Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Informatik
FH Zentralschweiz

Certificate of Advance Studies in Big Data Analytics

Adverse Media Search

Tracking Politically Exposed Persons in the Banking Industry

Project

Submitted on the 20th of March 2019 from:

Block-Kaefer Barbara, Finnova AG

Alvarado Liliana, IT Evotion GmbH

Worthington William, Sequoia Intelligence

Table of Contents

Table of Figures

Management Summary

Finnova develops and operates software for Swiss cantonal banks. The cantonal banks do not have an automated solution for the early detection of cases in which one of their clients is involved in corruption, especially if PEPs [1] (Political Exposed Person) are exposed to bribery. Banks receive random information about their clients' corruption cases. In the future, the banks want to receive this information as soon as it is known in the news media. This allows the banks to react more quickly, to check the customer and to quickly take action such as reporting or closing the account.

With this project a Proof of Concept (POC) is shown how news media articles can be searched and stored.

The approach was to be able to detect negative news with a sentiment analysis. One result of the POC, however, is that news articles have mostly neutral sentiment and the polarity does not change much from article to article. Thus, sentiment analysis is not suitable for detecting negative messages to customers. Therefore, a keyword list was used to detect the wanted news such as corruption, bribery, money laundering etc.

Since a lot of articles are published about the PEP, only articles that indicate a case of corruption (keyword) are searched for. In addition, the news will often be published multiple times, these duplicates, i.e. more than 80% of the article is the same, are excluded. Furthermore, articles that relate to a namesake are recognized based on specific attributes of the namesakes and excluded from further processing.

Often, several PEPs are mentioned in an article, identifying the PEP to which the keyword refers is a challenge. For this, the distance between the named PEPS and the key word is calculated and the article incl. keyword is assigned to the PEP which is closest to the keyword.
The correct recognition of negations, like the politician is against corruption is not recognized in the POC. Therefore, further analysis and development is needed. However, the use of a decision tree or a neural network is a recommended solution.

Forward

This topic is a customer request of the Finnova customers, and thus there is a concrete application possibility. Interesting is the challenge of searching for appropriate texts, and the processing of unstructured textual data. In addition, it is particularly appealing that, in contrast to today's areas of application of business intelligence, which usually has the emphasis on structured data numerical analyzes, unstructured news texts are now being analyzed.

[1] PEP [1] https://www.mybca.de/sites/bca_neu/files/media/GWG/PEP_Informationsblatt_BP.pdf

1 INTRODUCTION

Finnova developed a money laundering detection solution for Finnova banks. Banks now want more ways to quickly detect problems such as corruption or bribery of a customer. For this purpose, news articles should be evaluated for negative, unfavorable and risk-increasing information (adverse media search).

According to the Money Laundering Act, the Politically Exposed Person (PEP[2]) customer group is a natural person who holds an important political office, a public office in Switzerland or abroad, as well as their immediate family members and their known close associates are to be checked, since here bribery attempts are possible.

For this purpose, news articles that are available online should be read and evaluated. Measures may be based on information in the news, to check accounts and bookings of the customer for suspicious transactions, which may have to be reported, or the account frozen or closed.

The aim of this work is to create a proof of concept (POC) that shows how the data can be collected, evaluated and assigned to PEP customers.

The analysis options are visualized.

For reasons of discretion, it isn't possible to use data from a Finnova bank. Therefore, the POC was conducted with a freely available PEP list of Swiss national politicians.

With the implementation in Finnova, searching for news will be limited to existing customer connections with PEP customers. In addition, these messages (including historical ones) can be searched online and considered when opening an account.

Sentiment analysis is about large and unstructured data sets and thus about 'BIG DATA':

1. **Volume**: A large volume of data in the 'news' is analyzed

2. **Velocity**: The information must be quickly available to the customer advisor or the compliance department: Push messages

3. **Variability**: The test messages to be analyzed are unstructured

4. **Veracity:** The messages found to a customer are reported with a probability to indicate whether they concern the bank customer,

5. **Value**: The bank can monitor, avoid or minimize reputational risks.

Particularly exciting is the challenge of text recognition and text interpretation.

[2] PEP https://www.mybca.de/sites/bca_neu/files/media/GWG/PEP_Informationsblatt_BP.pdf

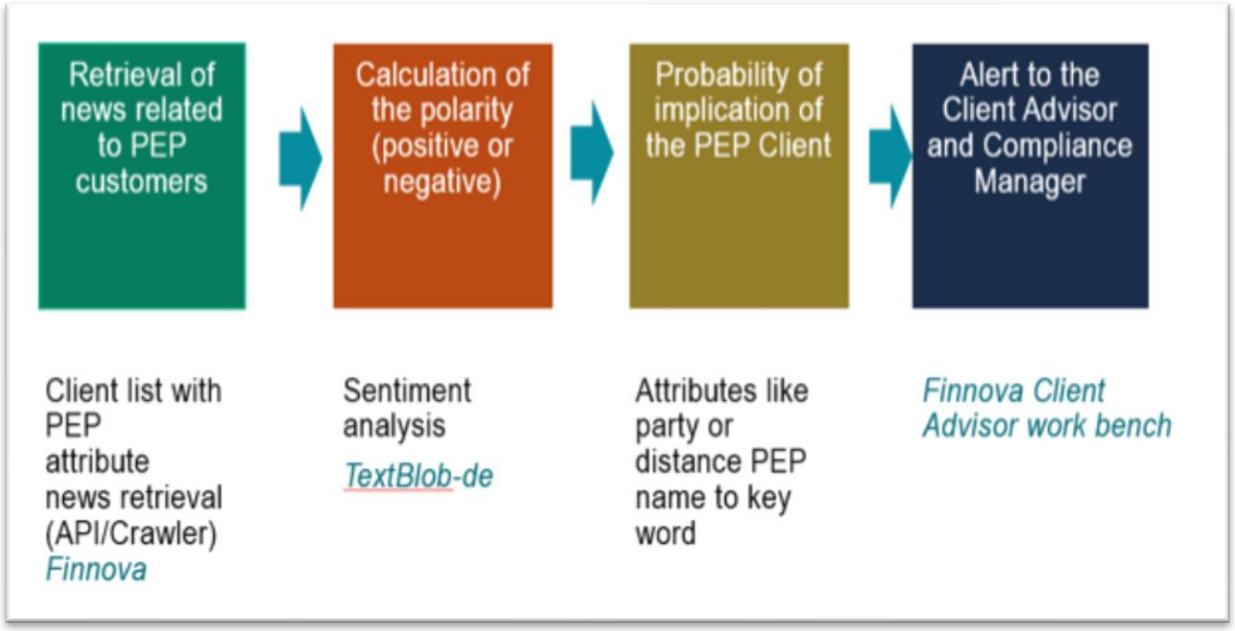


Figure 1 Adverse Media Search Process

2 OVERVIEW FINNOVA

The Finnova Banking Software provides banks and operating partners with an efficient, stable and flexible platform for modern end-to-end banking. At the heart of the strategic goal is the system-of- differentiation.

The Core Suite has the function of Systems of Record with the focus on reliable and efficient transaction processing and consistent primary data management. If the Core Suite can be understood as a core element in a transcendent sense, then the Systems of Insight, the Management Suite, with the controller, fulfill the brain function.

Control information, static and the longer the more algorithmic-dynamic, take center stage. Finnova continues to build the Management Suite with related analytics capabilities that will allow the bank to add value to their data, especially in their customer interactions as well as their controls.

With the Finnova Analytical Framework (FAF), the customers can answer individual questions such as fraud detection and prevention, customer profiling or simulation. However, Finnova's claim goes much further: Finnova offers banks the opportunity to expand the analytical framework step by step towards a holistic solution, according to their individual needs, their existing knowledge and their human, infrastructural and financial resources and potential.

Top strategic issues include improving the digital experience, using information from data analytics for decision making, and reducing operational and compliance costs. With strategic fields of action, Finnova sets clear development priorities in order to support their customers and partners in the new era of banking with an open ecosystem. This combines the efficiency advantages of a standard solution with the differentiation in the market through individual solutions.

With the technological possibilities, banks can gain a competitive advantage through the systematic analysis of large, structured and unstructured data volumes and the intelligent exploitation of results, be it to increase yields or to reduce risks in operational and strategic issues.

Thus, Finnova increasingly relies on the use of analytical tools and helps banks to maintain full transparency about their own business, as well as in the implementation of regulations, business hedging and settlement or the aggregation and provision of information for customers.

Adverse Media Search is part of the FAF (Finnova Analytical Framework) developed in the Development department.

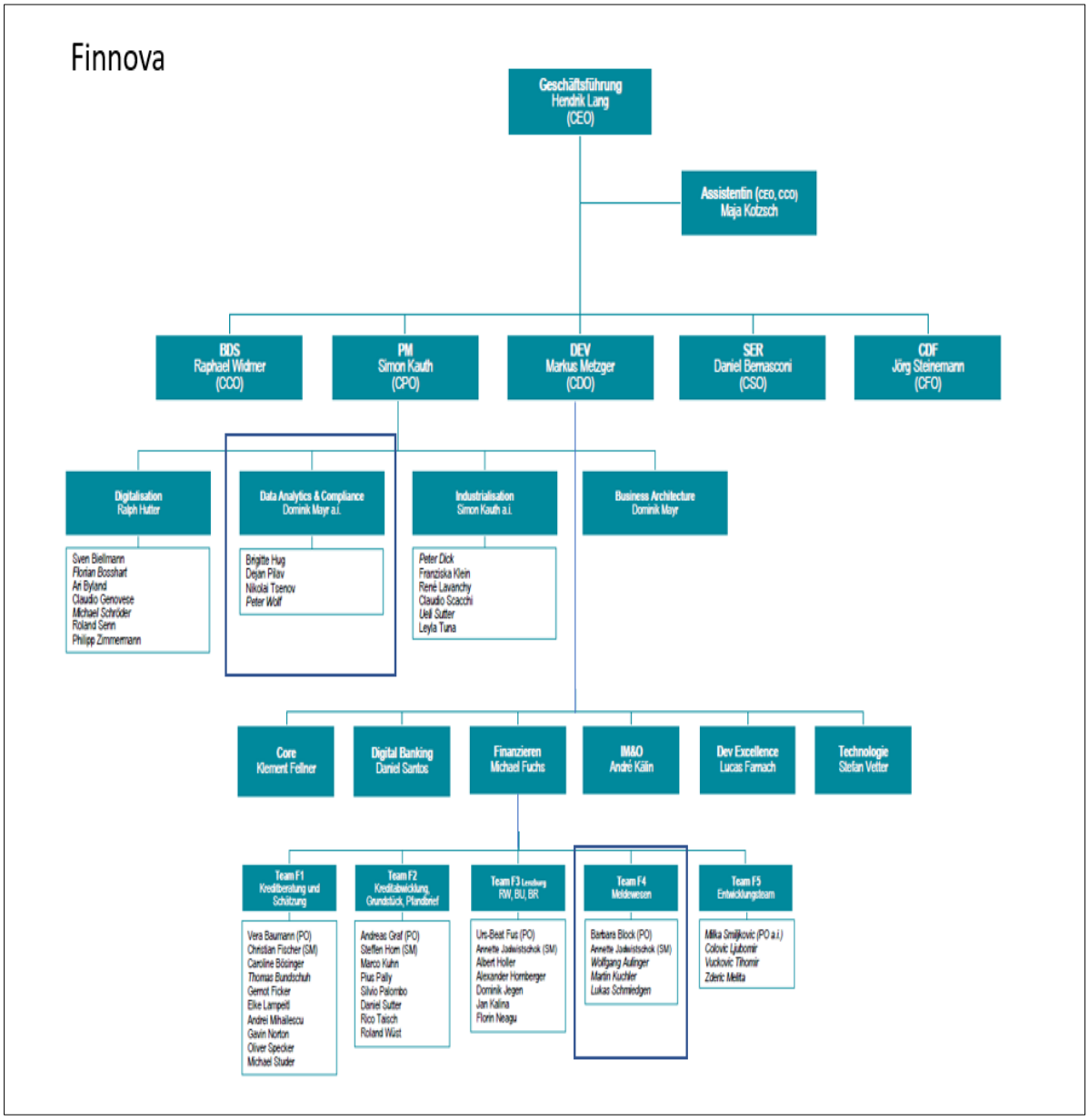


Figure 2 Organigramm Finnova

3 CURRENT SITUATION

• Today the Cantonal Banks using the Finnova Software have no information when a PEP is involved in a scandal relevant to the banking relationship. This includes bribery, corruption, money laundering, drug trafficking, insider trading or a pyramid / Ponzi scheme. They get the information only if customer advisors or compliance manager recognize the information by chance.

• The Cantonal Banks need to recognize quickly, as part of the 'Know your Customer' initiative in addition to money laundering detection, when a PEP is involved in a scandal relevant to the banking relationship.

• According to the Money Laundering Act, a PEP (politically exposed person) is a natural person who has exercised or exercises an important political office or public office in Switzerland or abroad, their immediate family members and persons known to be related. Immediate family members are spouses, partners, their children, parents, and siblings.

• Known persons are co-owners of a business or other close business relationships, or relationships in which the political person is the beneficiary.

• These include Members of Parliament, members of the Government, members of the Supreme Court and judicial authorities, ambassadors, chargé affairs, senior officers, religious leaders, members of the governing and administrative boards and supervisory bodies of state-owned companies.

Examples of corruption in Switzerland are the FIFA scandal in 2015 and the case of Pierre Maudet (Geneva City Council) in 2018. The Basel Cantonal Bank was involved in the 2012 ASE scandals.

30-Millionen-Attacke gegen Basler Kantonalbank

379 geprellte Anleger fordern Geld von der Bank. Das Geldhaus habe sich «bewusst blind» gestellt, als das Schneeballsystem des Vermögensverwalters ASE ins Rollen kam.

Eine Gruppe von Kleinanlegern bekam von der BKB bis heute kein Geld zurück. Foto: Gaëtan Bally (Keystone)

«Mindestens 28 Millionen Franken; plus 2,2 Millionen Euro; plus 2,4 Millionen US-Dollar» – exakt so viel wollen 379 Anleger von der Basler Kantonalbank (BKB) bezahlt haben. Am 19. August haben die Anwälte der Gruppe von Geschädigten einen Antrag beim Bezirksgericht Laufenburg eingereicht.

Hintergrund ist eine der grössten Schweizer Betrugsaffären – der Fall ASE Investments AG. Der Vermögensverwalter aus dem Aargauer Fricktal hatte von 2006 bis 2012 ein Schneeballsystem betrieben, ASE stand für «Anlage, Sicherheit, Ertrag». Die BKB amtierte als Hausbank der Firma. Im Frühling 2012 kollabierte das Konstrukt. Rund 1700 Personen verloren laut Anklage Geld, darunter Ärzte, Chiropraktiker oder Rentnerehepaare, aber auch Finanzberater und Bankangestellte.

ASE-Manager Martin S. hatte vorgegeben, er würde für die Kunden mit Devisen handeln. In Prospekten versprach er Renditen von 12 bis 18 Prozent pro Jahr. Tatsächlich schob S. über Jahre hinweg nur Gelder von Konto zu Konto und stopfte alte Löcher mit neuem Geld. Total 170 Millionen Franken fehlen heute laut Anklage. Voraussichtlich im November kommt es am Bezirksgericht Laufenburg zum Prozess gegen Martin S. und ASE-Präsident Simon M. – nach vier Jahren Ermittlungen.

Figure 3 ASE Example Tagesanzeiger 12.092016 complete article here

4 TARGET SOLUTION - REQUIREMENTS

This section describes the target solution requirements from the identification of business needs and requirements up to the specification of software requirements.

The sources to elicitate requirements have been principally four: Regulations for Swiss Financial Institutions, information searched in the internet, Developing a Proof of Concept (POC) and, last but not least, the IT experience of the team members.

The starting point was a brain-storming discussion around the idea of "searching for negative news related to PEP clients", while using a real case of potential corruption within Switzerland. Plenty news articles about corruption cases and negative news were found in the internet as well as quite a few open-source tools to retrieve that information, process and store it. This supported the setting of a POC for sourcing and processing news articles and executing the initial basic algorithms. The first results were produced, and the requirements could be enriched by testing. Indeed, this exercise facilitated the enhancement of the functionality and rules definitions.

For a good appreciation of this section, it has been subdivided as follows:

- Business Requirements
- Envisioned solution: Adverse Media Screening
- Specification of the solution sub-process: Data Sourcing, News Articles Processing and Alerts Visualization
- Conceptual Data Model of the solution
- Handling of False Negative and False Positive Alerts
- GDPR Requirements

4.1 BUSINESS REQUIREMENTS

Due to their position and influence, Politically Exposed Persons (PEP) represent a high risk in committing money laundering crimes and related predicate offences, including corruption and bribery[3]; hence Financial Institutions must undertake measures to prevent money laundering and corruption in their field of business[4]. PEPs can leverage their influence for personal gain instead of benefiting the public to which they have been entrusted. Likewise, they can favor business interests because of bribes that they may receive, and they may misuse funds for public works projects[5].

It is also the duty of Financial Institutions to keep records of the transactions and clarifications in a manner that other specially qualified persons are able to make a reliable assessment of the transactions and business relationships. That information shall be retained for a minimum of ten years after the termination of the PEP business relationship and be available within a reasonable time to any request made by prosecution authorities[6].

To illustrate PEP's influence associated to personal gains, it is the recent case of Mr. Pierre Maudet, Geneva State Councilor. He was accused of having accepted benefits for a controversial trip to Abu Dhabi in 2015. Crown prince of Abu Dhabi invited the politician and his family to watch a Formula 1 race.

Korruptionsverdacht: Aufstieg und Fall des Pierre Maudet

Reise nicht selbst finanziert

Die Staatsanwaltschaft wirft Maudet nun vor, der Kronprinz habe für die Flüge in der Business-Class und die Unterkunft bezahlt. Der Verdacht lautet auf Vorteilsnahme. Maudet hat zwar bereits offengelegt, dass er die Reise nicht selber finanziert habe. Bezahlt haben soll gemäss seiner Darstellung aber der Geschäftsmann Said Bustany. Der Aussage schenkt die Staatsanwaltschaft allerdings keinen Glauben.

Figure 4 Geneva State Council Pierre Maudet: Luxury trip to Abu Dhabi paid by crown prince. Published by Aargauer Zeitung 1.09.2019. Complete article here

In this regard, the business requirement can be summarized in:

- Support the identification, analysis and resolution of potential corruption cases where PEP clients may be involved. Detailed reports shall be timely provided to compliance analyst with relevant information to resolve signals of corruption cases.

[3] FAFT, Best Practices Paper, The Use of FAFT Recommendations to Combat Corruption

[4] Federal Act on Combating Money Laundering and Terrorist Financing, Chapter 2, Art. 8

[5] Why Relationships with PEPs May Represent an Increased Risk for Financial Institutions

[6] Federal Act on Combating Money Laundering and Terrorist Financing, Chapter 2, Art. 7

- The solution shall be an ongoing proactive process and shall raise alerts as soon as the news has been made public in the internet.
- Compliance Officers and Senior Management shall be enabled with a visualization tool to support the revision of PEP client relationships, considering not only current cases, but historical results of the news screening process.
- It shall be taken into consideration that Cantonal Banks, Finnova's customers, are swiss based Financial Institutions and their clients are only Swiss people and/or foreigners with permanent domicile in Switzerland.
- Record of transactions and relevant clarifications shall be kept in order to assist reliable assessment. This information shall be retained for a minimum of ten years after the termination of the PEP business relationship[7].
- The solution shall be compliant with Global Data Protection Regulation (GDPR), a mandate that financial Institutions must fulfil.

[7] Federal Act on Combating Money Laundering and Terrorist Financing, Chapter 2, Art. 8

4.2 ENVISIONED SOLUTION: ADVERSE MEDIA SCREENING

Based on the business requirements documented in the previous section, a solution has been conceived to retrieve news articles related to PEP clients from the World Wide Web (WWW) in order to identify potential wrong doings of the clients in questions. This section provides a high-level overview of the solution, named from now on "Adverse Media Screening" for PEP client. Details of software requirements, in the form of data processing and rules, are specified in the following sections.

Concepts used within the context of the project:

Adverse Media	Adverse media or negative news is defined as any kind of unfavorable information found across a wide variety of sources, from the traditional news media like newspapers in print or online or broadcast news across radio and TV, social networks[8]. For the purpose of the POC, only newspapers and magazines were taken into account
Alert	An Alert is a notice, generated as per defined conditions, of a potential corruption case in which a PEP client may be involved. Alerts shall be analyzed as per the compliance procedures of the organization.
Data Broker	A Data Broker is a business that aggregates information from a variety of sources; enriches, cleanses or analyzes it; and licenses it to other organizations. Data is typically accessed via an Application Programming Interface (API), and frequently involves subscription contracts. Data typically is not "sold" (i.e., its ownership transferred), but rather it is licensed for particular or limited uses[9].
Compliance Analyst role	The person in this role is in charge of analyzing the cases of potential wrong doings of the clients. In the case of Cantonal Banks, this role is performed by the Client Advisor as part of his/her duties.
Compliance Officer role	A compliance Officer, in corporate compliance is responsible for ensuring the company is in compliance with all federal, state and industry regulation and standards.
Push and Pull processing technology	Push processing technology is software that automates the proactive delivery of information to users or consuming applications. In contrast, "pull" processing requires a user or consuming application seeks for information

Figure 5 depicts the logical processing to identify adverse news related to PEP clients, from the retrieval of news articles, through the examination of their content to finally enable End-Users for the visualization of adverse news, in the form of Alerts.

