Instituto Tecnológico
y de Estudios Superiores de Occidente

Recognition of official validity of higher-level studies according to secretarial agreement 15018, published in the Official Gazette of the Federation of November 29, 1976.

Electronic, Systems and Computing Department

Master's Degree in Computer Systems

Recommendation System for Issues Found in R&D

RECEPTIONAL WORK TO OBTAIN THE MASTER'S DEGREE

IN COMPUTER SYSTEMS

Presents: **DIEGO PRECIADO BARÓN**

Adviser **DR. ALBERTO DE OBESO ORENDAIN**

Tlaquepaque, Jalisco. 16-July-2021.

Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

Sistemas de Recomendación para Problemas encontrados en Investigación y Desarrollo

---

TRABAJO RECEPCIONAL que para obtener el **GRADO** de

MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: **DIEGO PRECIADO BARÓN**

Asesor **DR. ALBERTO DE OBESO ORENDAIN**

Tlaquepaque, Jalisco. 16 de Julio de 2021.

ACKNOWLEDGMENTS

First, I would like to thank all the teachers who taught me and share with me their knowledge and experience through these years of study, especially to Alberto de Obeso who advised me on this project.

In addition, I am grateful for the financial support received from CONACYT, and ITESO for all the administrative and financial support that facilitated the achievement of my master's degree and my final project.

AGRADECIMIENTOS

Primero, me gustaría agradecer a todos los profesores que me enseñaron y compartieron su conocimiento y experiencia durante estos años de studio, en especial a Alberto de Obeso que me asesoró en este proyecto.

Además, agradezco el apoyo económico recibido por parte de CONACYT, **y** al ITESO por todo el apoyo tanto administrativo como económico que facilitó la consecución de mi maestría y mi proyecto final.

DEDICATION

I dedicate this project to my wife and family for their support during these two years of study. They are the foundation of my life for personal and professional development that allows me to contribute the best of me to the society.

DEDICATORIA

Dedico este proyecto a mi esposa y a mi familia por su apoyo durante estos dos años de maestría. Son la base de mi vida para mi desarrollo personal y profesional que me permite aportar lo mejor de mi a la sociedad.

SUMMARY

This project begins by the necessity to get insights and summarize a big number of issues found in the Software development phase for automotive modules. Part of this knowledge can be obtained based on past issues together with its own solutions. Due to technological growth, it is becoming easier to gather this information in different formats and process it.

This information grows every day, which is normally found in text format. Reading big quantities of text by a single or multiple persons to extract insights and visualize important facts, knowing that exponential growth of the information is unfeasible and probably impossible task to complete in an efficient way.

Through the new AI and Big Data technologies, it is possible to fulfill these objectives, especially with the Natural Language Process techniques from IA and the SQL and noSQL databases that will ease the analysis and process of this project.

This work consists mainly of six parts. First, there is a brief introduction to the project explaining with more clarity the background and objectives to aim, then the state-of-the-art is described related to those project objectives and it is followed by the theoretical explanation of the most important technical concepts.

Once defined the introductory part, we describe the project process where all the steps and route to follow are shown to explain the obtained results, and based on that, define our conclusions.

RESUMEN

Este proyecto nace a partir de la necesidad de encontrar conocimiento y resumir la gran cantidad de problemas descubiertos en las fases de desarrollo de Software para módulos automotrices. Parte de ese conocimiento se puede obtener con base en los problemas del pasado en conjunto con sus propias soluciones. Con el crecimiento de la tecnología es mucho más factible recopilar toda esta información en diferentes formatos y procesarla.

Esta información crece día con día, la cual se encuentra principalmente en forma de texto. Leer grandes cantidades de texto por una persona o incluso un conjunto de personas, para extraer información y visualizar datos importantes a la par de ese crecimiento de información es una tarea poco práctica o casi imposible de realizar de manera eficiente.

A través de las nuevas tecnologías de IA y Big Data, nos es posible cumplir con estos objetivos. En especial, las técnicas de Procesamiento Natural del Lenguaje por parte de IA y las bases de datos tanto SQL como noSQL nos facilitarán el análisis y proceso en nuestro proyecto.

Este trabajo consta principalmente de seis partes. Primero se dará una breve introducción al proyecto explicando con mayor claridad los antecedentes y los objetivos a alcanzar. Se describe después el estado-del-arte relacionado a los objetivos del proyecto y se prosigue con dar una explicación teórica de los conceptos técnicos más importantes.

Una vez definida la parte introductoria, se comenzará a describir el proceso del proyecto, donde se muestran los distintos pasos y la ruta a seguir para poder mostrar los resultados obtenidos y con base en eso definir las conclusiones.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

NLP	Natural Language Processing
ECR	Engineering Change Request
SOM	Self-Organized Map
TF-IDF	Term Frequency – Inverse Document Frequency
LDA	Latent Dirichlet Allocation
ARTM	Additive Regularization for Topic Modeling
SW	Software
HW	Hardware
LSA	Latent Semantic Analysis
IR	Information Retrieval
R&D	Research and Development
IMS	Integrity Management System
CSV	Comma Separated Value
EDA	Exploratory Data Analysis
KPI	Key Performance Indicator

1. INTRODUCTION

Summary: *In this chapter we briefly present the background of the project, the justification from the automotive parts manufacturing standpoint and we also define the problem to cover throughout this project.*

1.1. Background

Massimo Pacella, Antonio Grieco, and Marzia Blaco throughout their paper *On the Use of Self-Organizing Map for Text Clustering in Engineering Change Process Analysis* [1] presented an investigation concemed with a similar problem described in this project. They worked with many Engineering Change Requests (ECR) trying to find a solution that can provide useful insights. In their organization, as in any other, documents of this kind have relevant information related to problems faced in the past like solution proposals and lessons learned, however, the number of issues and solutions keep increasing in the development and production phases, and then stored in one or many databases, but barely consulted. Therefore, they proposed the use of Self-Organizing Map (SOM) as an unsupervised algorithm for text clustering that will be better explained in the upcoming sections.

Anastasia Ianina and Konstantin Vorontsov have a very interesting approach in two different papers, in the first one [2] they proposed Topic Modeling as a technique for exploratory search of relevant documents by long text queries. They did not implement Topic Modeling by itself but together with additive regularization which gets better results than other techniques like full-text search TF-IDF. In the second paper, they use the same technique plus a topical hierarchy approach that focuses on subtopics gradually discarding unnecessary information [3].

1.2. Rationale

An automotive part manufacturing company is dedicated to developing and producing embedded systems for multiple vehicle's functionalities. There is a standard procedure for the life cycle of the product but as with any other development, issues and errors are prone to occur at any phase. They could be related to software, hardware, mechanical design, etc.

When an error is found in the process, it dictates (in simple words) to track, analyze, and solve it, which depending on the results it may be converted into an engineering change. Any change request or problem report is not expected from the first design of the product, so an update to the product must be implemented to fix the issue. Depending on the severity of the issue an engineering change could take several days to be implemented from its first step of the process. Here we describe a general and brief process for an issue fix:

1. Analysis of the causes of the issue.
2. Identify the problem.
3. Propose a solution.
4. Test a new solution.
5. Deploy change into production.

The paper *On the Use of Self-Organizing Map for Text Clustering in Engineering Change Process Analysis* [1] describes that in modern industry, the development of complex products involves engineering changes that frequently require redesigning or altering the products or their components. In an engineering change process, Engineering Change Requests (ECRs) are documents (forms) with parts written in natural language describing a suggested enhancement or a problem with a product or a component. ECRs initiate the change process and promote discussions within an organization to help to determine the impact of a

change and the best possible solution. Although ECRs can contain important details, that is, recurring problems or examples of good practice repeated across several projects, they are often stored but not consulted, missing important opportunities to learn from previous projects. This paper explores the use of Self-Organizing Map (SOM) to the problem of unsupervised clustering of ECR texts. A case study is presented in which ECRs collected during the engineering change process of a railways industry are analyzed. The results show that SOM text clustering has a good potential to improve overall knowledge reuse and exploitation [1], it describes that about 35% of manufacturing resources are focused on fixing changes, from engineering drawing, manufacturing plans and scheduling requirements.

During the process of a project, the cost generated due to an error will vary depending on the current development stage. Finding and solving a problem at earlier stages may cause just an extra time rework by some developers but if this same problem is found when the product is already in the last line of production or even worse, already in the hands of a customer, this could cause a big impact to the company and its resources.

1.3. Problem Description

The automotive companies have plenty of information related to problems and solutions identified in the development phase of all their products. This information is typically found as text, hence, analyzing the complete database and any register from any product by a single person or even a group of people, can result in a very difficult task that could lead to a poor analysis output.

