

Using Fuzzy Logic for extracting department from email content

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

Abstract—Large organizations often struggle to deliver emails or form emails to their corresponding departments if they are not directly addressed to a department or employee. In this project a Fuzzy Logic approach is proposed for automatically determining the correct department.

A basic version of the approach is implemented with fuzzy logic. The set-up makes it possible to add more flexibility to determine departments not only on content based features but also based on more general features of the email content.

I. INTRODUCTION

MANY large organizations suffer from their own complexity. If an external party seeks contact with a specific person in an organization, this usually 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, simply because it is not clear what the right department is which should reply on such a message. At this moment emailing is the main way of communication to businesses, 120 billion emails a year are sent which figure includes a large portion of spam mail [1].

This project aims to solve this issue of low 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 department purely based on the content of the message.

This project aims to demonstrate that Fuzzy Logic has advantages toward other methods. Firstly, fuzzy logic deals well with incomplete 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, allowing to include expert knowledge into the system which is relatively easy to interpret.

The goal is determining the correct department from email content. Based on the cleaned word lists a feature vector will be determined for each email. For department determination content specific features are used. These features are used as inputs in the fuzzy logic system to finally determine as output the correct department. Our results will be compared to the given labels of the data-set.

II. LITERATURE REVIEWS

Fuzzy Logic has been used earlier for email classification. Ferolin [3] has used fuzzy logic to implement a anti-phishing tool using content- and non-content email parameters. A

RIPPER Classification Algorithm is used to learn relations of different phishing features, which translate into Fuzzy Logic rules. Santhi et al [2] 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 five linguistic variables which are input for the fuzzy logic algorithm. Ferolin [4] 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. None of these has utilized fuzzy logic for determining departments.

Douglass [4] developed an email priority setting learning system for G-mail based on social, content, search label features. These features are use in a statistical model which is parametrized for each user (recipient) separately. This research does not present a solution for situations in larger organizations where there is no information about the individual recipient.

III. APPROACH

For our department determination we follow the procedures proposed by Ferolin for Information Retrieval. The following approach is followed. After data-pre-processing the email words are ranked, resulting in a feature score per email. Words are ranked using a pre-compiled list of words. Then fuzzy logic is applied to classify the email to the correct department.

A. Data

A private dataset from “Gemeente Amsterdam” is used, containing 3589 emails with complaints from citizens. This dataset is a csv-file where each email has a correct department label, a description of the complaint, a description of contact with an employee and a proposed solution. An example:

```
Parkeren;Vrijdag voor een ...;Ja, contact  
gezocht ...;De 23 euro ...
```

B. Feature word-list preparation

In order to create a feature score for each email (refer to section Ranking), feature word-lists are necessary. 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: $T = [t_1, t_2, \dots, t_n]$. Other feature lists contain other themed words: U, V, W, \dots .

Feature word-lists are created using the term-frequency / inverse-document-frequency (tf-idf) method. [hier nog referentie toevoegen, Jim ?] From the corpus of training emails with the specific department label the following values are calculated for each term.

$$tdidf = tf * idf,$$

where $tf(t)$ equals the count of terms t , and idf give a ranking of the importance of the word in the whole corpus:

$$idf(t, D) = \log \frac{N(D)}{n_t(D)},$$

where $N(D)$ is the total number of documents and $n_t(D)$ equals the number of documents containing the specific term. Selecting those terms with a $tdidf > threshold$, the feature-word lists are created, with signal words for the specific department.

C. Data preprocessing

The data needs to be cleaned and filtered.

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 next operation will perform an intersection between the words and a list of predefined relevant words. Words that are not contained in the word list are removed. As the last filtering step, the words are counted, and a corpus is created.

D. Ranking

For this experiment the content type of feature determines the subject of the email.

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.

So for each email a feature vector is determined, containing the score between 0 and 1 against the features, as defined by the feature word-lists.