[8] ComplyAdvantage: https://complyadvantage.com/knowledgebase/adverse-media/

[9] Gartner IT Glossary

The adverse Media Screening solution, delimited by a blue dashed rectangle, depends on structured data from the Client Relationship Management (CRM) system, where the client data is managed, as well as unstructured data from the internet for digital news articles. A third linkage, between the sub-process Alerts Report and Banking Transactions, shown as a grey dashed line, has been introduced to only picture a dependency of information and the Alerts analysis as part of a compliance business process.

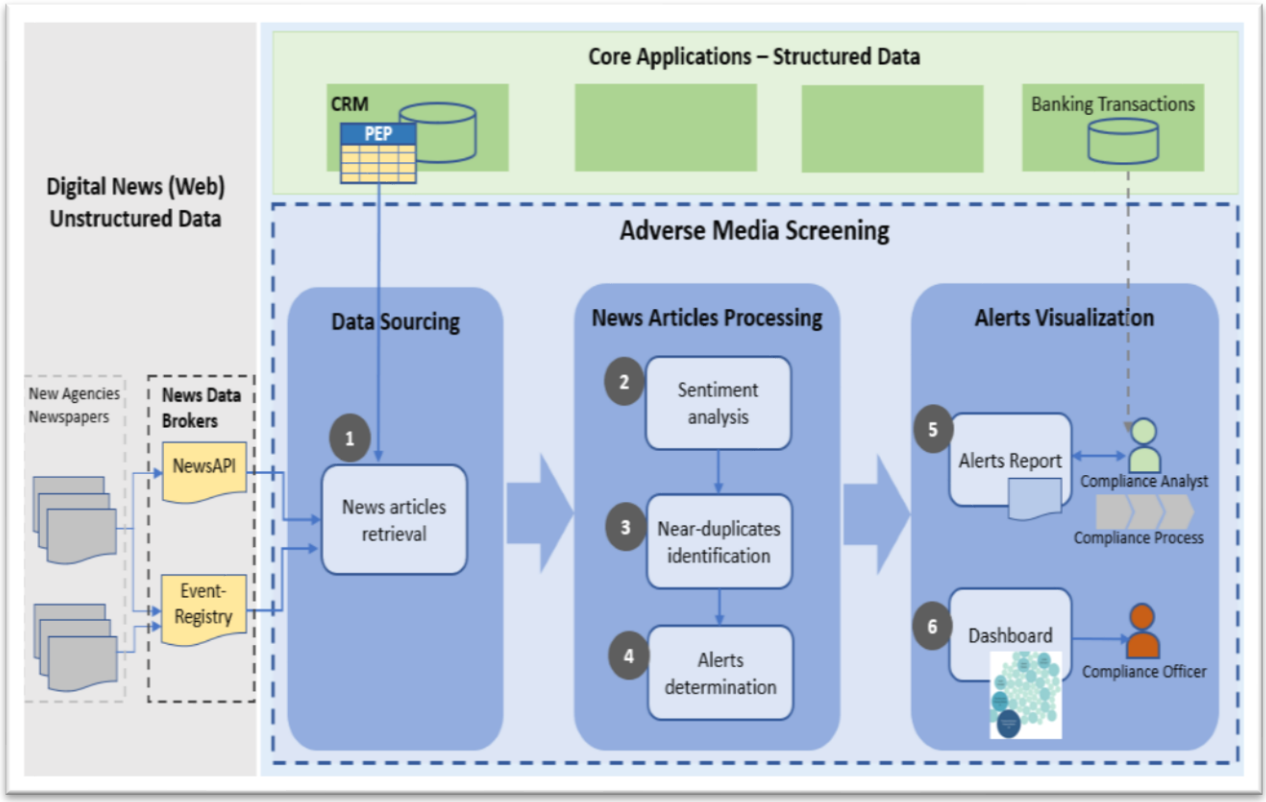


Figure 5 Adverse Media Screening - High Level Overview

Every subprocess is specified int the following sections, though a brief description is provided below:

1. News articles related to PEP clients are retrieved from the digital world, then stored in the company system for further processing.
2. The sentiment expressed in every retrieved article is analyzed using a Python library, TextBlob
3. Since it is usual that news articles are published by different news agencies and newspapers, there is a need for identifying and handling near-duplicates articles
4. Alerts are then determined utilizing a set of German keywords, e.g. Korruption, Bestechung, Schmiergelder.
5. News articles related to PEPs and contains at least one of the keywords are reported. This report shall support analysis and resolution of the Alerts
6. Compliance officers and Senior Managers are enabled to visualize the full collection of Alerts in different dimensions, for instance per specific PEP Client, Period-of-time, Client Advisor.

4.3 DATA SOURCING – NEWS ARTICLES RETRIEVAL

For the purpose of identifying adverse news related to PEP clients, it is required to source news articles from the World Wide Web (WWW). Figure 6 illustrates the input data, processing/selection criteria, output as well as the process frequency is specified in this sub-section.

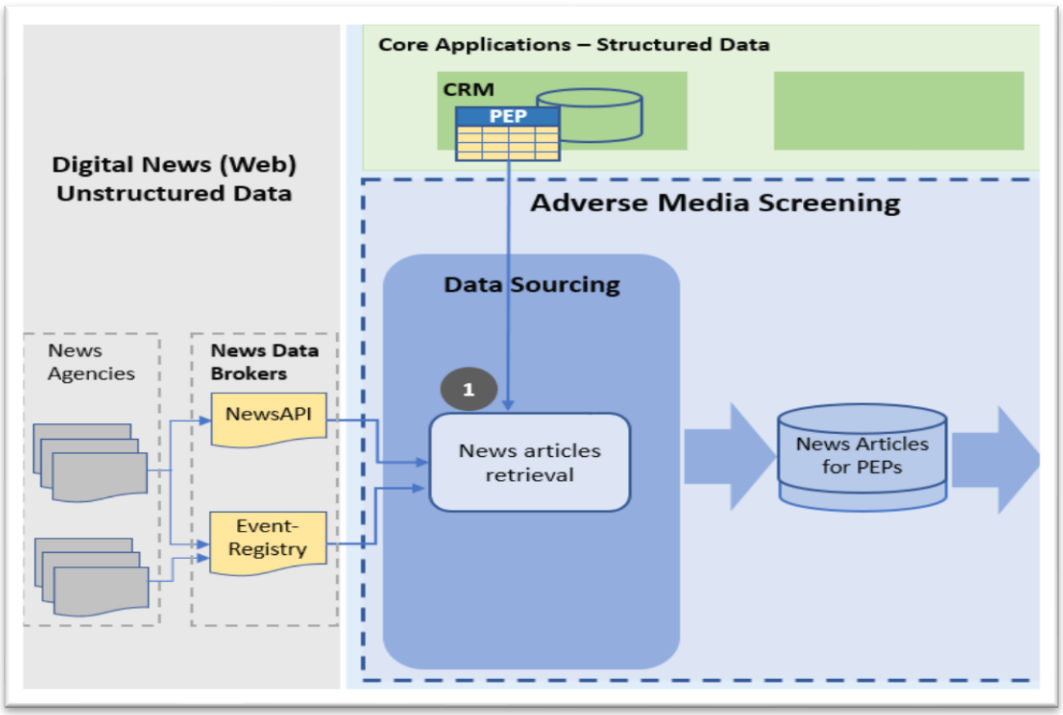


Figure 6 Data Sourcing – Retrieval of News Articles

4.3.1 Input Data

The retrieval of news articles is fundamentally based on two data sources, the PEP clients list and the digital news articles available in the internet.

a. **PEP clients list**: This file contains all PEP clients which have been already identified by the organization and it is sourced from the CRM system. It is important to point out that the Adverse News Screening solution relies on the content of this file, it is therefore the responsibility of the CRM system to ensure the completeness and integrity of the PEP client data. The required PEP client information is specified in section 4.4.6.

b. **Digital news articles**: There are plenty of newspapers, magazines, and other media types available in the digital world, where the existence of Data Brokers facilitate the retrieval of information from the WWW via an Application Programming Interface (API). A valid license with relevant data brokers is a pre-condition to start retrieving the news articles for the proposed solution, Adverse Media Screening.

4.3.2 Quality of the Input Data

The data quality of the PEP client information shall be ensured by the data provider, in this case the CRM System, the golden source for client data.

In relation to the quality of News Article, for example in terms of probability to get fake news, the Data Broker guarantees the completeness and consistency of the data.

4.3.3 Selection Criteria for retrieving news articles

The retrieving criteria shall take into consideration basically two aspects: PEP clients and news

PEP Clients	News Articles
Since news articles about people convey information mainly based on names (first name and family name) only, there is a high likelihood of getting namesakes of PEP Clients. Thus, the retrieval arguments shall consider the following attributes: • Client Name • Public function • Political Party • Client Date of Birth (DOB) • Gender Though that criteria may not ensure identifying the PEP clients uniquely because the metadata of news articles may not contain all information. Hence the introduction of additional criteria to filter out irrelevant information may be needed. During the POC, the following exclusion was considered: Fussball, Bayern, Bundesliga. The actual list is provided in the technical solution specification.	**Language** Being Switzerland a country with multiple official languages as well as an international key financial player, it is required to retrieve news in at least four languages: • German • French • Italian • English **News** Swiss and European newspapers and magazines The Adverse Media Screening solution shall nonetheless enable the capability to customize the preferred language(s) and news sources as per the organization's needs. For POC purposes, only news in German has been used.

Note: Namesake as defined by the Cambridge dictionary and within our context, is a person having the same name as another person. An example of namesakes, which are frequently in the news, are Mr. Thomas Müller the Swiss National Councilor, and Mr. Thomas Müller the German football player.

4.3.4 Retrieval process type and frequency

The news shall be examined as soon as they are made available to the public in the WWW. This implies that the news shall be pushed into the Adverse Media Screening solution. However, considering that the high number of news sources and the high frequency of news articles being published make complex a push process type, Data Brokers only offer pull retrieval of news articles[10]. In order to replicate a push processing, the news retrieval pull process can be executed constantly. News data consumers are indeed enabled to retrieve news articles even per minute[10].

[10] EventRegistry, talk with Mr Gregor Leban, CEO of EventRegistry

4.3.5 Output Data: PEP Client News

The output of the news retrieval sub-process are the PEP Clients, for whom at least a news article was found. The information shall be stored in the company system and shall contain details of the news article itself such as news agency source, publishing date and time. The relevant attributes of articles, held in the PEP News object, are described in section 4.4.6 Conceptual Data Model of the Adverse Media Screening Solution.

4.4 NEWS ARTICLES PROCESSING

Once the news articles related to the PEP clients have been retrieved and stored in the company system, they are ready for analysis and the creation of alerts, when appropriate. Figure 6 shows three sub-processes which are performed sequentially.

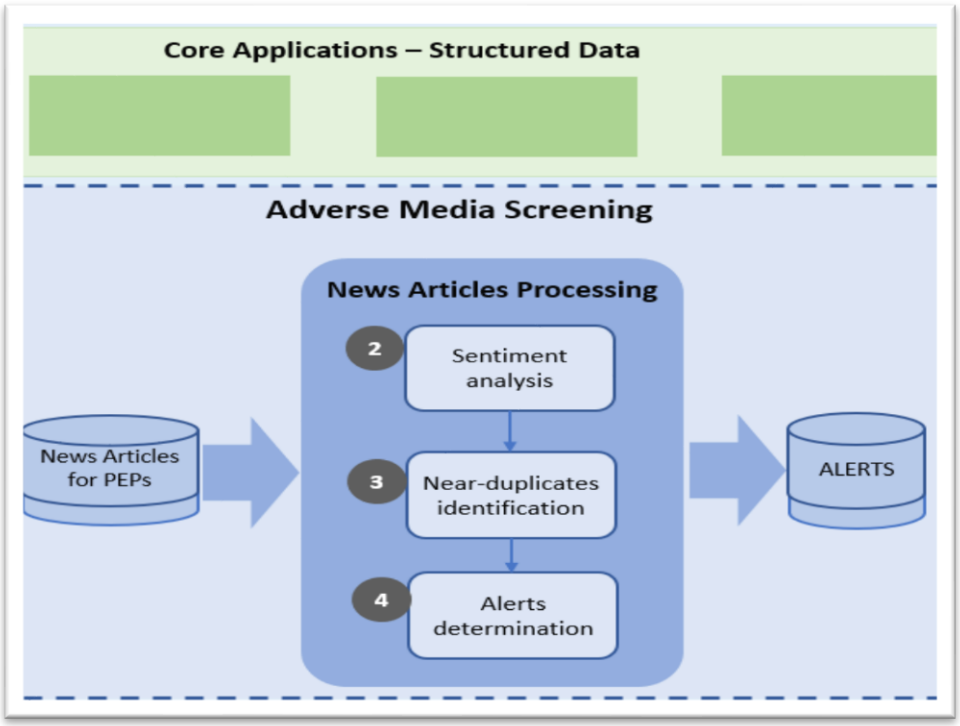


Figure 7 News Articles processing

4.4.1 Sentiment Analysis

The first processing step is to analyze the sentiment expressed in the content of the retrieved news articles. This shall give an indication whether the news article states positive or negative information about the PEP client.

Due to time constraints, the project team decided to utilize an open-source Python library, TextBlob, to perform the sentiment analysis.

TextBlob-de is a library for processing textual data known as Natural Language Processing (NLP) tasks, among these is sentiment analysis[11]. This Phyton library expresses the sentiment of a text in terms of polarity, which is a float within the range -1.0 to 1.0. Negative values denote negative sentiment in the text. Positive sentiment is represented by positive values, 0 indicates neutral.

Executing TextBlob on news articles however demonstrated that the sentiment of the news articles was mostly neutral, evidencing that TextBlob is not the appropriate algorithm for the purpose of this project. For instance, the Python library determined as neutral sentiment, 0.02369, for an evident corruption case reported in a news article. Please refer to Figure 8 for the news article in question.

[11] TextBlob: Simplified Text Processing: https://textblob.readthedocs.io/en/dev/

Erschütterung in Zug und in Afrika

Der Schweizer Geschäftsmann Jean-Claude Bastos verliert sein Angola-Mandat. Und Glencore steht im Kongo unter grossem Druck.

ie Paradise Papers haben entscheidend zum Absturz des Milliarden-nperiums von Jean-Claude Bastos beigetragen. Vor gut zwei Wochen verfügte in britisches Gericht die Einfrierung von drei Milliarden Dollar auf den Konten es umstrittenen Schweizer Geschäftsmanns und dessen Firmen. Betroffen sind uch zwei Schweizer Firmen aus Bastos' Quantum-Global-Gruppe mit lauptsitz in Zug.

«Korruption, Geldwäscherei»

In Angola lebt jeder zweite Bewohner mit weniger als zwei Dollar am Tag. Bastos hingegen machte mit dem Staatsfonds-Mandat innert kurzer Zeit ein Riesenvermögen, wie Dokumente aus den Paradise Papers zeigen. Einerseits kassierte Quantum Global pro Jahr 60 bis 70 Millionen Dollar Managementgebühren, die teils umgehend auf ein privates Konto von Bastos weiterflossen. Andererseits liess Bastos den Staatsfonds mehrere Hundert Millionen in Projekte investieren, die ihm selber gehören.

In einer Eingabe ans Gericht vom 9. April 2018 schreiben die mauritischen Finanzermittler, Bastos habe von Investments des Staatsfonds «persönlich profitiert». Es gebe ein erhebliches Risiko, dass Gelder des Staatsfonds widerrechtlich verschoben würden. Konkret verdächtigt die mauritische Financial Intelligence Unit Bastos der «Korruption, Veruntreuung und Geldwäscherei». Es gilt die Unschuldsvermutung.

Figure 8 Corruption/Money Laundering article. TextBlob determined it as neutral news. Published by Berner Zeitung on 15.05.2018. Complete article here

TextBlob is still being used, and its results are only used as reference. The sentiment analysis sub-process will be replaced by a proper algorithm to analyze the story itself narrated in a news article.

Sentiment analysis is a large growing topic and it is worth to illustrate it for further development of the Adverse Media Screening solution. Section 6 Sentiment Analysis provides an academic explanation how sentiment analysis is performed.

4.4.2 Near-Duplicates processing

News articles appear usually on different news agencies in either almost identical or revised form. This is because the news stories frequently consist of syndicated information, which is the supply of news for reuse and integration with other text. For instance, replacement of headlines, captions, and sometimes local-relevant content is added[12]. In the technical jargon, these documents are near-duplicates of each other.

Near-duplicates shall be detected by comparing the body text of the articles. If the similarity is higher than 80%, the near-duplicate articles shall be evidenced by indicating the related news articles and the corresponding level of similarity.

Although near-duplicate articles, which are regularly almost identical, may overstate the screening process, all of them shall be kept for data completeness reasons and GDPR requirements. For Alerts processing and reporting purposes, only the article that was published first shall be considered.

The example below illustrates a case of near-duplicate articles published by the Luzern and Aargauer newspapers:

Armee sorgt für Missklänge: 300'000 Franken für Posaunen aus Ausland - Schweizer Angebot war günstiger

Die Militärmusik der Armee kauft 48 Posaunen in den USA statt aus einheimischer Produktion. Im Kanton Thurgau ist die Politik erbost. Verteidigungsministerin Viola Amherd soll sich erklären.

Dissonanzen im Zusammenhang mit der Instrumentenbeschaffung für die Schweizer Militärmusik sind nicht neu. Schon vor zwei Jahren wurde bekannt, dass sich die Armee reichlich unpatriotisch mit Instrumenten aus dem Ausland eindeckt - statt die kleinen, aber feinen Schweizer Hersteller zum Einsatz kommen zu lassen. Diese konnten in der Vergangenheit nicht einmal Offerten einreichen.

Figure 9 Aargauer Zeitung, 16.01.2019, 15:02Hrs

Armee sorgt für Missklänge: 300'000 Franken für Posaunen aus Ausland – Schweizer Angebot war günstiger

Die Militärmusik der Armee kauft 48 Posaunen in den USA statt aus einheimischer Produktion. Im Kanton Thurgau ist die Politik erbost. Verteidigungsministerin Viola Amherd soll sich erklären.

Dissonanzen im Zusammenhang mit der Instrumentenbeschaffung für die Schweizer Militärmusik sind nicht neu. Schon vor zwei Jahren deckte diese Zeitung auf, dass sich die Armee reichlich unpatriotisch mit Instrumenten aus dem Ausland eindeckt – statt die kleinen, aber feinen Schweizer Hersteller zum Einsatz kommen zu lassen. Diese konnten in der Vergangenheit nicht einmal Offerten einreichen. Das hat sich bei der jüngsten Beschaffung – 300000 Franken für 48 Bassposaunen – zwar geändert. Das Resultat blieb dasselbe: Der Auftrag geht ins Ausland, in die USA und nach Frankreich. Auch der Weinfelder Hersteller «Blaswerk-Haag» ging leer aus, obwohl er sogar günstiger offerierte.

Figure 10 Luzerner Zeitung 16.01.2019, 16:03 Hrs

[12] Duplicate News Story Detection Revisited – Conference paper 2013 - SpringerLink

Note: the full content of the both news articles can be found in: AargauerZeitung and LuzernerZeitung

Comparing both articles resulted in 87% of similarities. The 13% text differences were not really substantial. For instance, the Aargauer Zeitung stated: «**Schon vor zwei Jahren wurde bekannt, dass**.», while the Luzerner Zeitung wrote: «**Schon vor zwei Jahren deckte diese Zeitung auf, dass**…»

The articles in question are presented following. The differences have been highlighted in orange.

News article published by the Aargauer Zeitung:

Die Militärmusik der Armee kauft 48 Posaunen in den USA statt aus einheimischer Produktion. Im Kanton Thurgau ist die Politik erbost. Verteidigungsministerin Viola Amherd soll sich erklären.

Dissonanzen im Zusammenhang mit der Instrumentenbeschaffung für die Schweizer Militärmusik sind nicht neu. Schon vor zwei Jahren wurde bekannt, dass sich die Armee reichlich unpatriotisch mit Instrumenten aus dem Ausland eindeckt - statt die kleinen, aber feinen Schweizer Hersteller zum Einsatz kommen zu lassen. Diese konnten in der Vergangenheit nicht einmal Offerten einreichen. Das hat sich bei der jüngsten Beschaffung - 300'000 Franken für 48 Bassposaunen - zwar geändert. Das Resultat blieb dasselbe: Der Auftrag geht ins Ausland, in die USA und nach Frankreich. Die Schweizer Unternehmen gingen leer aus. Darunter auch das renommierte Thurgauer Unternehmen "Blaswerk Haag" - und das obwohl die Weinfelder Firma sogar günstiger offeriert hat.

Diese Beschaffungspolitik der Armee, die sich auf das Gesetz beruft, stösst nicht nur vielen Parlamentariern sauer auf, sondern jetzt auch der Thurgauer SP-Regierungsrätin Cornelia Komposch. Die Chefin des Departements für Justiz und Sicherheit hat deshalb der neuen Verteidigungsministerin Viola Amherd geschrieben. Die Melodie des Briefes: Auch wenn am Entscheid nicht mehr gerüttelt werden könne, bitte man dennoch um Transparenz und eine Stellungnahme. "Es interessiert uns, zu wissen, weshalb die USA und Frankreich den Zuschlag erhielten", heisst es im Schreiben vom 4. Januar an die frisch gewählte Bundesrätin.

