Instituto Superior de Engenharia do Porto

Application of Clustering techniques in Group Decision Making context

Vasco Veiga Martins Rodrigues

Dissertation to obtain a master's degree in

Informatics Engineering, Information and Knowledge Systems

Advisor: Goreti Marreiros

Co-Advisor: Luís Conceição

Júri:

Presidente:

Dr. Isabel Azevedo

Vogais:

Dr. Goreti Marreiros

Dr. Ricardo Santos

Porto, 15th October 2022

“I have not failed. I’ve just found 10,000 ways that won’t work.”

Thomas Edison

Acknowledgments

First, I would like to show my sincere appreciation toward Professors Goreti Marreiros and Luís Conceição. They provided mentorship and supervision and helped me produce the work displayed here. I also appreciate them for allowing me to join the GECAD research center under the Smart Travel project. Also, I would like to thank Fundação para a Ciência e Tecnologia (FCT), which funded my research grant to develop this work.

Second, I would like to thank João Carneiro and Jorge Meira for providing the most scientific and technological knowledge I needed, giving me feedback and ideas on moving forward when I was not achieving the best results.

Third, I would like to thank GECAD for supplying me with the necessary conditions to develop this work in a good environment with great success.

Lastly, I would like to thank everyone who helped me directly or indirectly to overcome all the obstacles this academic journey provided, especially my family and friends.

Abstract

Nowadays, decisions made by executives and managers are primarily made in a group. Therefore, group decision-making is a process where a group of people called participants work together to analyze a set of variables, considering and evaluating a set of alternatives to select one or more solutions. There are many problems associated with group decision-making, namely when the participants cannot meet for any reason, ranging from schedule incompatibility to being in different countries with different time zones. To support this process, Group Decision Support Systems (GDSS) evolved to what today we call web-based GDSS.

In GDSS, argumentation is ideal since it makes it easier to use justifications and explanations in interactions between decision-makers so they can sustain their opinions. Aspect Based Sentiment Analysis (ABSA) is a subfield of Argument Mining closely related to Natural Language Processing. It intends to classify opinions at the aspect level and identify the elements of an opinion. Applying ABSA techniques to Group Decision Making Context results in the automatic identification of alternatives and criteria, for example. This automatic identification is essential to reduce the time decision-makers take to step themselves up on Group Decision Support Systems and offer them various insights and knowledge on the discussion they are participants. One of these insights can be arguments getting used by the decision-makers about an alternative.

Therefore, this dissertation proposes a methodology that uses an unsupervised technique, Clustering, and aims to segment the participants of a discussion based on arguments used so it can produce knowledge from the current information in the GDSS. This methodology can be hosted in a web service that follows a micro-service architecture and utilizes Data Preprocessing and Intra-sentence Segmentation in addition to Clustering to achieve the objectives of the dissertation. Word Embedding is needed when we apply clustering techniques to natural language text to transform the natural language text into vectors usable by the clustering techniques. In addition to Word Embedding, Dimensionality Reduction techniques were tested to improve the results. Maintaining the same Preprocessing steps and varying the chosen Clustering techniques, Word Embedders, and Dimensionality Reduction techniques came up with the best approach. This approach consisted of the KMeans++ clustering technique, using SBERT as the word embedder with UMAP dimensionality reduction, reducing the number of dimensions to 2. This experiment achieved a Silhouette Score of 0.63 with 8 clusters on the baseball dataset, which wielded good cluster results based on their manual review and Wordclouds. The same approach obtained a Silhouette Score of 0.59 with 16 clusters on the car brand dataset, which we used as an approach validation dataset.

Keywords: Group Decision-Making; Dynamic Clustering; Natural Language Processing; Argumentation

Resumo

Atualmente, as decisões tomadas por gestores e executivos são maioritariamente realizadas em grupo. Sendo assim, a tomada de decisão em grupo é um processo no qual um grupo de pessoas denominadas de participantes, atuam em conjunto, analisando um conjunto de variáveis, considerando e avaliando um conjunto de alternativas com o objetivo de selecionar uma ou mais soluções. Existem muitos problemas associados ao processo de tomada de decisão, principalmente quando os participantes não têm possibilidades de se reunirem (Exs.: Os participantes encontram-se em diferentes locais, os países onde estão têm fusos horários diferentes, incompatibilidades de agenda, etc.). Para suportar este processo de tomada de decisão, os Sistemas de Apoio à Tomada de Decisão em Grupo (SADG) evoluíram para o que hoje se chamam de Sistemas de Apoio à Tomada de Decisão em Grupo baseados na Web.

Num SADG, argumentação é ideal pois facilita a utilização de justificações e explicações nas interações entre decisores para que possam suster as suas opiniões. *Aspect Based Sentiment Analysis* (ABSA) é uma área de *Argument Mining* correlacionada com o Processamento de Linguagem Natural. Esta área pretende classificar opiniões ao nível do aspeto da frase e identificar os elementos de uma opinião. Aplicando técnicas de ABSA à Tomada de Decisão em Grupo resulta na identificação automática de alternativas e critérios por exemplo. Esta identificação automática é essencial para reduzir o tempo que os decisores gastam a customizarem-se no SADG e oferece aos mesmos conhecimento e entendimentos sobre a discussão ao qual participam. Um destes entendimentos pode ser os argumentos a serem usados pelos decisores sobre uma alternativa.

Assim, esta dissertação propõe uma metodologia que utiliza uma técnica não-supervisionada, *Clustering*, com o objetivo de segmentar os participantes de uma discussão com base nos argumentos usados pelos mesmos de modo a produzir conhecimento com a informação atual no SADG. Esta metodologia pode ser colocada num serviço web que segue a arquitetura micro serviços e utiliza Preprocessamento de Dados e Segmentação Intra Frase em conjunto com o *Clustering* para atingir os objetivos desta dissertação. *Word Embedding* também é necessário para aplicar técnicas de *Clustering* a texto em linguagem natural para transformar o texto em vetores que possam ser usados pelas técnicas de *Clustering*. Também Técnicas de Redução de Dimensionalidade também foram testadas de modo a melhorar os resultados. Mantendo os passos de Preprocessamento e variando as técnicas de *Clustering*, *Word Embedder* e as técnicas de Redução de Dimensionalidade de modo a encontrar a melhor abordagem. Essa abordagem consiste na utilização da técnica de *Clustering KMeans++* com o *SBERT* como *Word Embedder* e *UMAP* como a técnica de redução de dimensionalidade, reduzindo as dimensões iniciais para duas. Esta experiência obteve um *Silhouette Score* de 0.63 com 8 clusters no *dataset* de *baseball*, que resultou em bons resultados de cluster com base na sua revisão manual e visualização dos *WordClouds*. A mesma abordagem obteve um *Silhouette Score* de 0.59 com 16 clusters no *dataset* das marcas de carros, ao qual usamos esse *dataset* com validação de abordagem.

Palavras-chave: Tomada de Decisão em Grupo, Clustering Dinâmico, Processamento de Linguagem Natural, Argumentação

Index

List of Figures

List of Tables

List of Acronyms and Symbols

AE	Autoencoder
AI	Artificial Intelligence
AM	Argument Mining
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under Curve
BIO	Beginning-Inside-Outside
CBOW	Continuous Bag-of-Words
CRISP-DM	CRoss Industry Standard Process for Data Mining
CSS	Cascading Style Sheets
DM	Decision Making
DNN	Deep Neural Network
EM	Expectation-Maximization
FAST	Function Analysis System Technique
FCM	Fuzzy c-means
FCT	*Fundação para a Ciência e a Tecnologia of the Portuguese Ministry of Science, Technology and Higher Education.*
FFE	Fuzzy Front End
FN	False Negative
FP	False Positive
FURPS	Functionality, Usability, Reliability, Performance, Supportability
GDM	Group Decision Making
GDSS	Group Decision Support Systems
GECAD	Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development

HITL	Human-in-the-loop
HP	Hewlett-Packard
HTML	HyperText Markup Language
ICT	Information and Communication Technology
IDF	Inverse Document Frequency
ISEP	Polytechnic of Porto - School of Engineering
JSON	JavaScript Object Notation
MADM	Multi-Attribute Decision Making
MAE	Mean Absolute Error
MAS	Multi-Agents System
MCDM	Multiple Criteria Decision Making
MEI	Master's in Informatics Engineering
ML	Machine Learning
MODM	Multi-Objective Decision Making
MSE	Mean Squared Error
NCD	New Concept Development
NER	Named Entity Recognition
NLP	Natural Language Processing
NP	Noun Phrase
P.Porto	Polytechnic of Porto
PCA	Principal Component Analysis
PHP	PHP: Hypertext Preprocessor
POS	Part-of-Speech
PRAW	Python Reddit API Wrapper
PSAW	Pushshift.io API Wrapper
QA	Quality Assurance

R&D	Research & Development
RDF	Resource Description Format
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristics
ROI	Return of Investment
RSE	Relative Squared Error
SA	Sentiment Analysis
SAE	Stacked Autoencoder
SA-MpMcDM	Sentiment Analysis based Multi-Person Multi-criteria Decision Making
SEMMA	Sample, Explore, Modify, Model, and Assess
SG	Skip-gram
SQL	Structured Query Language
SVO	Subject-Verb-Object
TDMEI	Thesis/Dissertation
TDSP	Team Data Science Process
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
TN	True Negative
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
TP	True Positive
t-SNE	t-Distributed Stochastic Neighbor Embedding
UC	Use Case
UML	Unified Modeling Language
UMAP	Uniform Manifold Approximation and Projection
VP	Value Proposition

VP	Verb Phrase
WSGI	Web Server Gateway Interface
XML	Extensible Markup Language

1 Introduction

This dissertation describes the project developed during the time the author was with a research grant at GECAD with the project theme “Application of Clustering techniques in Group Decision Making context” and this section aims to make a global description of the area in which this dissertation has its focus. It contains the context of the work, description of the problem, objectives, primary contributions, and expected results. The work methodology is also presented and ends on the dissertation's adopted structure.

1.1 Context

Nowadays, decisions made by executives and managers are primarily made in a group (Carneiro et al., 2014). Therefore, group decision-making is a process where a group of people called participants work together to analyze a set of variables, considering and evaluating a set of alternatives to select one or more solutions. There are many problems associated with group decision-making, namely when the participants cannot reunite for any reason, ranging from schedule incompatibility to being in different countries with different time zones. To support this process, Group Decision Support Systems (GDSS) evolved to what today we call web-based GDSS. Making these systems capable of supporting groups in decision-making is a complex task. The most common type of GDSSs are the ones that supply a way to solve multi-criteria problems since it is the most common cause of decision-making in humans (Majumder, 2015). These problems have a set of alternatives to choose from, in which an alternative can be evaluated based on a set of criteria (Majumder, 2015). Therefore, to use these systems, users must initially set themselves in the web-based GDSS by either choosing their preferences (Palomares et al., 2014) or rating issues by many criteria like importance, urgency, and seriousness (XLeap, 2021). This preference setup was not enough for GDSS solutions to be widely accepted in organizations because it would miss out on many vital points of face-to-face meetings (Conceição et al., 2017). Hence, the field took a step further and started using Multi-Agent Systems (MAS) to simulate the dialogue between the discussion participants, empowering those agents with argumentation capabilities. This method is one of the available methods for automatic negotiation. These agents would have features of the participant they were representing, like styles of behavior, mood, personality types, and others (Carneiro, Saraiva, et al., 2018; Marreiros et al., 2010; Santos et al., 2010). These features potentiate the quality and realism of the GDSS compared to face-to-face meetings. The agents construct arguments in favor/against statements, evaluate each argument's strength, determine the different conflicts between arguments, assess the arguments' acceptability, and compare decisions based on relevant previously accepted arguments (Carneiro, Martinho, et al., 2018; Marreiros, 2008). These advancements in the GDSS field by using these agents that use a structured way of building an argument leads to information exchange and decision evolution over time. However, it is still tricky for a person to express their preferences in such a structured way, giving numerical values to available alternatives and criteria in the GDSSs nowadays. The decision-makers feel uncomfortable following such an unnatural way of structuring interactions. Making the jump into unstructured ways of interaction could be the next step to consider when evolving the actual GDSSs. Due to the unstructured nature of human dialogue, it could feel more natural to express the decision-maker's preferences about specific topics via text or voice. That is the preferred method in social network platforms like Facebook or Twitter. With those forms of communication, important information can be taken from those dialogues without quantifying preferences. That change turns this process into a natural discussion with fewer restrictions between the participants. Suppose that dialogue is well formed and argued. In that case, that consequently could lead to successful essential decisions in organizations, which lead to these types of systems being more accepted in ever-evolving organizations. Another aspect being considered is that today's GDSSs have difficulties extracting knowledge from dialogues based on their context, which could be vital in supporting group decision-making. Same way in psychology, when the psychologist listens to a person and gets knowledge from that person's situation by perceiving things from an outside perspective that a person might not perceive on its' on, by just listening to that person (US Bureau of Labor Statistics, 2019) or in sports where a coach during a game observes it from an outside perspective, allowing the coach to objectively view game and perceive aspects related to the game in hands that the players playing it might not perceive, it leads to the making of decisions that the coach feels like are the best to enhance the performance of the team and reach the team's objective for the game (Lyle, 2005), machine learning models could try to achieve the same thing automatically in the group decision making context, allowing those models to make predictions, classify the dialogues of the discussion based on Alternatives that it supports or argues against with which Criteria is used, the importance of an Alternative or Criterion and other vital aspects from unstructured discussions.

