Email content based addressee finder using fuzzy logic

Peter Heemskerk, Stefan Schenk, Jim Kamans
11988797, 11881798, 10302905

***Abstract*—Large organizations have problems with their customer service, since due to their complexity, they cannot answer messages from external parties in a quick manner. This project aims to demonstrate that Fuzzy Logic can be used to solve this problem, by determining the correct addressee within the organization in an automatic way.**

In this project it has been demonstrated that Fuzzy Logic can successfully be implemented. (to be extended).

***Index Terms*—Fuzzy Logic System (FLS)**

I. Introduction

MANY large organizations suffer from their own complexity. If an external party seeks contact with a specific person in an organization, this works fine, but if a party seeks contact about a subject (without knowing whom to talk to), it usually takes more time before the party gets a good answer. At this moment emailing is the main way of communication to businesses (120 bilion email a year), which include a large portion of spam mail [**?**].

Who does not have experience with this complexity or large organizations? Suppose you have a question on [example], we all have been facing either that we are sent from one person to another department or perhaps worse that we do not get an answer at all, since the message is lost somewhere. Organizations aim to get better on customer service, but tools that support this are building up.

[bovenstaande mag nog meer scientific en met meer referenties, Peter]

A. Problem

This project aims to solve this issue of customer service in a complex organization. We present software based on fuzzy logic which aims to bring a message of an external party to the correct internal addressee, being a department or a person, purely based on the content of the message. Furthermore based on the same content a priority score is provided.

B. Objectives

The experiment has the goal to determine two outputs based on email content. First the correct addressee needs to be determined. Secondly a priority score. Based on the cleaned word lists a feature vector will be determined for each email. For each of the desired outputs we designed a types of feature. For addressee determination content specific features are designed. For priority determination two general features are designed. For each email a feature vector is determined consisting of content and general features. These feature are used as inputs in the fuzzy logic system to determine the two outputs: addressee and priority.

This project aims to prove that Fuzzy Logic works for this problem. Why fuzzy logic ? Firstly, fuzzy logic deals well with incomplete or difficult to interpret data. Since there is a variety of email messages, short and long, specific and vague fuzzy logic better deals with these different sources. Secondly, fuzzy logic uses linguistic terms. With this we can include expert knowledge into the system which is relatively easy to interpret. This proves to be important for determining priority score from content, since no labeled data sets are available.

For our first goal addressee determination, our results can be compared to a given labeled data-set. Our second goal, using the same algorithms for determining a priority setting this experiment should be regarded as a proof-of-concept.

II. Literature reviews

Fuzzy Logic has been used earlier for email classification. Ferolin [**?**] has used fuzzy logic to implement a anti-phishing tool using content- and non-content eamil parameters. A RIPPER Classification Algorithm is used to learn relations of different phishing features, which translate into Fuzzy Logic rules. Santhi et al [**?**] determined the degree of dangerousness of spam email with a different method. A Fuzzy Logic system is used to categorize words that are spam in the degree to which these words are considered dangerous. The words are labeled to the names of five linguistic variables before they are fed to the particular input that corresponds with the name. Ferolin [**?**] introduced a fuzzy logic based ranking function for efficient Information Retrieval. A fuzzy approach was used to rank words based on term-weighting schemes such as term frequency, inverse document frequency and normalization. The term frequency and inverse document frequency and normalization of the query and document are fed to their Fuzzy Logic Controller, whose outputs are fed to the main Fuzzy Logic Controller, which outputs a relevance score.

III. Approach

For email classification the following approach is followed. After data-preprocessing the email is classified. Then fuzzy logic is applied to determine addressee and priority.

A. Data preprocessing

[Dit stuk herschrijven - Jim ?? references in a footnote ref]We used the Enron Email Dataset, May 7, 2015 version, available at https://www.cs.cmu.edu/ enron/. This dataset contains around 500,000 emails. Each email looks roughly like this:

[dit soort dingen in een figuur]

```
Message-ID: <10929741.1075855668115.JavaMail.
    evans@thyme>
Date: Mon, 25 Sep 2000 07:04:00 -0700 (PDT)
From: phillip.allen@enron.com
To: christopher.calger@enron.com
Subject:
Mime-Version: 1.0
Content-Type: text/plain; charset=us-
    asciihttps://www.sharelatex.com/project/5
    a17f18fb1d4a54ac4184a1d
Content-Transfer-Encoding: 7bit
X-From: Phillip K Allen
X-To: Christopher F Calger
X-cc:
X-bcc:
X-Folder: \Phillip_Allen_Dec2000\Notes Folders
    \All documents
X-Origin: Allen-P
X-FileName: pallen.nsf

Chris,

 What is the latest with PG&E?  We have been
     having good discussions
regarding EOL.
 Call me when you can. X37041

Phillip
```

[Jim herschrijven tot hier]

The data needs to be cleaned filtered and ranked.

