Journal of Computer Science

Original Research Paper

Improving Email Response in an Email Management System Using Natural Language Processing Based Probabilistic Methods

Abdulkareem Al-Alwani

Department of Computer Science and Engineering, Yanbu University College, Yanbu, Saudi Arabia

Article history
Received: 06-03-2014
Revised: 10-03-2014
Accepted: 04-08-2014

Abstract: Email based communication over the course of globalization in recent years has transformed into an all-encompassing form of interaction and requires automatic processes to control email correspondence in an environment of increasing email database. Relevance characteristics defining class of email in general includes the topic of thee mail and the sender of the email along with the body of email. Intelligent reply algorithms can be employed in which machine learning methods can accommodate email content using probabilistic methods to classify context and nature of email. This helps in correct selection of template for email reply. Still redundant information can cause errors in classifying an email. Natural Language Processing (NLP) possess potential in optimizing text classification due to its direct relation with language structure. An enhancement is presented in this research to address email management issues by incorporating optimized information extraction for email classification along with generating relevant dictionaries as emails vary in categories and increases in volume. The open hypothesis of this research is that the underlying concept to fan email is communicating a message in form of text. It is observed that NLP techniques improve performance of Intelligent Email Reply algorithm enhancing its ability to classify and generate email responses with minimal errors using probabilistic methods. Improved algorithm is functionally automated with machine learning techniques to assist email users who find it difficult to manage bulk variety of emails.

Keywords: Automated E-Mail Reply, E-Mail Classification, Machine Learning, Algorithm, Text Classification

Introduction

Email has evolved as the most simple and fastest means of communication between people around the world. Since its inception up till now, email has found its place in multiple applications for a variety of reasons. Ever-increasing use of email has resulted in overflowed, uncalled-for and mismanaged email accounts.

Internet has provided a potent platform for users to interact from anywhere in the world. E-mail technology as per now holds a major share in online communication and is being hosted by multiple online services.

As per recent instrument surveys carried out, 30-150 average emails are received every day by users. Frequency of emails varies from user to user but email flooding problems are reported in both professional and personal email accounts. An error in allocating important mail to right category or delay in replying to a critical email can cause serious consequences for users at both ends. This problem requires an effective solution that can employ email processing techniques for optimal management providing improvement in classifying emails and generating appropriate responses.

Email classification is imperative for a robust email management system and improves sorting of incoming emails into relevant groups. Text Classification algorithms such as Bayesian Theorem, Neural Network are actively being used for email classification. These classification techniques can be supervised or unsupervised improving categorization of emails (Liu *et al*., 2010).

Natural Language Processing (NLP) is another approach involving computational analysis of human language text and using this analysis for subsequent classification based on text context and definition. NLP research has been active since early 1950s, but its use in internet technology exponentially increased due to digital nature of technology (Rosenfeld, 2000). In the start, Linguistics techniques based NLP tasks were actively researched, but recently, machine learning and statistical methods have become the focus of research for implementing NLP techniques.

This research is presented as an extension of authors' previous study (Al-Alwani, 2014) on information extraction using Markov probabilistic methods model to train the email system facilitating prompt generation of an email reply. Previous research is improved upon by integrating Natural Language Processing to optimize information extraction by filtering overhead textual information. An Email structure consists of textual content which can be processed using NLP techniques for information extraction and building a dictionary consisting of significant words. Dwelling on this concept, this research presents an improved intelligent email reply algorithm using Natural Language Processing for classification and reply automation. The proposed research in addition to classifying emails will also have the ability to utilize functional mapping using extracted dictionaries from email sample space assisting in generating an automated email response. Dictionaries are created by removing redundant words using probabilistic estimation with NLP techniques providing us with most significant words for each class of emails. Proposed algorithm is broadly applicable to a wide range email classification problems and improves upon time efficiency and response accuracy of email management tasks.

Following hierarchy is followed in this report to provide better understanding of this research. Section-II comprises of a comprehensive review of literature pertaining to email classification techniques and influence of NLP methods in text classification. This is followed by a discussion in section-III on salient objectives of this research and subsequent improvements. A detailed overview and workflow of algorithm is presented in section-IV. This section will also include more explanation on algorithm parts related to email classification and automatic reply using information extraction. Section-V will conclude this study with concise comments on future propositions applicable to this specific research.

Review of Emailclassification Techinques

A major issue when dealing with text data in email body is what attributes of the text content is to be selected to ascertain the context and meaning of the email (Fong *et al*., 2008). These attributes of an email specifically define the words and their context with the neighborhood text, for which feature extraction methods using text classification are actively used. But there are numerous possibilities that cannot be accounted for. For example, some attributes of an email will be more useful than others but that also depends on which class of email is being processed. Moreover, there will be a lot of unnecessary text data will not assist us with learning, which should therefore not included in the machine learning process (Ayodele *et al*., 2007).