1.2 Problem Description

Today's GDSSs still have some limitations that make them not entirely accepted by organizations (Conceição et al., 2017). The reasons are the lack of time higher-ups have due to busy schedules to fill the preferences in these types of systems or feeling unnatural to express their preferences through numerical values instead of natural language, in which they are used to express their opinions. A shift from this structured way of representing the decision-maker preferences in a GDSS to a less strict form could be the next step for GDSS. Also, GDSSs nowadays have difficulties getting knowledge from the context that is getting discussed where this knowledge can be vital in supporting the group decision-making task. If a GDSS could automatically understand how the decision is flowing over time, which alternative is the most discussed, and what criteria are getting the most usage in the discussion, to name a few, the GDSS would have the knowledge needed to guide the conversation to reach a consensus faster with great satisfaction from the participants.

Said participants would normally interact via text or voice, with this GDSS working in the background to obtain meaningful knowledge about the discussion automatically. This knowledge would then be presented to the decision-makers to help form consensus, for example, by showing who agrees or has similar opinions with one decision-maker. However, to achieve this, one key aspect to consider is the platform that allows said discussion and its' structure. How a public or private forum is structured can differ from how Reddit, Facebook, or Twitter are structured. While all of them have one thing in common: an initial post or message that triggers the discussion, the platform's structure can highly affect the discussion route. Suppose a platform allows for messages of messages, as in, a person replies to a message another person sent instead of the main post. In that case, micro-discussions could be created about a specific alternative that could be different from the one exposed in the main post. A platform that allows this is valuable since it introduces another degree of discussion and increases argumentation. However, on the other hand, it can cause problems for decision-makers, like losing track of the main discussion if multiple micro-discussions coexist or the unwillingness to catch up with the discussion so far if doing such a task is not user-friendly. To overcome these problems, a careful definition and understanding of the platform structure could be the answer. Machine learning techniques like Clustering could be used to obtain meaningful knowledge, for example, how many participants have the same or similar opinions. This knowledge would be given afterward to the participants to help reach a consensus or machine learning models with specific datasets for alternative detection or using pre-trained Named Entity Recognition models for the same task mentioned earlier.

1.3 Objectives

The objectives defined for this project was the application of Clustering techniques in the Group Decision Making context, in which the following objectives are inserted:

- Investigation and concretization of a state-of-the-art study on Clustering in the Natural Language Processing area, in addition to other techniques considered relevant to reach the proposed objectives of the project
- Proposal and implementation of a methodology that uses Clustering to group arguments on topics
- Realization of tests to the developed models
- Evaluation of the results obtained in the tests

- Creation of the web service that hosts the Clustering models

1.4 Main Contributions

One of the initial contributions of this dissertation was a study of the available datasets that could be used in the group decision-making context. A review of the available datasets in platforms like Kaggle[1] and NLP Index[2] was performed. It led to the conclusion that there is a lack of datasets, especially in this context. A methodology was created to annotate unstructured text-based discussions from extraction to creating a new dataset in this context. The paper that describes said methodology was accepted in the Information and Knowledge Management theme of the international conference WorldCist for its' 2022 edition.

- Cardoso, T., Rodrigues, V., Conceição, L., Carneiro, J., Marreiros, G., Novais, P. (2022). Aspect Based Sentiment Analysis Annotation Methodology for Group Decision Making Problems: An Insight on the Baseball Domain. In: Rocha, A., Adeli, H., Dzemyda, G., Moreira, F. (eds) Information Systems and Technologies. WorldCIST 2022. Lecture Notes in Networks and Systems, vol 469. Springer, Cham. https://doi.org/10.1007/978-3-031-04819-7_3

The methodology, experiments, and results obtained from work developed during the thesis and presented in this document were published in the MDPI open-access journal Applied Sciences, an international, peer-reviewed, open-access journal on all aspects of applied natural sciences published semimonthly online.

- Conceição L, Rodrigues V, Meira J, Marreiros G, Novais P. Supporting Argumentation Dialogues in Group Decision Support Systems: An Approach Based on Dynamic Clustering. Applied Sciences. 2022; 12(21):10893. https://doi.org/10.3390/app122110893

1.5 Work Methodology

The solution from this dissertation is a web service with the application of Clustering techniques in a Group Decision Making context. Two development cycles should be considered, the first being the Clustering models that respond to the objectives proposed and the second being the web service that contains those Clustering models.

For the development of the web service, an agile development lifecycle is put in place using Agile Methodology. This methodology was created in 2001 by a group of 17 developers focusing

[1] kaggle.com
[2] index.quantumstat.com

on four fundamental values and twelve principles in the Agile Manifesto[3] (Cycle, 2018). The lifecycle in Fig. 1 comprehends the following stage (Clesham, 2020):

1. Project Initiation – Also known as the inception or envision phase, this initial stage discusses the project vision and the return on investment (ROI) justification. This phase is a high-level feasibility discussion and does not delve into specific details. It is also identified team members and determines the time and work resources that are required to execute the project until its' completion
2. Planning – Release planning is where the team gets together with their client or product owner and identifies what they are looking for in terms of functional and non-functional requirements. They discuss how this is made possible by building the story-level backlog of how an end-user would describe the feature or product. In this stage, risks should be estimated and milestones developed, leading to a backlog of requirements. That backlog is ordered in terms of business value and dependency based on feedback from the team, product owners, and stakeholders
3. Development – After the requirements have been defined, agile product development kicks in. It delivers a high-quality working product in incremental phases, sprints, or iterations while testing it continuously. This development cycle ensures that by the end of each sprint, there is a working, usable product that stakeholders and product owners can revise. After many sprints, from a minimally viable product to a fully functioning solution, the product is ready to be released into production. Still, before that, final testing and acceptance are done by quality assurance (QA) to detect bugs, in which QA might involve end-users. After some rework on bugs that might appear, the product is ready for production
4. Production – In this stage, the product is deployed and accessible to the end users. Close monitorization is required in these early times for bugs and defects missed in testing, offering continued support to these end-users.
5. Retirement – The final stage of the Agile lifecycle is where the product enters its "end of life" stages and is pulled from production and decommissioned. To achieve this, end-users who still use the product must be notified about possible migration to newer releases or alternative options. This stage typically happens when a more recent release deploys, leading to an older release not receiving support anymore. It is not cost-efficient for the current business model, and its termination is necessary.

[3] agilemanifesto.org

Fig. 1 - Agile Life Cycle (Clesham, 2020)

A well-defined process like the previous one should be considered for developing Clustering models and extracting meaningful knowledge from them. Since Clustering can be regarded as a part of data mining and, subsequently, data science (Sharma, 2020), there are specific frameworks for this type of project. Team Data Science Process (TDSP) and "Sample, Explore, Modify, Model, and Assess" (SEMMA), or the chosen one for this project is the CRISP-DM framework since it is the most commonly used framework for data science projects (Saltz, 2020). CRoss Industry Standard Process for Data Mining (CRISP-DM) emerged in the nineties from the necessity of standardizing the lessons learned into a standard methodology. CRISP-DM defines six significant steps (SridHaran, 2018):

1. Business Understanding – This initial step focuses on understanding the project objectives and requirements from a business perspective, formulating this knowledge into a data mining problem, and developing a preliminary plan
2. Data Understanding – After the business understanding step is done, data understanding consists in an initial data collection where the developer gets familiar with the data, identifies its' quality, problems and discovers first insights into the data, which could lead to detecting exciting subsets to form a hypothesis for hidden information
3. Data Preparation – Covers all activities to construct the final dataset from the initial raw data
4. Modeling – After constructing the final dataset, the developer can start selecting and applying different and appropriate modeling techniques to the data. If in the data preparation step, this next step was not taken into consideration, some modeling techniques require specific data labeling leading into a loop back to the previous step
5. Evaluation – The developer builds and chooses models that appear to have high quality based on metrics previously defined by the team, product owner, or stakeholders. Tests against new data are performed to see if the model can generalize and if it covers sufficiently all the critical business issues leading to the selection of one or more models
6. Deployment – After selecting one or more models, the code representation of them and the data preparation code is sent into deployment, where the models are rebuilt

in a production environment where a mechanism to categorize new unseen data is needed too to help aid these models since it should use new unseen data in the solution of the original business problem to validate them

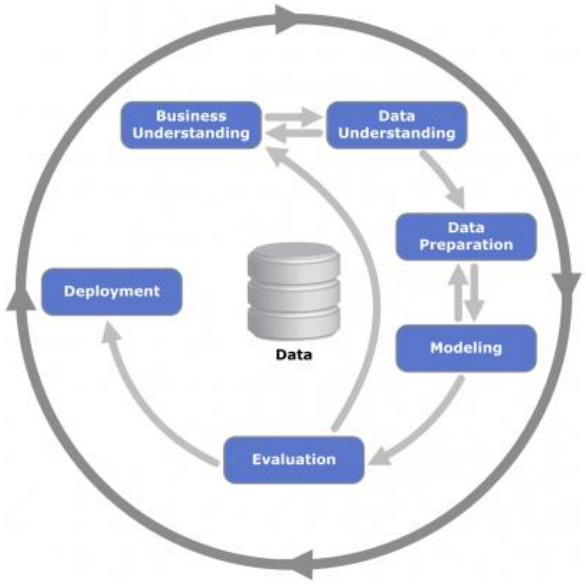


Fig. 2 – CRISP-DM cycle (SridHaran, 2018)

1.6 Document Structure

This last section of the Introduction aims to briefly highlight this dissertation's structure and content, which is divided into seven main sections: Introduction, Context and State of the Art, Value Analysis, Proposed Solution, Implementation of the proposed solution, Experimentation and Evaluation, and Conclusion.

In section 1, the Introduction, the context of the dissertation was exposed as well as its' problem description, objectives, main contributions, and work methodology.

Section 2, State of the Art, intends for the reader to further understand the problem's contextualization and critical theoretical concepts to comprehend the project better. This section has exposed some approaches to similar issues and ends with final considerations about the state of the art.

In section 3, Value Analysis, the value analysis of the project is performed. In it, the innovation process is exposed and detailed and analyzed the value concept. After this, a functionalities analysis using FAST and an alternative analysis using TOPSIS are performed.

Section 4, Proposed Solution, aims to describe the design step to obtain a solution to the proposed problem that can answer all the possible needs of the project.

Section 5, Dynamic Organization of Conversations using Clustering, reveals how the designed solution was implemented.

In section 6, Experimentation and Evaluation, some case studies and experimentations are described to evaluate the proposed solution leading to the detailing and analysis of the achieved results by the proposed solution.

Lastly, section 7 showcases the conclusions, in which the findings of this project are presented and detailed.

2 State of the Art

This section describes the theoretical context behind the problem, facilitating the perception of some concepts of the problem. Furthermore, it is detailed the primary components that contributed to the discovery of suitable solutions where existing solutions are exposed, and some technologies are described.