Erstaunt zeigt man sich im Kanton des ausgebooteten Unternehmens nicht zuletzt über die Gewichtung der Vergabekriterien: 70 Prozent betreffen die Qualität und lediglich 30 Prozent den Preis. "Es ist nicht nachvollziehbar, wieso die vom Thurgauer Instrumentenbauer blaswerk Musik Haag eingereichte Offerte betreffend die Posaunen bezüglich Qualität so viel schlechter bewertet wurde", schreibt Regierungsrätin Komposch. Zumal das Unternehmen national als auch international einen sehr guten Ruf geniesse. "Die Instrumente bürgen für hohe Qualität." Auch heisse es in Artikel 37 der Verordnung über das Beschaffungswesen, dass das wirtschaftlichste Angebot den Zuschlag erhalten müsse. "Aus unserer Perspektive, die zugegebenermassen eine einseitige ist, ist die Vergabe also kritisch zu hinterfragen", so Komposch.

Sie sei als Militärdirektorin mehrfach auf "diese sehr wohl befremdliche Vergabe angesprochen worden", begründet Komposch ihre Intervention. Dass ein äusserst qualifizierter Thurgauer Instrumentenbauer mitofferiert habe, der Auftrag jedoch zu einem bedeutend höheren Preis an die USA gegangen sei, "hat auch mich persönlich sehr gestört". Im Schreiben wird ausserdem auf die aktuelle Politik der USA verwiesen. Schliesslich erinnert Komposch Bundesrätin Amherd daran, dass der Kanton Thurgau "seit jeher treu und eng mit der Armee verbunden" sei. "Es geht in dieser Sache um Transparenz und Vertrauen."

Ein Antwortschreiben aus Bern ist bis gestern noch nicht im Kanton Thurgau eingetroffen. Dass die neue Bundesrätin der Militärmusik den Marsch blasen wird, ist allerdings kaum zu erwarten. Wahrscheinlich muss sich die einheimische Blasmusik an folgendes Motto halten: Stetes Posaunen höhlt vielleicht den Stein.

News article published by the Luzerner Zeitung:

4.4.3 Determination of Alerts

Since the polarity, calculated in the previous sub-process, is the verbal representation of the writer's sentiment, this value does not give a legitimate indication of a potential corruption case. It is therefore necessary to search for specific criteria, industry and context relevant, to identify potential negatives news. This shall support the recognition of news articles for compliance analysis. These are named as "Alert Keywords" within the Adverse Media Screening solution.

Alert Keywords
Korruption
Schmiergelder
Bestechung
Schneeballsystem
Vorteilsnahme
ungetreue Geschäftsführung
Betrug
Schneeballsystem
Insiderhandel
Steuerhinterziehung
Käuflich
schwarze Kasse
Schwarzgeld
Veruntreuung
Unterschlagung
Beeinflussung
Ponzi

Alert Keywords support the identification of news articles that may reveal a potential wrong doing of PEP clients.

A set of initial Alert keywords, in German language, has been introduced to test the POC, analyze the results and derive processing rules.

It is important to remark, that the Adverse Media Screening solution shall provide the capability of customization the Alert Keywords as per business needs. For instance, include keywords in different languages.

Having set the Alert Keywords, the screening process shall search for these in the news article. The news articles where at least one Alert Keyword appears, either in the title or body, shall be flagged as Alert. In case more than one Alert Keyword were found, all of them shall be referenced for further use in the reporting feature.

4.4.5 Special Case: More than one PEP client was found in the same news article

In cases where more than one PEP client was found in the same news article, that contains a least an Alert Keyword, it is required to denote which PEP client(s) is/are involved in the potential corruption case.

Although the Adverse Media Screening solution does not perform a semantic analysis of the news articles, the solution shall still support the compliance analyst in the analysis and resolution of the potential corruption case.

Based on the testing cases performed, POC wise, it is required that the solution determines the identification of the position within the text of both the PEP client(s) and the Alert Keyword(s). The bigger the difference between the position of the PEP client and the Alert Keyword is, the lower the probability that the PEP client in question would be a concern of the compliance control. For Alert reporting purposes, only the news where the difference is less than to 50 should generate an Alert.

To sum up, the Compliance Analyst needs to analyze the news article and judge the case.

Further analysis of the cases and the implementation of machine leaning algorithm is necessary to really examine the news content and provide which PEP may be implicated in a potential corruption case.

4.4.6 Conceptual Data Model of the Adverse Media Screening Solution

Based on the requirements documented in previous sections, the below conceptual data model has been conceived. Figure 11 shows the relevant objects within the dashed rectangle, and dependency of the PEP Client object with the Client object of the CRM system.

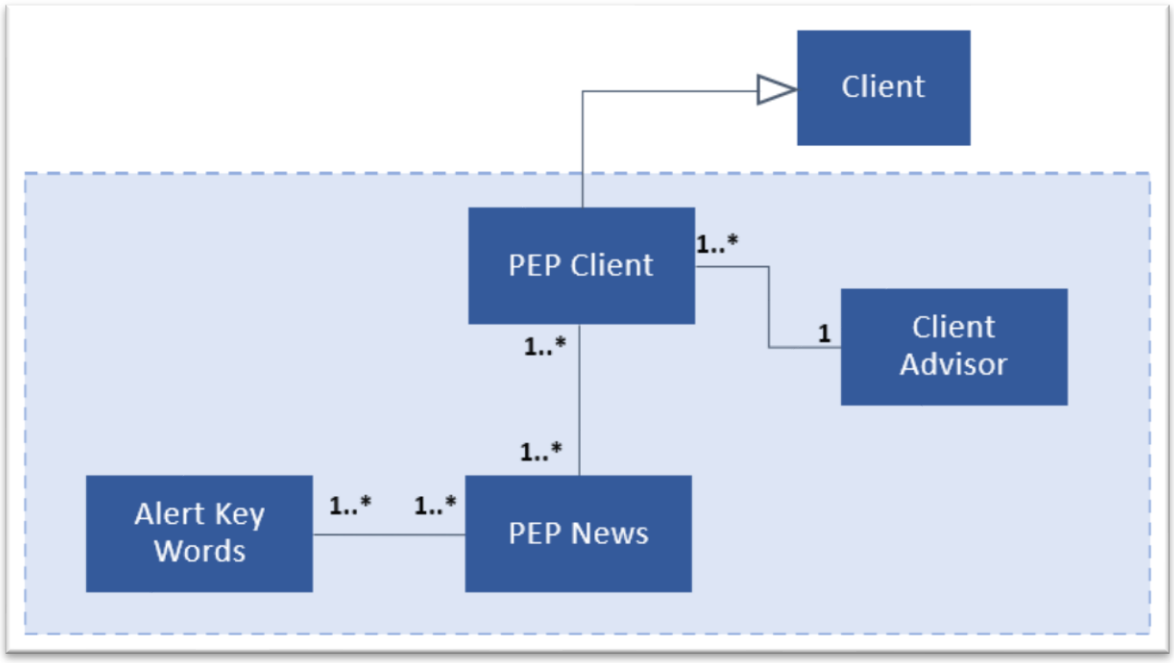


Figure 11Conceptual Data Model of the Adverse Media Screening Solution

It is important to highlight that conceptual model only takes into consideration the objects and attributes required by the Adverse Media Screening solution. Each Object is briefly described below:

PEP Client object This object contains all PEP clients identified by the CRM system. For POC purposes, a list of PEP clients was created with basic information.		
	Attribute Name	**Description**
1	ClientId	Client ID as managed by the CRM system
2	Name	Name of the client: first name and second mane
3	Client gender	Gender of the client
4	DOB	Date of Birth of the client
5	ClientAdvisorId	Id of the Client advisor responsible for the client
6	PoliticalParty	Political party to which the client belongs, if any
7	PublicPosition	Public position entrusted to the client
8	PublicPositionFromDate	From Date of the entrusted public position
9	PublicPositionToDate	To Date of the entrusted public position

PEP News

This object shall contain all news retrieved for PEP clients, regardless whether they represent a negative news or not. The negative news shall be flagged for reporting alerts.

#	Attribute Name	Description
1	ClientId	Client Id as managed by the CRM system
2	Name	Name of the PEP clientt: first name and second mane
3	RelatedNewsLink	Link of the news article found
4	RelatedNewsPublishTS	Timestamp on which the news was made available by the news agency
5	RelatedNewsPublishSource	Source of the news
6	SubjectDomainClass	News category assigned by the news agency, e.g. finance, political, natural disasters
7	RelatedNewsCreationTS	Timestamp on which the news was created in the system
8	PolarityLevel	Level of polarity calculated by the Sentiment Analysis algorithm
9	PolarityAlgorithmName	Name or identification of the algorithm utilised to calculate the news polarity level
10	PolarityAlgorithmPropriet	Owner of the polarity algorithm
11	AlertKeyWord	Alert key word with the highest rank found in the news
12	PositionPEPName	Position of PEP name within the article
13	PositionAlertKeyword	Position of the Alert Keyword within the article
14	PEPAlertDistance	Distance between position of PEP Name and Alert word

Alert Keywords

These words support the identification of negative news, and shall be maintained (created/modified/deleted) by the compliance business unit

#	Attribute Name	Description
1	AlertKeyword	Alert Keyword
2	AlertKeywordSeverity	The severity is intended to support the prioritization of the cases. This attribute has not been used in the POC

4.5 HANDLING FALSE NEGATIVE AND FALSE POSITIVE ALERTS

As per the set of conditions to generate Alerts, a lot of false Alerts were produced. In order to reduce them, a calculation of the distance (in terms of how many words) between the PEP client name and the Alert Keyword was introduced. The results were analyzed in order to probe the effectiveness of the rules and to eventually modify the processing rules or create new ones. The confusion matrix was utilized to classify the false alerts as illustrated in Figure 12.

		Actual Alert	
		Positive	**Negative**
Predicted Alert	**Positive**	**True Positive** A news article was predicted as adverse news and it is true. The PEP client is potentially involved in a corruption case	**False Positive** A news article was predicted as adverse news, but it actually is not.
	Negative	**False Negative** A news article was predicted as non-adverse news, but it actually is.	**True negative** A news article was predicted a non-adverse and it is true. News article related to a PEP client does not contain an Alert keyword and it is not reported.

Figure 12 False Negative and False Positive – Confusion Matrix for adverse news screening

4.5.1 False Negative Alerts

Cases where a news article was predicted as non-adverse news, but it actually is.

Within the context of the solution alert criteria a news article should not be reported. However, reading and analyzing the article in question, it was established that the it denoted adverse news. Following it is presented the focus of analysis and the actions to be taken:

Focus of Analysis	**Action**
PEP Client List	Verify that PEP client name is correctly spelt, and it exists in the PEP Client List. If this is not the case, the issue shall be notified to the PEP client data sourcing application, i.e. CRM system, for the corresponding changes.
Alert Keywords	Verify if the Alert keyword(s) identified by analyzing the news exist in the Alert Keywords List. If not, the new keyword(s) shall be added to the list.
News retrieval criteria	Verify the criteria to retrieve news and adapt the conditions to filter in the news correspondingly.
News Data Broker/News Agency	Verify that a valid license with the news Data Broker is in place. Likewise, verify with the Data Broker the reason of not getting the news article in question. It may be for example that the newspaper is a news source of the data provider.

4.5.2 False Positive Alerts

A news article was predicted as adverse news, but it actually is not. That is, a news article related to a PEP client contains an Alert keyword, though the PEP client is not involved in a potential corruption case, or even the news is not about corruption.

Focus of analysis	Action
Alert Keywords. homonyms	As per analysis of the test results, it was identified that the Alarm keyword "Betrug" had the homonym "betrug" which does not indicate corruption in any sense. The case is illustrated bellow. Enhancements to handle such cases are planned.
News story and/or negation of Alert keywords	It may be that the news is about an opinion, for example, about money laundering. Or the PEP client is the news because his/her open fight against corruption. Further enhancements shall be considered to analyze such cases.

Example of the "Betrug" Alert Keyword and its homonym "betrug", a past tense of the German verb "betragen" whose meaning in English is "to be", "to account".

Keller-Sutter (FDP) in St. Gallen gewählt - zweiter Wahlgang nötig

Die St. Galler Ständerätin Karin Keller (FDP, bisher) ist klar wiedergewählt worden. Mit 103'258 Stimmen lag die 51-Jährige klar über dem absoluten Mehr. Paul Rechsteiner (SP, bisher) und Thomas Müller (SVP) müssen in den zweiten Wahlgang.

Rechsteiner erhielt 62'944 Stimmen, während sein Herausforderer von der SVP, Thomas Müller, auf 50'629 Stimmen kam. Beide verpassten das absolute Mehr von 76'367 Stimmen klar. Die Wahlbeteiligung betrug 49,4 Prozent. Um den zweiten Sitz kommt es am 15. November zum zweiten Wahlgang.

Figure 13False Positive Alarm: homonym of Alert Keyword "Betrug". Full article here

A second example of false positive, an Alarm was generated for PEP client Jacqueline Badran. Figure 14 False Positive Alert: An opinion expressed by a PEP client:

SBB verkaufen Immobilien für 1,5 Milliarden Franken

Die SBB stossen Land und Gebäude ab – und bessern damit ihr Jahresergebnis auf.

Die **SBB** hat seit 2007 Land und Immobilien im Wert von über 1,5 Milliarden Franken verkauft. Alleine im letzten Jahr stiess die SBB Immobilien im Wert von knapp 204 Millionen Franken ab, berichtet der «Sonntagsblick». Die Verkäufe hätten dazu beigetragen, die Jahresergebnisse zu verbessern – und vielleicht sogar dazu beitragen können, Bonus-relevante Ziele zu erreichen, so die Zeitung.

Die SP-Nationalrätin Jacqueline Badran spricht von «Veruntreuung von Volksvermögen». Es störe sie, dass viele Immobilien an kommerzielle Privatanleger verkauft würden. Mit der Volksinitiative «Mehr bezahlbare Wohnungen» fordert die Politikerin zusammen mit Mitstreitern unter anderem, dass Kantone und Gemeinden in Zukunft ein Vorkaufsrecht haben, wenn die SBB oder andere bundesnahe Betriebe wie die Post Grundstücke verkaufen wollen.

Für 204 Millionen Franken verkauften die SBB 2017 Immobilien und Grundstücke. Seit dem Jahr 2007 waren es solche im Wert von über 1,5 Milliarden Franken. Bild: Christian Beutler/Keystone (4 Bilder)

Figure 14 False Positive Alert: An opinion expressed by a PEP client

4.6 ALERTS VISUALIZATION

News articles were retrieved and analysed, Alerts were generated and stored in the file system. Now those Alerts shall support the compliance professionals to analyse their PEP clients associated to potential corruption cases.

Two roles have been identified as End-Users of the Adverse Media Screening solution: The Compliance Analyst, who analyses the reported cases and the Compliance Officer or Senior Managers responsible for ensuring that the company is in compliance with all federal, state and industry regulation and standards.

The reports content, in terms of PEP client and article attributes, are described in following sections.

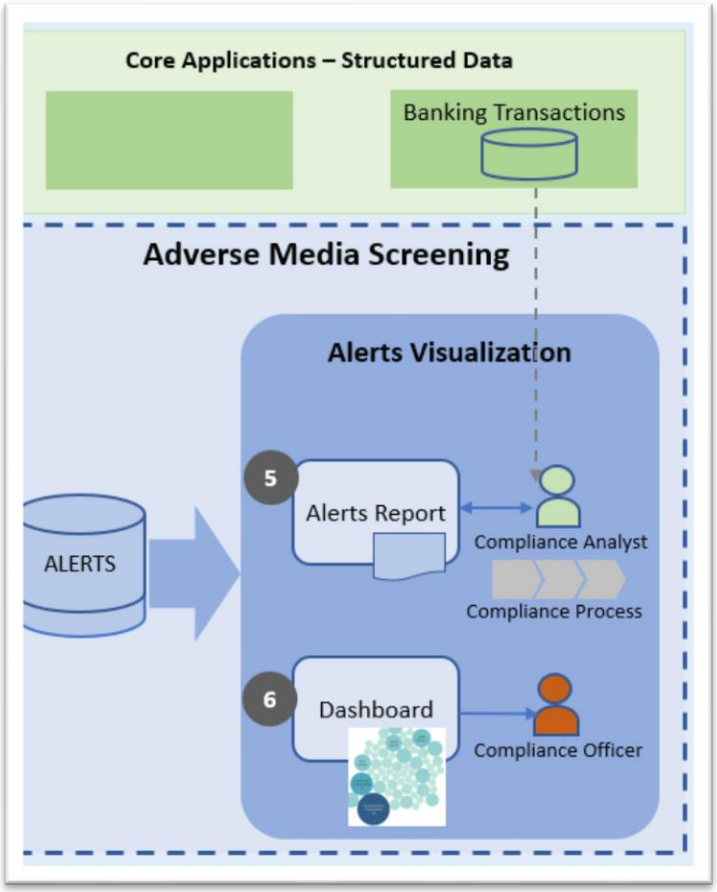


Figure 15Alerts Visualization

4.6.1 Alerts Report

As soon as an Alert is generated, the Compliance Analyst shall count on details of both the PEP client and the related negative news. This information is essential for him/her to examine the potential case of wrong doing of the PEP Client. He/she will use the news articles(s) couple with internal information provided by core systems. It is important to point out that compliance professionals follow a specific procedure of the organization, which is not matter of this project.

In Cantonal Banks, the compliance analyst role is performed by the Client Advisor.

Using an example of news about Mr. Pierre Maudet, Geneva State Councilor, back in May 2018, when the story of his potential wrong doing, pictured below

Libanesischer Geschäftsmann zahlte laut Maudet Reise nach Abu Dhabi

Der Genfer Staatsrat Pierre Maudet hat in Zeitungsinterviews vom Donnerstag eingeräumt, dass er seine umstrittene Reise nach Abu Dhabi im November 2015 nicht selbst bezahlt hatte. Ein libanesischer Unternehmer vor Ort sei für die Rechnung aufgekommen.

17.05.2018 06:35

Figure 16 Geneva State Councilor, case reported as misuse of his public function. Reported by Cash on 17.05.2018. Full article here

#	Attribute	Description
1	PEP Client Name	Name of the PEP client
2	PEP Client Id	Client Id of the PEP client
3	PEP Client Advisor	Name of the Client Advisor, who manages the business relationships with the PEP client
4	News Source	Name of the newspaper
5	News Article Header	Title of the news article
6	News Article Type	News article type, if available
7	Publishing Timestamp	Timestamp when it was published
8	Alert Keyword	Alert word found in the news article
9	Status	Values to indicate: New case, case under analysis and closed
10	Polarity	Polarity calculated by the sentiment analysis
11	Number of articles found (excluding near duplicates)	Number of articles found for the given PEP client.
12	Link	Link of the article(s) found
13	Comments	A free text box shall be provided to allow recording comments of the reported case

4.6.2 Dashboard

While Alerts are reported to compliance analysts as soon as they are generated for immediate actions, a dashboard shall be provided to Compliance Officers and Senior Managers. These Users shall have the possibility to see the complete collection of Alerts in different dimensions. For instance, Alert for specific Client Advisor, or per specific Period-of-time FROM ddmmyyy TO ddmmyyyy.

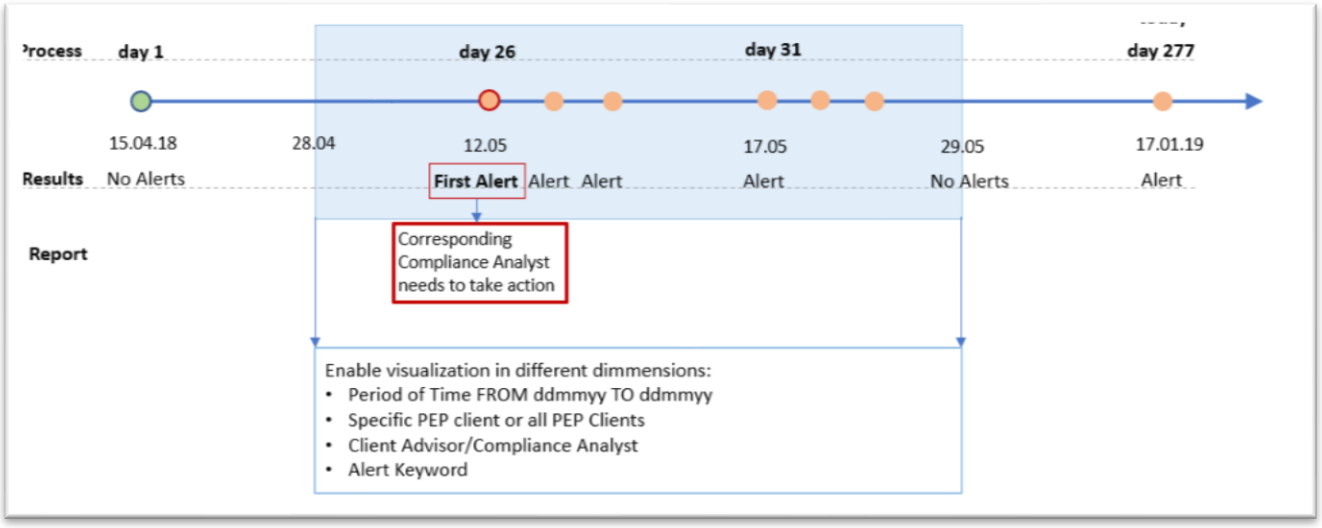


Figure 17Example of alerts generated chart, used for Dashboard requirements discussions

Examples of the Alert visualization is included in the technical solution section.