Some attributes are very usable in classifying an email and subsequently proper categorization. Proper selection of attributes in an algorithm for email classification can also be used as decision variables for the automatic reply.

Another bottleneck is multiple email formats which occurs due to lack of standardization. Users can communicate with each other following different formal and informal standards and an algorithm for accurate classification in such circumstances must adhere to adaptive classification to work in such a complex environment (Carmona-Cejudo *et al*., 2011).

Chan *et al*. (22004) proposed that there may be an attribute that could be extracted to relate text sections of an email together, but this is yet to be completely optimized due to limitations of Natural Language Processing. But still a degree of accuracy can be achieved depending upon the attributes for classification decisions.

Among a range of attributes to select from, it is clear that the size of the attribute sample space that is extracted can get very large. If simple bag of words approach is used, we have one address in the sample space for each unique word. This can reach up to thousands of words. This can cause problems for the classification algorithms being used (Ayodele *et al*., 2007). Wang *et al*. (2006) conducted an email spam filtering study and reported that majority of machine learning algorithms are unable to handle large number of attributes and their relative weights. Therefore removal of redundant attributes is most imperative in cases of information extraction and performance of any email classification algorithm is directly related to its ability to extract most relevant attributes of email text content.

Related mainstream algorithms used for email classification are investigated and a brief overview of these methods is as follows.

RIPPER Text Classification

RIPPER classification algorithm (Provost, 1999) is often used in automatic email filtering processes. Its architecture is based on rule based framework to sort email. RIPPER has the ability to automatically generate rules for selecting keywords instead of manual selection. Its main advantage is that it is one of the few machine learning algorithms that is capable of dealing with text natively. It is fast and able to deal with a large set of email attributes.

However keyword extraction rules have to be constructed for every possible class and it is easy for emails to be mixed up or irrelevant attribute information is extracted. These extraction rules are very stringent binary rules as they discard information like no. of words in an email. Since they only make binary decisions, so there predictions are not wholly deterministic as strict rules may cause emails to get mixed up in wrong classes. This is caused by attributes competing against each other for possession of an email message. As a direct result, such a system is also unable to learn adaptively. Whenever the attributes of an email are changed, the rules will be recreated from ground up in order accurately assign an email class. This task needs time to complete and other dependent tasks like automatic email reply are directly affected.

Naïve Bayes Classification

Naïve Bayes is an algorithm based on statistical analysis, with decisions and rules being made using numeric data (Haiyi and Li, 2007). It processes an email to match words chosen at random from the total words present in each folder. The words chance of being matched is proportional to the probability of finding that word in the all the classes. Bayes classifier is then used in the next step to determine likelihood that the email being considered belongs to the right class or not.

Naive Bayes is an efficient statistical classifier. It works well with statistical data analyzed in an email. In contrast to rule based algorithms, it can be done incrementally and the additional preprocessing step that is needed to create a word frequency feature vector is quite small. However, the size of the feature vector can get quite large and thus extra steps need to be taken to reduce its dimensionality.

Nearest Neighbour Classification

This approach is explored in a study (Weijie *et al*., 2012) based feature selection using mutual information. Nearest neighbour is a very simple numeric based algorithm which simply treats the feature vector as a vector inn-dimensional space and finds the nearest matching vector in terms of distance. This is calculated in the usual Pythagorean way, but generalized to n dimensions. Boone found that nearest neighbour is particularly effective when only examples of each folder are presented to the algorithm.

Neural Networks and Machine Learning Based Classification and Email Management

Neural networks have yet to fully mature in problems related to intelligent email filtering. Neural networks are a highly sophisticated network of artificial neurons that mimic the way neurons in the human brain work. They can be trained and retrained on given examples of data and then make predictions about the class of untrained examples. In the domain of email filtering, a feature vector must first be created in the same manner as the other numerical algorithms. The network can be trained using a technique known as back propagation and given enough examples, the network will be able to generalize to be able to predict other, unseen attribute vectors.

A NN algorithm is explored in (Arevian, 2007) for robust text classification deals with specifically for email classification. Numerous methods fail in accurate text classification in a scenario with multi-disciplinary viewpoint such as natural language processing and artificial intelligence. The results demonstrate that these recurrent neural networks can be a viable addition to the many techniques used in web intelligence for tasks such as context sensitive email classification and web site indexing.

There have been a lot of advances in the field of intelligent email filtering since the simple application of handmade filtering rules. However there are many things which have yet to be tested. The various preprocessing steps that have been taken suggest that methods that deal within formation gain statistics are very useful and can eliminate the need for other techniques such as stop−word lists. It is apparent that a lot more work needs to be done in the Natural Language Processing field before its practical application is viable. Email reply and template selection is another feature for current research. Yang and Kwok (2012) proposed machine learning methods can be used with superior performance for automation of electronic mails as an email using text classification. In another study authors investigated user-email interface and proposed machine learning routines to support real-time operations like reply prediction, attachment prediction and summary

keyword generation (Dredze, 2009). It was observed that automated email reply algorithms require text classification and subsequent attribute extraction to train decision variables in the system to generate an email reply whose accuracy is dependent on the features being extracted. Inorder to maitain a scalable workflow for analyzing emails for relevant content, extraction of data can be segregated into separate layers. Each layer performs different functions and as a whole can greatly improve the extraction process.