2.1 Group Decision Making

To understand the concept of Group Decision Making (GDM), first, we need to understand what Decision Making (DM) is. Human beings are fundamentally decision-makers. Everything we do, consciously or unconsciously, results from making decisions (Saaty, 2008). Those decisions can range from a simplistic choice of a restaurant to dine at to more important ones like the place to live or the car to acquire. A decision process comprises a series of steps, starting with information input and analysis, resulting in selecting an alternative from several available alternatives (Eilon, 1969). From an outside look, it seems like an easy process, but reality proves otherwise (Carneiro et al., 2016).

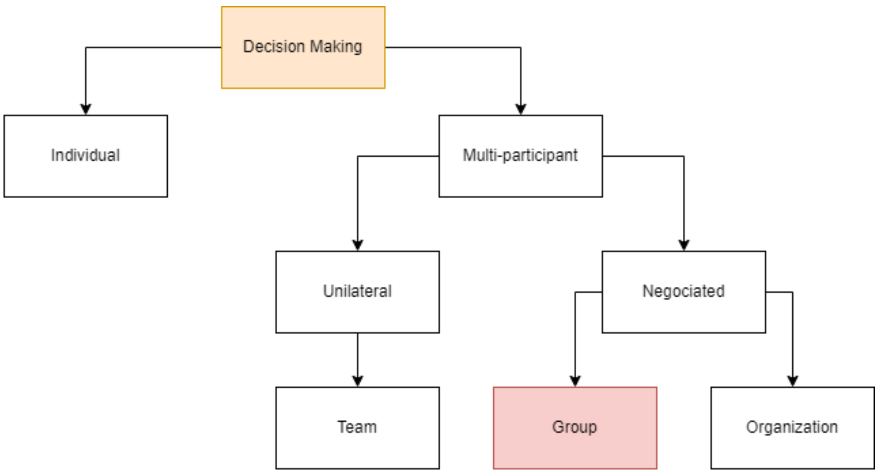


Fig. 3 – Classification of decision-making (H. Hamdani et al., 2018)

GDM is a multi-participant negotiated decision-making process, as we can see in Fig. 3. It has been studied in the last decades, and it is usually utilized in organizations to make decisions (W. Liu et al., 2017; Moon et al., 2003; Rees & Koehler, 2002). The success of an organization depends on the quality of the decisions made (Miller, 1997), meaning that decisions made in a group usually achieve better results (Conceição et al., 2017) since the chance of identifying a problem is higher, allowing decision-makers to work together to solve those problems (Carneiro et al., 2021). The number of these decision-makers can vary as well as being in the same place and at the same time to even scattering geographically at different times (Carneiro et al., 2019). In this context, GDM is needed because of the increasing competitiveness between organizations, making the teams multidisciplinary, and the organization structure forcing decisions to be made in a group (Bryant & White, 2019). It also allows less experienced people in said groups to learn from the process (Conceição et al., 2021), in which we pass said decisions to autonomous or semiautonomous task groups (Hirokawa & Rost, 1992). Still, there is a need to have the right conditions so that these groups can take advantage of the GDM process. It enables them to perform specific tasks like generating new alternatives and criteria to the problem they are facing through group interaction (Carneiro et al., 2021) and some associated difficulties like time consumption, costs, and improper usage of group dynamics (Santos et al., 2010).

Authors of the area started to consider other aspects that could influence the DM process. It is widely known that cognitive elements are taken into consideration by the decision-maker during the process (Schwarz, 2000), enabling them to enhance their learning ability and stimulating their mental level (Carneiro et al., 2021). In a DM process, participants like to recall pertinent information from memory. Still, their current feelings influence that, and then they use their apparent affective response as a base of judgment, making an introspective thought about how they feel about the situation. As a new response appears on the DM process, correlating how the decision-makers feel with the mood, evaluating a new response positively if the decision-makers are in a happy mood and negatively if they are in a sad mood, influencing said person's evaluation in a certain period (Schwarz, 2000).

Additionally, mood affects how a particular individual processes new information given by other participants. People in a happy mood typically “adopt a heuristic processing strategy characterized by top-down processing, with high reliance on pre-existing knowledge structures and relatively little attention to the details at hand.” (Schwarz, 2000). However, people in a sad mood usually “adopt a systematic processing strategy characterized by bottom-up processing, with little reliance on pre-existing knowledge structures and considerable attention to the details at hand” (Schwarz, 2000).

Another aspect to take into consideration is the participants' personalities. Suppose the objective of one participant is for the alternative that supports winning. In that case, if that person tries to predict each participant's personality, the participant could find the best arguments to us in that situation and reach a consensus faster or a better decision in the shortest amount of time (Santos et al., 2010). That action plan could reveal a dominant role or personality, recognizing them as dominators. If the rest of the group yields their choice to the group, determining them as followers and most likely having a submissive personality, a dominator could have his way (Y. Zheng, 2018). Still, if there are more dominators in the group, a clash could result in the group not reaching a consensus because neither dominator could give in to the other. Another personality trait to take into consideration is the extraversion of the participant. It influences the level of participation of a participant. It positively correlates to the number of verbal exchanges of said person, being in the presence of an alternative and its' criteria or the counter argumentation of another alternative (Thatcher & De la Cour, 2003).

Furthermore, the individual judgment of a participant takes into consideration rational and irrational aspects of behavior (Simon, 1986). Psychological factors, interpersonal relationships, and social comparisons affect the process (Lerner et al., 2015) and conflict of interest between the participants in which they could have different objectives to pursue and attain (Lewicki & Litterer, 1985). Although, in the end, each participant's opinion could be judged in terms of that person's importance, rank, expertise level, and credibility. In an extreme and authoritative environment, the opinion of the highest graduated specialist could be taken as a "rule of law", limiting any other opinion from other participants (Carneiro, Saraiva, et al., 2018).

2.1.1 Group Decision Support Systems

The introduction of the concept of Group Decision Support Systems (GDSS) was initially done by Huber (1984) as "a set of software, hardware, and language components and procedures that support a group of people engaged in a decision-related meeting". He also defined three critical activities for a GDSS: information retrieval, sharing, and usage. All key activities work hand in hand by initially extracting information from the organization's database or polling the decision group's opinions so they can be shared with everyone involved. In the end, with all this information, a GDSS should be able to reach a final decision using procedures and techniques of group problem-solving, even if it is an unstructured problem where the information could be incomplete or ambiguous (Huber, 1984).

In the same year, DeSanctis & Gallupe (1984) described GDSS as "an exciting new concept was emerging in the decision support area.". They also defined it as "an interactive computer-based system which facilitates the solution of unstructured problems by a set of decision-makers working together as a group.". They also defined five critical characteristics of a GDSS that are (DeSanctis & Gallupe, 1984):

- "The GDSS is a specially designed system, not merely a configuration of already-existing system components."
- "A GDSS is designed to support groups of decision-makers in their work. The GDSS should improve the decision-making process and outcomes of groups over what would occur if the GDSS were not present."
- "A GDSS is easy to learn and easy to use. It accommodates users with varying knowledge of computing and decision support."

- “The GDSS may be "specific" (designed for one type, or class, of problems) or "general" (designed for a variety of group-level organizational decisions).”
- “The GDSS contains built-in mechanisms which discourage the development of negative group behaviors, such as destructive conflict, miscommunication, or "groupthink".”

The same authors also proposed architecture for these systems shown in Fig. 4 and composed of (DeSanctis & Gallupe, 1984):

- Group members interact with a “group facilitator” and with the user interface, retrieving information stored in databases and other means of information available
- “A "group facilitator" coordinates the group's use of the technology, and there is a flexible, friendly user-interface language available for use by the facilitator or each group member.”
- A user interface responsible for allowing communication between the group members, facilitating the information exchange between group members and the application
- A processor to handle the group member's requests, accessing the database and using the available models at hand
- I/O devices for when the members are locally dispersed

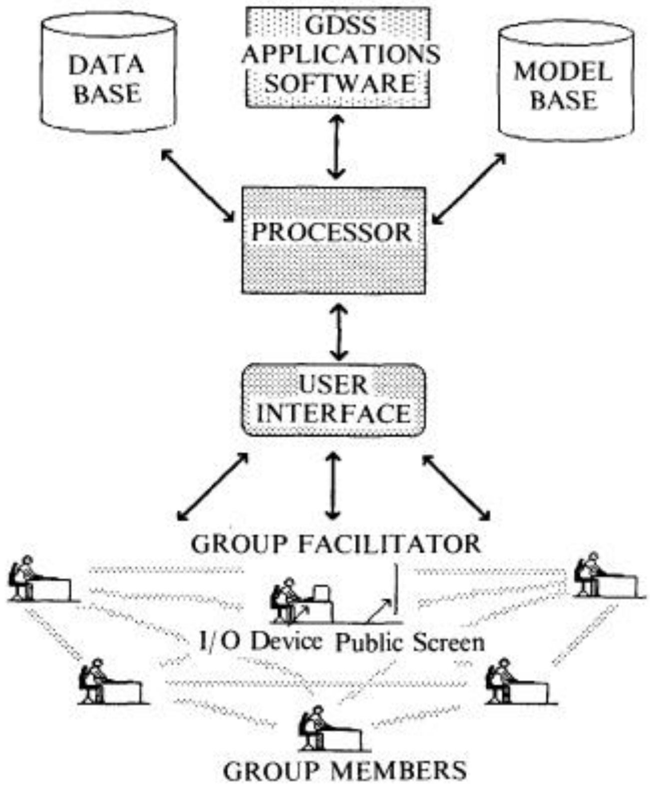


Fig. 4 – The proposed model of a GDSS (DeSanctis & Gallupe, 1984)

GDSS are divided into four categories according to Fig. 5. The authors divided GDSS according to the duration of the session and dispersion of the group members, resulting in four types: Decision Room, Local Decision Network, Teleconferencing, and Remote Decision Making.

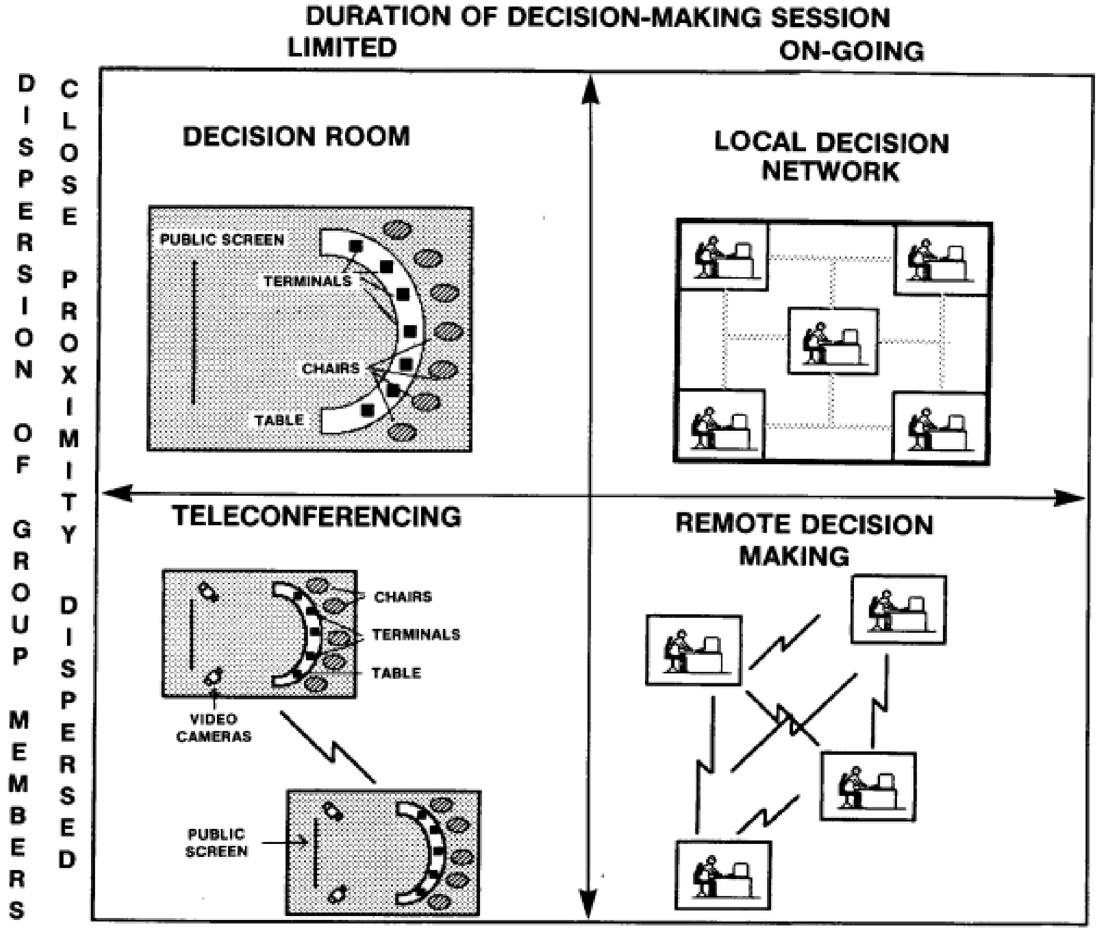


Fig. 5 – Framework for GDSS (DeSanctis & Gallupe, 1984)