4.7 GENERAL DATA PRIVACY REQUIREMENTS (GDPR)

#	GDPR Principles	Requirement
1	Personal Data shall be processed lawfully, fairly and transparently	The PEP client shall be informed what kind of data will be collected, processed and reported, and ensured that those are aligned with the implementation of the new Adverse Media Screening solution. This requirement shall be included in the Terms of Services, Allgemeine Geschäftsbedingungen (AGB).
2	Purpose Limitation	Personal data can only be obtained for specified, explicit and legitimate purposes. It cannot be other purposes, except with the consent of the PEP client. Therefore, the company Internal Policies shall be enhanced to make aware of the new data collection and processing.
3	Data Minimization	The data collected of the PEP Clients shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed.
4	Accuracy	Along with the generated Alerts, indications and/or evidences of the alert accuracy shall be provided. Inaccuracies of the Alerts shall be adequately documented. Therefore, the solution shall enable the compliance analyst recording comments and/or actions taken when the Alert was a false notice.
5	Storage Limitations	Personal Data shall be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed;

5 SCOPE OF THE POC

This chapter describes the prerequisites, the procedures and the findings of the POC. Without any data from a Cantonal Bank, a list of National Councils and Councilors, supplemented by Geneva City Council Pierre Maudet was used. The reason for adding the Geneva City Councilor Pierre Maudet was, that it was the only positive fraud case with the National Councils and Councilors during the observation period. The data volume of the articles found, in particular the articles with alerts, is too low for training a decision tree or a neural network. There are several, different cases necessary. The period for collecting data was too short for that.

5.1 PROCEDURE OF THE DATA MINING FOR THE POC

The CRISP Modell (CRoss-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining) was followed to understand the data and to get insights and findings

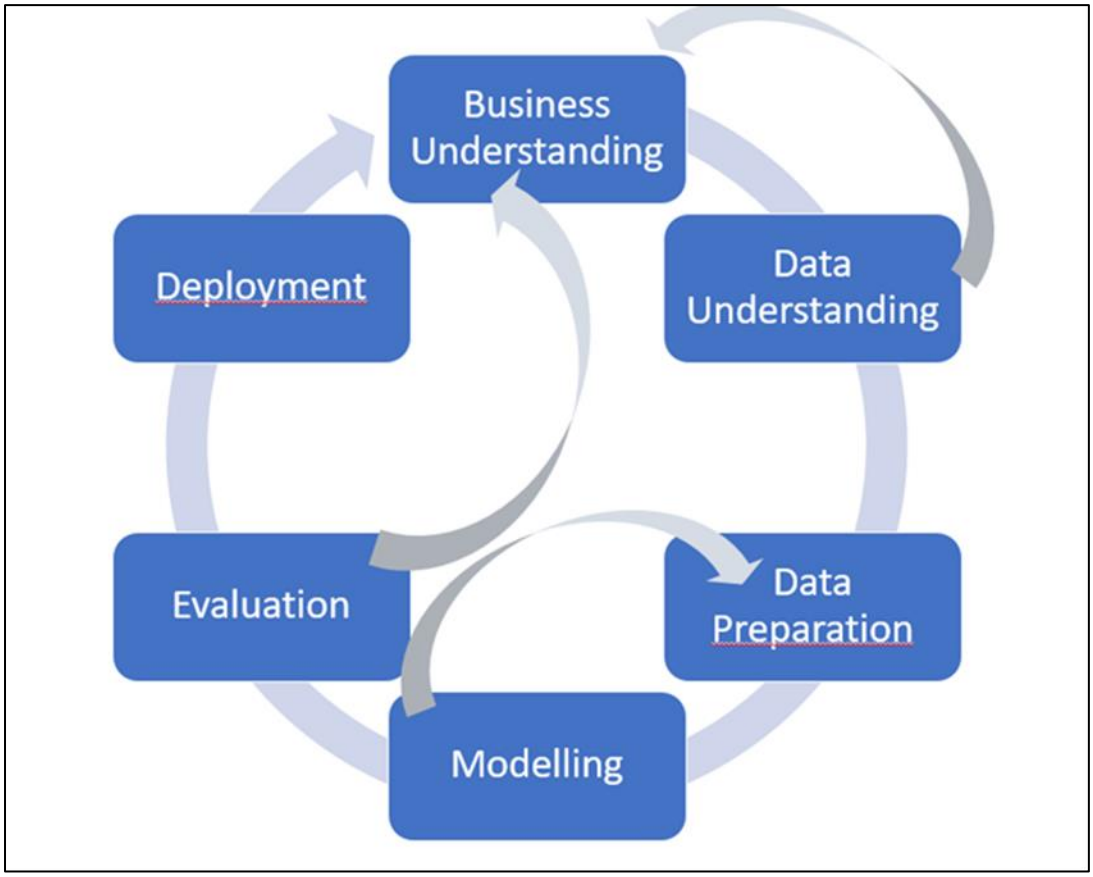


Figure 18 CRISP Model 1

- Business Understanding: Definition of the Target and requirements of the POC, Deployment of the task and the procedure

- Data Understanding: Collection of data, check data and data quality,
- Data Preparation: improve the data quality and prepare data for modelling
- Modeling Data Mining-Procedures, Optimization, check different models
- Evaluation define the model to use, with the best results.
- Deployment: Preparation and Presentation of the results

5.2 FINDINGS OF THE POC

Findings	Action
News articles have a neutral or positive sentiment	Key word list to identify the cases like Corruption
Namesakes	Identify the namesakes and exclude the articles
More than one PEP is mentioned in an article	Calculation of distance to the keyword
Duplicate articles	Exclude duplicate articles if only the web address is different
Near duplicates	Save, but exclude near duplicates
False alerts	open

6 SENTIMENT ANALYSIS

6.1 OVERVIEW SENTIMENT ANALYSIS

In this chapter an overview of the sentiment analysis, the methods of sentiment analysis and examples, in German are given.

(; ; Liu, 2012)

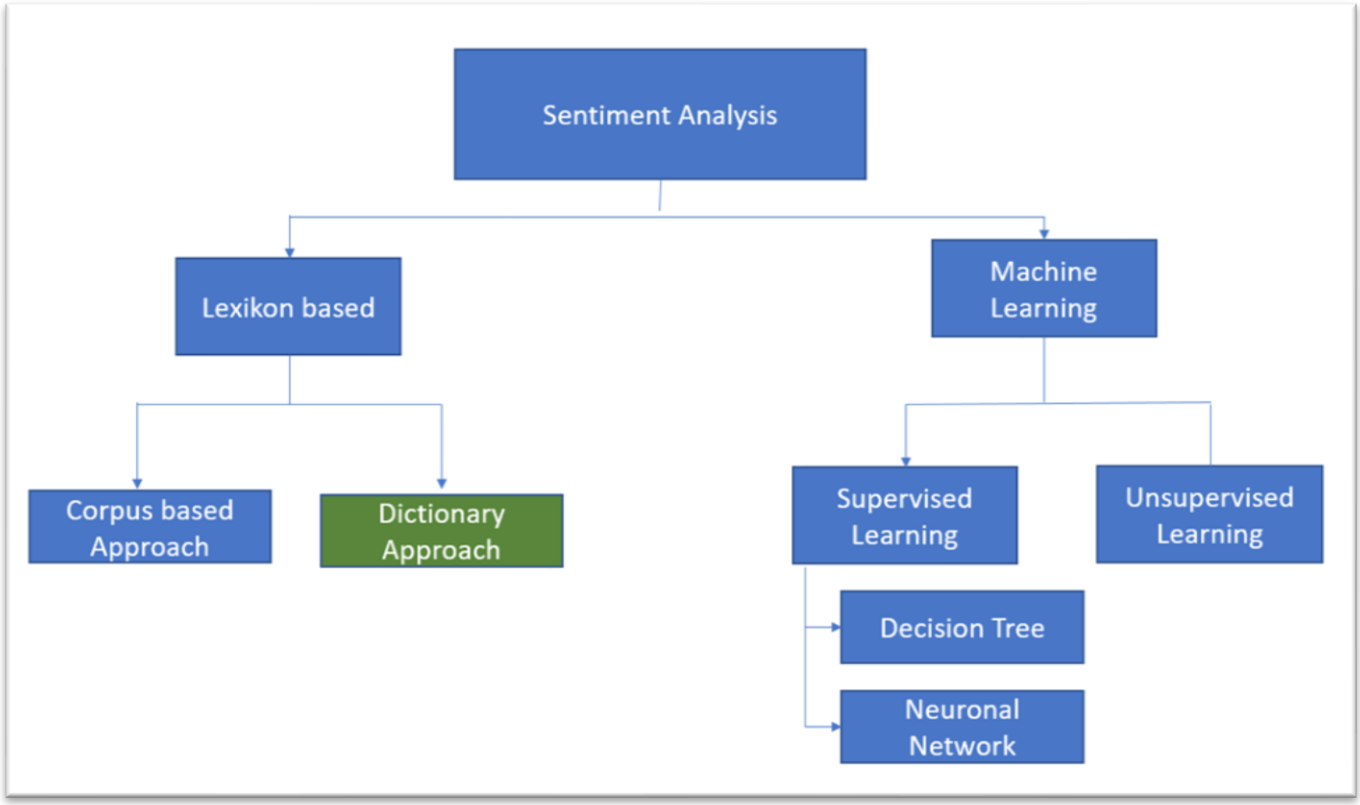


Figure 19 Sentiment Analysis Overview

6.2 METHODS OF SENTIMENT ANALYSIS

The Sentiment can be performed on words, phrases, sentences, chapters or on whole articles.

6.2.1 Lexicon Based Approach

This method assigns each word a polarity. All values of the words of the test are summed up to polarity value of the test.

A problem of the lexicon-based approach is that domain-specific words are often missing from existing dictionaries. In addition, often only base words are included. Adjectives that change the meaning / weight are not considered.

Furthermore, negations are difficult to recognize.

Examples

Negation	With the lexical approach the different conjunctions are difficult to recognize
Korruption ist ein Problem	Herr X reiste nach Saudi-Arabien
Korruption ist eine grosses und verbreitets Problem	Herr X lehnte die Reise nach Saudi-Arabien ab
Korruption ist kein Problem	Herr X war nach Saudi-Arabien gereist
Herr X ist korrupt	Herr X würde nach Saudi-Arabien reisen
Herr X ist der Korruption verdächtigt	Herr X hätte nach Saudi-Arabien reisen können

Ambiguity, irony and bipolar words are difficult to recognize and evaluate in text analysis:

«Der Skandal ist eine schöne Bescherung»

Due to time constraints the lexicon-based approach was chosen. For better results especially for the recognition of negations the creation of an individual phrase-based lexicon could be interesting.

Sentiment analysis on sentences or on chapters without consideration of negations would not get better results, because the appreciation of the separate words is the same.

6.2.2 Corpus -, Event – or Phrase Based Approach

With Corpus -, Event – or Phrase Based Approach word sequences or phrases are analyzed, so for example negations can be recognized. The approaches require the construction of a specific phrase dictionary and are therefore much more expensive.

Entity based is the evaluation of polarity related to an entity, a person or an enterprise. Event based is the investigation in terms of an event like a consumption.

Further analysis possibilities for the sentiment analysis are the use of a decision tree or a neural network. Like the lexicon-based analysis, the text can also be prepared and reduced (Stemming, Lemmatization) for machine learning, additional Features like corpus- or phrase-based approach are possible.

The result of a sentiment analysis is the value of the polarity, e.g. positive, negative, or neutral.

Other analysis tools of text analysis are word counts and word bags per text. Both can be visualized well and used for analysis.

An example is the analysis of the news about Pierre Maudet

Figure 20 Word bags example of Pierre Maudet (acceptance of benefits)

6.3 PROCEDURE IN SENTIMENT ANALYSIS

1,) Removing stop words (Stemming)
Stop words are words which are used frequently in text, however the single word only has no content or statement.
Examples of stop words are: ein, eine, einer, der, die, das und, oder, weil, während, in……

2.) Lemmatization:
Lemmatization means that words are needed in their basic form.

Example:

Läuft: Laufen	Ging: Gehen
Schneller: Schnell	Bäume: Baum

3.) Match synonyms
Replacing of synonyms with a unique word

Examples:

Gebäude=Haus	Klinik=Krankenhaus=Hospital

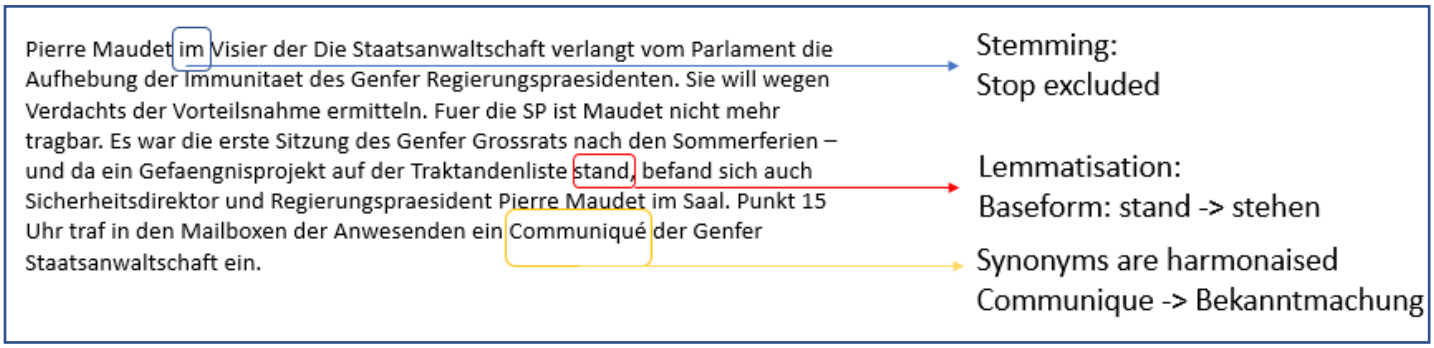


Figure 21 Sentiment Analysis example

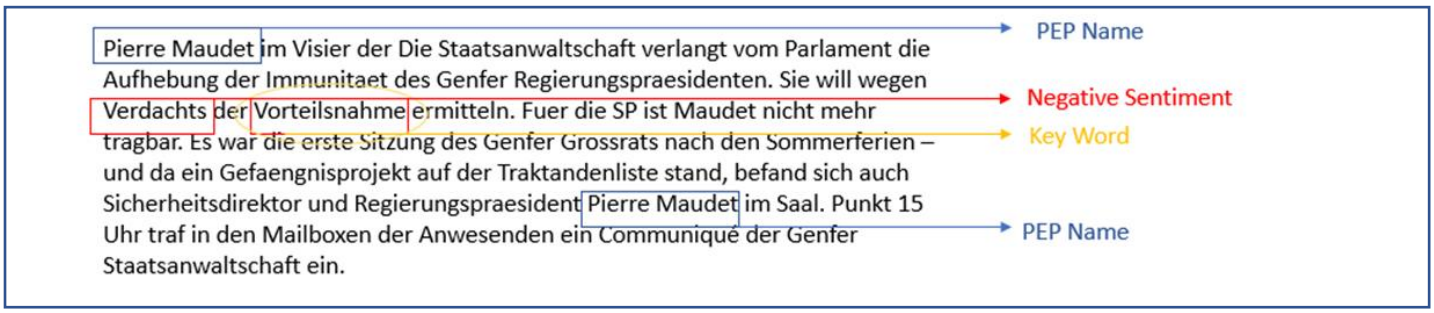


Figure 22 Sentiment Analysis example 2

6.4 EXAMPLE OF SENTIMENT ANALYSIS AND POLARITY

How the sentiment analysis is carried out is shown by the following example. One of the results of the sentiment analysis is that the news articles in total are positive. The negative effect of the words in one article is eliminated by the positive effect of the other words in the sum.

Pierre Maudet geht auf Konfr mit der Mutterpartei

Die FDP Schweiz hat den angeschlagenen Genfer Staatsrat zu eine ausserordentlichen Sitzung nach Bern aufgeboten – doch dieser w Am Ergebnis wird sich nichts ändern: Die Partei dürfte ihn offiziell auffordern.

Antonio Fumagalli, Lausanne
27.11.2018, 17:48 Uhr

Pierre Maudet hat keine Zeit für den Vorstand der FDP Schweiz. (Bild: Jean-Christophe Bott / Keystone)

In einer an Wendungen ohnehin schon reichen Politaffäre überschlagen sich derzeit die Ereignisse: Genfs Staatsrat Pierre Maudet, gegen den ein Strafverfahren wegen Verdachts auf Vorteilsannahme läuft, wird sich am Mittwoch nicht nach Bern begeben. Dies richtet seine Pressestelle auf Anfrage aus.

Damit schlägt er eine Aufforderung des Vorstandes der FDP Schweiz in den Wind. Dieser hatte sich am Montag zu einer ordentlichen Sitzung getroffen und dabei auch die Causa Maudet thematisiert. Beschlossen wurde dann, den letztjährigen Bundesratskandidaten anlässlich einer ausserordentlichen Vorstandssitzung am Mittwochnachmittag anhören zu wollen, um ihm «alle Fragen stellen zu können». Die Einladung wurde brieflich und elektronisch nach Genf übermittelt.

The words get a polarity value between -1 (negative) and +1 (positive) the neutral words are 0 The sum of all polarity values in the text is the value for the whole text. If a word is used more than once, the polarity value is counted several times. In sum, it gives a polarity for the text.

Since most of the news articles have a positive polarity, the polarity is not a good criterion for recognizing the searched texts, so a key word list was implemented. In the example below, using the keyword list, the word benefit is found, and an alert is generated.

Häufigkeit	Wort	Polarity
1	Krisensitzung	-1
1	Vorteilsannahme	-1
1	Strafverfahren	-1
1	angeschlagen	-1
1	Konfrontation	-1
1	Flexibilität	1
1	Verurteilung	-1
1	Aufforderung	1
1	Politaffäre	-1
1	beschlossen	1
1	Ereignisse	1
1	Demission	1
1	Verdacht	-1
1	Rückhalt	1
1	Ergebnis	1
1	Respekt	1
1	Lösung	1
1	Gründe	1
	Total	**0**

1.

Figure 23 example Pierre Maudet

6.5 PROBLEMS WITH DATA COLLECTION AND VALUATION

6.5.1 Namesakes

With the collection of news articles per PEP articles of namesakes of PEPs for Thomas Müller have been found, Thomas Müller is National Councilor of the FDP and his namesake is a German footballer for Bayern Munich.

So that articles of namesakes are not analyzed, and incorrect messages are displayed, they are eliminated with a filter.

Figure 24 Namesake

For the exclusion of articles related to soccer player Thomas Müller, is filtered out, if the following words are found in the article*:*

Fußball	Bayern
Bundestrainer	Nationalmannschaft
Bundesliga	

Some Other national councilors who could be mentioned in the News article have a namesake:

- Lukas Reimann, the namesake from Horb southern Germany, is active in the youth council, but hadn't any news articles during the observation period.
- Matthias Aebischer from Wünnewil was once the top scorer, also did not have any news articles during the observation period.

Not on our list of current national and councilors is former Federal Councilor Samuel Schmid who has the Schwinger Samuel Schmid as a namesake.

6.5.2 News articles with more than one Pep Person mentioned

Another problem in correctly allocating the article to the right PEP, that often several PEPS are mentioned in an article.
If several PEPS are mentioned in one article, the article is assigned to all named peps. If the article has found an alert (keyword), the keyword will be assigned to the closest PEP. In order to assign the article to the correct PEP, it is checked for the traffic light (alert) which pep is called in the sentence with the key word list word or which PEP is closest to the key word list word.

6.5.3 Treatment of wrong Alerts (Key Word is mentioned, eventually with negation)

Problematic are articles in which a PEP opposes an offense sought with the key word list.