In a semantic based approach published by (Beseiso *et al*., 2012), a method is proposed for extracting information from emails based on ontology based architecture. Their research showed an improvement in email communication along with a progressive utilization of time and resources. This architecture utilized four component layers for overall information extraction; Ontology Learning Component, the Management Component, the Semantic Email Component and the Client Side Plugin.

Statistical methods are also being used for mining for relevant attributes. An important advantage of statistical methods for attribute extraction is that such methods are not dependent on language. Internet content is mostly influenced by use of English language, where communications via email can and are performed in many languages other than English. Statistical methods are therefore able to work effectively in multiple languages that have not been developed for. Another important point to consider in current research is ignoring diversity of features contained in emails, features such as attachments and quoted text. Just like an email written in another language, agents should be prepared to deal with messages containing unusual embedded content such as attachments, or html-formatted text.

The statistical algorithms are able fill gaps that exist in the rule based methods, but at the cost of more processing time. But one area where research is lacking in application is Natural Language Processing (NLP) for insignificant feature selection. While being tedious to apply but offers the potential to classify more effectively on unclassified emails as information extraction using text classification provides not only relative weights between attribute words but also helps in finding attribute for reply template.

Proposed algorithm utilizes NLP and probabilistic techniques for email classification, dictionary building and generating a reply with accurate content. Details on the workflow of proposed algorithm is presented in the next section.

NLP Algorithm: Design and Methodology

This section will discuss application of NLP techniques used in the proposed algorithm for classification of emails along with methodology used for generating an email reply.

Classifying an email has many similarities with text categorization and NLP processes ensure accuracy of text classification. Salient features of proposed algorithm based on the NLP attribute prioritization are as follows:

- Accurate information extraction (attribute extraction)
- Weight assignments (weighting of extracted attributes)
- Discarding redundant attributes (feature reduction)
- Relating attributes using probabilistic methods (attribute selection)
- Template selection
- Template filling

All of these features constitute overall architecture of the algorithm. Algorithm workflow than consists of three major parts which are given as follows.

Algorithm for Information Extraction

Attributes of an email govern the decision process for classification. As discussed in section-II, attributes can vary from email to email typically extracted from the body of the email using text classification. Email text content of in general include sender, recipient, title and body of the email. Text in an email is initially transformed into bag of words representation, in which word definition and relative context is significant and accordingly processed. Main strength of this algorithm is its ability to remove redundant words.

First phase of the algorithm will comprise of development of a dictionary. When a new email arrives, email text is extracted initially and words are classified into non-negative and negative words. Next, verb words are removed and plural words are converted into singular words. After this step propositions, pronouns, interjections, conjunctions are also removed.

In second phase, to minimize dictionary overhead, all remaining words are replaced with synonyms where required. Table 1 and 4 shows an example of synonym database and synonym lookup table respectively. Optimized bag of words is then classified as an Email body. Depending on which algorithm takes a decision to extract all the words

from the bag of words representation in case an email body is detected and only email title words in a case if an email body is not detected. Each word from the bag of words from both cases is then checked in the dictionary to ascertain whether it is present or not. For elaboration a sample dictionary is shown in Table 2 containing significant words and relevant frequency of repitition of these words under each class type. If the word is already in the dictionary, predetermined class of email is allocated. A typical class structure for 'Order' word is shown in Table 3. Incase algorithm detects a new word, a new class is initiated against the word and added in the decision table. Flowchart for creating a dictionary is shown in Fig. 1.

Algorithm for Email Classification

Next part of algorithm consists of email classification and categorization. In email classification part of the algorithm, initial steps are same as first phase for developing a dictionary. After first phase, email body is processed to determine following probabilities for two cases Email Body or No Email Body:

$$PB = \frac{DB_{ji}}{\sum_{k=1}^{W} DB_{ki}} \quad (1)$$

$$PT = \frac{DT_{ji}}{\sum_{k=1}^{W} DT_{ki}} \quad (2)$$

Table 1. Synonym table

Word	Synonym 1	Synonym 2
Order	Buy	Purchase
Receive	Post	Deliver
More words…	…	…

Table 2. Dictionary for order class

Word\frequency	Class 1	Class 2	Class 3
Order	950	500	300
Receive	200	300	100
NOT-order	50	50	900
NOT-receive	100	800	300
Other words…	…	…	…