Later, DeSanctis & Gallupe (1987) defined a decision-making group as composed of two or more people jointly responsible for detecting a problem, elaborating on the nature of the problem, generating possible alternatives to solve it, and evaluating them or formulating ideas for implementing the alternatives. With this new concept in mind and considering the previous work done by the authors, the GDSS framework became three-dimensional, now taking into consideration the type of task too, as we can see in Fig. 6. They also defined three levels of GDSS systems (DeSanctis & Gallupe, 1987):

- "Level 1 GDSSs provide technical featured aimed at removing common communication barriers, such as large screens for instantaneous display of ideas, voting solicitation and compilation, anonymous input of ideas and preferences, and electronic message exchange between members". These features are found in meeting rooms, known as computer-supported conference rooms. For example, they intend to improve the decision process by facilitating information exchange among the group members.
- "Level 2 GDSSs provide decision modeling and group decision techniques aimed at reducing the uncertainty and "noise" that occur in the GDM process", improving from the communication medium offered by Level 1 GDSSs. GDSSs of this level may also

provide planning tools or other tools found in individual DSSs for group members to work and view them together using a larger standard screen

- "Level 3 GDSSs are characterized by machine-induced group communication patterns and can include expert advice in the selecting and arranging rules to be applied during a meeting.". At this level, each group member is considered a node in the communication network, and technology imposes deliberate communication patterns on them.

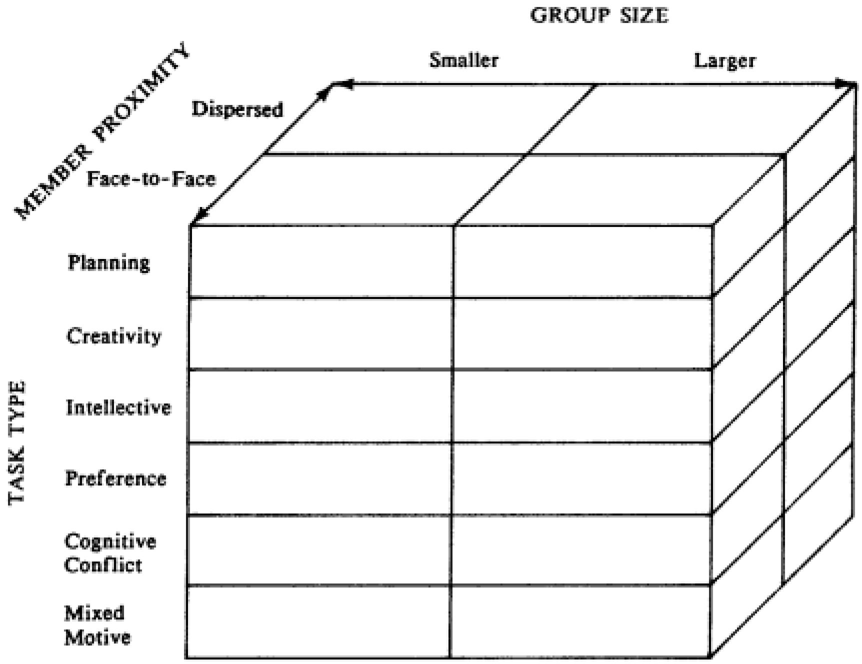


Fig. 6 – Framework for GDSS (DeSanctis & Gallupe, 1987)

In the 90s, the majority of the existing GDSS were synchronous (Turban, 1995), and there was a need to create GDSS that did not restrict the group members in terms of decision place and meeting times, creating a need for the study and creation of asynchronous GDSS (Carneiro et al., 2021).

In the 2000s, the focus was on the development of asynchronous GDSSs (Becker & Bacelo, 2000; Costa et al., 2003; Lee et al., 2003; Zamfirescu, 2003), with many studies comparing synchronous and asynchronous GDSSs to try to understand which type would bring more advantages to the GDM process (Carneiro et al., 2021). Other aspects that started to be considered in these systems were how to replicate types of communication, the interaction between decision-makers, social and affective aspects, and the participants' behavior in face-to-face meetings (Carneiro et al., 2021). However, with the start of market globalization, asynchrony was not enough for these systems since decision-makers started to be scattered worldwide with different time zones between them (Carneiro et al., 2021).

So to fight this, in the last decade, GDSS evolved into what we know today as web-based GDSS was created (Iwai & Sado, 2010; Qin et al., 2010; Siddiqui et al., 2018). These systems would need to adapt constantly and be available initially in a web browser and afterward in other devices like smartphones and tablets (Carneiro et al., 2021).

Nevertheless, there is always room for improvement. GDSS did not stop there and started using architectures for multi-agent systems (Carneiro, Andrade, et al., 2020; Husain, 2014; Marreiros et al., 2009; Santos et al., 2006) and ubiquity (Carneiro et al., 2012; Carneiro, Martinho, et al., 2018; Martinho et al., 2017). That allows decision-makers to add their ideas to the GDM process anywhere and anytime (Carneiro et al., 2014). In addition to this, GDSS started to incorporate cognitive aspects like emotions, personality, mood, and credibility (Carneiro et al., 2017; Marreiros et al., 2010; Santos et al., 2010), improving their qualities and reducing the gap between face-to-face meetings and the usage of a GDSS.

In Fig. 7, there is a summarization of the evolution of GDSS, starting with individual DSS when there was no concept of GDSS yet, to GDSS systems using ubiquity, where they are more portable and work wirelessly through the web.

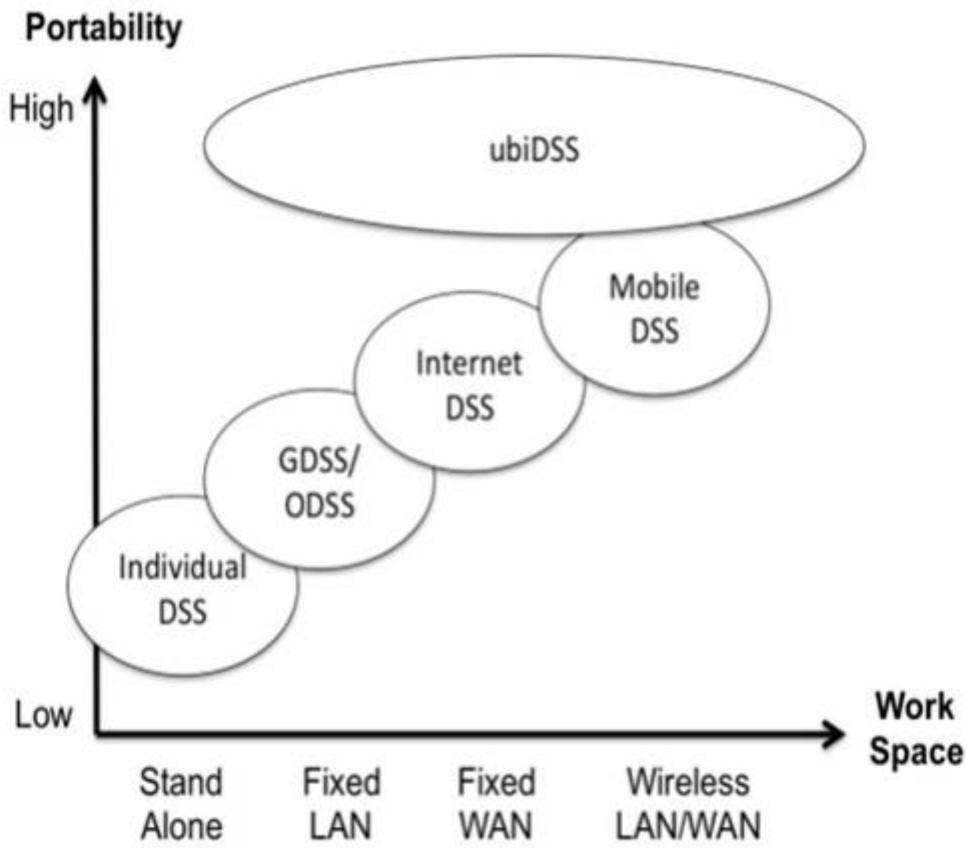


Fig. 7 – Evolution of the GDSS (Carneiro et al., 2012)

2.1.2 Methods

After exposing what a GDSS is and its evolution through time, we need to understand how these systems solve their problems. There are two effective methods categories for these systems, the first one being the mathematical aggregation of preferences of the decision-makers and the other one being automatic negotiation. Both are going to be detailed in the following subsections.

2.1.2.1 Mathematical aggregation of preferences

Mathematical aggregation of preferences consists in when considering n alternatives. There is a need to order that set of alternatives and respective contradicting criteria or various individual opinions to select the best one (Regenwetter, 2009; Tanguiane, 2012). Since there is disagreement in the decision-making process, Brown (1975) assumes that the first step should be de agreement of an ethical and institutional acceptable aggregation procedure since it might be easier to reach a consensus on that aspect faster. Another way to do this is by utilizing algorithms to combine preferences in systems for connecting search results on different search engines or collaborative filtering problems like ranking something for a user based on other users' rankings (Agrawal & Wimmers, 2000; Freund et al., 2004). However, the most used method for this category is multi-criteria analysis. This type of analysis can be subdivided into two types: multi-attribute decision-making (MADM) and multi-objective decision-making (MODM) (Hwang & Yoon, 1981).

An MODM must have measurable objectives, even if they are yes/no scales and their outcomes provide a base for comparison between alternatives, facilitating the selection of one (Brauers et al., 2008). Frameworks that imply this method can range from simple ones with low information needed to exhaustive ones with the implementation of mathematical programming techniques with extensive knowledge (Brauers et al., 2008). Examples of this type of problem framework are Goal Programming (Charnes et al., 1968), Genetics (D. Goldberg & Holland, 1988), and Ant Colony (Dorigo & Stutzle, 2004).

A MADM makes a preferable choice from the available alternatives obtained in a scenario with multiple complex attributes (Hwang & Yoon, 1981). One of the significant problems with these systems is that different methods give different results when applied in the same manner, under the same assumptions, and by a single DM (Zanakis et al., 1998). Examples of frameworks using these problem-solving methods are AHP (Saaty, 1988, 1990), TOPSIS (Hwang & Yoon, 1981), and PROMETHEE (Brans & Vincke, 1985).

2.1.2.2 Automatic Negotiation

Negotiation is the process in which two or more participants reach a consensus or outcomes under conditions of strategic interaction or interdependent decision-making. All sides usually negotiate face-to-face and typically sign an agreement to confirm the success of the negotiation process (Zhu et al., 2005). This negotiation process became automatic and is usually done through agents (Jennings et al., 2001; Morge & Beaune, 2004; Rossi et al., 2016a; Zhu et al., 2005). There are three automatic negotiation mechanisms: Game Theory, Heuristics, and Argumentation (Jennings et al., 2001).

Game Theory (Von Neumann & Morgenstern, 1953) is a type of mechanism based on mathematical analysis focused on decision-making in strategic situations. The success of each player in the decision that it makes depends on the decision of others, providing a mathematical process to select an optimum strategy against another player with its' strategy (Ray et al., 2014). In automatic negotiation, game theory has been used in the negotiation process of some systems so they can improve and help those systems reach a final decision in the decision-making process (Rajavela et al., 2021; Ray et al., 2014; X. Zheng et al., 2010).

