Folder Classification of Urdu and Hindi Language E-mail Messages

M. Tariq Banday  
Department of Electronics and Inst. Technology  
University of Kashmir  
Srinagar, India  
E-mail: sgrmtb@yahoo.com

Shafiya Afzal Sheikh  
Department of Electronics and Inst. Technology  
University of Kashmir  
Srinagar, India  
E-mail: sheikh.shafiya@gmail.com

Abstract — Automatic classification of e-mail messages into folders help recipients to organize messages in their local mail store. Both web and desktop e-mail client applications permit customization to support e-mail classification by defining rules to handle incoming e-mail messages. However, automatic language based folder classification of e-mail messages is neither supported by commercial web mail applications nor by desktop client mail applications. This paper reviews e-mail folder classification and language detection techniques. It proposes the use of Microsoft language translator API service to classify incoming e-mail messages into language specific folders. The evaluation of the proposed procedure by experimentations through a compatible e-mail client utilizing this service developed for the said purpose has shown significant classification accuracy.

Keywords – E-mail Classification; Language Detection; Folder Classification; E-mail Client; E-mail Language

I. INTRODUCTION

Automatic filtering, classification and organization of data with reasonable degree of accuracy is crucial in the current digital world where very large amount of data is generated, transmitted, processed and stored. The exponentially increasing volume of e-mail messages constitutes one of the major portions of this enormous data. With the standardization of the representations of non-English languages in computers and support for their processing in growing number of applications, e-mail messages are also composed in languages other than English language. As such, government and corporate officials have to process and respond to such e-mail messages especially in those parts of the world where official language is not English or where multiple languages are officially recognized. From marketing point of view it is also desirable to communicate with the clients in their preferred language. But, it may not be possible for an official to understand and respond to e-mail messages composed in languages not known to him, therefore, automatic classification of such e-mail messages in language folders is desired so that these can be forwarded to relevant official for their satisfactory disposal.

An e-mail message can contain diverse types of objects that include documents, pictures, audio, video, etc., but its primary constituent element is the text message itself. Though e-mail classification is not exactly same as traditional text categorization as e-mail messages are poorly structured in comparison to longer texts and often may be shorter and written in informal ways, however, the official and business communication is often formal, therefore, automatic classification of unseen e-mail message into most probable folder to which it belongs for such communications is essentially a text classification problem. This paper reports the usage of an online language detection and translation service for folder classification of Urdu and Hindi language e-mail messages.

II. LITERATURE REVIEW

Classification of e-mail messages into folders can help its users to manage and organize their growing number of received e-mails, however, such a classification is tedious and cumbersome if performed manually. A semi-automated classification by defining rules as permitted by various mail applications for sorting messages into user created folders can make this classification easier but not dynamic. Further, creation of rules for such a semi-automated classification requires technical skills which lower its effective usability. Machine learning techniques can be employed to analyze the structure of messages to suggest the most appropriate folder from the existing folder structure. This suggestion can help users to decide about most appropriate folder for each message individually. A fully automated classification system learns from the user’s past behavior to predict future classification choice and move messages to proper folders in automated manner. Classification of e-mail messages into classes is a well-researched area, however, most of such works have undertaken its classification into spam [1] and ham classes. This section presents an appraisal of some works carried out for language and folder based classification of e-mail messages.