Table 3. Email classes for orders

Class 1	Class 2	Class 3
Making order	Order problems	Cancelling order

Table 4. Synonym lookup table

Simplest word	Part of speech	Synonym 1	Synonym 2
Happy	Adjective	Jolly	Cheerful
Run	Verb	Sprint	Race
Run	Noun	Sprint	Journey
More words…	…	…	…

These probabilities calculated and are subsequently used for calculating total probability of Email class given as Equation 3:

$$P = P_T . P_B \quad (3)$$

Total probability will determine the class of Email. Class giving the highest probability will be selected and processed Email will be assigned to that class. As it can be seen in Equation 1 and 2, Dictionary for Body and Title will determine the probabilistic relation of an Email to a predefined class. A detailed flowchart for Email classification is shown in Fig. 2.

In removing redundant words, a synonyms lookup table is utilized to consider all synonyms as one word, so that the probabilities for all of them are aggregated and it will make the classification process more precise. WordNet is a good source for extracting the synonyms data. Here is an example of synonyms table.

There is no need to store prepositions, pronouns, conjunctions and interjections in dictionary because they don't help us in classifying emails as they exist on almost all texts regardless of the subject. We store both positive and negative values of each word in dictionary to make the classification more powerful by understanding if something is true or not. For example if the product has been received or not. The dictionary may look like this:

An Example

Assume that we have these 2 emails and we want to classify them:
Title: New Order
Hi,
I would like to order Cloudy with a Chance of Meatballs 2. I want the Blu-ray disk to be delivered to my friend's place.
Best Regards
Title: Product not Received
Hi,
I ordered Cloudy with a Chance of Meatballs 2 a few days ago, but I've not received it yet. How can I track my order?
Best Regards
Let's say we have the following classes of emails:

The following steps should be done both when we're making the dictionary and when we're classifying. (both for body and title):

i. Find any negative auxiliaries (i.e., not, can't, doesn't, didn't) and mark them, so that they won't be removed on next steps.

ii. Stem all the verbs.
iii. Remove all prepositions, pronouns, conjunctions and interjections.
iv. Replace plural forms of words by singular forms.
v. Using the synonyms table, replace synonyms with the simplest possible word.