Currently there is no tool that could help to learn from the past mistakes in the development phase to provide potential issues due to the integration of a specific part number, software change or less likely, propose a solution from a similar problem identified years before.

1.4. Hypothesis

There is no hypothesis for this project.

1.5. Project Goals

1.5.1. General Objective

Based on stored information about problems, analysis and solutions that occurred in previous stages of any project, the objective of this work is threefold: (i) analyze and describe a set of issues stored in the SW database; (ii) create a recommendation system able to suggest solutions to new problems found, and (iii) help managers to prevent possible mistakes by acting proactively.

1.5.2. Specific Objectives

- Provide information insights about the common issues presented in the life cycle of a product from the SW standpoint through a topic model.
- With a given problem, perform a semantic search for similar problems from the database and provide their respective solutions. (Note: As I'm not longer employee of the automotive company, this section will have only a general explanation about the process, and it won't show any example of the obtained results since the data is confidential and no longer available).

- Visualize through a graph database the interaction of problems, causes, solutions and other important features from an issue's databases.

1.6. Original Scientific Outcome

Many companies have many databases and search engines, but they only operate on a keyword frequency level with small queries. Furthermore, there does not exist any tool or methodology within the company which performs a qualitative analysis for a large corpus. Topic modeling can provide many useful features for management operations focusing on analysis and semantic search.

2. STATE OF THE ART

Summary: *This chapter presents some articles related to the state-of-the-art of Natural Language Processing and recommendation systems. NLP has a broad range of applications; however, this project focuses especially on topic modeling and semantic search. Regarding recommendation systems, content-based and knowledge-based are the models to be applied.*

2.1. Text Clustering

As explained in the first section, a similar approach of this project is described in the paper of Massimo Pacella, Antonio Grieco, and Marzia Blaco [1]. Even though their proposal is not the state-of-the-art of NLP techniques, they consider a similar goal as this project.

They focus on change management in an engineering process, meaning all changes to a product design regarding fixing a quality problem or to meet new customer requirements. However, once the implementation of a change is completed, engineering change requests are no longer consulted. If there were a method to review all these documents, it could improve the quality and design of any product.

For this purpose, they make use of the machine learning method Self-Organizing Maps to create a cluster of documents (the engineering change requests). This method was originally proposed by Kohonen [4] that maps M-dimensional input vectors into 2-dimensional neurons. In summary, the main idea is to cluster all requests based on their similarity and map them into a bidimensional array for consulting and visualization.

One of the advantages of using this method is that the similarity of the texts is preserved in the spatial organization of the neurons. This can help as a visualization tool because, for example, it can highlight any document that is very different from others. On the other hand, SOM has some limitations that we want to avoid, that is, the similarity method can be obtained by calculating the cosine distance (or any other metric) between two bag-of-words or TF-IDF vectors, keeping out the semantic relationship that could exist between two documents.

2.2. Semantic and Exploratory Search

Research has found that Keyword-based has limitations in terms of Information Retrieval (IR), so more research has been conducted. One improvement lies in the field of semantic search that generates better results when also searching with longer queries. Minuri Rajapaksha and Thushari Silva proposed a hybrid model which combines LDA, community detection method and collaborative filtering.

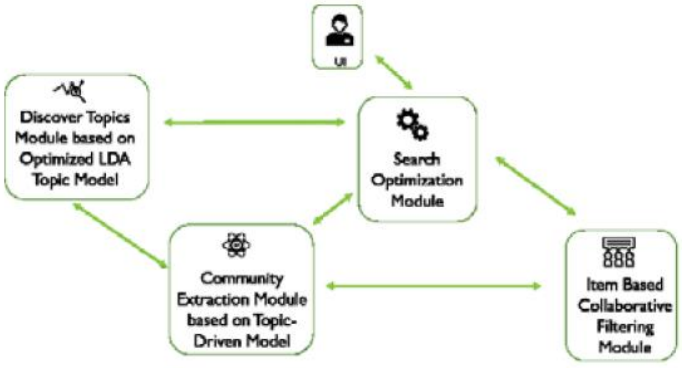

Figure 1 Topic modeling for search diagram

A similar idea is conducted in the paper of Anastasia Ianina and Konstantin Vorontsov [2], they use topic modeling for exploratory search but using a multi-objective approach called additive regularization of

topic models (ARTM) to have a highly sparse vector space of the documents to make the inverted index as compressed as possible. ARTM learns models with desired properties by maximizing a weighted sum of the log-likelihood and additional regularization criteria. In their results, these methods significantly improve the model and the precision of the exploratory search.

They also created another similar methodology for this kind of search called hierarchical topic-based exploratory search [3]. By having a long text query, the system learns its topic vector in the same way as it was done in the document collection (as we obtained LDA as explained in development section) through the ARTM framework applying a hierarchical approach on which a topic model divides into subtopics recursively, then, the system ranks documents vectors by their similarity to the query and presents top k-results to the user.

2.3. Vector representations of Words, Documents and Topics

Regarding deep learning models, Word2Vec, the Efficient Estimation of Word Representations in Vector Space [5] introduced by Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean is the state-of-the-art to get the syntactic and semantic properties of words. They proposed two architectures for learning distributed representations of words that try to minimize computational complexity, the first one is the continuous bag-of-words which predicts the current word based on its context, and the second one, the continuous skip-gram model which predicts surrounding words given the current word.

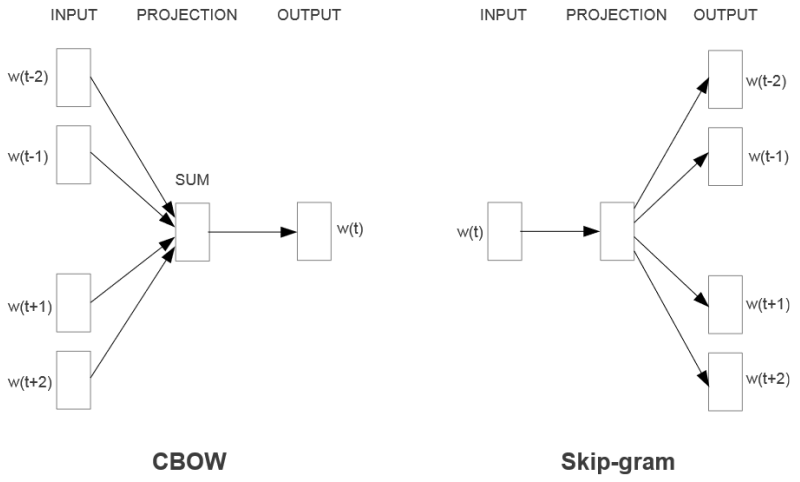

Figure 2 CBOW vs Skip-gram

In the paper *Distributed Representations of Sentences and Documents*, Quoc Le and Tomas Mikolov proposed the Paragraph Vector, an unsupervised framework that learns continuous distributed vector representations of text that can be variable-length like sentences or documents [6]. This approach overcomes the limitations of the common fixed-length feature representations like bag-of-words.

Finally, Top2vec presented by Dimo Angelov [7] leverages joint document and word semantic embedding to find topic vectors. This model will be described in more detail in the next section since it is the main approach taken on this project.

3. CONCEPTUAL FRAMEWORK

Summary: *This chapter presents the theoretical and conceptual bases about Topic modeling and Exploratory search for the recommendation system and knowledge graph.*

3.1. Natural Language Processing

The field of linguistics includes subfields that concern themselves with different levels or aspects of the language structure, as well as subfields dedicated to studying how linguistic structure interacts with human cognition and society. There are subfields under the natural language that can be represented by phonetics, phonology, morphology, syntax, semantics, and pragmatics. At each of those levels of linguistic structure, linguistics finds systematic patterns over enumerable units where both the units and the patterns have both similarities and differences across the language. [8]

From the subfields enumerated above we will make use mainly of two: morphology and semantics. Morphology is the way a word can change its form. For example, play, played, playing, plays. Before applying any model, it is important to have a preprocessing dedicated to convert all words to its base.

Syntactic level is the way how the language is structured. Although this subfield is important to understand text, the models applied to this project do not use it. Instead, to have a better performance we will focus on semantics, specifically semantic similarity for the engine search. Semantic similarity is based on the actual meaning of words, and it can be determined using hyponymy and meronymy. One simple example is that a car is a type of vehicle, the same as a bicycle so in this case car and bicycle can be considered similar.

