TUDaThesis – Abschlussarbeiten im CD der TU Darmstadt

LaTeX using TU Darmstadt's Corporate Design
Bachelorarbeit im Studienbereich Computational Engineering von Marei Peischl
Tag der Einreichung: 2. Februar 2023

1. Gutachten: Gutachter 1
2. Gutachten: Gutachter 2
3. Gutachten: noch einer
4. Gutachten: falls das immernoch nicht reicht
Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Studienbereich
Computational Engineering
Institut
Arbeitsgruppe

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 und § 23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Marei Peischl, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß § 23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 2. Februar 2023

M. Peischl

Inhaltsverzeichnis

1 Introduction

This work will be dedicated to the problem, which is very common to the business. Nowadays almost every (if not every) company that's coming from such fields like B2B (business-to-business), B2C (business-to-consumer), E-commerce, service-based, healthcare, NPOs (Non-Profit Organizations), Financial institutions, Telecommunications, Retail, Real estate or Transportation [FIND LITERATURE] uses one or several CRM Systems (Customer Relationship Management System) to manage internal and external organizational relationships, which also includes storing (and protecting) all sorts of the data, including the data about its products, activities, customers and shareholders etc. This data then could be utilized to provide a broad source of applications within of the company. One of such possible applications is building prediction or suggestion models to better understand the behaviour of the customers, product on the market or pricing development.

While to solve those problems there exists a broad set of tools, they would be useless unless there's a valid data to get insights from. For it one would first have to collect the data over some span of time and to prepare it to use. And this very point of collecting and processing data is the standing ground for the whole application to work accordingly, since it will have to rely on this data through its whole life-time.

Let's imagine a company hosting various events (also online-events) during some period of time, while also gathering the data about the participants in form of some registration form, that also includes the name of the participant. The use case that will appear at some point using this data would of course include the validation of such data.

This brings us to the point, where there's only one possible way to do it properly - match this data from registration sheets to the data of the CRM System. While we can be (almost) sure about validity of the CRM data, there's no way to be sure, that data from registration sheets is of the needed quality. Unconscious mistakes such as misspelling or getting typing errors in the own name are often a very common in the data like this. But also conscious misspelling such as writing the name in a different way or even using a nick-name instead of real name might coexist it the same data. And lastly, the events might also include

participants from outside of the company, which makes the data not only chaotic, but also sparse in the sense of possible matches.

Of course this is only one example of this type of problems, there're many more, just name a few: matching product names made by company using CRM Data and product names, which have been sold by a retailer, written by hand by the sales workers; matching therapeutic area of a doctor from the CRM system to a therapeutic area collected by sales employees etc.

All this cases are similar in few ways:

- it's a matching problem including two (often very big) data sets, which leads to performance issues;
- the data consists of character strings without any context;
- the data contains proper names, which don't really have or should have semantic sense;
- the collected data includes mistakes and mismatches with regard to a valid data.

We will call this set of problems **fuzzy name matching**.

Using this knowledge we now can define our goal:

1. define mapping, that matches a fuzzy name to its valid analogue,
2. create the architecture of the solution,
3. perform the matching task in polynomial time (???)

Here comes a section about following chapters.

2 Definitions

All the needed definitions should be stored here.

2.1 Theory and previous work on the matter

Include Theorems and Conclusions.

2.2 Mapping possibilities

2.2.1 Using training set

2.2.2 Using bounded training set

2.2.3 No training set

Architecture

Implementation

Performance

2.3 Overall performance + acceleration

2.3.1 Implementation

2.3.2 Performance

2.4 Use case

2.4.1 Implementation

2.4.2 Evaluation

2.5 Conclusion

2.6 Literature