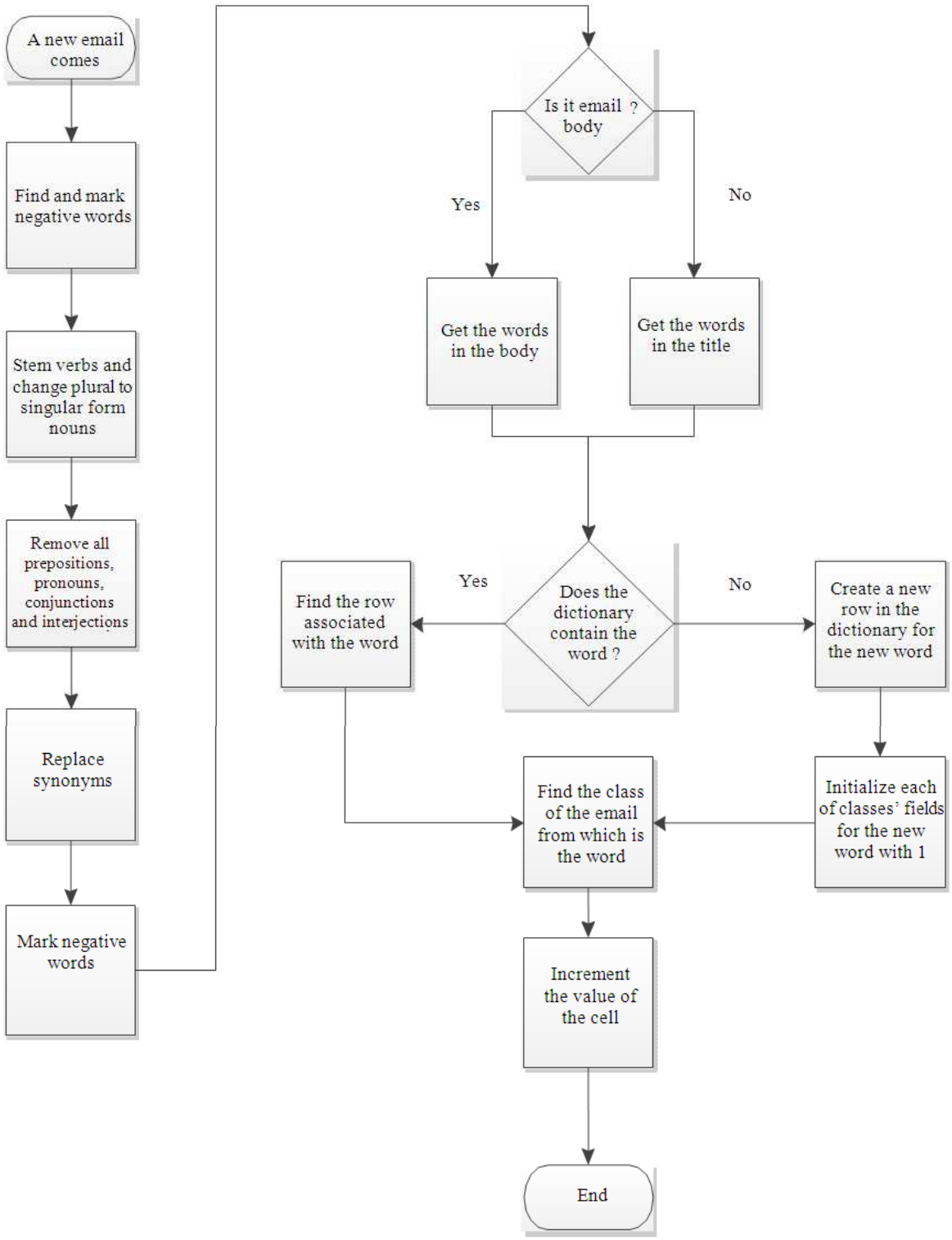

Fig. 1. Flowchart for extracting a dictionary word

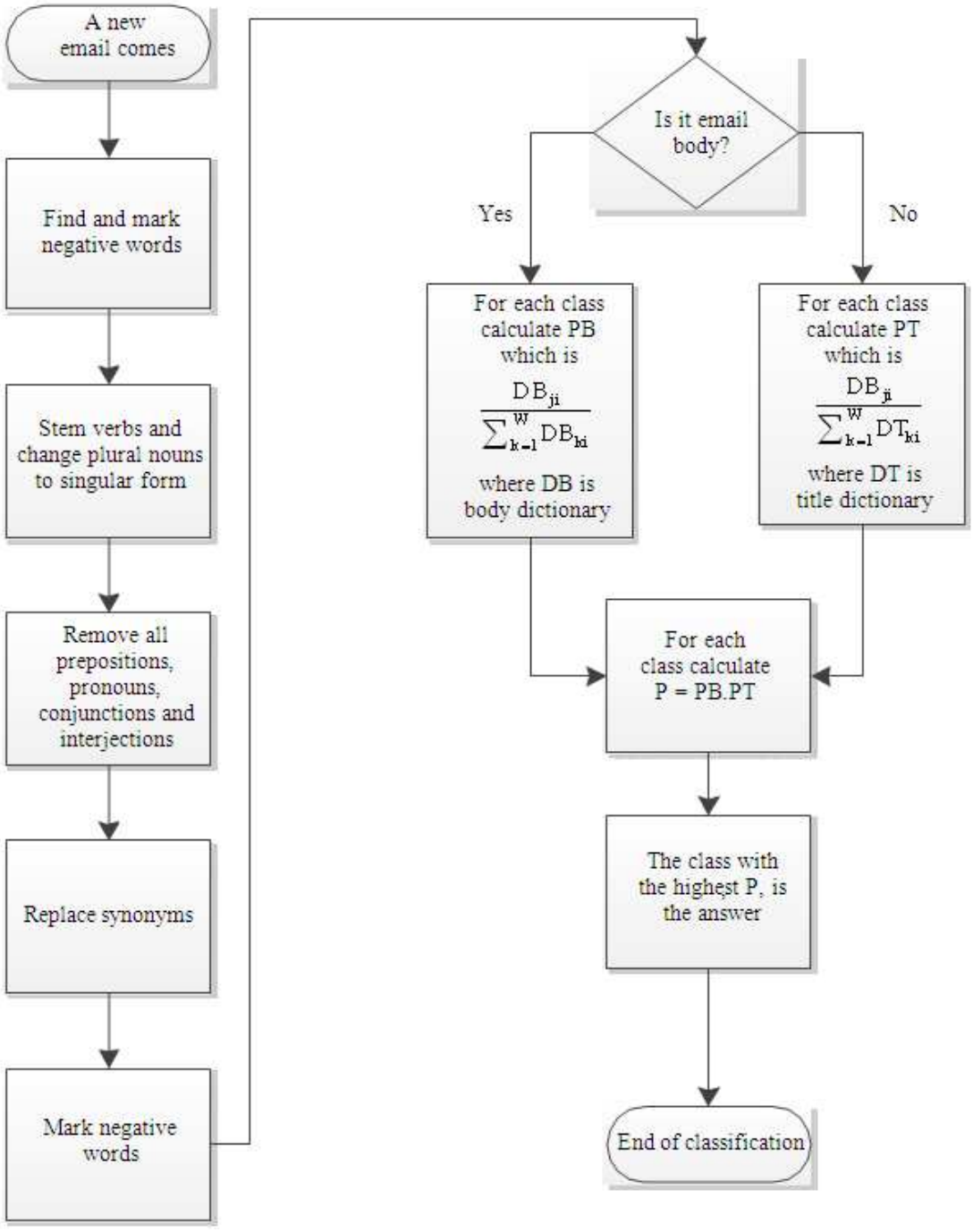

Fig. 2. Flowchart for email classification

Email will be transformed to following format after being processed using abovementioned process.