3.2. NLP Pipeline

The following flow chart represents a common NLP pipeline described in the book *Practical Natural Language* [9]. These steps may vary in some respects between some other representations however the main fields are covered.

1. Data acquisition: Where the data is taken from? Is it taken from a public dataset found on the internet? Is the data size enough or Data Augmentation techniques should be applied to have a more extensive data set?
2. Text cleaning: Extract raw text from the input data by removing non-textual information. It takes any data source such as PDF, HTML, Excel sheets etc. Techniques like spelling correction can be applied on this stage.
3. Pre-processing:
    a. Sentence segmentation and word tokenization: Separate text into sentence or text into words only.
    b. Stop word removal: Eliminate words like a, an, the, of, etc. that do not carry any content on their own.
    c. Stemming: Removes suffixes and reduces a word to some base (e.g., cars -> car).
    d. Lemmatization: Maps all different forms of a word to its base word, called lemma (e.g., better -> good).
    e. Other common steps: Removing digits, lowercasing, language detection.

4. Feature engineering: Also known as feature extraction or in NLP tasks, text representation. The main idea is to convert the raw text or data into a format that can be consumed by a model.

5. Modeling: Depending on the data size and approach this step can vary. For instance, it's always better to start a heuristic approach before jumping to a machine learning one. In our case, the topic modeling called LDA and Top2Vec are used as main models.
6. Evaluation: To measure how good the model is to unseen data. For topic modeling, perplexity can be a good metric.
7. Deployment: The NLP module can be deployed as a web service. This step and the next one is out of the scope of this project.
8. Monitoring and model updating.

3.2.1. Topic modeling

To have a summary of any text or document, people want to know about the concepts it covers instead of the words that make it up. For that reason, topic modeling is a powerful technique for analysis of a huge collection of documents, it is used for discovering hidden structure text. We can say that a topic is viewed as a recurring pattern of co-occurring words, simply put, a topic includes a group of words that often occurs together. This means that it can link words with the same context and differentiate across the uses of words with different meanings [10].

Typically, most industries have huge volumes of unlabeled text documents. In case of an unlabeled corpus to get the initial insights of the corpus, a topic model is a great option because it does not just give us topics of relevance, but also categorizes the entire corpus into a few topics given to the algorithm.

On this area of study, there are mainly five methods to discover topics from a corpus (collection of documents) but only the last two will be covered:

- Vector Space Model (VSM),
- Latent Semantic Indexing (LSI)
- Probabilistic Latent Semantic Analysis (PLSA)
- Latent Dirichlet Allocation (LDA)
- Top2Vec

3.2.2. LDA

LDA is a generative probabilistic model of a corpus and as an unsupervised learning algorithm has a semantic approach rather than a syntactic one since the order of the words are not important. The main idea behind LDA is that documents are represented as random mixtures over latent topics and each topic is a distribution of a fixed vocabulary. It has been tested under different scenarios like document modeling, text, classification, and collaborative filtering.[11]

Knowing all the previous information about LDA, it can help to capture the synonyms, hypernyms, and hyponyms of any word. For example, the words "person" and "Diego" would be clustered into the same topic. In addition, the words "component" and "piece" would also be clustered into the same topic. The

topic model improves the performance of a classification model by reducing the number of features or dimensions.

According to the book *Natural Language Processing in Action*, LDA topics are better to explain than LSA, for this reason, LDA was the first approach to generate topics in this project however many challenges were faced, like knowing the optimal number of topics to select for every dataset, which could lead to a multiple iterations and time consuming even though a coherence model was proposed to help select the best number of topics, and also, the selection of many stop-words that must be performed manually by a developer and a group of experts.

The best solution to avoid the problems stated above was to use another algorithm called Top2Vec, explained below.

3.2.3. Word2Vec, Doc2Vec and Top2Vec

Word vectors are numerical vector representations of word semantics or meaning. For example, a word can be grouped in an animal, concept, or person cluster. This vector representation is a dense vector of floating-point values with no zeros. These numbers are trainable, meaning that there are weights to be learned during training.

The other name that is commonly used is word embedding and this dense representation will lead to have similar encodings with similar words. Word2Vec is a family of model architectures that can be used to learn word embedding from large datasets.

The same idea can be extended to paragraphs or documents instead of words. For this case, the model is called Doc2Vec, and so the document, paragraph or sentence is represented by a vector as well.

The spatial representation of words and documents is called a semantic space. This space can show many aspects of the vector representation, like the word vectors closest to the document vector will be semantically related. In the same way, if there are a bunch of documents close to each other within an area indicate that they are semantically similar. Including a document vector used for the word prediction, the already trained document vector can be used in tasks to find similar documents in the corpus, by either calculating the cosine distance or grouping the documents into clusters.

For the Top2vec topic vectors are calculated as the centroids of each dense area of document vectors. A dense area is an area of very similar documents, and the centroid, or topic vector, can be thought of as the average document most representative of that area We leverage the semantic embedding to find the words which are most representative of each topic vector by finding the closest word vectors to each topic vector.

Removing stop-words, lemmatization, stemming, and a priori knowledge of the number of topics are not required for top2vec to learn good topic vectors. This is a huge advantage and that is the reason Top2Vec was chosen instead of LDA.

As mentioned before, a document can be represented by a set of topics so the topic vector in Top2Vec is obtained by calculating the centroid of the collection of common document vectors, as described in the picture below. With this idea, it can be explained that Top2Vec does not need to remove stop words since the common words shared by the documents will have a long distance from the topic vectors and equally distant from all documents.

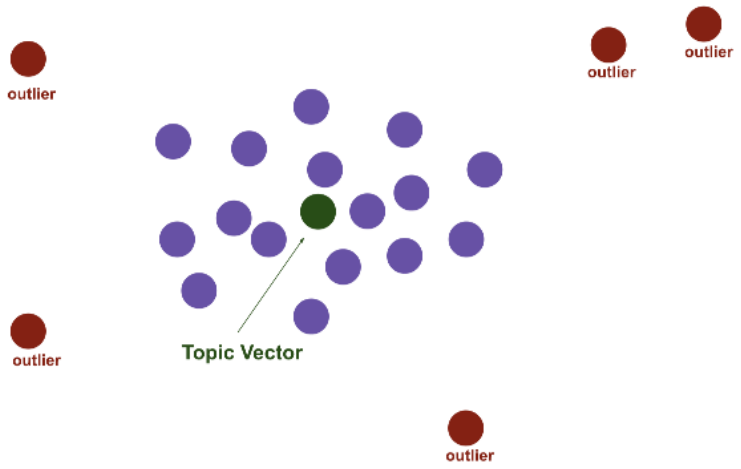

Figure 3 Topic Vector

Additionally, Top2Vec make use of a dimension reduction algorithm before finding all the dense areas or dense clusters of documents through the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), this algorithm is called UMAP, stands for Uniform Manifold Approximation and Projection for Dimension Reduction. In summary, the goal of these two algorithms is to label each document to the corresponding dense cluster to which it belongs in the semantic embedding space [7, p. 2]

3.2.4. Exploratory Search Engine

Due to the continuous increment of data worldwide, more research has been performed in the field of Information Retrieval (IR), which covers the methods and techniques used in search engines. Most of them are keyword based and they are useful for short queries, but when we need more accurate results those queries may be limited so we may need to enter an extended query as our project needs (complete explanation of an issue in a SW component or module). For these reasons, exploratory search emerges as a new paradigm for search engines in Information Retrieval.

TF-IDF is a very used method to extract better semantic information than a common bag-of-words and it is often used in search engines but, in comparison with LDA, LDA has the advantage to look for the topic representation within a text (most of our queries will be long requirements, that can be long fragments of text, a whole document or even a set of documents) so at the end of a semantic search it will retrieve the most similar documents with more accuracy. LDA can be highly improved by the regularization according to the paper of *Artificial Intelligence and Natural Language* [12]. However, this will be out of the scope of this project, but it might be considered in a second stage depending on the efficiency of the current models.

Another advantage of the topic-based search in comparison to TF-IDF search is that the low-dimensional sparse topical representation of documents can be converted into a highly compressed inverted index, lowering the cost implementation. [12]

Word2Vec is another method that can be used in exploratory search which can provide better results according to the paper of Md. Hasan, Md. Motaher Hossain, Adnan Ahmed, and Mohammad Shahidur Rahman [13]. In the same way, we can use Top2Vec to find similar documents considering the semantic features of topics.

3.2.5 Knowledge Graph

A knowledge graph can be defined as a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities [14]. As illustrated in Figure 4, knowledge graphs are composed from different sources so it can be very diverse in structure, for that reason, a schema is defined to represent a high-level structure of the knowledge graph.

