UNIVERSITY OF AMSTERDAM

DRAFT REPORT FUZZY LOGIC

November 25, 2017

Authors: Peter Heemskerk, Stefan Schenk, Jim Kamans
Studentnrs: . . . , 11881798, 10302905
Course: Fundamentals of Fuzzy Logic

Contents

1 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 is has been demonstrated that Fuzzy Logic can successfully be implemented. The team implemented software based on a Python implementation.

2 INTRODUCTION

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

2.1 Problem

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.

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

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 department or person fur further action, purely based on the content of the message.

Why fuzzy logic ?

- First, with fuzzy logic methods, we can better deal with the uncertainties from the real world.

- Second, fuzzy logic deals well with incomplete or difficult to interpret data.

- Finally, fuzzy logic uses linguistic terms. With this we can include human knowledge into the system which is relatively easy to interpret.

These fuzzy and linguistic features support dealing with unstructured messages.

2.2 Objectives

Via several steps our software will bring a unstructured message (e-mail) to the correct department. For achieving this goal several steps are taken:

1. from an email a list with relevant words is produced, irrelevant words are filtered out

2. the words are matched and scored (word count) against a feature list

3. with fuzzy logic these features are matched with a department.

4. so the end result is a department attached to an email

This project aims to prove that Fuzzy Logic works for this problem. Due to time-contraints the project has limitations:

- Only written e-mails are used as input. At this moment e-mailing is the main way of communication to businesses (120billion/day) **??**, and far more used that other ways like social media or message apps. This will change , so in later phase we envisage this to be expanded to messages in any other 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

3 LITERATURE REVIEWS

Provide a coherent and synthesized summary of relevant work in the literature, present the supporting/related evidence for your project.

3.1 A Proactive Anti-Phishing Tool Using Fuzzy Logic and RIPPER Data Mining Classification Algorithm

A Proactive Anti-Phishing Tool Using Fuzzy Logic and RIPPER Data Mining Classification Algorithm [2] is a paper that explains how phishing emails are detected using content- and non-content based approaches in the form of a combination of the RIPPER Classification Algorithm and a Fuzzy Logic system. The RIPPER Classifier is used to learn relations of the different phishing features, resulting into rules that are used by the Fuzzy Logic System.

While designing our system, the implementation of the content-based classification approach gives us a starting point for designing membership functions, linguistic variables and rules. However, the characteristics used to determine the crisp inputs aren't mentioned.

3.2 A Content Based Classification of Spam Mails with Fuzzy Word Ranking

A Content Based Classification of Spam Mails with Fuzzy Word Ranking [1] introduces a different method by which 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. After that, the Fuzzy Logic System outputs a degree of

dangerousness.

This work seemed relevant before we finished our detailed design, but as our team gained more knowledge, it seemed that this approach misses all the value a Fuzzy Logic System is supposed to add.

3.3   A new fuzzy logic based ranking function for efficient Information Retrieval system

A new fuzzy logic based ranking function for efficient Information Retrieval system [3] explains how "conventional ranking functions fail to capture inherent features of documents and queries due to subjectivity involved in natural language text", and offers a fuzzy approach to rank words based on term-weighting schema's such as term frequency, inverse document frequency and normalization. The tf/idf and normalization of the query and document are fed to their FLC (Fuzzy Logic Controller), whose outputs are fed to the main FLC, which in turn outputs a relevance score.

As we are using a content based approach in classifying emails, we need to determine whether words in the emails are related to words in a pre-assambled list.

4   APPROACH

In this section . . .

4.1   Data

4.2   Design

The solution consists of four main operations: Cleaning, Filterig, Ranking, and Classifying. These three terms are descriptive, but contain more additional steps that are somewhat corrolated to the meaning of the terms.

All actions in each step are represented by the colored arrows.

1. Cleaning

   An email is read from the file system as plain text, containing mostly irrelevant information for classification. The text is tokenized, meaning that individual terms are stored as individual values. After that, capital characters are converted to lower case, punctuation and special characters are removed, resulting in an array of words.