False alerts are currently being reported in these cases.
A solution must be developed, some keywords to recognize the articles with negation are:

Bekämpfung	gegen
Kampf	Massnahmen
bekämpfen	Gesetz
Prävention	Anti

7 COMPLIANCE BUSINESS PROCESS

The workflow describes the decisions of the compliance manager in the bank

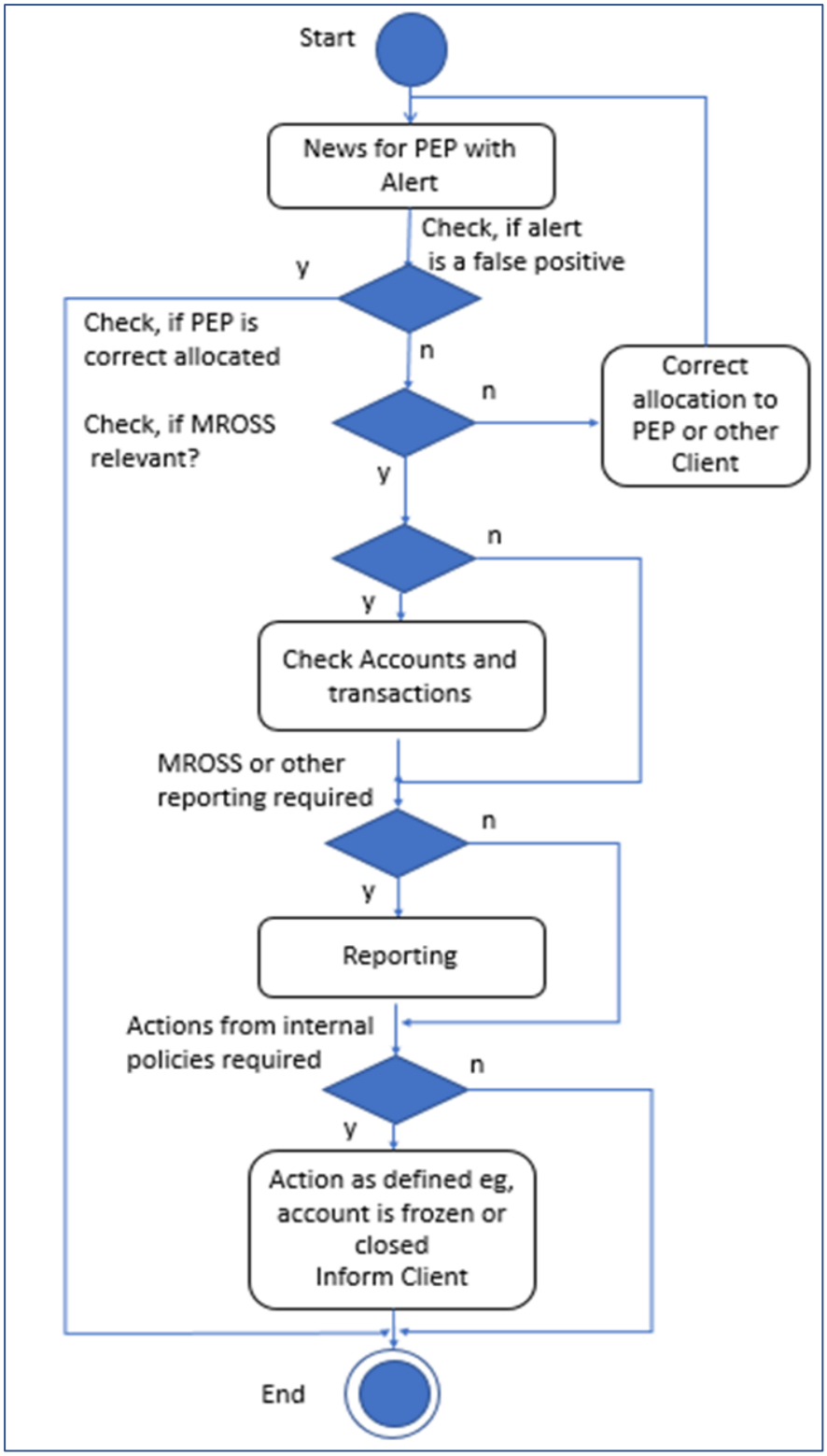


Figure 25 Compliance Business Process

8 TECHNICAL SOLUTION

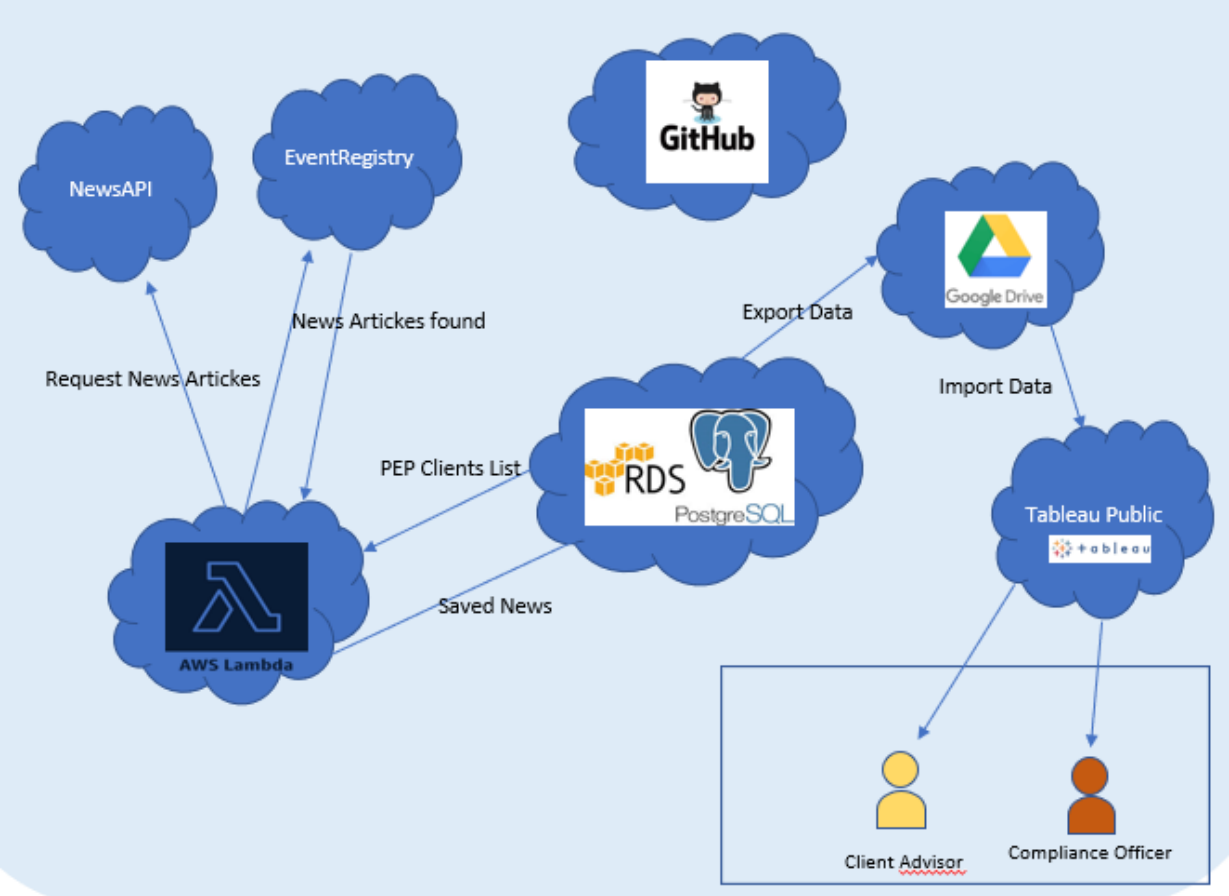


Figure 26: Over view of technical solution.

With limited resources it was important to keep the technical solution as simple as possible. There are many services available on the internet to retrieve news articles. It was decided to use one of these services instead of writing a custom web crawler, which would have been very time consuming. The project needed a server to run code for retrieving the articles and a place for a database to store those articles. The best solution for a server was to implement the project in the cloud. For the final presentation there needed to be a web site that anybody could access to view the data. Due to resources and time constraints a custom web site for the project was not developed; instead Tableau Public [13] was used to display the project results. The following sections describe in detail the project technical solution.

[13] https://public.tableau.com/en-us/s/

8.1 AWS RELATIONAL DATA BASE SERVICE (RDS)

Amazon Web Services (AWS) is basically an Infrastructure as a Service (IaaS) that offers many different Platform as a Service (PaaS) options. One of those options is their Relational Database Service (RDS):

> ***„Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need."***
>
> ***https://aws.amazon.com/rds/***

The Amazon RDS is very fast and easy to set up. With just a few mouse clicks it is possible to have a database up and running. The big advantage of RDS is it is not necessary to configure any of the database system files. A big disadvantage of RDS is it is not possible to configure any of the database system files. This made it impossible to use any custom dictionaries for full text searches. It is also not possible to create or partition table spaces on the RDS. All the database objects are stored in one default tables space:

> ***"...since all storage is on a single logical volume, tablespaces cannot be used for IO splitting or isolation. We have benchmarks and practical experience that shows that a single logical volume is the best setup for most use cases."***
> ***https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_PostgreSQL.html#PostgreSQL.Concepts.General.FeatureSupport.Tablespaces***

One of the biggest advantages of having a cloud based data base is elasticity[14]. That is the resources can quickly adapt to the workloads. Amazon RDS has many instance types[15] to choose from. The project used the smallest and cheapest instance, the t2.micro, which is one virtual CPU and 1 GB Memory. The "t" instances are burstable performance instances. When not in use the instance idles down and starts to collect credits. When needed it can "burst" up above its baseline performance and use the credits that were collected during the idle times. There are also instances optimized for memory or performance. Currently the largest instance is a db.r5.24xlarge, which has 96 virtual CPUs and 768 GB memory. Just like renting a high-performance car for an hour to drive around, it is possible to kick your instance up to whatever level your heart desires.

[14] Elasticity https://en.wikipedia.org/wiki/Elasticity_(cloud_computing)

[15] Amazon RDS Instance Types https://aws.amazon.com/rds/instance-types/

Amazon now offers a Serverless Data Base:

> ***"Amazon Aurora Serverless is an on-demand, auto-scaling configuration for Amazon Aurora (MySQL-compatible edition), where the database will automatically start up, shut down, and scale capacity up or down based on your application's needs. It enables you to run your database in the cloud without managing any database instances. It's a simple, cost-effective option for infrequent, intermittent, or unpredictable workloads."***
> ***https://aws.amazon.com/rds/aurora/serverless/***

The Aurora Serverless Postgres-compatible edition is currently in beta testing.

For future projects or for a production project, the serverless database would be a better option.

8.2 POSTGRESQL DATABASE

Reasons it was decided to use Postgres as the project database:

- Open Source no licence fees.
- Atomicity, Consistency, Isolation, Durability (ACID)[16] Compliant
- Traditional Relational Data Base Management System (RDBMS[17])
- Support for NoSQL JSON data types[18].
- Support for Full Text searches.[19]
- Special Full Text indexing: GiST and GIN.[20]
- Support for Materialized Views[21]
- Support for user defined Stored Functions

> ***"PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. The origins of PostgreSQL date back to 1986 as part of the Postgres project at the University of California at Berkeley and has more than 30 years of active development on the core platform."***
> ***https://www.postgresql.org/about/***

For writing user defined stored functions Postrges supports in addition to SQL and C a variety of procedural languages:

- PL/pgSQL[22]
- PL/Tcl[23]
- PL/Perl[24]
- PL/Python[25]

Amazon RDS does not support PL/Python because it is only available as a "untrusted" language[26].

[16] ACID https://en.wikipedia.org/wiki/ACID_(computer_science)

[17] RDBMS https://en.wikipedia.org/wiki/Relational_database_management_system

[18] Postgres JSON https://www.postgresql.org/docs/10/datatype-json.html

[19] Postgres Full Text searches https://www.postgresql.org/docs/9.5/textsearch.html

[20] Postgres GIN and Gist Indexes https://www.postgresql.org/docs/11/textsearch-indexes.html

[21] Postgres Materialized View https://www.postgresql.org/docs/9.3/rules-materializedviews.html

[22] PL/pgSQL https://www.postgresql.org/docs/9.5/plpgsql.html

[23] PL/Tcl https://www.postgresql.org/docs/9.5/pltcl.html

[24] PL/Perl https://www.postgresql.org/docs/9.5/plperl.html

[25] PL/Python https://www.postgresql.org/docs/9.5/plpython.html

[26] Untrusted PL/Python https://dba.stackexchange.com/questions/132352/why-is-pl-python-untrusted

PL/pgSQL was used to write the following user defined stored functions for the project:

- AddArticle – With a PEP_ID and an articles JSON object that was returned from EventRegistry will try to add the article to the data base. First checks if any existing articles are 80% or more similar to the new one. If a similar article is found, the new one will not be added, the PEP_ID will be associated with the existing article that is most similar to the new one. If no similar articles are found the article will be inserted into the "Art" table.
- UpdatePep_Art – Used to make sure all PEPs within an article are correctly associated with the article in the PEP_ART table along with the number of times they appear in the article and their word positions in the article.
- Find_PEP_Art - is used to find the closest PEP to an alert word within an article, at the end the "ALERTWORD_ART" table is updated with the IDs of the alert word, the PEP, and the article. The word positions of the PEP and alert word within the article is also written to the table.
- Lexeme_Occurrences – is used to find how often where a word appears within a tsvector. It returns the count of how often it appears and an integer array of its positions.

The sql code for the above functions can be seen in the appendix and in GitHub.

An ER diagram for the data base can be seen in the appendix. The full data base documentation can be seen in GitHub.

8.3 AWS LAMBDA SERVERLESS COMPUTING

The project needed a server to run the code to retrieve news articles and other jobs once or twice a day. It did not need a full time sever for these actions. The perfect solution is serverless computing in the cloud. Amazon's solution for this is called Lambda.

> ***"AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of the Amazon Web Services. It is a computing ser-vice that runs code in response to events and automatically manages the computing resources required by that code. It was introduced in November 2014.***
> ***The purpose of Lambda, as compared to AWS EC2, is to simplify building smaller, on-demand applications that are responsive to events and new information. AWS targets starting a Lambda instance within milliseconds of an event."***
> ***https://en.wikipedia.org/wiki/AWS_Lambda***

Development of the code was done on the developer's personal computer. To upload the code, all supporting libraries, and the needed configurations, the Serverless Framework was used:

> ***"The Serverless Framework is a free and open-source web framework written using Node.js. Serverless is the first framework that was originally developed for building applications exclusively on AWS Lambda, a serverless computing platform provided by Amazon as a part of the Amazon Web Services. Currently, applications developed with Serverless can be deployed to other function as a service providers, including Microsoft Azure with Azure Functions, IBM Bluemix with IBM Cloud Functions based on Apache OpenWhisk, Google Cloud using Google Cloud Functions, Oracle Cloud using Oracle, Kubeless based on Kubernetes, Spotinst and Webtask by Auth0."***
> ***https://en.wikipedia.org/wiki/Serverless_Framework***

8.4 PYTHON CODE

The Python programming language was chosen to retrieve news articles from the internet and save them to the database.

> ***"Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales."***
> ***https://en.wikipedia.org/wiki/Python_(programming_language)***

One of the strengths of Python is its very large standard library. Libraries used for this project are:

- psycopg2[27] – connecting, reading, writing to the PostgresSQL database.
- requests[28] - sending requests to and receiving data back from a Web Sites API.
- json[29] – working with json objects.
- os.environ[30] – retrieve configuration variables from environment variables.
- traceback[31] – exception handling / error messages
- logging[32] – logging errors and exceptions
- eventregistry[33] – eventregistry.org wrapper for API calls.
- bs4.BeautifulSoup[34] scrape and format html text from URL.
- textblob-de[35] German Version NLP TextBlob – Sentiment Analysis.

Python applications written for this project:

- QueryNews – Requests articles from NewsAPI.org and writes them to the database.
- QueryHTML – Used after the QueryNews was run to get / save the URL html text and calculate sentiment polarity for the text.
- QueryRegistry – Requests articles from EventRegistry.org and calls the database stored function addArticle to save the article to the database.
- QuerySentiment – Used after QueryRegistry is run. Uses TextBlob-De to assign sentiment polarity to the news articles.

[27] Psycopg2 https://pypi.org/project/psycopg2/

[28] Requests http://docs.python-requests.org/en/master/user/intro/

[29] Json https://docs.python.org/3/library/json.html

[30] Os.environ https://docs.python.org/3/library/os.html

[31] Traceback https://docs.python.org/3/library/traceback.html

[32] Logging https://docs.python.org/3/library/logging.html

[33] Eventregistry https://github.com/EventRegistry/event-registry-python/wiki

[34] beautifulSoup https://pypi.org/project/beautifulsoup4/

[35] Textblob-de https://pypi.org/project/textblob-de/

The code was written on the developer's PC and then with Serverless Framework uploaded with supporting libraries to Amazon Lambda to be run on a scheduled basis.

For future projects it is suggested to use the AWS Cloud 9[36]. It is a cloud-based integrated development environment (IDE). It makes it easy for the developer to change development PCs and to share his work with other developers, because everything is done in an internet browser window.

The code for the above applications can be seen in GitHub (see Section 9.7) and in the appendix.

8.5 DATA SOURCES

There are many internet services offering Application Programming Interfaces (APIs) to retrieve news articles. Below are the two that were used for this project

8.5.1 News API

News API[37] was chosen because they have a free developer version that at first looked like it would meet the needs of the project.

One limitation of the developer version was the limit of 1000 API calls per day. When working with it, it was discovered this was farther broken down to just 250 calls every six hours. The project's python code was written to do one API request per PEP. With over 170 PEPs the 250 limit would be reached if the code was executed more than once in a six-hour period, which for developing and testing was the case. Development and testing had to be done with a limited PEP list because of this limitation.

Another limitation of the developer version was it would not return the full text of an article. An additional python application was written to retrieve the text of the article from the URL that was returned.

The developer version limited the searches to just the last 30 days. The paid version would allow for searches up to one year in the past.

After a month of using News API, 350 articles had been downloaded from 15 different news sources. Most of the news sources had only one article and just about all the other articles came from just one source. The paid version for News API started at $500(USD) a month, which was out of the budget range for the project. A new source for the news articles had to be found.

8.5.2 Event Registry

Event Registry[38] is a powerful news media API service that bills by tokens per month. They have a 30-day trail that allocates 2000 tokens. A subscription for the project of 10000 tokens a month at

[36] AWS Cloud 9 https://aws.amazon.com/cloud9/

[37] News API https://newsapi.org/

[38] Event Registry http://eventregistry.org/

$150(USD) was started for the project. With the paid subscription it is possible to search for articles back to 2014.

Event Registry returns a wealth of meta data for each article found. Some of these were only returned for English articles, the project was only searching for German language articles, so never received meta data for example location or sentiment. For an example of the data returned for an API request see the appendix.

A total of 18,454 articles were downloaded from the Event Registry, of those 1557 had at least one of the alert words that were being searched for, that is 8.4% of the articles.

Total number of words 10,282,691, which 1,989 were one of the alert words, of those only 179 could be associated with a PEP that was with in 50 words of the alert word.

The articles came from 357 different news sources, e.g. Aargauer Zeitung, Basellandschaftliche Zeitung, Frankfurter Allgemeine, etc.

The python code that used Event Registry to retrieve and save articles can be seen in the appendix.

8.6 PRESENTATION OF DATA

Tableau Public[39] was used to show the data that was collected and processed.

To view interactive charts and graphs of the data collected, please click on the link below:

https://public.tableau.com/profile/bill.worthington#!/vizhome/shared/YHYQBJ66C

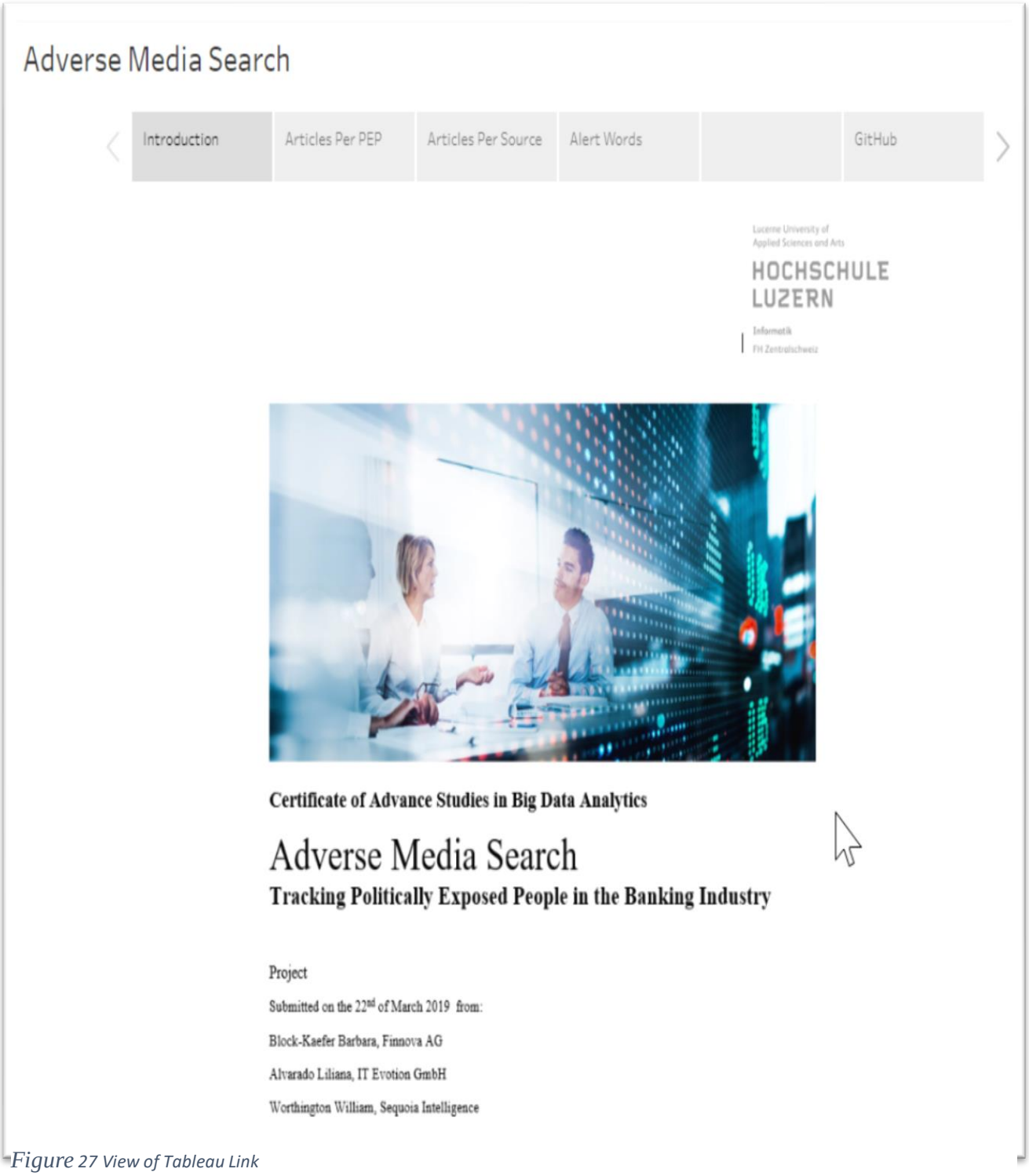


Figure 27 View of Tableau Link

[39] Tableau Public https://public.tableau.com/en-us/s/

8.7 GITHUB

Python code, data base scripts, CSV exports of data, and full data base documentation can be viewed at: https://github.com/SequoiaIntelligence/UHL