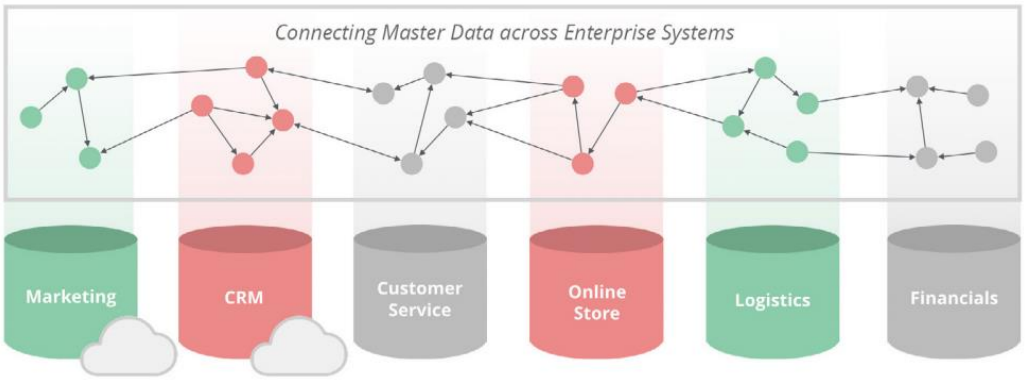

Figure 4 Neo4j Integration Example

Knowing the diversity of information on R&D departments, from issues, requirements, lessons learned to employees, customers, partners, etc. Corporate knowledge graphs can be used to support the semantic relationship of contents and perform many of the common graph algorithms for centrality, community detection, similarity etc. to get more insights besides the NLP models.

Even though on this project an extensive Knowledge Graph will not be constructed, it will be the starting point for future development with the objective of having powerful one that covers many areas of information through many of databases, since currently, the company lacks intercommunication and tools for knowledge extraction and discover hidden patterns in data and its relationships.

4. METHODOLOGY

Summary: *In this section, we will cover all the steps needed to fulfill the requirements and get the expected results. A road map will be shown to depict the entire data processing pipeline, model training and graph database integration.*

4.1. Requirements

The overall objective of this project is to cover the following features, so in this section we are going to show the procedure to cover them:

- LDA and Top2Vec model generation
- Topic visualization
- Top Topic representation
- Top Topic timeline
- Semantic search
- Graph representation

The company manages databases which contain all the information of issues found in R&D. Jira is a tool that manages and stores the information of many projects and is currently becoming the main database across all the company replacing IMS that for many years was the main change management tool. IMS still has a lot of old information that can be treated and analyzed however, for this project and as part of a demo to the company only Jira was used for two different projects.

4.2. Road Map

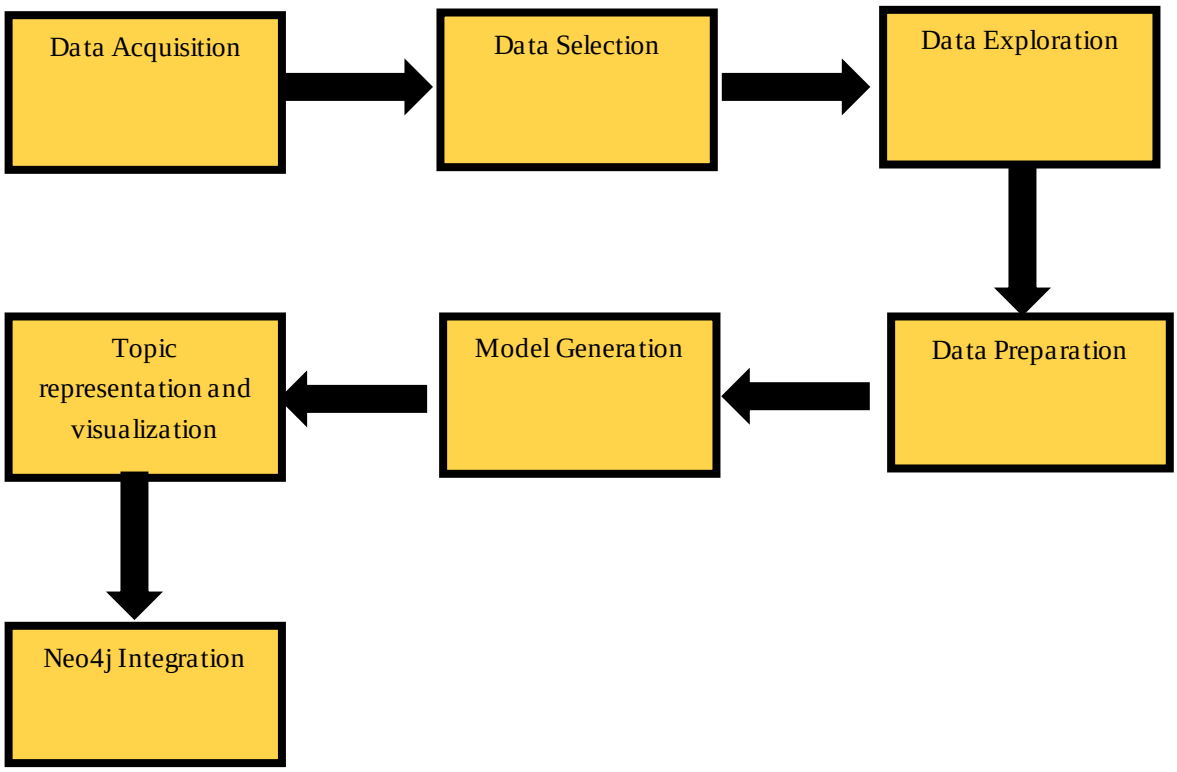

Figure 5 Road Map

4.3. Data Acquisition

As mentioned before, issues or problem reports of two projects are found in Jira. It is possible to export the main information and metadata as csv files. Then we proceed to import the data to the environment, explore it and transform it.

4.4. Data Selection

Through the pandas' library we are going to load the csv data and transform it to a pandas' data frame and then proceed with the exploration and manipulation of the data.

It is important to note the first discrepancy we found between the two datasets. Even though these files come from the same database, they differ somehow in format (mainly column names) and the fields used. This happens because although all projects start with a common format, it is possible to customize and adapt the structure according to the needs of each project. Therefore, in the data selection and data preparation phases more adjustments are going to be implemented to follow a single schema. Anyway, this problem will be indicated to project management to try to keep the same schema as possible between projects which will ease future analysis.

4.5. Data Exploration

On this step, we explore the data to answer some questions like what data we have? What could be the possible inputs for the model? What kind of data cleaning and transformation is needed? How much data is missing?

If we get the information of project #1, it shows:

#	Column name	Non-null counter	Dtype
0	Summary	954 non-null	object
1	Issue Type	954 non-null	object
2	Issue id	954 non-null	int64
3	Status	954 non-null	object
4	Issue key	954 non-null	object
5	Project name	954 non-null	object
6	Parent id	442 non-null	float64
7	Priority	954 non-null	object
8	Created	954 non-null	object
9	Resolved	770 non-null	object
10	Description	895 non-null	object
11	Custom field (Analysis Results SW)	190 non-null	object
12	Custom field (Analysis Results System)	30 non-null	object
13	Custom field (Caused by)	209 non-null	object
14	Custom field (Proposed Solution)	222 non-null	object
15	Custom field (Detected by)	213 non-null	object
16	Custom field (Error Root Cause)	212 non-null	object
17	Custom field (Occurrence)	157 non-null	object

18	Custom field (Requirements IDs)	168 non-null	object
19	Fix Version/s	865 non-null	object
20	Component/s	747 non-null	object
21	Component/s.1	186 non-null	object
22	Watchers	951 non-null	object
23	Watchers.1	545 non-null	object
24	Watchers.2	187 non-null	object
25	Watchers.3	65 non-null	object
26	Watchers.4	27 non-null	object
27	Watchers.5	15 non-null	object
28	Watchers.6	4 non-null	object
29	Watchers.7	2 non-null	object
30	Assignee	913 non-null	object

Table 1 Information of Project #1

The Summary and Custom Fields carry the information for the topic modeling generation, on which there are many null values. These null values will just be ignored.

The Created field corresponds to the date of the issue creation but is not detected by pandas a datetime type. This field will be used when plotting topics over time.

