more and more businesses are moving towards a global, database-driven e-commerce solution for their customers’ needs. This paper demonstrated about the techniques that can be applied to real time e-commerce websites to make it an appealing one to the users.

The key advantage of our system is that it facilitates the customer in a user friendly manner by letting them view the content in their native language, protecting their personal and financial information and helping them to search product with ease.

**BIBLIOGRAPHY**


8. Using a Relational Database for an Inverted Text Index (Steve Putz Xerox Palo Alto Research Center).