Title: New Order

like order Cloudy Chance Meatball 2. want Blu-ray disk receive friend place.

Best Regard

Title: Product not Receive

Order Cloudy Chance Meatball 2 day ago, not receive. track order?

Best Regard

vi. Find the words that are preceded by the marked negatives and change them to negative form (one-word negative which is included in dictionary). Then remove the corresponding negative words.

For example in the second email, 'receive' will be changed to 'not-receive' and the marked (highlighted) 'not' will be removed

The following steps will be done only when we are making the dictionary:

vii. If the word is in the dictionary go to step xi.
viii. Otherwise, create a new row in the dictionary associating with the new word
ix. Initialize each of the fields (one for every email class) with 1 (Laplacian smoothing parameter)
x. Go to step xii
xi. Find the row which is associated with the word
xii. Find the class of the email from which is the word
xiii. Increment the value of the corresponding cell.

The following steps will be substituted in the process of classifying emails:

xvii. For each class, calculate PB = for body and PT = for title. DB is the body dictionary and DT is the title dictionary
viii Calculate P=PB.PT for each class
ix The class with the highest P is the answer

Abovementioned steps are executed in the algorithm to classify emails using highest probability to assign an email to a specific class.

Following the same approach, methodology for automated email reply is developed and details are presented in the next subsection.

Algorithm for Automated Email Reply

Like classifying part, NLP is used on the email before processing it in order to have a better understanding of the text. The POS tagger is used to help us find the correct synonym of the words. For example if a word is processed which is both a verb and a noun, the POS data helps us find the correct synonym for the word. The other advantage of POS is that it can be used as a parameter to calculate probabilities more efficiently. For example if a word extracted is a noun for a certain class, we have the possibility to decrease probabilities for non-noun words when extracting the information from that class to generate an email.

Before running the algorithm we need to have a synonyms data structure to be used on all emails. Its job is to replace all synonyms to one specific word always. For example quantitative adjectives like awesome, prefect, superb, will be all changed to good! This would help us making a better dictionary. A process flowchart for automatic email reply is shown in Fig. 3. It can be seen from the flowchart that that it mainly depends on probabilities of occurrence of words in the nearest proximity of decision variable. This process is explained using following example.

(1) An Example

Let's say we want to extract product name from this email:

Hi,

I want to order Awesome Programming Languages. Please send me more details.

Best Regards

The algorithm will process this email to generate a reply in the following manner.

i. Tokenize the email body using a tokenizer able to distinguish compound nouns (for example open NLP). Mark the compound nouns as one-word tokens, so that they would be considered as single words on the rest of algorithm. For example, there would be one dictionary record for each compound noun
ii. In our example "Programming Languages" is marked as compound noun.
iii. Run a POS tagger (like Stanford POST) on email body and find and store part of speech for each word
iv. Make a copy of the original email for future use, because we will change the text on next steps. We need the copy to retrieve the original text when filling the template
v. Stem all verbs
vi. Replace plural forms of words by singular forms.

Our example would be:

Hi,

I want to order Awesome Programming Language. Please send me more detail.

Best Regard

vi. By having part of speech for each word found on step 2, we can use our synonyms data structure to change all synonyms to one word. Do it for all words which have a synonym

Our example would become:

Hi,

I want to buy Good Programming Language. Please send me more detail.

Best Regard

vii. The information needed from the email will be called decision variable. Using the dictionaries for the words before and after our decision variable and the probability functions, we calculate probability of each word to be the answer. Let's denote P_B the *i.* probability for a word to appear before decision variable, P_A the probability for a word to appear after decision variable and P_D the probability of a word to be our decision variable. The probability functions should be implemented based on the information we want to extract. The dictionaries for the words before and after the decision variable have been built for each information type using training data set. We calculate $P_B(w_{t-1})P_D(w_t)P_A(w_{t+1})$ for each word w_t and assign it to P_t.