For the project #2 the csv had 164 columns and 2110 rows. With that number of columns pandas was not able to display it in the Jupyter notebook, so a subset was created considering only the most important columns:

#	Column name	Non-null counter	Dtype
0	Project	2110 non-null	object
1	Key	2110 non-null	object
2	Summary	2110 non-null	object
3	Issue Type	2110 non-null	object
4	Status	2110 non-null	object
5	Resolution	2110 non-null	object
6	Assignee	2110 non-null	object
7	Created	2110 non-null	datetime64[ns]
8	Resolved	1806 non-null	datetime64[ns]
9	Affects Version/s	2063 non-null	object
10	Fix Version/s	1428 non-null	object
11	Component/s	1971 non-null	object
12	Description	2110 non-null	object
13	Resolution Description	1629 non-null	object
14	CCB Decision	1534 non-null	object
15	Testcase IDs	1111 non-null	object
16	Requirements IDs	1440 non-null	object

17	Error Root Cause	1800 non-null	object
18	Occurrence	1562 non-null	object
19	Caused by	1801 non-null	object
20	Detected by	2106 non-null	object
21	Proposed Solution	269 non-null	object
22	Analysis Results SW	185 non-null	object

Table 2 Information of Project #2

As you can see, the column names are different between projects so in the next section this data frame will converge into a unique one, for two reasons, standardize the inputs for the topic model generation and the graph database integration depending on the schema that will be presented later.

4.6. Data Preparation

For LDA and Top2Vec data preparation, a unique data frame was created integrating the two project's data. This data frame will consist in the following schema including only the problem reports since Jira also contained change requests and other types of tickets:

IssueID	Problem	Analysis	Solution	RootCause

Figure 6 Dataframe Header

Besides topics, the requirements are an important entity that could be extracted and visualized in the graph database. These requirements could generalize the feature that belongs to the issue, and it could be found in a specific field from Jira or other cases be in the problem description, therefore, a regular expression was created to extract these requirements from the text like R:2.1.23.6 or R:6.7.1.53:

Regular expression: *r'(R:\s*[0-9]+\.[0-9]+)'*

4.6.1 LDA

First, we are going to briefly describe the data preparation for the LDA model. We will notice the disadvantages using this approach which led us to propose another model for the topic generation. These are the stages in the NLP pipeline:

1) Convert text to lower case.
2) Tokenize
3) Stop-words removal.
4) Lemmatize.

Converting to lowercase all words eases the manipulation of the text for NLP techniques like stop-words removal and bag-of-words.

In the tokenization step, one word will be considered one element (spaces, marks and punctuation are removed) so at the end a list of tokens is created.

We decided to have stop-words removal always upgradable, this means that words that need to be removed are taken from a specific csv document that is editable by any person who finds any word that cannot bring meaning to the analysis.

Then we proceed to lemmatize the tokens through the spacy library for English words.

After applying the "process" method we could see the following output:

```
['malfunction', 'problem', 'mobile', 'problem', 'radio']
```

The LDA model is generated through a library called Gensim, which requests as input a specific format called dictionary and corpus. A dictionary will look like this:

```
{'malfunction':0, 'problem':1, 'mobile':2, 'radio':3}
```

As shown above, a dictionary takes as key all unique tokens and assigns a unique ID as value. After that, the corpus is created for each document which represent an array of tuples where the first element is the token ID followed by a counter of the token within the document:

[(0,1), (1,2), (2,1), (3,1)]

4.6.2 Top2Vec

This library already has built-in the minimum preprocessing steps required for the model, so once we have the raw data frame including both projects, then we proceed at this stage by only removing the blank registers otherwise the model would throw an error.

Here the main advantage using this approach is that we avoid adding an extra manual step (stop-word removal) that a user can modify and can affect the model output if not done properly.

4.7. Topic model generation

In this section, we are going to describe some of the parameters needed for the generation of LDA and Top2Vec models

4.7.1. LDA

For the LDA, one of the main parameters to set is the number of topics that LDA will produce. This is not an easy task, since that number will depend on multiple variants like the quality of the data, the number of documents etc. Fortunately, there is a measure of topics' coherence that can help to get a good approximation.

To get the optimal number of topics for current analysis, we propose a routine that will iterate from the variable "start" to the "limit" with "steps" and on each of these steps a coherence value will be computed and plotted. At some point, as the number of topics is increasing the coherence value will stop varying, hence, the optimal number of topics can be obtained by selecting the minor value on which this behavior starts.

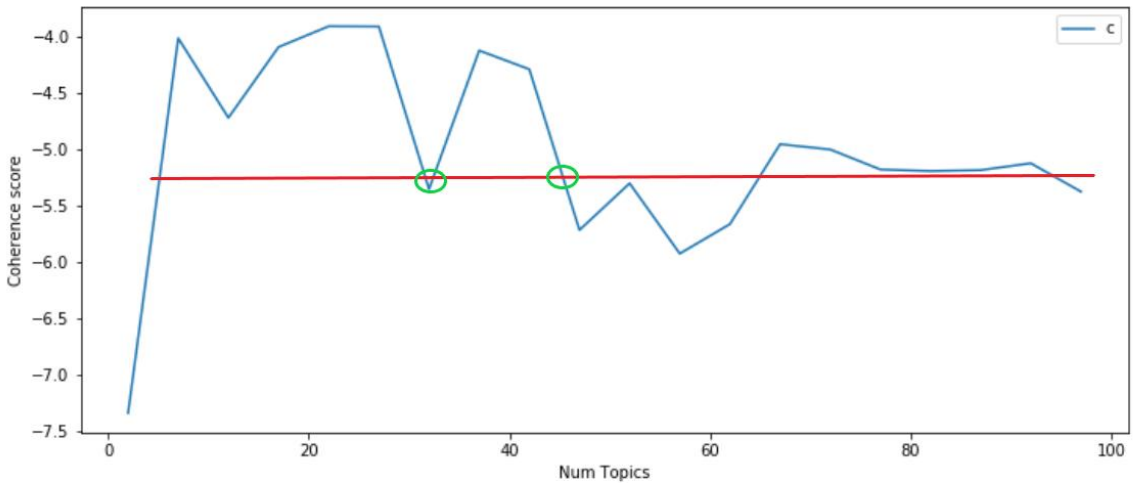

Figure 7 Coherence Score

The red line is the base on which the coherence value stabilizes as the number of topics increases. In this scenario we could experiment with two values, 32 and 45 topics, but the 45 value is closer to a better coherence stabilization.

4.7.2. Top2Vec

In contrast, Top2Vec does not need to specify the number of topics for the model output since the library has built-in algorithms to reduce dimensionality, find clusters and calculate its centroids as explained in the last section.

The unique parameter to include is the embedding model. We decided to include an embedding model because this is a prototype project with a small data set, so implementing the vector representations only using our data may not lead to an efficient result. Although it will be a good idea to use only the doc2vec approach to train a larger dataset when more data is available since there is a very specific vocabulary in the automotive sector, and the pretrained models are based on large public datasets. [Top2Vec]

4.8. Neo4j Integration

Before importing the data into the graph database, it is important to have a clear view of the schema to be used. Besides multiple features collected from Jira, The Topic entity obtained with the topic modeling will be essential for this graph. The schema was developed based on the table below, so the data was manipulated to follow this structure.

Columns	Name	Type	Relationship
Summary	Summary	Property	Issue
Issue Type	IssueType	Node	Issue - OF_TYPE -> IssueType
Issue id	IssueID	Property	Issue
Status	Status	Property	Issue
Issue key	Issue	Node	Issue
Project name	Project	Node	Project - HAS_ISSUE -> Issue, Project-HAS_COMPONENT-> Component
Parent id	ParentIssue	Relationship	Issue - HAS_CHILD -> Issue
Priority	Priority	Property	Issue
Created	CreatedDate	Property	Issue
Resolved	ResolvedDate	Property	Issue
Components	Component	Node	Issue - FOUND_IN -> Component
Description	Problem	Node/Property	Topic/Issue
Custom field (Analysis Results SW)	Analysis	Node/Property	Topic/Issue
Custom field (Analysis Results System)	Analysis	Node/Property	Topic/Issue
Custom field (Caused by)	Source	Node	Issue - CAUSED_BY -> Source
Custom field (Proposed Solution)	Solution	Node/Property	Topic/Issue
Custom field (Detected by)	Detected	Node	Issue - DETECTED_BY -> Detected
Custom field (Error Root Cause)	Error	Node	Issue - ROOT_CAUSE -> Error
Custom field (Occurrence)	Ocurrence	Node	Issue - OCURR_TYPE -> Ocurrence
Custom field (Requirements IDs)	Requirement	Node	Issue - HAS_REQ -> Reqs
DTCs	DTC	Node	Issue - HAS_DTC -> DTC
Fix Version/s	Version	Node	Issue - FIX_IN -> Version
Assignee	Assignee	Node	Issue - ASSIGNED_TO -> Assignee

Table 3 Neo4j Table Schema

Columns: Name of each column to get from the csv files.