Classification of Polish language e-mail messages into user specific folders employing several machine classification techniques on self-created e-mail corpus has been undertaken by Stefanowski and Zienkowicz [2]. This study has shown that language processing techniques have little influence on the classification accuracy in comparison to appropriate feature selection. A study by Crawford et al [3], [4] focused towards the management of e-mail messages developed an interface for e-mail management to automatically classify messages into their archive folders. The study employed learning algorithms to predict appropriate class of e-mail message. Alaa El-Halees [5] applied machine learning techniques namely maximum entropy, decision trees, artificial neural nets, Naïve Bayes, support system, machines and k-nearest neighbor to filter English and Arabic e-mail messages. It was found that the performance of filtering Arabic messages is improved when
they are stemmed before classification and classifier employing feature selection are more accurate than those which are not stemmed. Al-Radaideh et al [6] experienced the use of both Graham statistical and rule-based filters to detect and filter Arabic e-mails for alerts about government and political issues, breaking news, and criminal attacks. The designed filter has obtained a reasonable performance in filtering alert Arabic e-mail messages. Park and UnAn [7] proposed e-mail classification methods using Latent Semantic Analysis (LSA) and Nonnegative Matrix Factorization (NMF) wherein first multi-category labels are extracted from the received e-mail messages and then classified into multi-category labels. These methods also support directory searches by using term e-mail frequency matrix. The results have shown high degree of flexibility, efficiency and effectiveness. Krzywicki and Wobcke [8] used the concept of clumping (exploration of local, often abrupt but cohesive changes in data) in their proposed e-mail classification methods: Local Term boosting (LTB) and Weighted Simple Term Statistics (WSTS). It was reported that experiments with Euron corpus showed very high accuracy of these methods. They concluded that these methods are viable alternatives to machine learning methods commonly used in text categorization, such as Naïve Bayes, k-NN, SVM, LDA, LSI and MaxEnt which are more complex and resource demanding. It was also reported that the combination of LTB and WSTS were more accurate than STS or LTB alone. Khusainov [9] has studied e-mail classification into categories using back propagation multi-layer neural network technique based on TF-IDF heuristics. Chakraverty, Venkatachalam and Telang [10] studied the use of graph mining techniques for multi-folder classification of e-mail messages. Unlike other techniques such as n-gram the study took into consideration the positions of the word in the messages to extract representative substructures and rank them for efficient classification. It was concluded that this approach showed significant improvement in performance over Naïve Bayesian method. Sappelli, Verberne and Kraaij [11] converted documents (.doc, .docx, .txt, .pdf) in the existing directory structure on the file system into text and used them for training classifiers (SVM, Naïve Bayes, Decision Tree, KNN and ZeroR) for categorizing e-mail messages but found that this approach cannot be effectively used for classifying e-mail messages, however, the study has found that person names are important features for classifying e-mail messages. Pujara et al [12] identified granular and load-sensitive classification scenarios where intelligent feature acquisition improved classification efficiency. The study employed Adaptive Classifier Cascades (ACC) that combined a series of base classifiers. Tammela et al in a recent survey [13] on the automatic management of e-mail messages by e-mail clients and webmail services concluded that automatic categorization is not supported by these clients or services. The study also found that the use of such tools can be effective for the management of e-mail messages. Ross et al [14] built a custom workflow management system for a specialist engineering company by developing an e-mail classifier to support automatic handling and routing of routine e-mail enquiries. The system operated in corporate environment where information regarding the clients was available. They concluded that the use of Naïve Bayes classifier was more effective than vector-based approaches for badly skewed training data. Weng and Liu [15] used text classification techniques for classifying e-mail messages and accordingly recommending template suggestions for customer service personnel to respond to customer query quickly and effectively. In a similar research, Aloui and Neji [16] proposed the combined use of text-mining and ontological techniques to mine and classify e-mail messages, fetch, generate, and send answers automatically to learners.

III. LANGUAGE DETECTION

Automated language detection of text is a technique whereby written language of an electronic document is identified out of a predetermined set of natural languages. As a preprocessing step, language identification plays a vital role in many Natural Language Processing (NLP) applications like machine translation, text mining, information retrieval, email routing, language based search engines etc. With ever increasing Internet users across globe, huge amounts of data is being processed in different languages for extending NLP and other services for their users. This has significantly boosted the research on automated language identification of text. Automated language identification of text is considered a problem of categorization or classification. The language of text of a document can be determined only when the system has a prior knowledge of this language. In the general scheme of things, a language model is created for each language we are interested in during a training phase. The text of the input document is then compared with each of these models. Minimal distance of the input text with a language model determines its language.