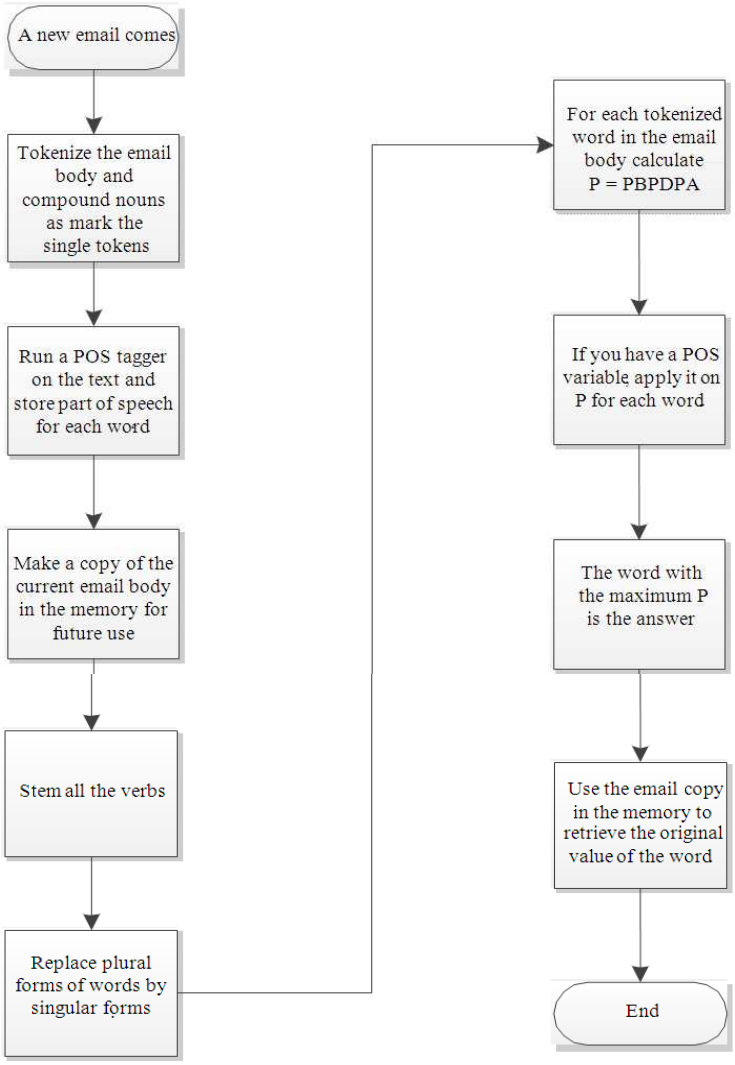

Fig. 3. Flowchart for automatic email reply

viii. In this step, we can apply POS variable effect to the calculated P_t for each word. For example when we want to extract product name, we give much more priority to noun phrases rather than verbs. Let's say we divide P_t by 2 for non-nouns in this case.
ix. The w_t corresponding to the maximum P_t will be our answer.
x. After finding the best w_t, we use the copy on step 3 and replace all the words to their original form in w_t. We would find that "Good Programming Language" is the product name. Then we use our copy to change it back to "Awesome Programming Languages" and fill our template using the latter value.

Results

Precision and Recall percentages are caluclated as evaluate the performance and the quality of our proposed algorithm. Based on four cases: True Positive case is where an email reply is generated positive and it is tested positive. False Positive case is in which an email is generated positive but when tested it turns out to be negative. True Negative case caters for discarded email reply which in testing proved noncompliant also and False Negative.case deals with discarded email replies which when tested, turned out to be poistive with corrrect operation. 400, 200 and 400 emails were selected related to Meeting and Scheduling, Discussion and Comments and Technical Support and Requests respectively. Table 5 shows the results for such an evaluation.

After generating replies with positive/negative predictions for all the emails under test and getting a poitive or negative response from the receipent, Precision and recall are calculated.

Table 5. Example of Precision and Recall for generated email replies in different domains

Domains (English emails)	Number of emails	Precision (%)	Recall (%)
Meeting and schedule	400	80	69
Discussion and comments	200	72	60
Technical support and requests	400	76	65

For meeting and schedule precision is 80% and recall is 69%. 69% of positive email replies were caught positive with a correct positive email reply prediction of 80%. For other two domains also the accuracy measures are reasonably good at more than 60%. The emails in Meeting and Schedule domain, in general, are more formal in nature. This minimizes the ambiguities related to natural language. Discussion and Comments category is prone variety of words and contexts making it somewhat less accurate as compared to the other two groups of emails whose emais are more quantitaive and easy to analyze. This is why we are getting best results for domain of Meeting and Schedule.

Conclusion

When dealing with excessive flux of emails, classifying emails into groups and generating automatic email replies can be time-consuming and vulnerable to mistakes. A novel algorithm proposed in this research solves this problem by using NLP based processing to classify relevant text attributes for email class and make template decisions by utilizing most relevant features and discarding overhead attributes. This NLP algorithm complements probabilistic routines to select email messages that require immediate reply and permit users to manage email messages using NLP based classification. Most challenging assignment in an accurate classification system is identifying the most precise way extract attributes from an email i.e., identifying minimum attributes that are sufficient to describe an email context and requirements. Initial optimized classification can greatly extend ability of email response architecture by removing unnecessary redundancy. Other email classification tasks typically use bag-of-words representations of message bodies and subject lines, with quoted text from previous messages removed, combined with features that indicate trivial information such as message sender and recipients. Similarly, rule based approaches binary decisions for classification can induce a serious lapse in classification process when categorizing emails.