Heuristics are optimization methods that exploit domain-specific knowledge and give no guarantee of finding the optimal solution. They are defined by knowledge of high-quality solutions and rules of thumb, obligating the persons that want to implement a heuristic to understand what makes the quality of the solution for a particular problem what distinguishes high-quality solutions from low-quality solutions (Rothlauf, 2011). In automatic negotiation, heuristics are used to help try to reach an agreement based on the user's preferences and reduce the time in the negotiation step (Kraus et al., 2008; Rossi et al., 2016a, 2016b).

Argumentation is a multidisciplinary field of study containing aspects from philosophy, communication, and psychology and has been widely used in Artificial Intelligence (Rahwan & Simari, 2009). It uses language to justify or refute a stance to secure agreement in views (van Eemeren et al., 2015). In GDSS with a multi-agent system (MAS), argumentation is an asset to them since it facilitates the usage of justifications and explanations in dialogues between decision-makers. The dialogues can influence the preferences of other decision-makers and, consequently, the result of the decision-making process (Carneiro et al., 2012; Carneiro, Alves, et al., 2020; Santos et al., 2010). A taxonomy was proposed (Walton & Krabbe, 1995), where the dialogues are classified based on their primary objective, each participant's aim, and the dialogue participants' initial knowledge. Tab 1 exposes each dialogue class and its characteristics based on the three aspects mentioned previously.

Class	Initial Situation	The objective of each participant	The objective of the dialogue
Inquiry	Lack of knowledge	Assess the integrity of the knowledge	Prove something
Persuasion	Conflict between opinions	Persuade the remaining peers	Reach a consensus
Negotiation	Conflict of interests	Achieve own objectives	Reach an agreement
Deliberation	Predicament of choice	Influence the result	Choose the best course of action
Information Seeking	Unbalanced knowledge division	Obtain or supply information	Diffusion of knowledge
Eristic	Personal conflict	Verbally attack your opponents	Reveal subjacent motives to the conflict and reach a certain degree of accommodation

Tab 1 – Dialogue Taxonomy (Walton & Krabbe, 1995)

After this, argument taxonomy started to consider their emotions, how they could be added to MASs (Norman et al., 2003), and why a new taxonomy was born (Kraus et al., 1998). It defined six argument types that are present in human negotiations and are the following:

- “Threats to produce goal adoption or goal abandonment on the part of the persuadee.”
- “Enticing the persuadee with a promise of a future reward.”
- “Appeal to past reward.”
- “Appeal to precedents as counterexamples to convey to the persuadee a contradiction between what she/he says and past actions.”
- “Appeal to “prevailing practice” to convey to the persuadee that the proposed action furthers their goals since it has furthered others’ goals in the past.”
- “Appeal to self-interest to convince a per”

This taxonomy has been accepted and used in the auto-negotiation process that uses argumentation to negotiate between decision-makers (N. Hamdani & Hamdadou, 2019; Marreiros, Santos, Freitas, et al., 2008; Marreiros, Santos, Ramos, et al., 2008; Matsatsinis & Tzoannopoulos, 2008; Santos et al., 2011).

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that builds a mathematical model based on receiving data, commonly known as training data. This data allows them to make predictions or decisions without explicitly programming it to perform the job, unlike Rule-based engines (Zhang, 2020). ML is slowly starting to be used in the argumentation module of a GDSS to classify the relationship between two arguments given by participants of a decision-making process in a dynamic argumentation framework (Conceição et al., 2021). Their system utilizes dynamic argumentation that consists of a social network where participants can express their opinion about one or more Alternatives and criteria (Carneiro et al., 2016). ML comes into play to classify the direction of such a statement if it is against or in favor of one or more Alternatives or criteria (Conceição et al., 2021). Another way GDSS have been using ML in their favor is through Argument Mining (AM). AM consists of the automatic identification and extraction of the structure of inference and reasoning expressed as arguments presented in natural language (Lawrence & Reed, 2019). AM is helping GDSS to become more attractive to organizations by automatically obtaining meaningful information from unstructured text like aspect terms, aspect category, and polarity detection (Zuheros et al., 2021). This aspect reduces the time participants spend setting themselves up in a GDSS since AM can obtain that information automatically from unstructured text in natural language, also known as his written opinion. This process can be done without the participants needing to interact with the GDSS actively, making that system passive and external help to the participants.

2.2 Data Management

Data Management is collecting, keeping, and using data securely, efficiently, and cost-effectively. Its’ goal is to help people, organizations, and connected things to optimize the usage of data within the rules and policies so that they can make decisions and take actions that maximize their benefit (Oracle, 2021).

The management of digital data involves a vast range of tasks, policies, procedures, and practices, and its' work involves things like (Oracle, 2021):

- Create, access, and update data across a diverse data tier
- Store data across multiple clouds and on-premises
- Provide high availability and disaster recovery
- Use data in a growing variety of apps, analytics, and algorithms
- Ensure data privacy and security
- Archive and destroy data following retention schedules and compliance requirements

This section intends to go through some of the tasks needed in Data Management used in these projects, starting with extraction, then labeling, and ending in preprocessing such intelligent models can use such data.

2.2.1 Data Extraction

GDM needs argumentation to be successful. Therefore, the extraction of data for this context needs to be from a platform that enables the user to have discussions, like forums, passing up on platforms that only allow users to give a quick review on something, removing the argumentation aspect needed for GDM. Platforms like Kaggle[4] and NLP Index[5] offer public datasets to be used, but after a careful search, they only have available quick review datasets from TripAdvisor, IMDB, and other platforms (Cardoso et al., 2022).

So, to obtain this type of data with argumentation, a careful search was done, and three platforms came up as top contenders for getting that type of data. Those platforms, Reddit, Twitter, and Facebook, are presented next.

2.2.1.1 Reddit

Reddit is a popular social bookmarking and microblogging platform that constantly updates user-submitted content such as news, images, videos, blogs, and books. It is home to thousands of communities ranging from breaking news to sports and TV fan theories, always having a community about one's interests. This content is divided into areas of interest called "reddits" or "subreddits" which give the platform stability due to mature moderation processes and subreddits having to be approved manually before being created officially (Barbaresi, 2015; Nguyen et al., 2016; Reddit, 2021). Reddit also allows users to start discussions about any subreddit about a topic of their choosing, leading to argumentative dialogues (Cardoso et al., 2022).

Reddit's official API is free to use but requires approval for commercial purposes (Reddit, 2016). Since its' API does not have a streaming functionality, third-party implementations need to be used for that purpose. Some implementations are the Python Reddit API Wrapper (PRAW),

[4] kaggle.com
[5] index.quantumstat.com

which abides by Reddit's API usage rules (Boe, 2014) or the PSAW[6] library that uses the Pushshift platform, which allows obtaining historical data and real-time data (He, 2021). If a traditional approach is preferred and the discussion is closed, Reddit allows users to add ".json" after the URL, transforming the page into a structured JSON file that can be downloaded.

2.2.1.2 Twitter

Twitter is famous as a social bookmarking and microblogging platform advertised as a service for friends, family, and coworkers to communicate and stay connected by exchanging quick, frequent messages. These messages are called "tweets" and can contain photos, videos, links, or text (Twitter, 2020). People share their opinions by "tweeting" their thoughts, but they are restricted to Twitter's character limit of 280 characters (J. Kumar & Kumari, 2018). That might seem like many characters, but tools like TwitLonger[7] enable users to share even longer messages on Twitter.

Twitter's official API has three types of products Essential, Elevated, and Managed, where the first two are free. They all differ, varying the number of tweets an application can access monthly, the number of app environments, and API version access (Twitter, 2010). Some third-party libraries help in that task, like Tweepy[8] , which uses the official API, and Twint[9] , which does not use the official API, removing the authentication and limits from the official API.

2.2.1.3 Facebook

Facebook is a social networking platform that helps you connect with friends, family, and a community of people who share the same interests (Meta, 2021). Access to Facebook information is limited to approved apps written by registered developers. The information that can be accessed is public posts written by general users and posts on public pages (Franzoni et al., 2017).

To access this public data, the most common tool to do it is Meta's Graph API. It is the primary way for apps to read and write on the Facebook social graph (Meta, 2017). Facebook's social graph consists of nodes that are Facebook objects like photos, pages, or posts connected through edges (photos and their comments on the page). At the same time, fields include information about specific node attributes (Franzoni et al., 2017).

2.2.2 Data Labelling

Machine Learning models and training data are two things that go hand in hand. However, for those models to make decisions and take action, they require a large amount of training data (Appen, 2021). Yet, for those models to be trained to understand specific information, Data Labelling, also known as Data Annotation, is vital. Data Labelling is the process of identifying raw data like images, text, and videos and adding one or more meaningful and informative

[6] pypi.org/project/psaw
[7] twitlonger.com
[8] tweepy.org
[9] github.com/twintproject/twint

labels that provide context and categorization of the raw data. This data must be appropriately organized and annotated for each use case application (Appen, 2021; IBM Cloud Education, 2021). Text annotation is the most common case of Data Labelling, and that consists of highlighting keywords, phrases, or sentences using metadata tags. Text annotation can be done in different forms (Appen, 2021):

- Sentiment Annotation – Consists of assessing attitudes, emotions, and opinions so, in the end, it can provide meaningful insight that could drive serious business decisions
- Intent Annotation – Consists in understanding both natural language and user intent. Since humans are conversing more with human-machine interfaces, machines need to understand the intention of the human behind their speech.
- Semantic Annotation – Consists of enriching the raw data with meaningful and descriptive concepts to increase the depth and meaning of the text
- Named Entity Annotation – This consists of identifying certain entities within the text to detect critical information, such as formal names, places, and brand names.
- Aspect-Based Sentiment Analysis Annotation – Consists of identifying specific entities and their aspects so an opinion can be mined and summarized (Apidianaki et al., 2016). This type of annotation combines others mentioned before, like named-entity for identifying entities and sentiment annotation to determine whether an opinion is favorable based on the emotions it transmits.

In an area where accuracy means everything and inaccurate labeling can lead to misinterpretation and difficulty understanding words in a particular context (Appen, 2021), two different ways of doing Data Labelling exist doing it manually or automatically. These two different ways of doing Data Labelling are explained next.

2.2.2.1 Manual Data Labelling

Manual Data Labelling consists of asking a person to make judgments about some unlabeled data and hand-labeling it according to a set of rules, where a person doing that task is called a “labeler” or “annotator” (IBM Cloud Education, 2021). This way of labeling data may seem outdated, but the vast majority of labeling is still done manually. Hand-labeling data has significant issues, like bias and lack of explanation or interpretation. Also, the costs associated with it, be it financial costs and time that needs to be dedicated to the task, are not perfect manually labeled datasets. Most known hand-labeled datasets have error rates of at least 5% (Datagen, 2021; Mohanty & Bowne-Anderson, 2021).

In some types of annotation, like Named Entity and Sentiment Analysis, manual labeling is starting to get outpaced by machine learning models that do that task more efficiently in terms of time and cost. However, in the case of Aspect Based Sentiment Analysis, the job is still done manually since there is a lack of datasets in that area to train a machine to learn models is an issue. There is a workshop that incentivizes teams from all over the world to join them in the tasks they create, in which Aspect Based Sentiment Analysis annotation is one of them. In the edition of 2016, 29 teams made 245 submissions in seven domains and eight languages, which resulted in 19 training and 20 testing datasets (Pontiki et al., 2016). This annotation methodology might not seem enough to create a global machine-learning model that can automatically annotate the raw data since there are many domains on which an opinion can be made. It is a nice start to move this type of annotation away from being done manually.

2.2.2.2 Automatic Data Labelling

Machine learning models are being used to label data automatically. This automation helps reduce the cost and time it takes to label raw data compared to the previously mentioned method (SageMaker, 2019). It requires Human-in-the-loop (HITL) to combine manual and automatic data labeling by giving the human labelers the power to create, train, fine-tune, and test machine-learning models. They guide the process by deciding what datasets can be given to machine learning models most applicable to a given project (IBM Cloud Education, 2021). These models are first trained on a subset of the raw data that humans have labeled. Considering the model's confidence in the results, it can start automatically applying labels to the rest of the raw data when it has high confidence. However, when it has low confidence in the results, it passes the data to human labelers. Over time, with the model increasing its' confidence by receiving labeled data from the labeler whenever it is not sure about the results, the ability to automatically label the next set of raw data improves. Ultimately, it can label more and more data automatically and reduce the time taken to create training datasets (IBM Cloud Education, 2021; SageMaker, 2019).