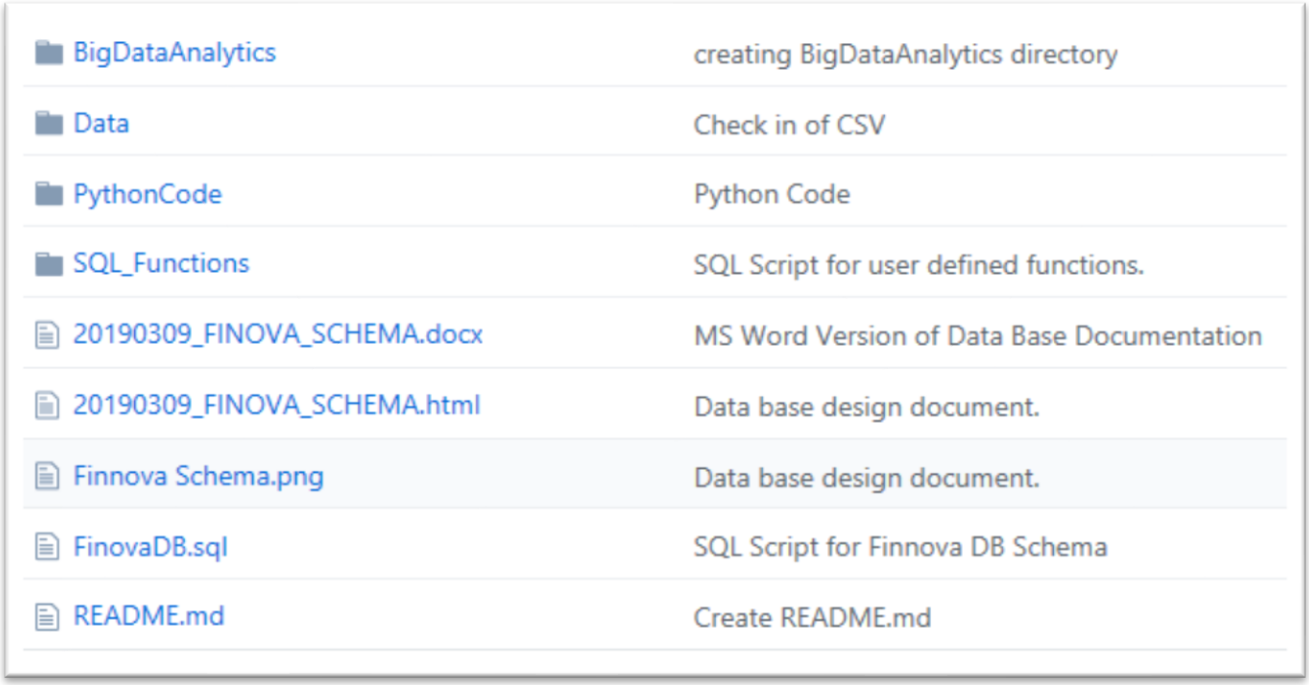


Figure 28 View of GitHub contents

9 IMPLEMENTATION IN FINNOVA

A prerequisite for the implementation is the extension of the customer CRM with the attributes PEP, direct family members and other relatives. The functionalities realized in the POC, are collecting the news on the PEP as well as the sentiment analysis. Additionally, the dashboard can then be implemented in the Finnova Core system.

Furthermore, the relations - Spouses, friends, relatives and close associates can also be included in the review.

9.1 CRM

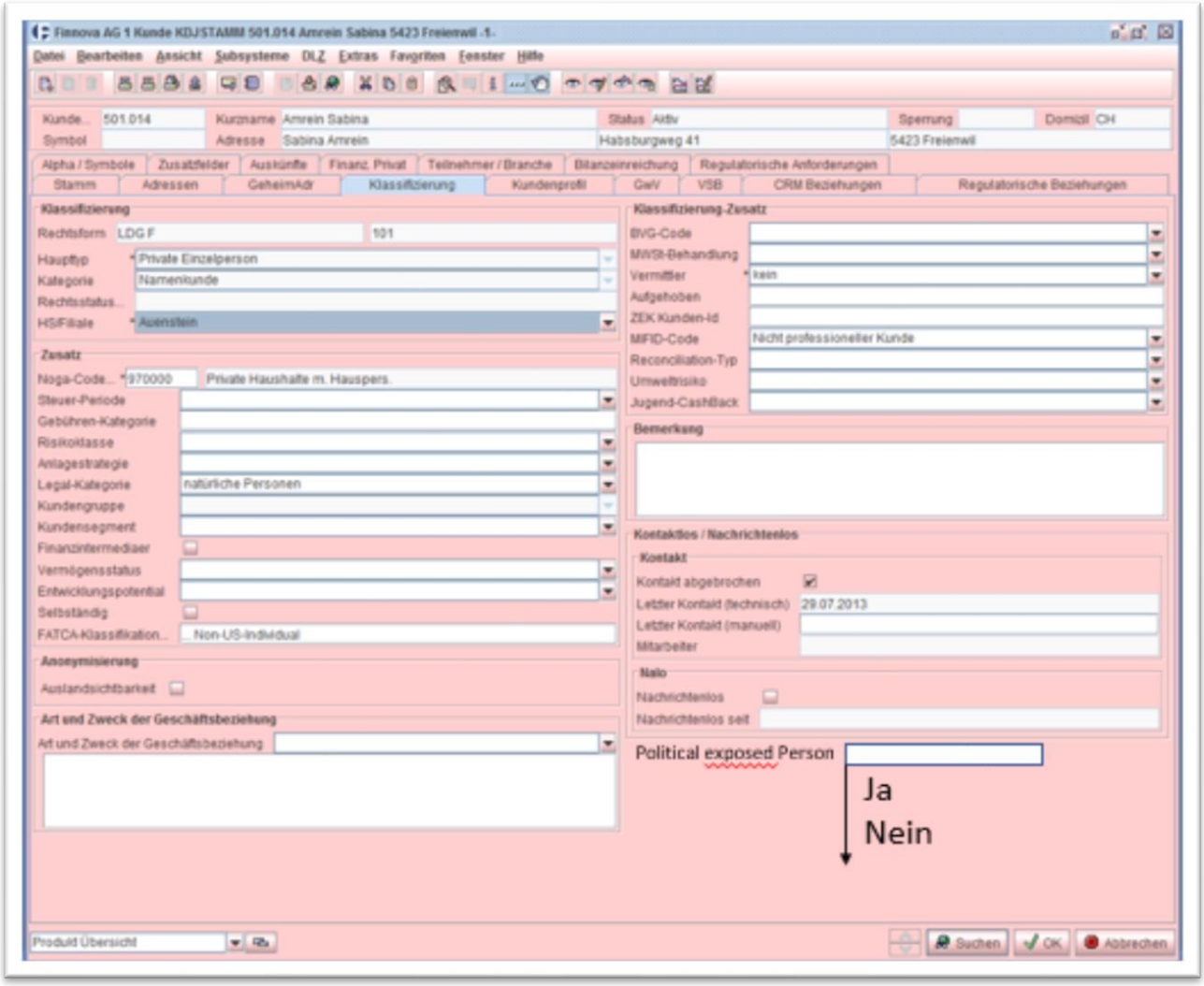


Figure 29 CRM 1

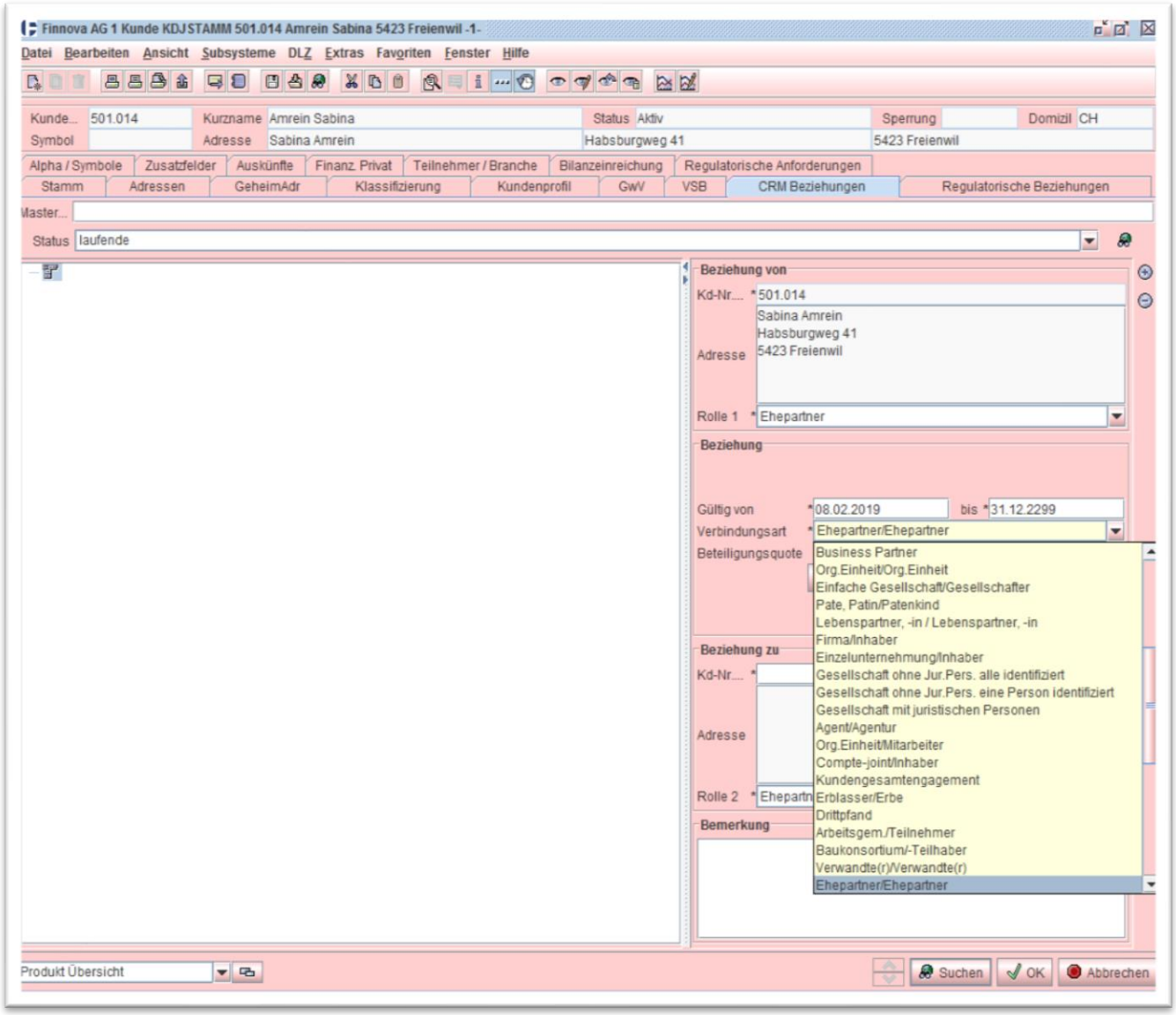


Figure 30 CRM 2

9.2 IMPLEMENTATION PLAN

Alternative 1

1.Using the lexical approach and expanding logic to avoid false alerts

2. Extension of the customer attributes with the attribute PEP

3. Extension of the customer connections to the connections with PEP

4. Installing a web crawler to collect the news

5. Sentiment analysis as described and performed in the POC

6. Detection of PEP

7. Key word list must be available, must be changed (deleted, added)
8. Exclusion list must be available, must be changed (deleted, supplemented)
9. Overview for customer advisors with the messages and alerts
10. Overview for the Compliance Manager with the messages and alerts

Alternative 2:

- Collecting data: different examples with alert are needed
- Installation of alternative 1
- Continuous cleanup of the data so that training data with results is available
- As soon as enough data is available, the neural network is trained
  and the neural network is implemented as an extension to Adverse Media Search.

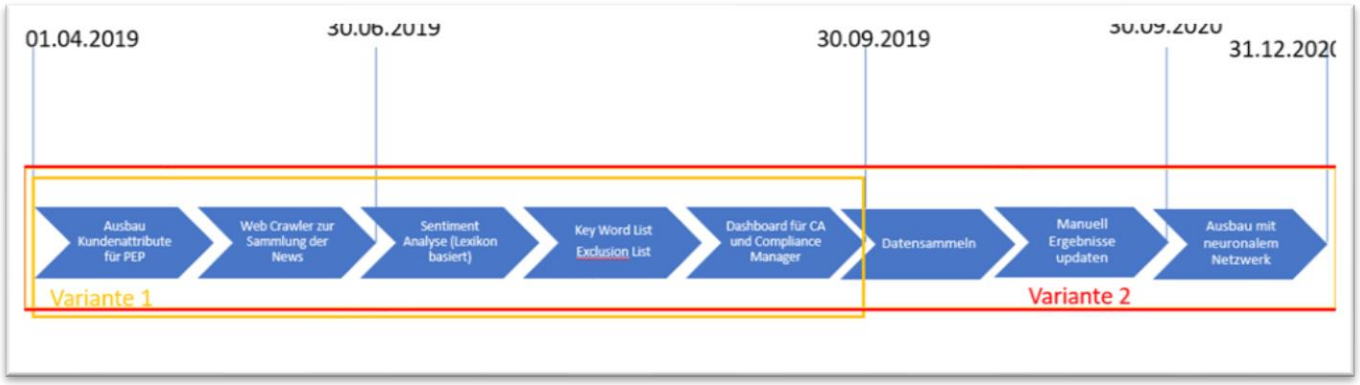


Figure 31 Implementation Plan

10 Options for Future Enhancements

In this chapter, the procedure and possible extensions as well as other uses of text analysis are shown.

The collection of texts and the detection and assignment to the PEP customers is good. The problem is the detection of news in which the PEP expresses that he is against corruption or wants to act against it. These articles generate incorrect alerts.

Since the data volume of the articles correctly identified as positive alerts is too small, a larger database must first be created. Then the data can be used to train a decision tree or neural network. Better results and avoiding false alerts can then be expected.

A big advantage of news articles is that the data is usually spelled correctly,

The procedure for training is the following:

1.) Collecting the data

2.) Evaluate the data manually to supplemental result

3.) Removal of stop words (same as in POC)

4.) Stemming (same as in POC)

5.) Train the decision tree or neural network

6.) Reporting and visualizing the results (same as in the POC)

Advantages of using a decision tree or neural network is the possibility of recognizing and evaluating whether a statement of word phrases is negative and the recognition due to the connection to the PEP this is positively or negatively connected to the message

This can be done with methods such as the Vector analysis and Navies Bayes

10.1 USAGE OF A DECISION TREE

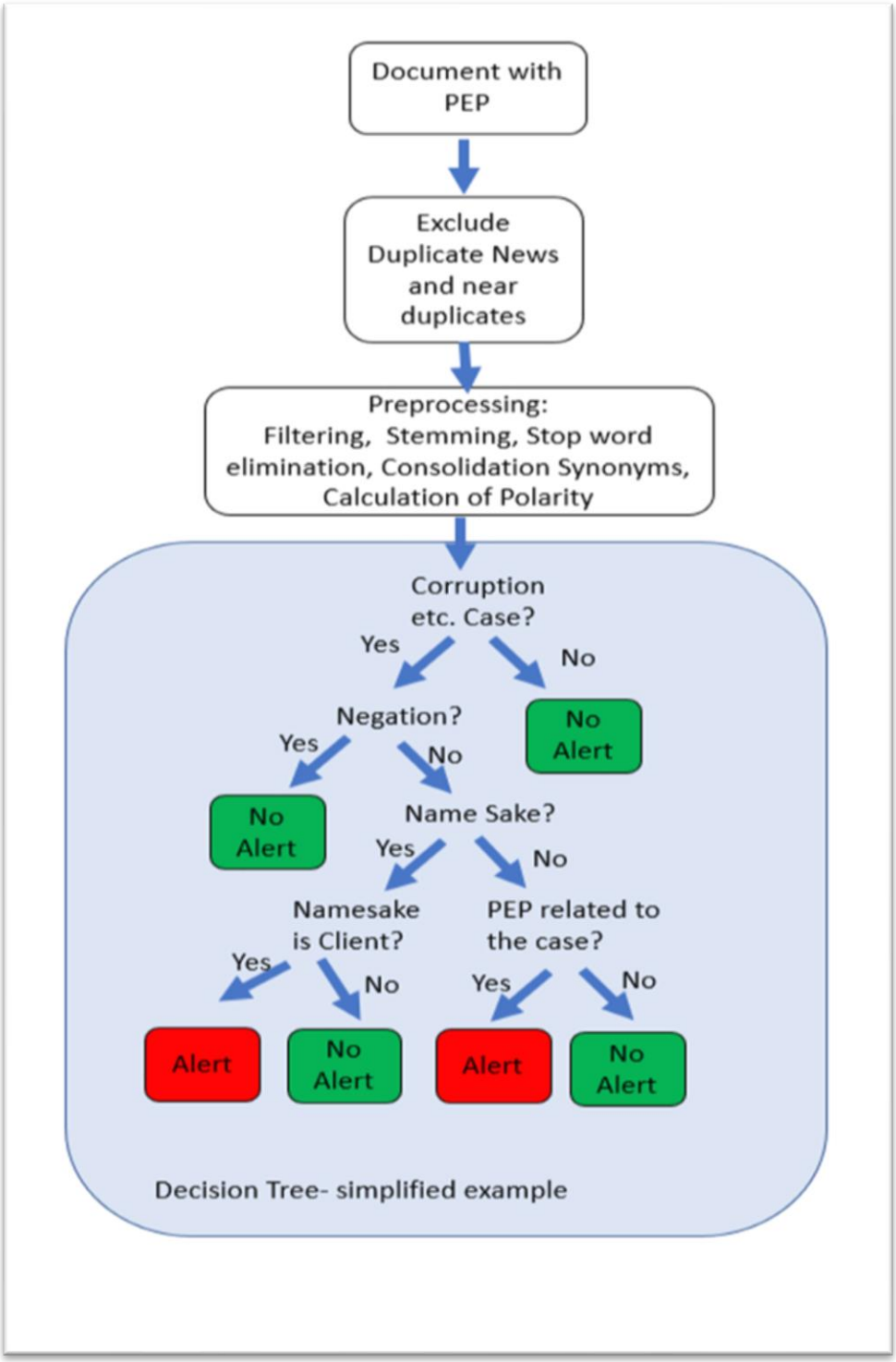


Figure 32 Decision Tree

10.2 USAGE OF A NEURONAL NETWORK

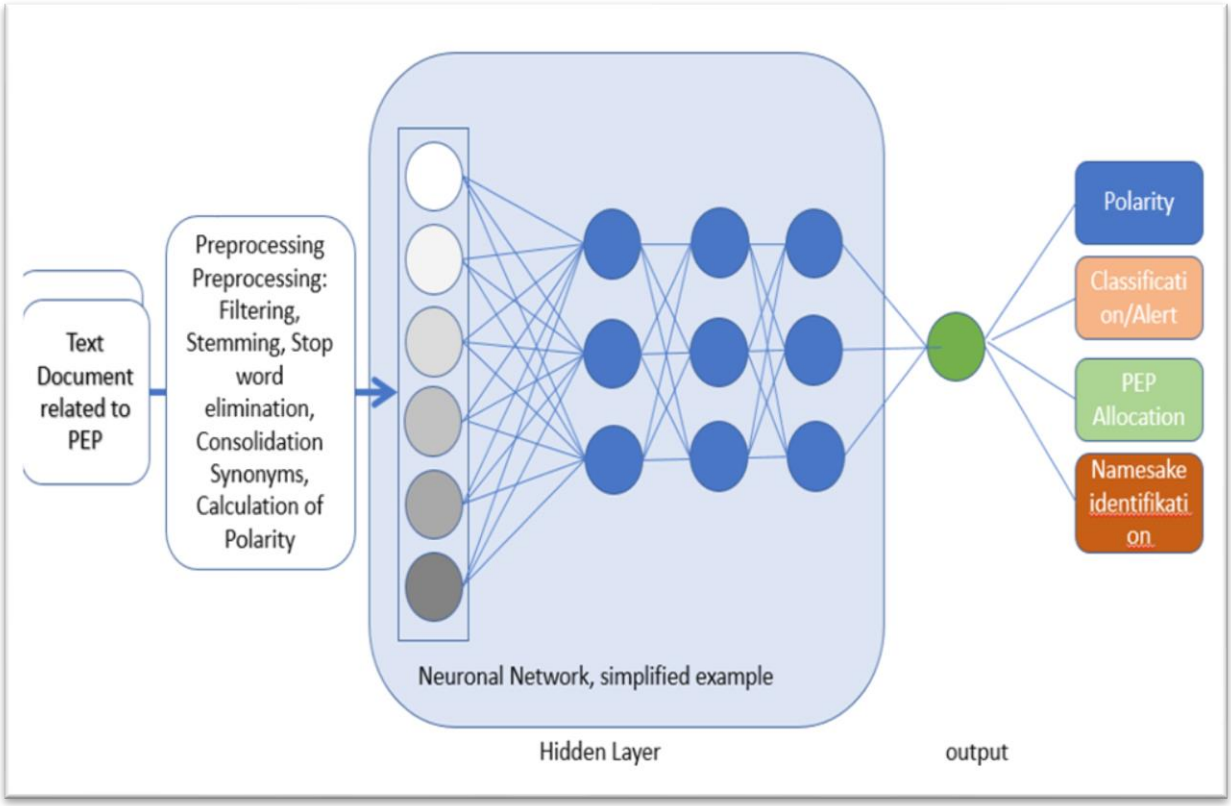


Figure 33 Neuronal Network

10.3 ADDITIONAL USE CASES/ FURTHER POSSIBLE USES

Text analysis can also be used for:

- Evaluations of customer feedback and complaints
- Collect articles to own bank
- Collecting articles on the evolution of the stock market, individual stocks, or the economy.
- Offer the solution to third party companies such as artist agencies who would like to be informed quickly about news about their clients.