E. Classification with Fuzzy Logic

For determining the output variable department four content input variables are defined, corresponding with the content features. Refer to table III-E. Since these are mostly chosen as

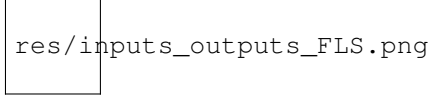


Fig. 1. Input and Output variables in Fuzzy Logic System

a default, also now for each of the input variables 3 triangular membership functions (ms) are chosen to represent a low, medium or high value. For the output, a variable is created for each of the four departments, each variable containing 3 triangular membership functions low, medium, high.

We have set a rule base for determining departments based on the content input variables and a rule base for determining priority based on the general input variables. An overview of Rules you can find in attachment. [Jim to create]

input variable	basisinformatie	3 ms on low, med, hi
	belastingen, werk en inkomen	3 ms on low, med, hi
	openbare ruimte	3 ms on low, med, hi
	parkeren	3 ms on low, med, hi
output variable	basisinformatie	3 ms on low, med, hi
	belastingen, werk en inkomen	3 ms on low, med, hi
	openbare ruimte	3 ms on low, med, hi
	parkeren	3 ms on low, med, hi

Input and Output variables in Fuzzy Logic System

figure with rules - JIM

F. Training and validation

For training and validation the data set is divided in a training set and validation set with a factor 0.7.

The training set of emails is used for the creation of the feature word-lists using tf/idf. The setting of the Fuzzy Logic rule base is done based on the expert vision of one of the team-members.

The validation set including the department label is used for validation.

G. Implementation

For cooperation purposes we used Github ¹ (for source control) and Trello ² (as scrum projectmanagement tool)

We used Python3 as programming language and Jupyter Notebook as development environment. The code is enlisted in Attachment [ref]. For data-preprocessing Python's NLTK module ³ is used. For classification a new algorithm has been developed. The fuzzy logic system itself is based on the Fuzzy Logic LAB ⁴, amended for using more than one output, a centroid defuzzifier and some more flexibility and error handling in management of fuzzy logic rules.

¹<https://github.com/Menziess/Fuzzy-Logic-Email-Classification>

²<https://trello.com/fuzzylogicemailclassification>

³<http://www.nltk.org/>

⁴https://blackboard.uva.nl/webapps/blackboard/execute/content/file?cmd=view&content_id=_6947429_1&course_id=_212301_1&framesetWrapped=true

IV. EXPERIMENT

A. Results

The automatic process of feature word-list creation, data-preprocessing, ranking and classification with fuzzy logic has been implemented. The process is tested and works.

Validating the calculated departments with known labels has resulted in a xx percent correctness. [Stefan / Jim]

B. Discussion

The project resulted in a baseline for the proposed approach. A feature vector is used which corresponds to the departments already, and we included a basic fuzzy logic implementation. Inclusion of a feature vector has an important advantage that also non-content features (like: "press sensitive") can be extracted which may result in determining a different department. The creation of a relevant feature vector based on a large email training set is work to do.

The fuzzy logic implementation has been basic. We included the feature scores of an e-mail as input, with basic triangular membership functions for low, medium and high. A basic rule set is included, with rules like 'if input feature "parkeren" is high, then output department "parkeren" is high. Work to do is to learn a rule based from a training data-set. For this the RIPPER algorithm, or others like Decision Tree or Neural Network may be considered. [6] [7]

[Opmerking van Muriel: vooral veel aandacht aan discussie, keuzes die gemaakt zijn geven. Dus deze paragraaf nu aandacht aan geven allemaal: - be critical about why our process does not work - could we decide whether the model or the data is the bottleneck ? - be critical about our fuzzy logic systems setting DUS JIM HIER ALS JE NOG WAT TOE TE VOEGEN HEBT. GRAAG]

V. ACKNOWLEDGMENT

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.
- [5] Douglas Aberdeen, (2010 approx.) *The Learning behind Gmail Priority Inbox*
- [6] J. Foernkranz, D. Gamberger, N. Lavrac (2012) *Foundations of Rule Learning, chapter 2, Rule Learning in a nutshell*.
- [7] M. Britsch, N. Gagunashvili, M. Schmelling (2008) *Application of the rule-growing algorithm RIPPER to particle physics analysis*