Name: Name of the graph entity.

Type: Type of the graph entity, it can be Node, Property or Relationship.

Relationship: Direction of the relationship between nodes.

To import the data to the Neo4j database we made use of the py2neo library which allow us to execute the same queries and instructions that can be perform directly in Neo4j but with the advantage to implement this in a python notebook together with the data processing code and to automate specific transactions through python functions. These functions were used to create nodes and relationships and to add or update properties to them.

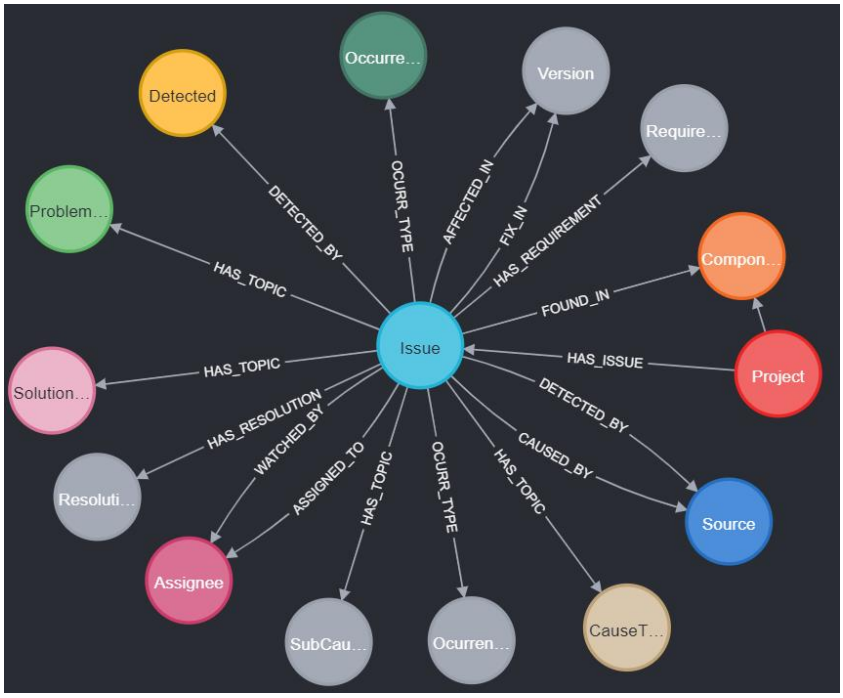

Figure 8 Graph Database Schema

We can observe in the schema above that almost all features are pointing in and out of the Issue entity. The topics are Solution, Problem, Cause and Sub-cause. This last one was obtained by regenerating the topics from the Cause topic #1 since the algorithm detected only 3 topics for the Cause records.

5. RESULTS AND DISCUSSION

Summary: *This chapter shows the results obtained through the entire development and a discussion about data and strategies implemented in this project.*

5.1. Results

It is important to point out that the results were obtained through the Top2Vec model only since this was chosen to work with the data due to its simplicity in the data processing phase.

Four Top2Vec models were created that represent the **ProblemTopic, SolutionTopic, CauseTopic and Sub-CauseTopic**. As an example, the table below describes the 26 problem topics with the following distribution. The topic cluster is already sorted by the total number of issues.

Topic	Number of issues
0	194
1	176
2	172
3	160
.	
.	
.	
23	33
24	24
25	23

Table 4 Number of issues in ProblemTopic cluster

For the other models, **SolutionTopic** also got 26 topics and **CauseTopic** only 3 topics having the topic cluster 0 1838 issues, for that reason, we proceeded to create another model specifically for that cluster on which we obtained 16 topics that we called **SubCauseTopic**.

5.2.1 Topic data acquisition for analysis

The information of the topics for each issue can be retrieved from the graph database by performing some queries, like the following:

MATCH (i:Issue)-[:HAS_TOPIC]->(t:ProblemTopic)

RETURN i.issueID as issueID, t.name as ProblemTopic

Having a query like the above one for all topic models we can obtain the next table (showing only the first 5 rows), where each issue is linked to a topic or in other cases not linked to any topic (like some cases in Cause and Sub-cause topic).

	issueID	ProblemTopic	SolutionTopic	CauseTopic	SubCauseTopic
0	917	6	24.0	NaN	11.0
1	3402	6	2.0	1.0	NaN
2	3456	6	9.0	NaN	2.0
3	3959	6	14.0	NaN	16.0
4	3960	6	24.0	NaN	16.0

Table 5 Issue-Topic Distribution

5.2.2 Topic representation and visualization for analysis

To describe the topic clusters to managers and leaders, we proceeded in two ways. First, as each topic is represented by a mixture of words, a word cloud was generated for each topic (we avoided showing the word cloud due to the protection of the data).

Then to go deeper in the description of the topics, we provided some real documents (as seen in the original CSV files) clustered in a specific topic of either Problem, Solution, Cause or Sub-cause. This can result in a better understanding of the output of the model.

Besides that, to answer some of the questions about the behavior of specific issues through time, we continued by plotting the main topics for each model, as shown below:

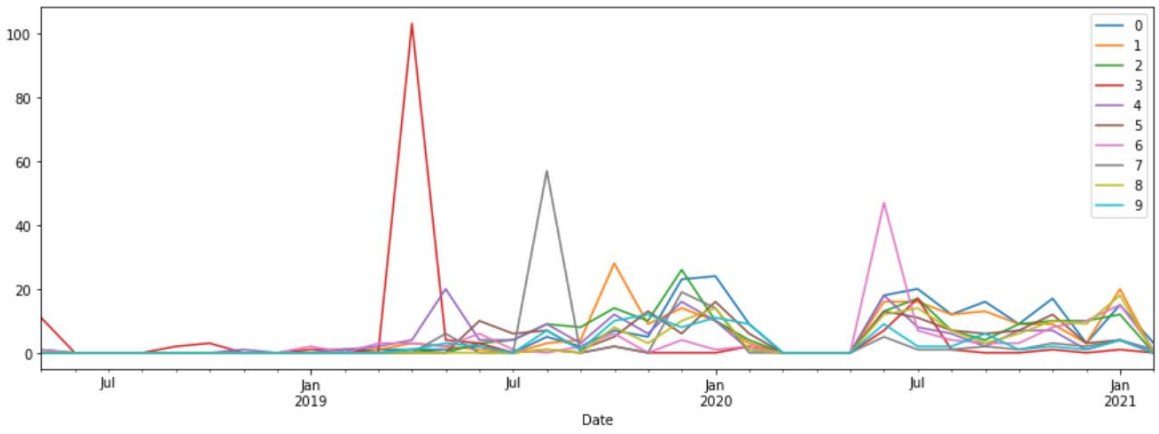

Figure 9 ProblemTopic Timeline

We can quickly note that there is a high peak in April 2019 related to the ProblemTopic #3, other important ones in August 2019 and June 2020. Probably, a deeper analysis could be started from here by reviewing the kind of issues at this time and the requirements related to those issues.

Although this kind of time base analysis is better on the specific issues, we developed two additional plots for the SolutionTopics and the Sub-CauseTopics, just in case it was needed:

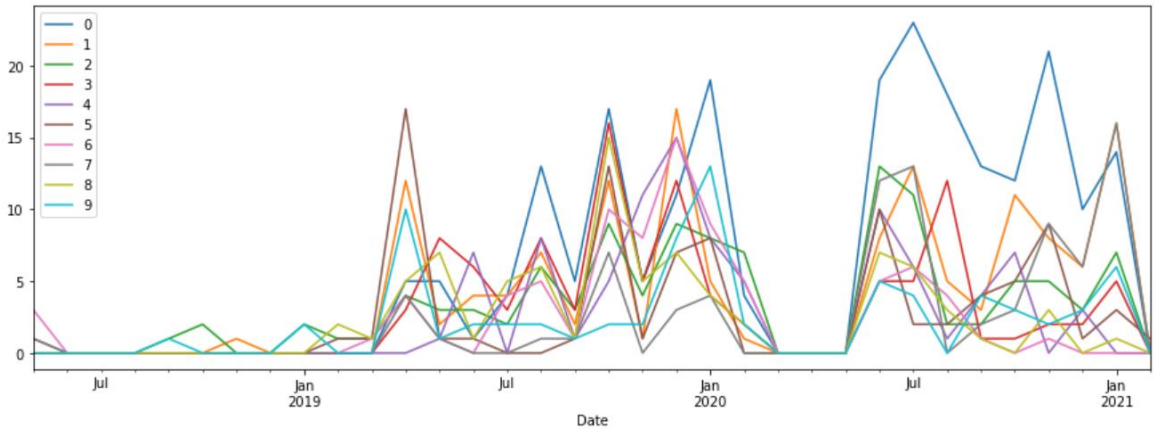

Figure 10 SolutionTopic Timeline

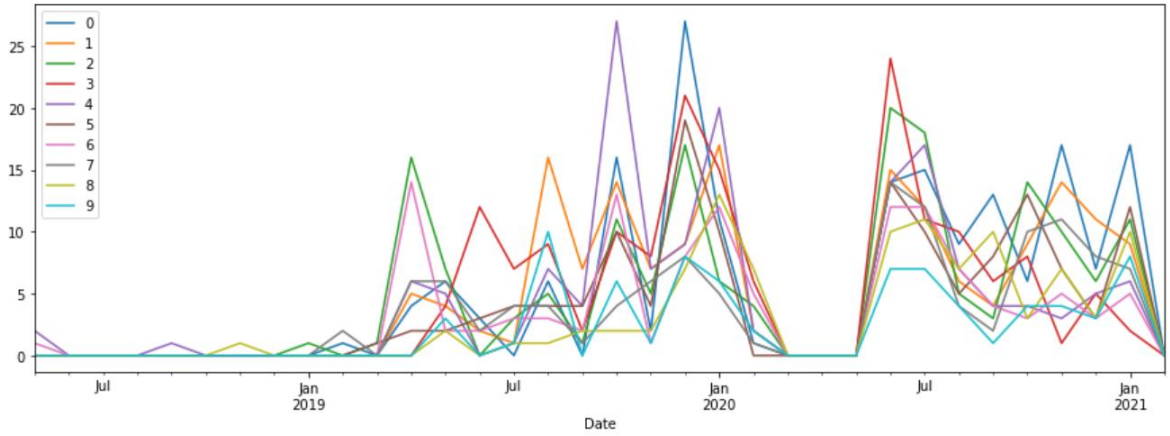

Figure 11 Sub-causeTopic Timeline

5.2.3 Semantic search

We can then proceed with the last feature we stated in the requirements, this is, look for similar issues in our database based on a long query or description, since we would like to get the optimal match by using a semantic search instead of a key-based search.

The Top2Vec model can already ease the procedure by first adding the new query or document to the model. Internally the model saves the entire document or query description and assigns it to a topic cluster. After this procedure, the document is already represented as an embedded vector so a calculation to search similar documents is straightforward. For both sequences, the model has the **add_document** and **search_documents_by_documents** methods.

5.2.4 Graph Database Visualization

To keep with analyzing, now our data is found in a graph database, so here we can explore and visualize the connections between the issues, topics and other features, like users, projects, places etc. As an example, here we are going to show some graphs.

The following figure is the result of a simple graph query that shows the two projects (red) rounded by relationships of their SW/HW components (orange). The two entities in the middle of the picture are the only components that are similar between the two projects. Ideally there should be more related components since although the product is intended for different customers, there are many features that are being used in both. Therefore, with this simple example we can show to managers that we will need to standardize some components, and this can allow us to have the data with more quality and ready to be analyzed to get better insights.

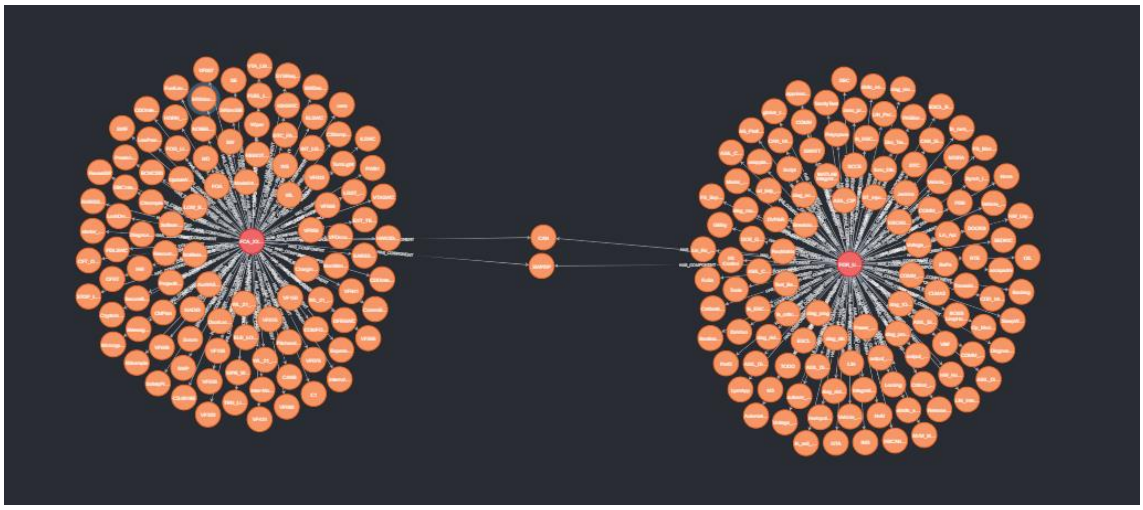

Figure 12 SW Components

In another example, we can get more clusters to analyze if we plot the relationship Project-Issue-Source, as shown in figure 13. The project (red entity) has many issues (light blue), and they are associated with a source (blue). The source entity means a classification of the cause of the issue, so in this visualization we can identify some of the most frequent sources of a specific project.

This approach can lead us to aim some important objectives since as we can see, the upper and lower sources have a greater number of issues, therefore, if we focus on these, we could improve in a significant way the status of the project.

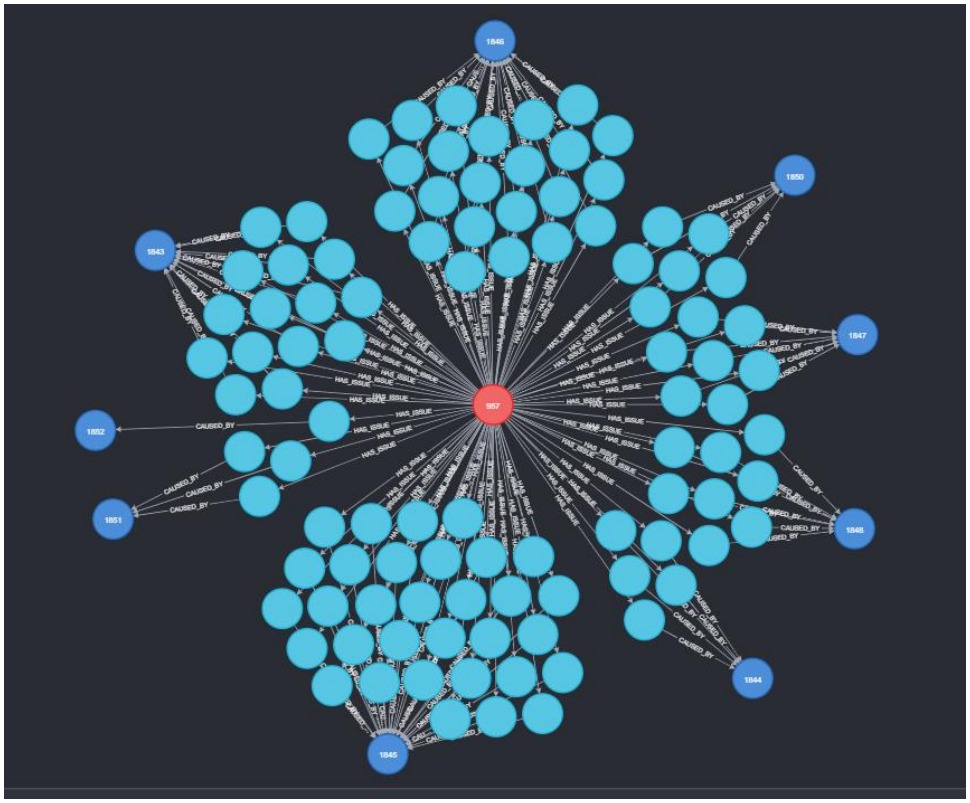

Figure 13 Project-Issue-Source

5.2. Discussion

One of the advantages we discovered through this project is that there are many ways to represent our data using NLP techniques and graph representation. In the following figure, we can see a combination between these two.