In addition, the analysis of the following data can provide important, advice-relevant information about the customer and customer consultation can thus be designed individually:

- Analysis of the text in payment orders to know the customer's use of the products such as mortgages, insurance, third-party investment products.
- Recognizing the purchase of a subscription / service that implies an investment plan, such as buying a home or car.

11 RECOMMENDATION

Conclusion and Recommendation is the alternative 2 for the Adverse Media Search of the PEP, but since a lot of data is needed for training, it makes sense to first start with alternative 1, collect the data, manually check the evaluation with the lexical approach and thus good training data to collect.

Once enough data is available, the solution can be expanded with a neural network or decision tree.

12 CLOSING REMARKS

With the POC it could be shown that the articles are collected and recognized, furthermore a result is that the sentiment analysis is not suitable to recognize the corruption cases. With an expansion of the solution, the desired reports can be made available to the banks.

In the context of the POC solutions for the recognition of the searched cases, the treatment of namesakes, as well as the detection of message duplicates were developed. Expanding the solution is necessary to detect the wrong alerts.

REFERENCES

Simon Clematide, Stefan Gindl, Manfred Klenner, Stefanos Petrakis, Robert Remus, Josef Ruppenhofer, Ulli Waltinger and Michael Wiegand: 2012 MLSA - A Multi-Layered Reference Corpus for German Sentiment Analysis. https://062c576c-a-62cb3a1a-s-sites.googlegroups.com/site/iggsahome/publications/125_Paper1.pdf?attachauth=ANoY7cpomWiZqYBV2F93AvCngr0TBbTg-_fPvS9y8yih1OT7sE0FUp-EYIwKxDeZRL8FWDQxoKdu11yhA_sEGCyajTK_EVQIHiYv-RZzJhqfYuXwwvObcInfL0c01zZfO0lKrISs7msw8TrAwKF97mF5iXRH2rKfejtQolMgipnI4vRk1Pf2O1qsfXSaLDh9sJfLJme0kLczQgPNL7WwfYhdhYwkgmMU6s9S0ZbyBsz_VPf47I4h33M%3D&attredirects=0 aufgerufen am 15.01.2019

Liu, B. (2012). Sentiment Analysis and Opinion Mining. St Rafael: Morgan & Claypol.

Appendix

Appendix A ABBREVIATIONS

ACID	**Atomicity Consistency Isolation Durability**
API	**Application Programming Interface**
AWS	**Amazon Web Services**
CRISP	**CRoss-Industry Standard Process**
FAF	**Finnova Analytical Framework**
FTS	**Full Text Search**
GIN	**Genearlized Inversed Index**
GiST	**Generalized Search Tree**
IaaS	**Infrastructure as a Service**
JSON	**Java Script Object Notation**
JSONB	**Java Script Object Notation Binary**
NLP	**Natural Language Processing**
OLAP	**Online Analytical Processing**
OLTP	**Online Transaction Processing**
PaaS	**Plattform as a Service**
PEP	**Politically Exposed Person**
POC	**Proof Of Concept**
RDBMS	**Relational Databasse Management System**
RDS	**Relational Database Services (AWS)**
URI	**Uniform Ressource Identifier**
URL	**Uniform Ressource Locator**
YAML	**YAML Ain't Markup Langauge**

Appendix B EVENT REGISTRY JSON OBJECT

Request to the EventRegistry for articles returns one large JSON object that contains up to 100 articles as sub-JSON objects.

Here is an example of an article's JSON object. For readability the body text of the article is not included, the "concepts" sub JSON object is broken out and displayed below, line breaks have been added.

{"sim": 0,

"uri": "1054718434",

"url": "https://www.suedostschweiz.ch/politik/2019-02-01/nationalratskommission-gegen-frauenquote-im-bundesrat",

"wgt": 286733400,

"date": "2019-02-01",

"lang": "deu",

"time": "16:10:00",

"image": "https://www.suedostschweiz.ch/sites/default/files/styles/np8_full/public/media/2019/02/01/20190201163723180.jpg?itok=5zVk3E1W",

"title": "Nationalratskommission gegen Frauenquote im Bundesrat",

"shares": {},

"source": {"uri": "suedostschweiz.ch", "title": "suedostschweiz.ch", "dataType": "news"},

"authors": [{"uri": "agentur_sda@suedostschweiz.ch", "name": "Agentur Sda", "type": "author", "isAgency": false}],

"dataType": "news", "dateTime": "2019-02-01T16:10:00Z",

"eventUri": null,

"location": null,

"storyUri": null,

"sentiment": null,

"categories": [{"uri": "news/Politics", "wgt": 78, "label": "news/Politics"}], "isDuplicate": false}

Note: "location" was very seldom given; "eventUri", "storyUri", and "sentiment" always returned "null", this is because only German articles were being searched for. If English articles were used, then EventRegistry would have returned information for these fields.

Below is an example of an article's "concept" object. In the database it is stored in its own column as a JSONB object.

[{"uri": "http://de.wikipedia.org/wiki/Andrea_Caroni", "type": "person", "label": {"eng": "Andrea Caroni"}, "score": 5},

{"uri": "http://en.wikipedia.org/wiki/Cédric_Wermuth", "type": "person", "label": {"eng": "Cédric Wermuth"}, "score": 4},

{"uri": "http://en.wikipedia.org/wiki/Politician", "type": "wiki", "label": {"eng": "Politician"}, "score": 4},

{"uri": "http://en.wikipedia.org/wiki/Switzerland", "type": "loc", "label": {"eng": "Switzerland"}, "score": 4, "location": {"type": "country", "label": {"eng": "Switzerland"}}},

{"uri": "http://de.wikipedia.org/wiki/Offizialdelikt_(Deutschland)", "type": "wiki", "label": {"eng": "Offizialdelikt (Deutschland)"}, "score": 3},

{"uri": "http://de.wikipedia.org/wiki/Aline_Trede", "type": "person", "label": {"eng": "Aline Trede"}, "score": 3},

{"uri": "http://en.wikipedia.org/wiki/Fokus", "type": "org", "label": {"eng": "Fokus"}, "score": 3},

{"uri": "http://en.wikipedia.org/wiki/Corruption", "type": "wiki", "label": {"eng": "Corruption"}, "score": 3},

{"uri": "http://en.wikipedia.org/wiki/National_Council_(Switzerland)", "type": "wiki", "label": {"eng": "National Council (Switzerland)"}, "score": 3},

{"uri": "http://en.wikipedia.org/wiki/Strafgesetzbuch", "type": "wiki", "label": {"eng": "Strafgesetzbuch"}, "score": 3},

{"uri": "http://en.wikipedia.org/wiki/Good_governance", "type": "wiki", "label": {"eng": "Good governance"}, "score": 3},

{"uri": "http://en.wikipedia.org/wiki/Bribery", "type": "wiki", "label": {"eng": "Bribery"}, "score": 3},

{"uri": "http://en.wikipedia.org/wiki/Alliance_'90/The_Greens", "type": "org", "label": {"eng": "The Greens"}, "score": 3},

{"uri": "http://en.wikipedia.org/wiki/Parliament", "type": "wiki", "label": {"eng": "Parliament"}, "score": 3},

{"uri": "http://en.wikipedia.org/wiki/Human_rights", "type": "wiki", "label": {"eng": "Human rights"}, "score": 3},

{"uri": "http://de.wikipedia.org/wiki/Maximilian_Reimann", "type": "person", "label": {"eng": "Maximilian Reimann"}, "score": 2},

{"uri": "http://en.wikipedia.org/wiki/FIFA_World_Cup", "type": "org", "label": {"eng": "FIFA World Cup"}, "score": 2},

{"uri": "http://en.wikipedia.org/wiki/Ecology", "type": "wiki", "label": {"eng": "Ecology"}, "score": 2},

{"uri": "http://en.wikipedia.org/wiki/Qatar", "type": "loc", "label": {"eng": "Qatar"}, "score": 2, "location": {"type": "country", "label": {"eng": "Qatar"}}},

{"uri": "http://de.wikipedia.org/wiki/Roland_Rino_Büchel", "type": "person", "label": {"eng": "Roland Rino Büchel"}, "score": 1},

{"uri": "http://en.wikipedia.org/wiki/BILANZ", "type": "wiki", "label": {"eng": "BILANZ"}, "score": 1},

{"uri": "http://en.wikipedia.org/wiki/Singapore", "type": "loc", "label": {"eng": "Singapore"}, "score": 1, "location": {"type": "country", "label": {"eng": "Singapore"}}}]

APPENDIX D PYTHON CODE

Python applications written for this project:

- QueryNews – Requests articles from NewsAPI.org and writes them to the database.
- QueryHTML – Used after the QueryNews was run to get / save the URL html text and calculate sentiment polarity for the text.
- QueryRegistry – Requests articles from EventRegistry.org and calls the database stored function addArticle to save the article to the database.
- QuerySentiment – Used after QueryRegistry is run. Uses TextBlob-De to assign sentiment polarity to the news articles.

APPENDIX D.1 QUERYNEWS

```
def newsquery(event, context):
    import psycopg2
    import logging
    import traceback
    import requests
    import pprint
    import json
    from os import environ
    from datetime import datetime, timedelta

    # These parameters are loaded into the
    #  into the environment variables with the
    #  Serverless Framework YAMIL file.
    endpoint=environ.get('ENDPOINT')
    port=environ.get('PORT')
    dbuser=environ.get('DBUSER')
    password=environ.get('DBPASSWORD')
    database=environ.get('DATABASE')

    countSQL = "SELECT COUNT(*) FROM finnova.\"Articles\";"
    querySQL = "SELECT vp.\"ID\" as ID, vp.\"searchname\" as searchname "
    querySQL = querySQL + " FROM finnova.\"V_PEP\" vp WHERE vp.\"Language\" = 'd';"


    insertSQL = "INSERT INTO finnova.\"Articles\ "
    insertSQL = insertSQL +  " VALUES (DEFAULT, %s, %s) ON CONFLICT DO NOTHING;"

    qFilter = 'AND ("SP" OR "GLP" OR "CVP" OR "FDP" OR "BDP" OR "Bundesrat" OR "Bundesrätin" '
    qFilter = qFilter + ' OR "Ständerräte" OR "Bundeshauses" OR "Reierungsrat" OR "Wahl" OR '
    qFilter = qFilter + ' "Wahlen" OR "Parlament" OR "Nationalrat" OR "Sozialdemokraten")'
    qFilter = qFilter + ' AND -"Fußball" AND -"Bayern" AND -"Bundestrainer"&'

        #NewsAPI.org string elements
    https = 'https://newsapi.org/v2/everything?'
    language = 'language=de&'  # Take only German articles
        # set time span for 3 days in the past to cover any articles that were added late.
    date = 'from=' + (datetime.today() - timedelta(days=3)).strftime('%Y-%m-%d&')
        # Take the max page size of 100
    pagesize =  'pagesize=100&'
    apiKey =    'apiKey=YourNewsAPIKey'

    logger=logging.getLogger()
    logger.setLevel(logging.INFO)

    def make_connection():
        conn_str="host={0} dbname={1} user={2} password={3} port={4}".format(
            endpoint,database,dbuser,password,port)
        conn = psycopg2.connect(conn_str)
        conn.autocommit=True
        return conn


    def log_err(errmsg):
        logger.error(errmsg)
        return {"body": errmsg , "headers": {}, "statusCode": 400,
            "isBase64Encoded":"false"}

    logger.info("Cold start complete.")

    try:
```

```
    cnxQuery = make_connection()
    cursorQuery =cnxQuery.cursor()

    cnxInsert = make_connection()
    cursorInsert = cnxInsert.cursor()

    try:
        cursorQuery.execute(countSQL)
        numArticles = cursorQuery.fetchone()
        print("Start Num Articles: ", numArticles,"\n")

        # Get list of PEPs
        cursorQuery.execute(querySQL)
        for result in cursorQuery:
            # Get articles for current PEP searchname
            q = 'q=\"%s\" ' % result[1]
            q = q + qFilter
            url = (https + q + language + date + pagesize + apiKey)
            response = requests.get(url)
            if  response.json()['status'] == 'ok' :
                article_list = response.json()['articles']
                print('PEP: ' + result[1] + '   TotalResults: '
                    + str(response.json()['totalResults']) )
                for article in article_list:
                # Insert current PEP_ID and articles JSON for current search name into DB
                    cursorInsert.execute(insertSQL, (str(result[0]),
                                        json.dumps(article) ) )
            else :# Not okay, log error
                logger.error(response.json())

    except:
        return log_err ("ERROR: Cannot execute cursor.\n{}".format(
            traceback.format_exc()) )


except:
    return log_err("ERROR: Cannot connect to database from handler.\n{}".format(
        traceback.format_exc()))


finally:
    try:
        cursorQuery.execute(countSQL)
        numArticles = cursorQuery.fetchone()
        print("End Num Articles: ", numArticles,"\n")

        cursorInsert.close()
        cursorQuery.close()
        cnxInsert.close()
        cnxQuery.close()
    except:
        pass


body = {
    "message": "Go Serverless v1.0! Your function executed successfully!",
    "input": event
}
```

APPENDIX D.2 QUERYHTML

```
#  pip3 install requests bs4 -t .
#  pip install html5lib -t .

def htmlquery(event, context):
   import psycopg2
   import logging
   import traceback
   import requests
   import json
   import imp
   import sys
   sys.modules["sqlite"] = imp.new_module("sqlite")
   sys.modules["sqlite3.dbapi2"] = imp.new_module("sqlite.dbapi2")
   import nltk
   from os import environ
   from bs4 import BeautifulSoup
   from textblob_de import TextBlobDE as TextBlob

 #  nltk.download('vader_lexicon')

   endpoint=environ.get('ENDPOINT')
   port=environ.get('PORT')
   dbuser=environ.get('DBUSER')
   password=environ.get('DBPASSWORD')
   database=environ.get('DATABASE')

   countSQL = "SELECT COUNT(*) FROM finnova.\"Articles\" a "
   countSQL = contSQL + " WHERE a.\"html\" IS NULL;"

   countSensitivity = "SELECT COUNT(*) FROM finnova.\"Articles\" a "
   countSensitivity = countSensitivity + " WHERE a.\"senisitivity\" IS NULL;"

   querySQL ="SELECT a.\"ID\", (a.data ->> 'url'::TEXT) AS URL "
   querySQL = querySQL + " FROM finnova.\"Articles\" a WHERE a.\"html\" IS NULL;"

   querySensitivity = "SELECT a.\"ID\", a.\"html\" FROM finnova.\"Articles\" a"
   querySensitivity = querySensitivity + " WHERE a.\"sensitivity\" IS NULL;"

   updateSQL = "UPDATE finnova.\"Articles\" a SET \"html\" = '{0}' "
   updateSQL = updateSQL + " WHERE a.\"ID\" = {1};"

   updateSensitivity = "UPDATE finnova.\"Articles\" a SET \"sensitivity\" = '{0}',"
   updateSensitivity = updateSensitivity + " \"polarity\" = {1} WHERE a.\"ID\" = {2};"

   logger=logging.getLogger()
   logger.setLevel(logging.INFO)

   def make_connection():
      conn_str="host={0} dbname={1} user={2} password={3} port={4}".format(
         endpoint,database,dbuser,password,port)
      conn = psycopg2.connect(conn_str)
      conn.autocommit=True
      return conn


   def log_err(errmsg):
      logger.error(errmsg)
      return {"body": errmsg , "headers": {}, "statusCode": 400,
         "isBase64Encoded":"false"}
```

```
def getTextFromURL(url):
    r = requests.get(url)
    soup = BeautifulSoup(r.text, "html.parser")
    text = ' '.join(map(lambda p: p.text, soup.find_all('p')))
    text = text.replace("'", " ") # strip out single auotes
    return text

def nltk_sentiment(sentence):
    from nltk.sentiment.vader import SentimentIntensityAnalyzer
    nltk_sentiment = SentimentIntensityAnalyzer()
    score = nltk_sentiment.polarity_scores(sentence)
    return score

logger.info("Cold start complete.")

try:

    cnxQuery = make_connection()
    cursorQuery =cnxQuery.cursor()

    cnxUpdate = make_connection()
    cursorUpdate = cnxUpdate.cursor()

    try:
        # Get list of articles without html text
        cursorQuery.execute(querySQL)
        print("Start Num Articles without HtML: ", cursorQuery.rowcount,"\n")

        for article in cursorQuery:
            # Get text from URL for current article
            text = getTextFromURL(article[1])

            if  len(text) > 10  :
                sql = updateSQL.format(text, str(article[0]) )
                cursorUpdate.execute(sql)

                print("Updated HTML for article id: " +  str(article[0]))

        # Get list of articles without a senistivity score
        cursorQuery.execute(querySensitivity)
        print("Start Num Articles without sensitivity score: ",
            cursorQuery.rowcount,"\n")

        for article in cursorQuery:
            # Get text from URL for current article
            blob = TextBlob(str(article[1]))
            polarity = blob.polarity
            print (polarity)
            nltk_results = nltk_sentiment(article[1])
            sql = updateSensitivity.format(json.dumps(nltk_results),
                                polarity, str(article[0]) )
            cursorUpdate.execute(sql)

            print("Updated sensitivity code for article id: " +
                str(article[0]) + " Poloarity: " + str(blob.polarity) )
    except:
        print ("Error: Article_ID " +  str(article[0]))
        print (sql)
        return log_err ("ERROR: Cannot execute cursor.\n{}".format(
            traceback.format_exc()) )
```

```
except:
    return log_err("ERROR: Cannot connect to database from handler.\n{}".
                format(traceback.format_exc()))


finally:
    try:
        cursorQuery.execute(countSQL)
        numArticles = cursorQuery.fetchone()
        print("End Num Articles without HTML: ", numArticles,"\n")

        cursorQuery.execute(countSensitivity)
        numArticles = cursorQuery.fetchone()
        print("End Num Articles without sensitivity code: ", numArticles,"\n")

        cursorUpdate.close()
        cursorQuery.close()
    except:
        pass


body = {
    "message": "Go Serverless v1.0! Your function executed successfully!",
    "input": event
}

response = {
    "statusCode": 200,
    "body": json.dumps(body)
}

return response
```

APPENDIX D.3 QUERYREGISTRY

```
from eventregistry import *
import psycopg2
import logging
import traceback
import requests
import pprint
import json
import eventregistry
from os import environ
from datetime import datetime, timedelta

endpoint=environ.get('ENDPOINT')
port=environ.get('PORT')
dbuser=environ.get('DBUSER')
password=environ.get('DBPASSWORD')
database=environ.get('DATABASE')

countSQL = "SELECT COUNT(*) FROM finnova.\"Art\";"
querySQL = "SELECT vp.\"ID\" as ID, vp.\"searchname\" as searchname "
querySQL = querySQL + "FROM finnova.\"V_PEP\" vp;"
addArticleSQL ="SELECT * FROM finnova.\"addArticle\"(%s, %s);"

logger=logging.getLogger()
logger.setLevel(logging.INFO)

def make_connection():
    conn_str="host={0} dbname={1} user={2} password={3} port={4}".format(
        endpoint,database,dbuser,password,port)
    conn = psycopg2.connect(conn_str)
    conn.autocommit=True
    return conn


def log_err(errmsg):
    logger.error(errmsg)
    return {"body": errmsg , "headers": {}, "statusCode": 400,
        "isBase64Encoded":"false"}

logger.info("Cold start complete.")

try:

    cnxQuery = make_connection()
    cursorQuery =cnxQuery.cursor()

    cnxInsert = make_connection()
    cursorInsert = cnxInsert.cursor()

    try:

        er =  EventRegistry(apiKey = "YourAPIKey")

        cursorQuery.execute(countSQL)
        numArticles = cursorQuery.fetchone()
        print("Start Num Articles: ", numArticles,"\n")

        # Get list of PEPs
        cursorQuery.execute(querySQL)
        for result in cursorQuery:
            # Get articles for current PEP searchname
```

```
cq1 = ComplexArticleQuery(
  CombinedQuery.AND([
    BaseQuery(dateStart = "2019-01-01",
        # dateEnd = "2014-04-01",
         lang = 'deu'),
    CombinedQuery.OR([
      BaseQuery(keyword = QueryItems.OR(["Volkspartei",
                        "Grünen",
                        "CVP", "SVP",
                        "SP", "GLP", "FDP",
                        "BDP", "Bundesrat" ,
                        "Bundesrätin",
                        "Ständerräte",
                        "Bundeshauses",
                        "Reierungsrat",
                        "Nationalrat",
                        "Sozialdemokraten" ]) )
    ]),
    BaseQuery(keyword = result[1],
        exclude= BaseQuery(keyword =
            QueryItems.OR(["Füßball",
                        "Bundestrainer",
                        "Bundesliga",
                        "Bayern",
                        "Nationalmannschaft"])))
  ])
)
q = QueryArticlesIter.initWithComplexQuery(cq1)
q.setRequestedResult(RequestArticlesInfo(
  returnInfo = ReturnInfo(
  articleInfo = ArticleInfoFlags(basicInfo = True,
                socialScore = True,
                storyUri = True,
                eventUri = True,
                categories = True,
                location = True,
                image = True, concepts = True))))

res = er.execQuery(q)

art_page = res["articles"]["page"]
art_pages = res["articles"]["pages"]
art_count = res["articles"]["count"]
art_totalResults = res["articles"]["totalResults"]

print('PEP_ID: ' + str(result[0]) + ' Page: ' + str(art_page) +
    ' of ' + str(art_pages) + ' Count: ' + str(art_count) +
    ' of ' + str(art_totalResults) + ' Total Results' )
# Loop through articles found for the current PEP and add them to the database.
for indx in range(art_count):
  # print ("Article %d" % (indx))
  art = res["articles"]["results"][indx]
  artdump = json.dumps(art)
  cursorInsert.execute(addArticleSQL, (result[0],   artdump)  )
  messg = cursorInsert.fetchone()
  #pprint.pprint (messg)

cursorQuery.execute(countSQL)
numArticles = cursorQuery.fetchone()
print("End Num Articles: ", numArticles, "\n")
```