An example of this method is the Amazon SageMaker Ground Truth[10] , which uses neural networks with active learning to improve their models. Still, they require a massive amount of data, imposing a minimum of 1250 data objects to use automated data labeling and suggesting the usage of the system with a minimum of 5000 data objects, providing their system for single-label text classification and others (SageMaker, 2019). Another one is TagTog[11] , a tool focused on text-based labeling, providing a platform to, for example, manage the work of labeling text manually or taking it into machine learning models to optimize the task. It supports machine learning, dictionary annotations, multiple languages, and formats, enabling team collaboration and quality management (Kaewsanmua, 2021). There are more straightforward options for more focused tasks like pre-trained models for named entity recognition like Spacy[12] and Stanford NLP[13] or sentiment analysis like TextBlob[14] and VADER[15].

By using this annotation way, it combines the best of the two worlds, reducing the financial and time cost. However, this is not a perfect method since the models are initially trained on data labeled by humans, which can also lead to the models having bias.

---

[10] aws.amazon.com/sagemaker
[11] tagtog.net
[12] spacy.io
[13] nlp.stanford.edu/software/CRF-NER
[14] textblob.readthedocs.io/en/dev
[15] github.com/cjhutto/vaderSentiment

2.2.3 Data Preprocessing

Natural Language Processing (NLP) is an area of research and application that tries to understand how computers can interpret and manipulate natural language text or speech to do valuable tasks (Chowdhury, 2003). Data preprocessing is a crucial step in text mining, a category of data mining to that NLP belongs. Since text documents are unstructured by nature, to properly use them in NLP, suitable preprocessing is needed to transform and represent those text documents in a more structured way so they can be used later on (Denny & Spirling, 2018). Besides that, it increases the quality of the final results when applied to classification problems, Clustering, and other issues (Ramasubramanian & Ramya, 2013; Weiss et al., 2004).

Online user-generated content, such as forums and social media discussions, is increasingly important since it can provide essential knowledge to companies and organizations. However, this type of content has lots of noise, such as abbreviations, non-standard spelling, a specific lexicon of the platform, and no punctuation. These problems reduce the effectiveness of NLP tools, hence why data preprocessing is needed (Čibej et al., 2016).

Next, a few data preprocessing tasks that can be used in text documents for NLP tools to be able to use those is explained

2.2.3.1 Sentence Segmentation

Sentence segmentation, also known as Sentence boundary detection and Sentence boundary disambiguation, consists of segmenting large paragraphs and documents into a sentence's fundamental unit of text processing. This task is usually the first step in many NLP pipelines since unstructured text rarely has marked sentence boundaries, and, in many languages, punctuation serves as a boundary delimiter. This dependency on punctuation can be flawed since punctuation can be ambiguous, like using acronyms and abbreviations (Wicks & Post, 2021). Therefore, the challenge lies in deciding where a sentence starts and ends.

2.2.3.2 Lowercasing

Lowercasing text data can seem like a simple task, but it can be a highly effective form of preprocessing. It can be used in most NLP problems, but in named entity recognition (NER) problems or Sentiment Analysis (SA), preserving the capitalization of words can help these models detect more accurate entities since capitalization indicates high relevance (Ganesan, 2019; S. Kumar, 2019).

2.2.3.3 Stop word removal

Stopwords are a type of word that does not have any linguistic value. Since they are considered low information, removing them allows for focusing on the essential terms of a text document (Ganesan, 2019). This task requires a list of stopwords to remove specific for each natural language, and this list is already compiled (S. Kumar, 2019).

2.2.3.4 Stemming

The stemming technique consists of reducing inflection in words to their base form. This base may not be the actual base of the word. It can be a canonical form of the original word since this task uses crude heuristics that removes the end of terms in hopes of accurately obtaining the root form, which sometimes does not work that well (e.g., troubled becomes “troubl” instead of trouble). However, not everything is terrible about this task since it helps deal with sparsity issues and standardizes the text document’s vocabulary (Ganesan, 2019).

2.2.3.5 Lemmatization

Lemmatization works similarly to Stemming in terms of reducing inflected words into their root form but varies in the fact that it tries to do it correctly without crude heuristics, making sure the word that resulted from the lemmatization (lemma) belongs to the language. By doing this, it becomes slower than Stemming in doing the same task, and some papers refute the usage of Lemmatization since they proved it has no significant impact on the accuracy of text classification with neural architectures (Ganesan, 2019; S. Kumar, 2019).

2.2.3.6 Normalization and Tokenization

Normalization and Tokenization are used to break a text document into tokens, commonly words, for easier text manipulation (Weiss et al., 2004). It includes all sorts of lexical analysis steps like removing punctuation, number, accents, extra spacing, removal or conversion of emojis and emoticons, spelling correction, removal of URLs and HMTL characters (Denny & Spirling, 2018; Ganesan, 2019; S. Kumar, 2019).

2.3 Intra-sentence segmentation

While Sentence Segmentation has tools like Gensim[16] and OpenNLP[17], they do not serve the purpose of intra-sentence segmentation since those tools' objectives are to detect when a new sentence starts and when it ends and divide a complete text of a document into sentences.

Long sentences are a critical problem in machine learning, especially in machine translation, due to their high complexity (S. D. Kim et al., 2000, 2001). In the GDM context, it is essential to divide a written argumentative opinion of a participant into significant bits that can portray what criteria or Alternatives a participant uses since one opinion can have multiple arguments about multiple Alternatives using various criteria. Intra-sentence segmentation consists in doing that division task. Intra-sentence segmentation consists of two steps: initially, it identifies potential segmentation positions in a sentence, and after, it is decided on the actual segmentation position among the possible ones. Accurate segmentation is crucial because incorrect segmentation can lead to a wrong parse tree or parsing failure (S. D. Kim et al., 2001). Some tools are available, and those are explored next.

[16] pypi.org/project/gensim/
[17] opennlp.apache.org/

2.3.1 ClauCy

ClauCy is a framework that implemented the ClausIE information extraction system using Python combined with SpaCy (Chourdakis & Thornton, 2021). ClausIE is a clause-based approach to open information extraction, which extracts relations and their arguments from natural language text. It separated from the rest because it divided the detection of valuable information expressed in a sentence from their representation in terms of extractions, exploiting linguistic knowledge about English grammar. First, it detects clauses and then identifies the type of clause according to the grammatical function of its constituents. It is based on dependency parsing and a small set of domain-independent words, operating sentence by sentence without any post-processing in addition to not needing any training data, independently if it was labeled or not (Del Corro & Gemulla, 2013).

ClauCy differs from ClausIE since it uses a different dependency parser, ClausIE uses Stanford Dependencies, and ClauCy uses SpaCy dependencies. This change in the dependency parser led to the improvement of the separation of embedded clauses and the ability to inflect verbs so that they can become helpful when generating propositions in the text (Chourdakis & Thornton, 2021).

2.3.2 Split and Rephrase

Sentence Split and Rephrase aims to transform complex sentences into several simple ones with their meaning preserved. There is no deletion and no lexical or phrasal simplification. The systems in the area need to learn how to split complex sentences into shorter ones and the respective syntactic transformations required by dividing the complex sentence (Narayan et al., 2017a). Narayan *et al.* (2017a) created a dataset called WebNGL[18] that is now used as a benchmarking dataset for models of this area and tested multiple models with it to see what would perform the best. With the tests, the model with the best results was a Seq2Seq model, also known as encoder-decoder, that consists of a Recurrent Neural Network (RNN). It converts a source sentence to a dense, fixed-length vector representation (encoder) that is then passed into another recurrent network (decoder) to convert that vector to a target sequence. The base Seq2Seq did not perform that well and only got better after the model learned a simple sequence-to-sequence that, given an RDF triplet, generates a text that is the more straightforward sentence that the complex sentence should become, making it supervised learning. An RDF (Resource Description Format) triplet is a triplet in the form of “subject|property|object” where the subject is a URI (Uniform Resource Identifier), the property is a binary relation, and the object is either a URI or a literal value such as a string, a date or a number (Narayan et al., 2017a).

A few years later, Guo et al. (2020) improved the best model of Narayan (2017b) since they identified that Seq2Seq learning had two significant limitations. The first is that it does not

[18] github.com/shashiongithub/Split-and-Rephrase

consider the facts stated in a long sentence, resulting in generating simple sentences that can miss or inaccurately say the points of the original sentence. The second is because the simple sentences derived from the long sentence might be in any order. This order variation may confuse a Seq2Seq model during training. To overcome this, they proposed a Fact-aware Sentence Encoding which enables the Transformer model to learn facts from the long sentence using a multi-task learning paradigm. The model does not only train in sentence splitting and rephrasing but training in judging if a given fact is true or false. This improves the precision of sentence split in addition to Permutation Invariant Training was introduced to find the best permutation of the simple sentences in the reference that yields the minimal loss for avoiding learning against the previously learned patterns hence alleviating the effects of order variance in seq2seq learning for this task.

2.3.3 Shallow Parsing

Shallow parsing, also known as chunking or light parsing, is a type of sentence analysis in which the essential elements of a sentence, like nouns, verbs, and adjectives, are first identified. Then they are associated with higher-order units with specific grammatical meanings like noun groups, verb groups, and phrases. Initially, the most straightforward approaches linked constituent parts based on basic search patterns specified by regular expressions. Recent systems started using machine learning techniques that can take contextual information into account and create chunks that reflect semantic relations between the essential elements of a sentence (Jurafsky & Martin, 2021).

Chunking can be broadly classified into two types, up and down. Chunking up means the lack of deep dive into the data, getting just an overview of the information and, therefore, a brief idea of the data or chunking down, which gives more detailed information (Nithyashree, 2021). Chunking is usually done using supervised learning, training a BIO sequence labeler, which consists in tagging the beginning (B) and inside (I) of each chunk type and tagging outside (O) for tokens outside a chunk. Since annotating BIO tags is expensive and time-consuming, chunking started using Part-of-speech (POS) tagging to obtain the syntactic phrases, also known as chunks, from the entire parse constituents of a sentence. Utilizing a POS tagger makes chunking prone to error since it relies on its accuracy (Jurafsky & Martin, 2021). POS tagging is explained next, as well as the process afterward.

2.3.3.1 Part-of-speech (POS) tagging

In any natural language, words are organized into grammatical classes or parts of speech, with all languages having at least the nouns and verbs categories. The number of categories a particular language has is not exact and depends on how the language is analyzed by a linguist (Weiss et al., 2004). Therefore, Part-of-speech (POS) tagging assigns grammatical categories to terms in a text depending on their definition and context (Escudeiro, 2012). One problem these systems have is the possibility of the same word belonging to multiple grammatical categories depending on the context, which can lead to errors in the tagging process. There are numerous approaches to this system, as shown in Fig. 8.

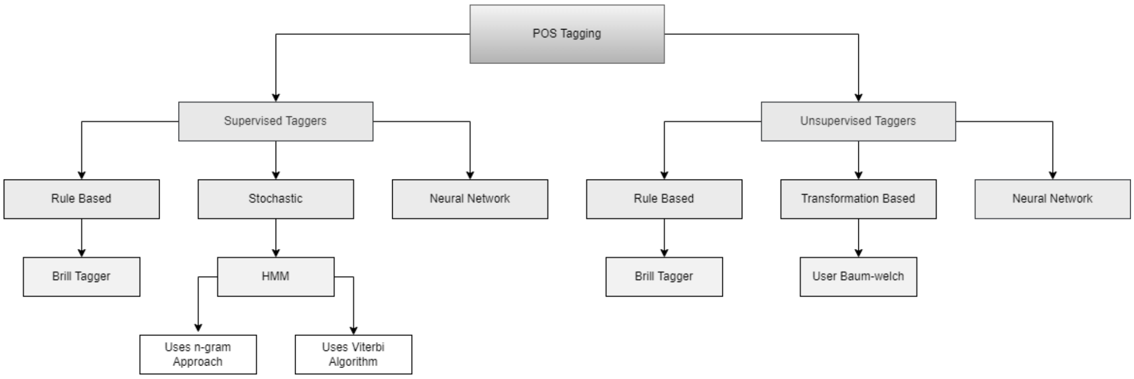


Fig. 8 – POS Classification, adapted from (Kumawat & Jain, 2015)

The tags a POS tagger depends on the tag set it follows. The most famous one is widely used in the Penn tag set from the Penn Treebank project, which is the most extensive set with 36 categories constructed from the Wall Street Journal dataset (Weiss et al., 2004). In the opposite part of the scale, Petrov et al. (2012) proposed a simplified tag set, calling it the universal tag set[19], with only 12 categories, removing the distinctions of verbs and nouns subtypes, for example. Therefore, depending on the necessity of the task, whether it needs broad tags or exact ones, there is a tag set for everyone.