Several techniques have been proposed for automated text detection in the last two decades. In the common words approach proposed by Grefenstette [17], the most frequent words, usually the words with a grammatical function, of languages are stored in the database and the input document is then compared with these word lists. The word list with which it shows maximum resemblances determines its language. In a similar technique, most frequent letter combinations [18] of languages are employed to identify the text documents. One of the important and highly employed techniques of text identification is the n-gram, introduced by Cavanar and Trenkle [19]. The technique works by identifying the most frequent character n-grams of a language. N grams are strings and sub-strings of words with n running from 1 to 5. The n-gram ranking profile which is a list of most frequent n-grams in descending order of frequency serves as the model for the language with which the n-gram profile of input document is compared. The model with which the n-gram profile of input document has minimum distance determines its language. Another significant statistical technique for text identification making use of Markov Models and Naïve Bayes theorem has been proposed by Dunning [20]. In this technique rather than tokenization of text to develop language models through n-grams, data is taken as sequences of bytes to develop character level language models. In yet another technique, language models are developed by compressing the text by using Prediction by Partial Matching (PPM) algorithm. Input text to be identified is also compressed and a comparison is made between the number of bits required for its encoding and the bits in existing language models.
Referring to Cavanar and Trenkle’s n-gram technique, Hayati [21], reports that the technique is not sufficiently powerful enough to discriminate between similar languages. Rather she suggests the use of Fischer discriminative function to select n-grams and cosine similarity for comparing the input text document with the language model.

Most language detection algorithms have been developed and trained on documents that are longer and better formulated. These language detection techniques have also been employed to detect language within short and long text communications such as Twitter and blogs. Hong et al. [22] used LingPipe and the Google Language API and Semicast [23] used an internal proprietary tool to determine the language of a tweet. Carter et al. [24] and Gottron and Lipka [25] identified several challenges with language detection in short texts. In a study involving very short text messages of Twitter and e-mail, Mayer [26] has reported that the efficiency of the text identification technique is enhanced substantially, if the algorithms used are chosen on the basis of the length of the text document. Milne et al [27] after applying the frequent word and trigram technique on two collections of data, the Wikipedia and Europarl, concluded that word based and n-gram techniques respond to different aspects of language. They also concluded that shorter texts need different techniques than longer ones. Carter et al. developed a link-based prior to consider the language of linked-to content and a blogger-based prior to aggregate tweets on a per account basis to form a larger document to enhance performance for classification. It was found that both improve accuracy. Hale [28] in combination with the presence of certain keywords used the Compact Language Detection (CLD) kit for detecting the language of blogs. He found these two methods in combination improve detection accuracy. Hale, Gaffney and Graham [29] studied the reliability of key methods used to determine language and location of content in Twitter. It compared three automated language identification algorithms namely Compact Language Detection Kit (https://code.google.com/p/chronium-compact-language-detector/), Alchemy API (http://www.alchemyapi.com), Xerox Open Source (http://open.xerox.com/Services/LanguageIdentifier) to Twitter’s user language interface setting and to a human coding of languages, and identified common sources of disagreement. In none of the above studies language identification methods were able to match the accuracy of human coding by multiple coders because significant challenges persist to accurately determining the language of tweets in an automated manner. The basic reasons for this included informal writing style, short length of tweets, use of multiple languages within a single tweet and the presence of non-language specific content such as URL’s and emoticons. However, it was concluded that Compact Language Detection Kit and Alchemy hold useful potential for language identification.

IV. LANGUAGE DETECTION AND TRANSLATION USING MICROSOFT TRANSLATOR API


Microsoft Translator is a hosted machine translation service from Microsoft Corporation, which can be accessed via an API to provide language translation and other language-translated services to applications, web sites, or utilities. The Microsoft Translator API can be integrated into web sites or applications, using: i) the Web Widget, ii) AJAX, iii) HTTP, or iv) SOAP interfaces. The Microsoft Translator Web Widget allows free translation of an entire website from one language to another. The AJAX interface permits text translation of website to another language while excluding specific sections from translation. Both web and desktop or client applications can use HTTP interface (HTTP POST and GET) from a variety of programming environments including .NET, Ruby, JAVA, PHP, etc. to call Microsoft Translator API functions for any language translation operation. Client applications can use SOAP interface for asynchronous translation through Microsoft Translator API so that these applications can continue with their user interface until a response is returned from the called translator API function. Examples of such applications include RSS reader, blog title reader, etc.