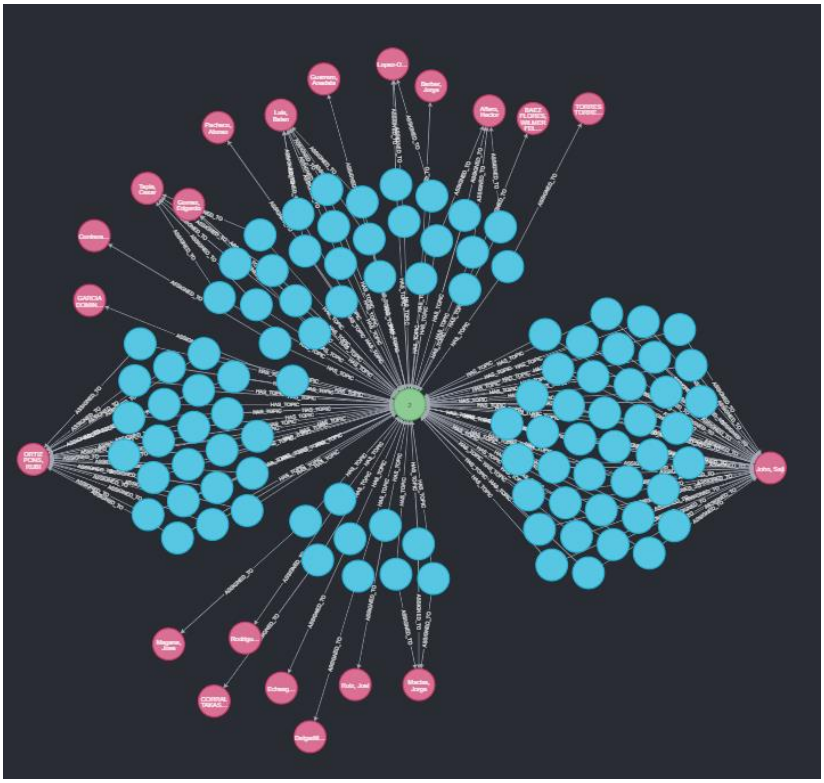

Figure 14 Project-Issue-User

In this example, we can see a more insightful graph where the ProblemTopic #2 (green) is surrounded by the issues related to that cluster and the assignees or users that worked on that kind of issue. At the left and right side, we can see two users associated with many issues of the same kind of topic. This is very important information that could lead to many project decisions.

There are plenty of graph queries that can be applied to watch different relationships between the entities, so this can be leveraged by the input of the project managers and leaders.

As a future idea, having this kind of insights could help when a new issue is found and loaded to the database. Once it is loaded it can trigger some indications like the most common past issue (from an automatic semantic search) and additional information like the actual person in the team that could solve this kind of issue, etc.

Lastly, the LDA vs Top2Vec algorithm can be further discussed since although the last one was chosen, LDA still has some relevant characteristics once the additional steps are set, and it is one of the most popular topic model techniques nowadays. LDA offers a topic distribution for each document or issue that can be quite useful for semantic search through some distance calculation like Hellinger. A comparison between these two can be a good approach to continue to discover the best model for semantic search.

6. CONCLUSION

Summary: *In this chapter, the conclusions are presented, and the future work related to some activity proposals to continue and improve this project.*

6.1. Conclusion

We have experimented the importance of using NLP techniques in the analysis of the issue's database related to Research & Development in the SW area, however, this kind of analysis can reach many other areas like HW, Systems or even issues found in the production line, claims, feedbacks, lessons learned, etc. As already explained, there is a lot of information in the text format that due to its big size, it is impossible that a single person can go through it and generate useful insights.

Topic modeling is a quite useful technique to get this kind of general insights, and the advantage is that it can bring other kinds of features besides Exploratory Data Analysis (EDA) but also semantic search. Besides, the timeline for topics was a good plot to present to managers to grasp the behavior of the project. It was ideal to compare two approaches in that matter, as occurs in the real life that not always we develop one algorithm from start to end, but we must create more models and test which one performs best with our data.

Another important point to remark is that the data processing stage was the one which took most of the time, for both the topic model and the graph integration, therefore, as everybody says "garbage in, garbage out" we must take special care of how we construct the data pipeline from the raw data, through the cleansing phase and the tools and techniques used during the complete process.

Finally, this project covered more than just the implementation of NLP techniques, when integrating the graph database, we realized the advantage of using it, for exploratory, visualization and analysis purposes. Knowing the relationships between many entities can reveal important information, as everything is related to something, but with the old tools and technologies it was hard to notice.

6.2. Future Work

The main activity to consider is to establish new requirements based on needs of project managers and leaders (e.g., know the input necessary to aim a specific KPI) because this project was proposed the other way around, meaning that a bunch of algorithms, features and technologies were used to try to find an application that could add value to the company projects. Anyway, this approach helped to let the management team know about the potential of this kind of analysis.

Some of the findings in this project is that we realized the data discrepancy between projects, so a good activity to follow as well is to standardize common data between projects to have consistency in the data and ease the process and analysis.

As already discussed, the semantic search feature has a big potential for the system recommendation. More effort can be applied to get the best semantic search selected based on a good metric, since this feature was not really tested and compared, only demonstrating that it can be quite useful once we already have the topic model. Additional algorithms can be used besides topic modeling search with LDA and Top2Vec.

There are many other NLP techniques that could be taken also into consideration like the Name Entity Recognition but were omitted on this project. This kind of token can be then added as well to the graph database.

Now we know the potential of this project and many of the activities and improvements that can be implemented. It is fundamental to design a scalable application going from the data architecture, data processing to the web application. However, to do so it is necessary to conform a specialized working team dedicated full-time on this project.

REFERENCES

[1] M. Pacella, A. Grieco, and M. Blaco, "On the Use of Self-Organizing Map for Text Clustering in Engineering Change Process Analysis: A Case Study," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–11, 2016, doi: 10.1155/2016/5139574.

[2] A. Ianina, L. Golitsyn, and K. Vorontsov, "Multi-objective Topic Modeling for Exploratory Search in Tech News," in *Artificial Intelligence and Natural Language*, vol. 789, A. Filchenkov, L. Pivovarova, and J. Žižka, Eds. Cham: Springer International Publishing, 2018, pp. 181–193. doi: 10.1007/978-3-319-71746-3_16.

[3] A. Ianina and K. Vorontsov, "Regularized Multimodal Hierarchical Topic Model for Document-by-Document Exploratory Search," in *2019 25th Conference of Open Innovations Association (FRUCT)*, Nov. 2019, pp. 131–138. doi: 10.23919/FRUCT48121.2019.8981493.

[4] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990, doi: 10.1109/5.58325.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *ArXiv13013781 Cs*, Sep. 2013, Accessed: Jun. 05, 2021. [Online]. Available: http://arxiv.org/abs/1301.3781

[6] Q. V. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," *ArXiv14054053 Cs*, May 2014, Accessed: Jun. 05, 2021. [Online]. Available: http://arxiv.org/abs/1405.4053

[7] D. Angelov, "Top2Vec: Distributed Representations of Topics," *ArXiv200809470 Cs Stat*, Aug. 2020, Accessed: Jun. 05, 2021. [Online]. Available: http://arxiv.org/abs/2008.09470

[8] E. M. Bender, "Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax," *Synth. Lect. Hum. Lang. Technol.*, vol. 6, no. 3, pp. 1–184, Jun. 2013, doi: 10.2200/S00493ED1V01Y201303HLT020.

[9] "Practical Natural Language Processing A Comprehensive Guide to Building Real-world Nlp Systems by Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana (z-lib.org).epub."

[10] B. V. Barde and A. M. Bainwad, "An overview of topic modeling methods and tools," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Jun. 2017, pp. 745–750. doi: 10.1109/ICCONS.2017.8250563.

[11] D. M. Blei, "Latent Dirichlet Allocation," p. 30.

[12] A. Filchenkov, L. Pivovarova, and J. Žižka, *Artificial Intelligence and Natural Language: 6th Conference, AINL 2017, St. Petersburg, Russia, September 20–23, 2017, Revised Selected Papers*. Springer, 2017.

[13] Md. Hasan, Md. M. Hossain, A. Ahmed, and M. S. Rahman, "Topic Modelling: A Comparison of The Performance of Latent Dirichlet Allocation and LDA2vec Model on Bangla Newspaper," in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, Sep. 2019, pp. 1–5. doi: 10.1109/ICBSLP47725.2019.202047.

[14] A. Hogan *et al.*, "Knowledge Graphs," *ArXiv200302320 Cs*, Jan. 2021, Accessed: Jun. 06, 2021. [Online]. Available: http://arxiv.org/abs/2003.02320