To resolve these problems, a NLP based email classification algorithm has been developed and presented in research which also have the provision of generating correct email replies using probabilistic machine learning template filling. This approach has the ability to extract relevant attributes, remove redundant attributes and used probabilistic measures to ascertain class of email. Extracted attributes are used to build a dictionary which is available in parallel for email template selection and attribute filling providing standalone architecture for Email classification and automatic email reply.

Regarding future work in NLP processing for Email classification, efficacy of NLP techniques is being improved with the passage of time. Improvised algorithms are being continuously published showing extensive utilization of machine learning and statistics approaches. In authors opinion, current NLP processes and practices provides us with plenty of opportunities for future classification research, as ever increasing email data requires investigation into more innovative techniques and models. Still, in current circumstances, persistent reevaluation of NLP methodology and Text Classification will allow for improved classification and generation of an accurate email response in minimum amount of time.

Funding Information

The authors have no support or funding to report.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Al-Alwani, A., 2014. A novel email response algorithm for email management systems. J. Comput. Sci., 10: 689-696. DOI: 10.3844/jcssp.2014.689.696

Arevian, G., 2007. Recurrent neural networks for robust real-world text classification. Proceedings of the International Conference on Web Intelligence, Nov. 2-5, IEEE Xplore Press, Fremont, CA, pp: 326-329. DOI: 10.1109/WI.2007.126

Ayodele, T., R. Khusainov and D. Ndzi, 2007. Email classification and summarization: A machine learning approach. Proceedings of the IET Conference on, Wireless, Mobile and Sensor Networks, Dec. 12-14, IEEE Xplore Press, Shanghai, China, pp: 805-808.

Beseiso, M., A.R. Ahmad and R. Ismail, 2012. A new architecture for email knowledge extraction. Int. J. Web Semantic Technol., 3: 1-10. DOI: 10.5121/ijwest.2012.3301

Carmona-Cejudo, J.M., G. Castillo, M. Baena-Garcia and R. Morales-Bueno, 2011. A comparative study on feature selection and adaptive strategies for email foldering. Proceedings of the 11th International Conference on Intelligent Systems Design and Applications, Nov. 22-24, IEEE Xplore Press, Cordoba, pp: 1294-1299. DOI: 10.1109/ISDA.2011.6121838

Chan, J., I. Koprinska and J. Poon, 2004. Co-training with a single natural feature set applied to email classification. Proceedings of the International Conference on Web Intelligence, Sept. 20-24, IEEE Xplore Press, pp: 586-589. DOI: 10.1109/WI.2004.10135

Dredze, M.H., 2009. Intelligent Email: Aiding Users with Ai. 1st Edn., University of Pennsylvania, ISBN-10: 1109227795, pp: 227.

Fong, S., D. Roussinov and D.B. Skillicorn, 2008. Detecting word substitutions in text. Knowledge Data Eng. IEEE Trans., 20: 1067-1076. DOI: 10.1109/TKDE.2008.94

Haiyi, Z. and D. Li, 2007. Naïve bayes text classifier. Proceedings of the International Conference on Granular Computing, Nov. 2-4, pp: 708-708.

Liu, W.Y., L. Wang and T. Wang, 2010. Online supervised learning from multi-field documents for email spam filtering. Proceedings of the International Conference on Machine Learning and Cybernetics, Jul. 11-14, IEEE Xplore Press, Qingdao, pp: 3335-3340. DOI: 10.1109/ICMLC.2010.5580676

Provost, J., 1999. Naïve-Bayes vs. Rule-learning in classification of email. The University of Texas at Austin.

Rosenfeld, R., 2000. Two decades of statistical language modeling: Where do we go from here. Proc. IEEE, 88: 1270-1278. DOI: 10.1109/5.880083

Wang, R., A.M. Youssef and A.K. Elhakeem, 2006. On some feature selection strategies for spam filter design. Proceedings of the Canadian Conference on Electrical and Computer Engineering, (ECE' 06), IEEE Xplore Press, Ottawa, Ont., pp: 2186-2189. DOI: 10.1109/CCECE.2006.277770

Weijie, L., H. Chen, W. Cao and X. Zhou, 2012. An idea of setting weighting functions for feature selection. Proceedings of the 2nd International Conference on Cloud Computing and Intelligent Systems, (CIS' 12), IEEE Xplore Press, Hangzhou, pp: 690-695. DOI: 10.1109/CCIS.2012.6664263

Yang, W. and L. Kwok, 2012. Improving the automatic email responding system for computer manufacturers via machine learning. Proceedings of the International Conference on Information Management, Innovation Management and Industrial Engineering, Oct. 20-21, IEEE Xplore Press, Sanya, pp: 487-491. DOI: 10.1109/ICIII.2012.6340024