1) Cleaning
   As an email body is read from the file system as plain text, individual terms are stored as individual values (’tokenized’). After that, capital characters are converted to lower case, punctuation and special characters are removed, stop-words are removed and the words are reduced to their base root form (’stemmed’).
2) Filtering
   The remaining words may be meaningful now, but most of them may not be relevant. So the next operation will perform an intersection between the words and a list of predefined relevant words. All the words that reside in both lists will remain, but unlike sets, the list contains duplicates. As the last filtering step, the words are counted, and a corpus is created.
3) Ranking
   Other predefined sets of words within the set of relevant words share the same characteristics. For example a set $T \subseteq R$ exists where T is the set of technical words, and R is the set of relevant words from before. Feature list of technical words: $T = [t_1, t_2, \ldots, t_n]$. Other feature lists contain other themed words: U, V, W.

   For every word in the email that is present in T, a score is calculated that takes the count of that word into account in relation to the total number of relevant words in the email. This calculation is made for all feature lists (T, U, V, W), for every word in the email.
   [Bovenstaande Nakijken Stefan]

B. Classifying

The cleaned words from an email are classified to a ’feature” by using word matching [better word ?] using predefined feature word-lists. The words are compared to a predefined set of words per feature, and a score is determined on each of the features. For this experiment two types of features are chosen.

A content type of feature determines the subject of the email. We worked with features: personal, financial, space, tax and ...[to confirm after experiment]. We have used our labeled training set to determine these feature lists.

The second type are two general features: ”agitation”, which determines the (negative) emotion in the mail. And a feature ”action” for addressing the more neutrally emergency of the mail. Since no labeled training-set was available for determining the feature word lists, we based our word lists based on a training set of 40 emails which were scored on a scale from 1 to 100 on the variables ”agitation” and ”action” by an experienced email reader [or more ?]

C. Fuzzy Logic

The following parameters have been set in our Fuzzy Logic system: [–¿ figure with input params, ruls and output parms]

D. Implementation

For cooperation purposes we used Github [ref] (for source control) and Trello [ref] (as scrum projetcmanagement tool)

We used Python3 as programming language and Jupyter Notebook as development environment. The code is enlisted in Attachment [ref]. For data-preprocessing Pythons NLTK module [ref] is used. For classification a new algorithm has been developed. The fuzzy logic system itself is based on the Fuzzy Logic LAB [ref], amended for using more than one output, a centroid defuzzifier and some more flexibility and error handling in management of fuzzy logic rules.

IV. Experiment

Our goal is two fold. Firstly, the use of fuzzy logic is tested to send an email to the right department. Secondly, with fuzzy logic we aim to solve how priority can be deducted from e-mail contents. Experimental approach for both is different.

A. The correct departments - content

We can make this a supervised learning experiment, since we have for each email department labels available. We divide our data-set in a training set and validation set with factor 0.6. We train the system with the training set and validate on the validation set. Our feature lists (lists with trigger words for each content feature) is trained. For the Fuzzy Logic rule set we used our own expert knowledge.

B. The correct priority - the notion of agitaion and action

To translate a received email based on its content to priority is innovative and our data-set does not give labels for the outcome. The following approach is taken.

For training the feature lists for input variables ”agitation” and ”action” 40 emails are randomly selected from our training-set and we have an independent experienced email reader score for these two input variables on a scale from 1 to 100. Based on these inputs, the feature lists are create resulting in the trigger words for ”agitation” and for ”action’.

For validation we randomly selected 40 emails from our validation set and had an independent experienced email reader score for the output ”priority” on a scale from 1 to 1000. These values are used for validation.

C. Results

First results are promising but much work is left to do.

D. Discussion

Due to time-constraints the project has limitations:

- Only written emails are used as input. At this moment emailing is the main way of communication to businesses (120 bilion email a year [**?**], and far more used than other ways like social media or message apps. This will change, so in a later phase we envision this to be expanded to messages in any format, like messaging via WhatsApp or LinkedIn.
- The “word-to-feature translator” is implemented with limited scope, currently a number of only (xx?) English words. We did not extend to much on this, since we would like to focus on the Fuzzy Logic part of the software.
- We used a theoretical organization for the proposed structure of department, with first testing in the real world, we should test and amend this structure.

This work is done as part of the autumn 2017 bachelor course Fundamentals of Fuzzy Logic by A. Bilgin (and M. Hol and V. Dankers) within the study Artificial Intelligence at University of Amsterdam.

References

[1] The Radicati Group, inc.(2015), *Email Statistics Report, 2015-2019,* .

[2] G.Santhi, S. Maria Wenish, Dr. P. Sengutuvan (2013), *A Content Based Classification of Spam Mails with Fuzzy Word Ranking,* Department of Information Science and Technology, Issue 3.

[3] Rosana J. Ferolin, (2010 - approx.) *A Proactive Anti-Phishing Tool Using Fuzzy Logic and RIPPER Data Mining Classification Algorithm,* Department of Computer Engineering University of San Carlos.

[4] Rosana J. Ferolin, (2014) *A new fuzzy logic based ranking function for efficient Information Retrieval system,* Department of Electrical Engineering Dayalbagh Educational Institute.