2.3.3.2 Chunking

So, after obtaining the tags of a long sentence, a lot can be done with them to achieve the chunking task. Next, some options for using these POS tags to accomplish the chunking task are presented.

RegexParser. This package in NLTK[20] verifies if POS tags satisfy a regular expression pattern, also known as grammar, which can be defined for the specific task. There can be multiple chunking tags in the grammar. I believe there are two ways of using this package, the first being to extract meaningful information that constitutes the chunks using a Subject-Verb-Object (SVO) approach similar to (D'Souza, 2017). Other options are Noun Phrase (NP) or Verb Phrase (VP) approach, similar to (Orăsan, 2000) or (Attardi & Dell'Orletta, 2008). It is known as **Constituency Parsing** (D. Sarkar, 2018). The second way would be removing non-meaningful information from the sentences, breaking the sentences into chunks naturally. This way would consider the removal of commas, dots, and semicolons in addition to the words tagged as CC from the Penn tag set that is the coordinating conjunctions and includes words like "and", "but", "or".

NgramTagger. An N-gram is an N sequence of words with a minimum of N being two, and this approach uses methods from the NLTK library[21,22] where it creates WTC triples (word, POS tag, chunk tag) to train NgramTagger models. One usage of this approach was training a

[19] github.com/slavpetrov/universal-pos-tags
[20] nltk.org/api/nltk.chunk.html?highlight=regex
[21] nltk.org/api/nltk.tag
[22] nltk.org/api/nltk.chunk

TrigramTagger as a chunker (Bogdani, 2016) or training a BigramTagger with a UnigramTagger as the backoff tagger (Bachani, 2020).

Classifier-based tagger. This package in NLTK[23] uses the BIO tags and the POS tags to train a sequential tagger model that uses a classifier to choose which tag for each word of the sentence can be used for chunking. This model can contain a backoff tagger if the ClassifierBasedTagger cannot determine a label for a given term (Bogdani, 2016).

Dependency Parser based tagger. This approach uses dependency-based grammar to analyze and infer structure, semantic dependencies, and relationships between words in a sentence. In any language sentence, all words except one have some relationship or dependency on other words in the same sentence. The word that has no dependency is called the root of the sentence in which the verb is usually the root. All the other words are directly or indirectly linked to the root using links, which are dependencies (D. Sarkar, 2018). Spacy allows using this with their two English dependency parsers[24] or this approach efficiently. Using BIO tags on top of the Dependency Parser results in a sequential tagger model that predicts BIO tags, which can be used for chunking (Lacroix, 2018).

2.4 Clustering

Clustering is the decomposition of an entity set into “natural groups” in which these groups capture the data's natural structure. There are two significant points to Clustering, the first being the algorithmic issues on how to find such data decomposition and the second being the quality of the computed decomposition (Gaertler, 2005). Clustering was initially introduced in data mining research as an unsupervised classification method to transform patterns into groups (Gaertler, 2005). Later, it was expanded into other fields like information retrieval (Bordogna & Pasi, 2011) and text summarization (Deshpande & Lobo, 2013). Its’ concern is to group a set of entities that are similar to each other from entities that belong to different groups (Bramer, 2007). In the case of intra-cluster density versus inter-cluster sparsity (Gaertler, 2005), the objective is to minimize intra-cluster distances and maximize inter-cluster distances (Fig. 9). Other paradigms exist like the Density-based paradigm, which is similar to human perception since we are used to grouping things into categories in our daily life (Gaertler, 2005).

[23] nltk.org/api/nltk.tag.sequential
[24] spacy.io/api/data-formats#section-dependency-parsing

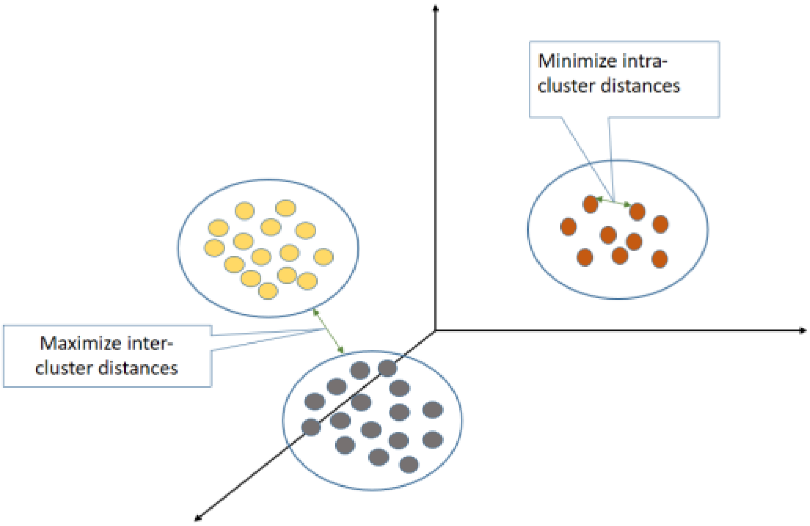


Fig. 9 – Clustering and its' basic paradigm (Villegas et al., 2018)

The main requirements a Clustering algorithm should have are (Matteucci, 2022):

- Scalability
- Dealing with different types of attributes
- Discovering clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to the order of input records
- High dimensionality
- Interpretability and usability

Clustering can be subdivided into two subgroups: hard and soft. In hard Clustering, each data point is clustered or grouped into one cluster. Each data point may either belong entirely to a cluster or not, while soft Clustering uses probabilities to indicate the degree of likelihood between a data point and a cluster, allowing a data point to be grouped into multiple clusters (S. Kumar, 2021). The visual representation of this difference between both can see in Fig. 10.

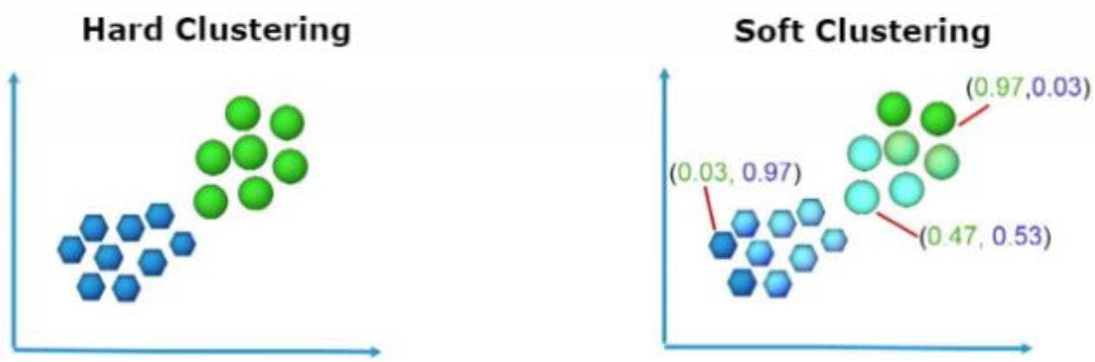


Fig. 10 – Hard Clustering versus Soft Clustering (S. Kumar, 2021)

Clustering can be done using many types of Clustering that use earlier paradigms and techniques, which are explained next. Afterward, an insight into Clustering in natural language processing is done since it is one of the objectives of this project.

2.4.1 Types of Clustering

The multitude of Clustering types varies in how they divide the data into clusters. Applying a Clustering type should conform with the problem it is trying to solve. Some of the most common approaches are shown ahead.

2.4.1.1 Hierarchical

According to Sonagara & Badheka (2014), hierarchical Clustering involves building a cluster hierarchy using a tree of clusters, commonly known as a dendrogram. There are two basic approaches to hierarchical Clustering:

- Agglomerative – Understood as a bottom-up approach, it begins with points as individual clusters and, at every step, merges the most similar or nearest pair of clusters, needing a definition of cluster similarity or distance.
- Divisive - Understood as a top-down approach, it begins with one cluster gathering all the data in it, and at every step, it splits the cluster until singleton clusters of individual points stay, needing at every step, a decision on which cluster to separate and how to perform the split.

2.4.1.2 Partitional

Partitional Clustering seeks to obtain a single partition of the data into a fixed number of clusters, utilizing an adequacy criterion function to optimize the partition (Carvalho & Lechevallier, 2009). An initial allocation of objects to clusters is followed by reassignment to new groups based on a measure of proximity between each object and each group, the adequacy criterion function. The process continues until all things have been assigned to their closest groups (Bogomolov et al., 2019).

2.4.1.3 Density-based

Reiterating what was mentioned earlier, this approach is popular because it resembles the human perception of things since we used to group items into categories naturally. It is a nonparametric approach where data objects are spread through data space over a contiguous region of a high density of objects. The clusters are considered high-density regions and are separated by having adjacent regions of low-density things between them, making full use of the density concept (Campello et al., 2020).

2.4.1.4 Grid-based

The grid-based approach uses a single uniform grid mesh to cluster the entire data into cells, getting represented by the cell using a set of statistical attributes from the data objects within the cell. After this, Clustering is applied to the grid cells instead of the data itself, reducing the task time since the grid size usually is smaller than the number of data and improving the processing speed (Liao et al., 2004).

2.4.2 Clustering Techniques

Following the types of Clustering shown previously, some techniques are derived from them, and some are presented now.

2.4.2.1 K-means

K-means is the most known Clustering technique widely used in multiple fields. It is a partitional type of Clustering published by Forgy (1965), and then a more efficient version of it was proposed and published by Hartigan (1977). This method works with a distance function between data points to decide the number of clusters needed (k). After this, the algorithm begins by selecting k points as starting centroids, also known as centers of the clusters, and then iteratively does these two steps. Each is the assignment step for assigning the remaining data points to the closest centroid by using the distance function indicated initially and the update step to calculate a new centroid which is the mean of all points in a cluster done to all clusters. After each iteration, the centers of the clusters move slowly, reducing the total distance from each point to its assigned center until it reaches convergence, when there are no changes to the clusters even after both steps are done in an iteration. Since this technique depends on the selection of the initial centroids for its' results and it is a hard-Clustering technique (one data point can only be in one cluster), in some cases, this can be seen as a problem. However, it is an effective technique used widely in multiple fields, becoming an excellent all-around Clustering technique (Iliassich, 2016).

K-means++ and k-means variants. K-means++ and Adaptative k-means both followed the same principles of K-means and were created to overcome the problems of the original method. **K-means++** was created by Arthur & Vassilvitskii (2007). The goal is to disperse the initial centroid by assigning the first centroid randomly and then choosing the rest of the centroids based on the maximum squared distance, pushing the centroids as far as possible from one another (Iliassich, 2016). On the other hand, **Adaptative k-means** was created by Bhatia (2004) and overcame the problem of the original variant by allowing the partitioning of a given data set without having to depend on the initial centers' identification. His technique is based on rearranging the clusters to reflect the partitions better when new elements are added. In addition, some clusters can be merged or created new. It can be based on a specified threshold, in which case the number of clusters is unknown until all the elements have been clustered, turning both ways of using this technique adaptive and hence the name of it (Bhatia, 2004). **Ckmeans**, Consensus K-Means, was created by (Monti et al. (2003). It consists of an unsupervised ensemble Clustering algorithm, combining multiple K-Means Clustering executions. Each K-Means is trained on a random subset of the data and a random subset of the features. The predicted cluster memberships of each single Clustering execution are then combined into a consensus matrix, determining the number of times each pair of samples was clustered jointly over all Clustering execution (Monti et al., 2003).

2.4.2.2 Fuzzy c-means

Fuzzy c-means (FCM), also known as soft Clustering or soft k-means, was initially published by Dunn (1973) and improved by Bezdek (1981). This technique diverges from the original by doing soft Clustering instead of hard Clustering, allowing data points to be in multiple clusters. It utilizes the likelihood or probability score of each data point. It consists of three steps: fixing the value of the number of clusters and selecting a value to initialize the partition matrix. After the centroids are calculated and then the partition matrix is updated. These three steps repeat until it reaches convergence (S. Kumar, 2021).