To use Microsoft Translator API service in websites or desktop applications, it is required to sign up for the service, register the application, and obtain the Client ID and Client Secret. Once registered for the Microsoft Translator service, the Translator API can be integrated into websites or applications, using Web Widget, AJAX, HTTP, or SOAP interfaces.

V. E-MAIL CLIENT FOR FOLDER CLASSIFICATION OF URDU AND HINDI LANGUAGE E-MAILS

This section discusses the usage and evaluation of Microsoft Translator API service for language detection of incoming e-mail messages and their automatic classification into folders based on the language of their content. An e-mail retrieval client to retrieve e-mail messages from users’ server mailboxes using POP3 and IMAP protocols was developed during the current study. It supports SSL connection establishment with the server and provisions for mail deletion on the server. The development was carried out using VB.NET programming language of Visual Studio 2012. SOAP interface was used to invoke Microsoft Translator API methods asynchronously for e-mail folder classification.
A. Working and Experimentation

The working of the proposed scheme for folder classification of Urdu and Hindi language e-mail messages is presented in figure 1. After fetching e-mail from the user inbox, it is converted into UNICODE and submitted online to translation or detection service, which returns a language code based on the language of most part of the message. The received code is used to classify the message into either Urdu or Hindi folders.

As shown in figure 4, the client can be configured for selecting either Hindi or Urdu or both languages for choosing preferred classification option. Further, configuration options also permit the classifier to make classification decisions on either body, subject or both body and subject.

Figure 4: Settings View

B. Tests and Results

The developed e-mail retrieval client was tested for correctness by sending e-mail messages from different e-mail accounts in Urdu, Hindi, English and other languages. Due to non-availability of e-mail corpora of real e-mail messages in Urdu and Hindi languages, the validity and efficiency of the developed client was undertaken by engaging 15 voluntary who were asked to send e-mails in real time to current author’s e-mail account which was configured for retrieval in the developed client. The voluntary were instructed to avoid sending very short text e-mails.

As many as 735 e-mails in different languages were received in the configured e-mail account. Out of these, 79 e-mails were received in languages which were not supported for classification by the client. Out of the remaining 656 e-mails, 580 were classified correctly in their Hindi and Urdu folders. Out of the remaining 87 e-mails, 30 were wrongly classified and 57 e-mails remained unclassified of which 20 were in Urdu language and 37 were in Hindi language.

On inspecting the unclassified e-mails it was found that these e-mails contained text that was too short for classification.
classification. Further, a majority of them were of length larger than that supported by the Microsoft Translator through one call. Images, URL’s and emotions were removed from the text before submitting for language detection and therefore did not bias classification.

**Figure 5: Classification Accuracy**

The wrongly classified e-mails were multi-lingual (e-mails composed in more than one language) and got classified in either of the three folders i.e. General, Urdu or Hindi folders depending upon the language used for most part of the message. The client rendered Hindi and Urdu language e-mails correctly in its interface. Though the client supports only Urdu and Hindi e-mail separation, it can be easily upgraded to support detection and classification of more languages.

**Conclusion and Future Scope**

Existing language detection services can be employed for realization of an e-mail system that can help easy and automatic organization of e-mail messages. In this study, automatic machine language based e-mail folder classification has been realized by use of MS translator service. A POP3/IMAP protocol based e-mail retrieval client was developed and successfully tested for automatic classification of Urdu and Hindi language e-mails. Most of e-mails retrieved by the client that were composed in Urdu or Hindi languages were correctly classified and e-mails composed in other languages were received in the general mail folder. Multi-lingual e-mails got classified on the basis of language of most part of the message. The classification efficiency of the proposed procedure has shown promising performance and therefore, it is suitable for use in existing mail clients. Future study may be conducted to use multiple language detection and translation services and for improving accuracy and response time.

**References**


[23] Semiocast, “Half of messages on Twitter are not in English: Japanese is the second most used language,” 2010, URL: http://semiocast.com/downloads/Semiocast_Half of messages on Twitter are not in English 20100224.pdf.