   After that, stopwords are removed. Some examples of stopwords are: "a", "and", "but", "how", "or", and "what". And finally the words are stemmed, reducing words to their base root form. For example: "waited" will become "wait".

2. Filtering

   The remaining words may be meaningfull 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.
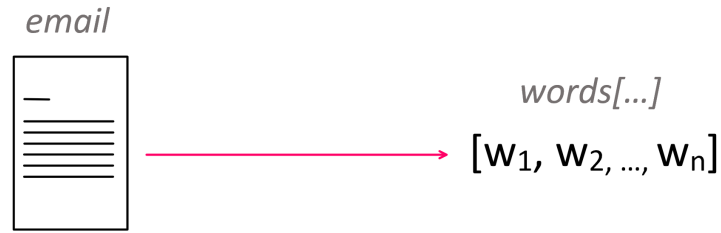
email

words[...]

$$[w_1, w_2, ..., w_n]$$

Figure 1: Cleaning one email resulting into an array of words

words[...]   *relevant_words[...]*

$$[w_1, w_2, ..., w_n] \quad \cap \quad [rw_1, rw_2, ..., rw_n]$$

$$[[w_1, w_{1count}],$$
$$[w_2, w_{2count}], \quad \textit{words[[...]]}$$
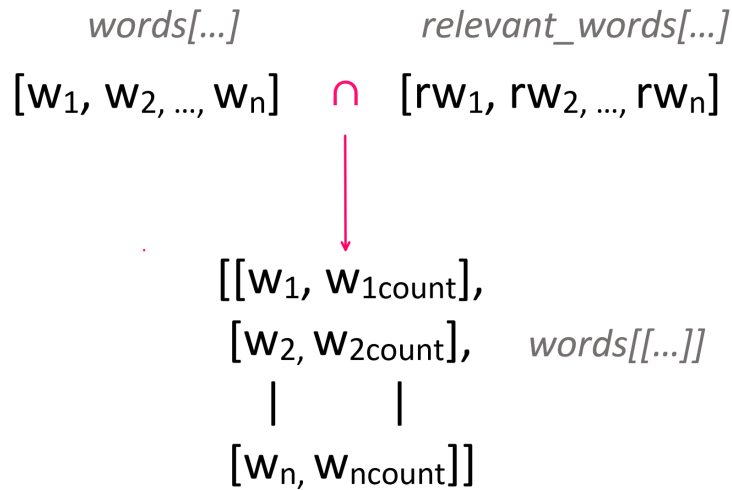$$|\qquad|$$
$$[w_n, w_{ncount}]]$$

Figure 2: Taking the intersection of the words with a relevant list of words

3. Ranking

Sets of words within the set of relevant words share the same characteristics. For example a set $T \subseteq R$ exists where R is the set of technical words, and R is the set of relevant words from before.

Feature list of technical words: $T = [t_1, t_2, \ldots, t_n]$

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, for every word in the email.

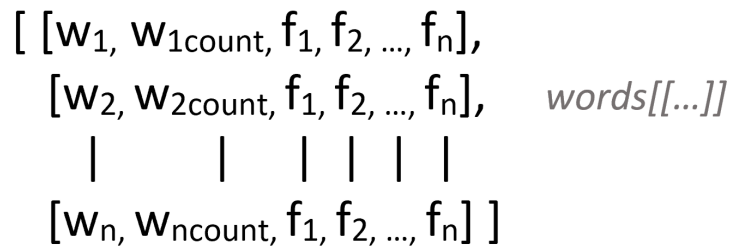Finally the scores are aggregated, resulting in the inputs for the Fuzzy Logic System.

$$[[w_1, w_{1count}, f_1, f_2, ..., f_n],$$
$$[w_2, w_{2count}, f_1, f_2, ..., f_n], \quad \textit{words[[...]]}$$
$$|\qquad|\quad|\ |\ |\ |$$
$$[w_n, w_{ncount}, f_1, f_2, ..., f_n]]$$

Figure 3: The scores for each words are added as features

4. Classifying

When we expand on the example of the technical word list, we may have a couple of other lists that are subsets of the relevant word list:

4.3 Implementation

REFERENCES

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

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

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