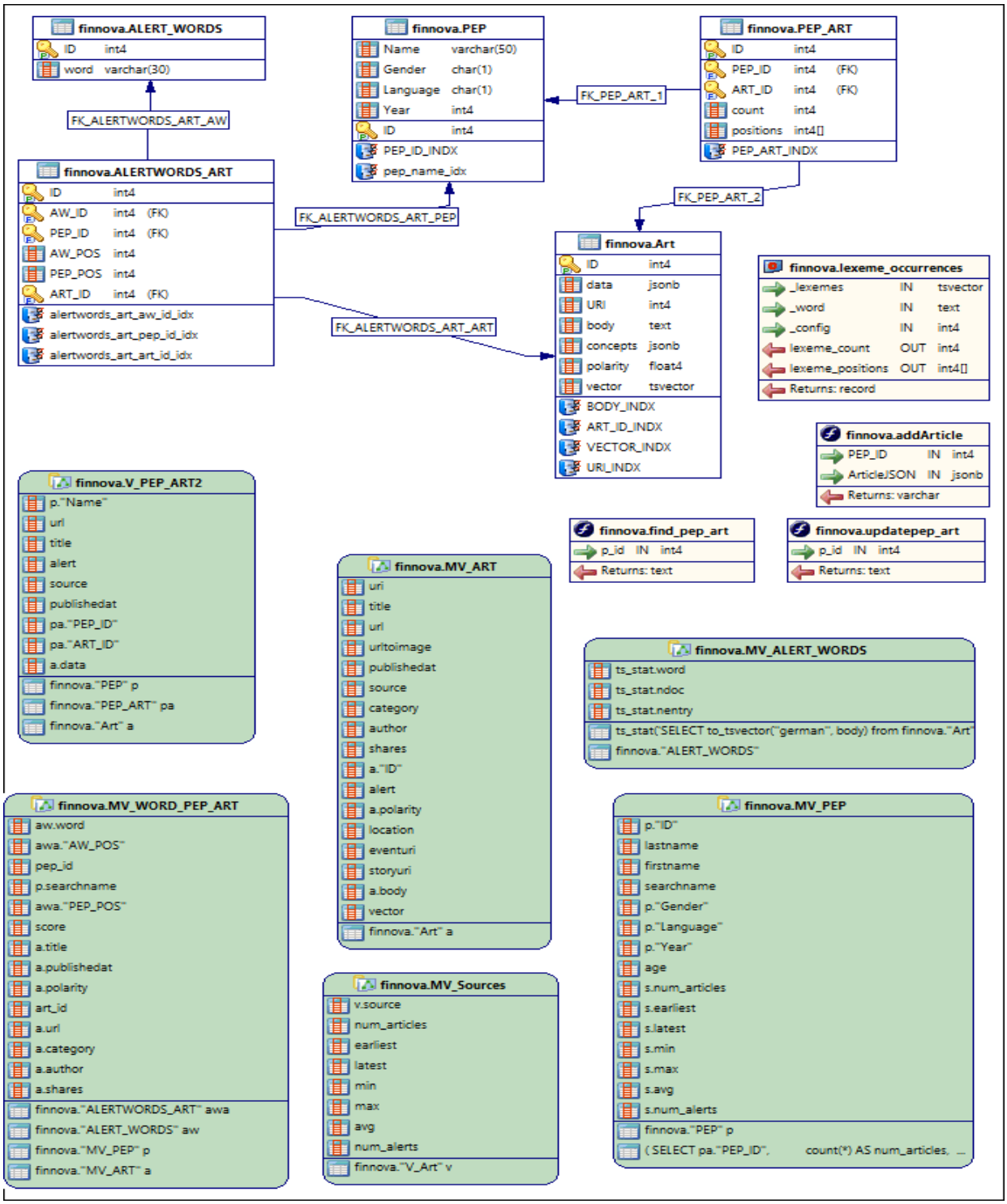
```
    except:
        print ("ERROR: Cannot execute cursor.\n{}".format(
            traceback.format_exc()) )

except:
    print("ERROR: Cannot connect to database from handler.")
    print(format_exc())

finally:
    try:
        cursorQuery.execute(countSQL)
        numArticles = cursorQuery.fetchone()
        print("End Num Articles: ", numArticles, "\n")

        cursorInsert.close()
        cursorQuery.close()
        cnxInsert.close()
        cnxQuery.close()
    except:
        pass
```

APPENDIX D.4 QUERYSENTIMENT

```
#   pip3 install requests bs4 -t .
#  pip install html5lib -t .


import psycopg2
import logging
import traceback
import json
import imp
import sys
sys.modules["sqlite"] = imp.new_module("sqlite")
sys.modules["sqlite3.dbapi2"] = imp.new_module("sqlite.dbapi2")
from os import environ

from textblob_de import TextBlobDE as TextBlob

 #  nltk.download('vader_lexicon')
endpoint=environ.get('ENDPOINT')
port=environ.get('PORT')
dbuser=environ.get('DBUSER')
password=environ.get('DBPASSWORD')
database=environ.get('DATABASE')

countSQL = "SELECT COUNT(*) FROM finnova.\"Art\" a WHERE a.\"polarity\" IS NULL;"
queryPolarity = "SELECT a.\"ID\", a.\"body\" FROM finnova.\"Art\" a WHERE a.\"polarity\" IS NULL;"
updatePolarity = "UPDATE finnova.\"Art\" a SET \"polarity\" = {0} WHERE a.\"ID\" = {1};"

logger=logging.getLogger()
logger.setLevel(logging.INFO)

def make_connection():
   conn_str="host={0} dbname={1} user={2} password={3} port={4}".format(
      endpoint,database,dbuser,password,port)
   conn = psycopg2.connect(conn_str)
   conn.autocommit=True
   return conn


def log_err(errmsg):
   logger.error(errmsg)
   return {"body": errmsg , "headers": {}, "statusCode": 400,
      "isBase64Encoded":"false"}


logger.info("Cold start complete.")

try:

   cnxQuery = make_connection()
   cursorQuery =cnxQuery.cursor()

   cnxUpdate = make_connection()
   cursorUpdate = cnxUpdate.cursor()

   try:
      # Get list of articles without a polarity score
      cursorQuery.execute(queryPolarity)
      print("Start Num Articles without a polarity score: ", cursorQuery.rowcount,"\n")

      for article in cursorQuery:
```

```
            # Get text body for current article
            blob = TextBlob(str(article[1]))
            polarity = blob.polarity
            print (polarity)
            sql = updatePolarity.format(polarity, str(article[0]) )
            cursorUpdate.execute(sql)

            print("Updated polarity code for article id: " +  str(article[0]) + " Poloarity: " + str(blob.polarity) )
        except:
            print ("Error: Article_ID " +  str(article[0]))
            print (sql)
            log_err ("ERROR: Cannot execute cursor.\n{}".format(
                traceback.format_exc()) )


    except:
        log_err("ERROR: Cannot connect to database from handler.\n{}".format(
            traceback.format_exc()))

    finally:
        try:
            cursorQuery.execute(countSQL)
            numArticles = cursorQuery.fetchone()
            print("End Num Articles without polarity: ", numArticles,"\n")

            cursorUpdate.close()
            cursorQuery.close()
        except:
            pass
```

APPENDIX E DATA BASE DESIGN



APPENDIX F DATA BASE FUNCTIONS

PL/pgSQL was used to write the following user defined stored functions for the project:

- AddArticle – With a PEP_ID and an articles JSON object that was returned from EventRegistry will try to add the article to the data base. First checks if any existing articles are 80% or more similar to the new one. If a similar article is found, the new one will not be added, the PEP_ID will be associated with the existing article that is most similar to the new one. If no similar articles are found the article will be inserted into the "Art" table.
- UpdatePep_Art – Used to make sure all PEPs within an article are correctly associated with the article in the PEP_ART table along with the number of times they appear in the article and their word positions in the article.
- Find_PEP_Art - is used to find the closest PEP to an alert word within an article, at the end the "ALERTWORD_ART" table is updated with the IDs of the alert word, the PEP, and the article. The word positions of the PEP and alert word within the article is also written to the table.
- Lexeme_Occurrences – is used to find how often and where a word appears within a tsvector. It returns the count of how often it appears and an integer array of its positions.

APPENDIX F.1 ADDARTICLE

```
CREATE OR REPLACE FUNCTION finnova."addArticle"("PEP_ID" integer, "ArticleJSON" jsonb)
 RETURNS character varying
 LANGUAGE plpgsql
AS $function$
DECLARE
    retMessg      CHARACTER VARYING;
    errMessg      CHARACTER VARYING;
    art_id        INTEGER;
    pep_art_id    INTEGER;
    uriInt        INTEGER;
    countDups     INTEGER;
    jsonURI       TEXT;
    jsonBody      TEXT;
    data          JSONB;
    concepts      JSONB;
BEGIN
    uriInt = ("ArticleJSON" ->> 'uri'::TEXT);
    jsonBody = ("ArticleJSON" ->> 'body'::TEXT);
    concepts = ("ArticleJSON" ->> 'concepts'::TEXT);
    retMessg = "ArticleJSON" -> 'title';

    -- see if any existing articles with the current PEP in the them are
    -- more than 80% similar to this new one,
    -- if so, consider it a duplicate
    -- get the art_id of the most similar article

     SELECT s."ID"
     INTO art_id
     FROM (
          SELECT "ID", similarity(body, jsonBody)
          FROM finnova."Art" a
          WHERE a."body" LIKE (SELECT '%' || lastname || '%'
                               FROM finnova."MV_PEP"
                               -- only check articles that have the current PEP in them.
                               WHERE "ID" = "PEP_ID")
          ORDER BY similarity DESC -- order results descending to find the most similar article
     ) s
     WHERE similarity > 0.80
     LIMIT 1;      -- take the top, most similar article_id

    IF (art_id IS NOT NULL) THEN
          retMessg = 'DUPLICATE: ' || retMessg;
    ELSE -- New article to be added.

         -- full original JSON object is too large for indexing
         -- remove the 'body' and 'concepts' elements to reduce its size.
         -- those elements will get saved in their own columns
         SELECT "ArticleJSON"::JSONB - 'body' - 'concepts'
           INTO data;

         INSERT INTO finnova."Art" ("URI", "body", "concepts", "data")
              VALUES (uriInt::INTEGER, jsonBody::TEXT, concepts::JSONB, data::JSONB);

         -- find the art_id of the article that was just inserted
         SELECT a."ID"
           INTO art_id
           FROM finnova."Art" a
          WHERE a."URI" = uriInt;
     END IF;
```

```
    -- Make an entry in the PEP_ART table for the pep and article
    -- if it was an article that was already in the database
    -- there is a chance that there is already an entry for it and the pep
    -- so on conflict do nothing.
    INSERT INTO finnova."PEP_ART" ("PEP_ID", "ART_ID")
          VALUES ("PEP_ID", art_id)
    ON CONFLICT DO NOTHING;

    RETURN retMessg;
EXCEPTION
    WHEN OTHERS
    THEN
        RAISE INFO 'Error Name:%', SQLERRM;
        RAISE INFO 'Error State:%', SQLSTATE;
        RETURN 'Error Will Robinson! ' || SQLERRM;
END;
$function$
;
```

APPENDIX F.2 UPDATEPEP_ART

```
CREATE OR REPLACE FUNCTION finnova.updatepep_art(p_id integer)
 RETURNS text
 LANGUAGE plpgsql
AS $function$
DECLARE
   pep                  RECORD;
   art_id               INTEGER;
   b_art_id             INTEGER;
   pep_id               INTEGER;
   lastname             TEXT;
   lexeme_count         INTEGER;
   lexeme_positions     INTEGER ||;
   _lexemes             tsvector;
   _searched_lexeme     tsvector;
   _occurences_pattern  TEXT;
   _occurences_list     TEXT;
BEGIN
   FOR art_id IN    SELECT a."ID" AS art_id
                     FROM finnova."Art" a
                    WHERE a."ID" > p_id
                 ORDER BY art_id
   LOOP
      RAISE NOTICE 'Art_ID = % ', art_id;

      -- For every Article ID
      -- See if PEP_ID exists for it in PEP_ART
      FOR pep IN    SELECT p."ID" AS pep_id, p."lastname", p."searchname"
                     FROM finnova."MV_PEP" p
                 ORDER BY pep_id
      LOOP
         --
         IF NOT EXISTS
               (SELECT *
                  FROM finnova."PEP_ART" pa
                 WHERE pa."ART_ID" = art_id AND pa."PEP_ID" = pep.pep_id)
         THEN
            -- The current PEP_ID is not associated with the current ART_ID
            -- in the PEP_ART Table
            -- See if the current PEP name is in this article

            IF EXISTS
                  (SELECT *
                     FROM finnova."Art" a
                    WHERE      a."ID" = art_id
                          AND a.body LIKE ('%' || pep.searchname || '%'))
            THEN
               -- PEP exists in this article
               -- add a new PEP_ART entry for it

               INSERT INTO finnova."PEP_ART" ("PEP_ID", "ART_ID")
                     VALUES (pep.pep_id, art_id);

               RAISE NOTICE
               'SearchName = % Art_ID = % PEP_ID = %',
               pep.searchname, art_id, pep.pep_id;
            END IF;                                  -- PEP exists in this article
         END IF;                        -- IF NOT EXISTS PEP_ID and ART_ID in PEP_ART

        -- Now update the count of the current PEP  for the current article
        -- in the PEP_ART table.
```

```
        SELECT a.vector
          INTO _lexemes
          FROM finnova."Art" a
         WHERE "ID" = art_id;

        -- If the last name starts with 'de ', just take the part after the 'de '
        -- Else try to split hyphinated names and just take the first part before
        -- the hyphination, if it is not hypinated it will take the whole thing.
        SELECT
          CASE
               WHEN substr(pep.lastname::text, 1, 3) = 'de '::text
               THEN substr(pep.lastname::text, 3)
               ELSE split_part(pep.lastname::text, '-', 1)
          END
          INTO lastname;

        _searched_lexeme := strip (to_tsvector ('german', lastname));
        _occurences_pattern := _searched_lexeme::TEXT || ':([0-9,|+)';
        _occurences_list := substring (_lexemes::TEXT, _occurences_pattern);

        SELECT count (a), array_agg (a::INT)
          FROM regexp_split_to_table (_occurences_list, ',') a
         WHERE _searched_lexeme::TEXT != ''          -- preventing false positives
          INTO lexeme_count, lexeme_positions;

        UPDATE finnova."PEP_ART" pa
           SET count = lexeme_count, positions = lexeme_positions
         WHERE pa."ART_ID" = art_id AND pa."PEP_ID" = pep.pep_id;

    END LOOP;                                                    -- through PEPs


    END LOOP;                                                    -- through Articles

    RETURN 'All done!';
END;
$function$
```

APPENDIX F.3 FIND_PEP_ART

```
CREATE OR REPLACE FUNCTION finnova.find_pep_art(p_id integer)
 RETURNS text
 LANGUAGE plpgsql
AS $function$
DECLARE
   aword                RECORD;
   pep                  RECORD;
   pepart               RECORD;
   art                  RECORD;
   list                 RECORD;
   art_id               INTEGER;
   b_art_id             INTEGER;
   pep_id               INTEGER;
   closest_pep          INTEGER;
   closest_pos          INTEGER;
   aword_count          INTEGER;
   aword_positions      INTEGER [];
   search_name          TEXT;
   _lexemes             tsvector;
   _searched_lexeme     tsvector;
   _occurences_pattern  TEXT;
   _occurences_list     TEXT;
BEGIN
   -- get list of articles that have alert words in them
   FOR art IN    SELECT a."ID", a."vector"
                   FROM finnova."MV_ART" a
                  WHERE a."ID" > p_id AND a."alert" = TRUE
               ORDER BY art_id
   LOOP
      RAISE NOTICE 'Art_ID = % ', art."ID";

      -- For every Article ID
      -- Check every alert word
      FOR aword IN SELECT *
                     FROM finnova."ALERT_WORDS" aw
      LOOP
         --
         SELECT finnova.lexeme_occurrences (art.vector, aword."word", 'german')
           INTO list;

         _lexemes := art.vector;
         _searched_lexeme := strip (to_tsvector ('german', aword."word"));
         _occurences_pattern := _searched_lexeme::TEXT || ':([0-9,]+)';
         _occurences_list := substring (_lexemes::TEXT, _occurences_pattern);

         SELECT count (a), array_agg (a::INT)
           FROM regexp_split_to_table (_occurences_list, ',') a
          WHERE _searched_lexeme::TEXT != ''       -- preventing false positives
           INTO aword_count, aword_positions;


         FOR counter IN 1 .. aword_count
         LOOP
           -- For the current alert word and its current position
           -- find the nearest PEP and its position.

          closest_pep := 0;   -- should end up with the ID of the closest PEP
          closest_pos := 0;   -- should end up with the position of the closest PEP

          -- Check all the PEPs that are associated with this article
```

```
            FOR pepart IN SELECT *
                        FROM finnova."PEP_ART"
                        WHERE "ART_ID" = art."ID"
            LOOP
                FOR pep_pos IN 1..pepart."count" -- loop t
                LOOP
                    RAISE NOTICE 'PEP POS : %', pepart."positions"[pep_pos];
                    IF pepart."positions"[pep_pos] < aword_positions[counter]
                    AND pepart."positions"[pep_pos] > closest_pos
                    THEN
                        -- if the current pep position is before the alert word position
                        -- and it is after the last found closest position
                        -- update the closest postion and closest pep
                        closest_pep := pepart."PEP_ID";
                        closest_pos := pepart."positions"[pep_pos];
                    END IF;
                END LOOP; -- positions of pep in article

            END LOOP; -- pep_arts

             IF closest_pep > 0 THEN
                -- if a pep was found
                -- Add an entry in the ALERTWORDS_ART table
                -- for the values found above.

                INSERT INTO finnova."ALERTWORDS_ART"
                ("AW_ID", "PEP_ID", "AW_POS", "PEP_POS", "ART_ID")
                VALUES (aword."ID", closest_pep, aword_positions[counter],
                        closest_pos, art."ID");
            END IF;

        END LOOP;   -- positions of alert words
    END LOOP;   -- alert words loop
END LOOP;       -- article loop

RETURN 'All done!';
END;
$function$
```

APPENDIX F.4 LEXEME_OCCURRENCES

```
CREATE OR REPLACE FUNCTION finnova.lexeme_occurrences
(_lexemes tsvector, _word text, _config regconfig,
OUT lexeme_count integer, OUT lexeme_positions integer[])
 RETURNS record
 LANGUAGE plpgsql
AS $function$DECLARE
--  _lexemes          tsvector := to_tsvector (_config, _document);
    _searched_lexeme       tsvector :=  strip (to_tsvector (_config, _word));
    _occurences_pattern    TEXT := _searched_lexeme::TEXT || ':([0-9,]+)';
    _occurences_list       TEXT
        := substring (_lexemes::TEXT, _occurences_pattern);
BEGIN
    SELECT count (a), array_agg (a::INT)
      FROM regexp_split_to_table (_occurences_list, ',') a
     WHERE _searched_lexeme::TEXT != ''             -- preventing false positives
      INTO lexeme_count, lexeme_positions;

    RETURN;
END
$function$
```

Declaration:
We hereby declare that we have completed the present work resp. the performance stated by us independently, without the assistance of third parties and only by using the specified sources authored resp.

Place, Date: ……

Signature: ……

Barbara Block

Place, Date: ……

Signature: ……

Liliana Alvarado

Place, Date: ……

Signature: ……

William Worthington