2.4.2.3 Expectation-Maximization

Expectation-Maximization (EM) algorithm is a partitional Clustering technique that was published by Dempster et al. (1977) and later on, Wu (1983) corrected the flawed convergence analysis made initially. It consists in calculating for each data point the probability of it belonging to each existing cluster (Expectation(E)-step). It can have more than one acceptable cluster, creating a multivariate Gaussian probability distribution over the existing clusters, making it a soft-Clustering technique. After the E-step, the Maximization(M)-step recalculates the parameters of each cluster. These parameters are the centroid (mean), covariance that enables elliptical clusters, and weight, that is, the size of the cluster. It does that by using the assignments of points to the previous set of clusters and weighting each data point by its' probability of belonging to the cluster obtained during the last step. Alternating between both phases increases the total log-likelihood until it converges, finding the maximum likelihood (Iliassich, 2016).

2.4.2.4 Girvan-Newman

The Girvan-Newman algorithm is a hierarchical Clustering technique published by the authors that named it (2002) and consisted in detecting communities by progressively removing edges from the original network. In this case, a network is a connected graph. The connected components of the network that remain are called communities, which can also be known as clusters, allowing us to perceive that the objective of this algorithm is not to decide what edges are the most central in which community. Instead, it focuses on determining what edges are most likely to be between communities. It uses Vertex Betweenness to indicate highly major nodes in networks. It expands this concept to edges (Edge Betweenness), which comprise the number of shortest paths between pairs of nodes that run along it. Edges with high edge betweenness in connecting communities are removed, revealing the underlying community structure of the network (Girvan & Newman, 2002).

2.4.2.5 Affinity Propagation

Affinity Propagation is a partitional Clustering technique that was first published by Frey & Dueck (2007), and its' concept is based on creating clusters by sending messages between data objects until convergence. It does not require a predetermined number of clusters before running this technique, using instead parameters like preference which controls how many clusters are used. It also uses a damping factor in attenuating the responsibility and availability of messages to minimize numerical oscillations that can happen when updating these messages. The input data is described using a small number of exemplars, which are members of the initial data input representative of clusters. The messages sent between pairs of data objects represent the suitability of one data object to be the exemplar of the other, which is updated in response to the change of the values in different pairs, happening iteratively until convergence. Finally, finishing the Clustering task, the final exemplars are chosen (Dey, 2019).

2.4.3 Clustering in Natural Language Processing

The next crucial step to achieve the Clustering task in Natural Language Processing (NLP) is to transform said data into numerical data to be applied to Clustering techniques. Word Embeddings convert text data into numerical representation, the so-called Vectorization. According to Goldberg (2017), Word Embedding, also known as Distributed Representations of Words, is the term used to represent the technique where individual words are meant into real-value vectors. These vectors often have a dimension number in the tens or even thousands scale. Each word is mapped to one vector, representing a sentence in a list of these vectors. The mapping of a word to a vector can be done through dictionaries. This mapping is better than using sparse word representations in the thousands or even millions of dimensions (Brownlee, 2019). In addition to Word embeddings, Dimensionality Reduction techniques are also advised to improve the Clustering accuracy when handling data with a high number of features, making it advantageous in terms of computational efficiency (Allaoui et al., 2020; Cunningham, 2008).

Next, some of the most used Word Embedding and Dimensionality Reduction techniques are explained, and some approaches to Clustering in NLP are presented.

2.4.3.1 Vectorization with Word Embeddings

TF-IDF. TfidfVectorizer[25] is a method in the scikit-learn framework that enables the conversion of raw documents into a matrix of TF-IDF features. This method performs two tasks, the first being the usage of CountVectorizer[26], which transforms the text into a matrix of token counts, creating a sparse representation of the counts of the words in a text. Afterward, it utilizes the TfidfTransformer [27] that transforms the token count matrix to a normalized TF-IDF representation. TF-IDF is the combination of the Term Frequency (TF) metric that represents the number of times a term occurs in a document versus the total number of terms in a document. In contrast, the Inverse Document Frequency (IDF) represents the number of documents that contain the term (Leskovec et al., 2014).

Word2Vec. Word2Vec[28] is a popular Word embedding technique developed by Tomas Mikolov (2013) while he worked for Google. It provides two methods to achieve this, either by using Continuous Bag-of-Words (CBOW) (Mikolov, Chen, et al., 2013) or the Skip-gram model (SG) (Mikolov, Sutskever, et al., 2013). The CBOW method takes the context of each word as the input and tries to predict the word corresponding to the context, while the SG method inverts the CBOW method, using the target word as the input and trying to predict the context. According to the author, CBOW is faster and has better representations for more frequent

[25] scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer
[26] scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer
[27] scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer
[28] github.com/tmikolov/word2vec

words, while SG works well with a small amount of data and represents rare words well (Karani, 2018). Other than the original GitHub implementation, the Word2Vec method is present in a commonly used framework called Gensim[29] or a less general framework called Magnitude[30]. With a straightforward way to train a Word2Vec model or load a pre-trained model, TensorFlow[31] shows how to fully implement a Word2Vec model that uses its tools.

GloVe. Global Vectors for Word Representation (GloVe) is the unsupervised learning algorithm for this task created by Stanford[32] (Pennington et al., 2014). It is available on GitHub[33], where they supply pre-trained models for the task depending on the requirements and the option of training your model with your own labeled data. Using the pre-trained models means that the GloVe model becomes a static dictionary since we obtain the word and its' vector representation by downloading a pre-trained model (Theiler, 2019). If the intention is going for a machine learning model for the task replicating the GloVe paper, Yan (2021) made one using PyTorch.

fastText. fastText is a library for efficient learning of word representation and sentence classification created by Meta Research[34][35] (Bojanowski et al., 2017). It is available on GitHub[36], where they offer their state-of-the-art model for English word vectors and word vectors for 157 additional languages. It took inspiration from the Word2Vec technique, so they are similar but diverged by this one using subword information on word similarity tasks to improve its results. Besides the original GitHub implementation, like Word2Vec, the fastText method is present in Gensim and Magnitude with a straightforward way to train a fastText model or load a pre-trained model.

BERT. Bidirectional Encoder Representations from Transformers (BERT) is a language representation model released by Google designated to pre-train deep bidirectional representations from an unlabeled text. It does that by jointly conditioning the left and right context in all layers (Devlin et al., 2019). That means it considers the context when creating word and sentence embedding vectors, where the exact two words can have two different vectors (McCormick & Ryan, 2020). In addition to the available pre-trained models, these models can be fine-tuned to create state-of-the-art models in different NLP tasks, as in the model's GitHub page[37].

[29] radimrehurek.com/gensim/models/word2vec
[30] github.com/plasticityai/magnitude
[31] tensorflow.org/tutorials/text/word2vec
[32] nlp.stanford.edu/projects/glove/
[33] github.com/stanfordnlp/GloVe
[34] opensource.fb.com
[35] github.com/facebookresearch
[36] github.com/facebookresearch/fastText
[37] github.com/google-research/bert

SBERT. Sentence-BERT (SBERT)[38] is a modification of the original pre-trained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine similarity (Reimers & Gurevych, 2019).

ELMo. Embeddings from Language Models (ELMo) is a state-of-the-art NLP framework developed by AllenNLP[39]. ELMo's representations differ from traditional ones because each token is assigned a representative that is a function of the entire input sentence. This way, word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus (Peters et al., 2018).

2.4.3.2 Dimensionality Reduction Techniques

PCA. Principal Component Analysis (PCA) was initially invented by Pearson (1901) and later independently developed and named by Hotelling (1933, 1936). It is a statistical process that converts a group of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. All principal components are orthogonal to each other. Each one is chosen in a way that represents most of the available variance, with the first component having the maximum variance in a way that it selects a subset of variables from a more extensive set, based on which original variables have the highest correlation with the principal amount (Akash Kumar, 2022; Kambhatla & Leen, 1997). PCA can be named differently depending on the field of application, whereas in the ML field, it is called PCA and uses Singular Value Decomposition (SVD)[40] (Stewart, 1993).

t-SNE. t-distributed Stochastic Neighbor Embedding (t-SNE) was initially developed by Roweis & Hinton (2002). They created the concept of Stochastic Neighbor Embedding, and later, Van Der Maaten & Hinton (2008) proposed the t-distributed variant. This variant is a nonlinear technique that converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. t-SNE has a cost function that is not convex, meaning that with different initializations, we can get different results (Van Der Maaten & Hinton, 2008). This technique has implementations in multiple technologies, making it widely available[41]. The t-SNE implementation in scikit-learn uses the Barnes-Hut approximation algorithm, which relies on quad-trees or octa-tree, which makes the maximum number of dimensions that can be used with t-SNE three.

UMAP. Uniform Manifold Approximation and Projection (UMAP) was developed by McInnes et al. (2018) with a theoretical framework based on Riemannian geometry and algebraic topology. It is based on three assumptions, the data is uniformly distributed on a Riemannian manifold, the Riemannian metric is locally constant (or can be approximated as such), and the manifold is locally connected. This way, it is possible to model the manifold with a fuzzy topological

---

[38] github.com/UKPLab/sentence-transformers
[39] allenai.org/allennlp/software/elmo
[40] scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA
[41] lvdmaaten.github.io/tsne

structure. The embedding is found by searching for a low-dimensional data projection with the closest possible equivalent fuzzy topological structure (McInnes et al., 2018)[42].

2.4.3.3 Approaches

Kim *et al.* (2018) applied Clustering with NLP in the biology field by extracting data from two diverse sources, microarray gene expression data and gene co-occurrences in the scientific literature from bioRxiv using NLP. After normalizing the microarray data and applying dimensionality reduction with Principal Component Analysis (PCA), they grouped this data into a different number of clusters by using the K-means technique. The resulting clusters were compared to the extracted gene co-occurrences pairs in the NLP data to evaluate the results of the steps taken. The evaluation was done using Entropy analysis on the combined data, comparing it to the maximum Entropy from the sole clusters. Their results approve the usage of NLP in this field to extract gene co-occurrences from the literature in which the usage of Clustering helped confirm this claim.

Sarkar et al. (2018) applied Clustering with NLP as an intermediary step in creating a model to predict occupational accident risk. After extracting the data from an integrated steel plant's safety management system database, pre-processing is done where duplicates, missing data, and inconsistent data are removed. The authors used EM-based text Clustering to build clusters with categorical attributes while using the Silhouette Coefficient to determine the optimal number of clusters. This data is then fed to a Deep Neural Network (DNN) model with a structure comprised of a stacked autoencoder (SAE) with an autoencoder (AE) and a SoftMax classifier. The AE is a feed-forward Artificial Neural Network (ANN) comprising one input layer, one hidden layer, and one output layer. Usually, it is trained to copy its input to its output so that the errors become minimum. Therefore, the dimension of the input must be the same as that of the output. Support Vector Machine and Random Forest was used to compare this approach. For DNN, the grid search technique was used to find the best hyperparameters.

Hema & David (2018) applied Clustering with NLP in the medical field as an intermediary step in creating a model to predict diseases based on symptoms. The data is collected using Medical Forums about various stomach disease symptoms, and an OWL file is created. After data preprocessing, stopwords, special characters, numbers, and white spaces were removed in addition to stemming words. Speech tagging is used to extract verbs, nouns, subjective words, adverbs, and others from the dataset, so afterward, Fuzzy c means can cluster the data into groups of common symptoms. RDF is then utilized for taxonomic relations, object relations, and data, while OWL is used for attribute relations. These relations are then mapped for the Genetic Algorithm to predict the disease of a customer based on the symptoms.

Dragos & Schmeelk (2020) applied Clustering with NLP in education to obtain meaningful information from student surveys. They obtain open-text survey answers from five cybersecurity courses and cluster the answers to each question on the survey. They first select the cluster number based on heterogeneity. This measure represents the sum of all squared distances between data points in a cluster and the centroids. After the number of clusters is

[42] umap-learn.readthedocs.io/en/latest/

decided for each question, they use TF-IDF as word embedding to cluster the data with k-means and obtain categories based on the top keywords made manually. With this, they aim to fill the gap in identifying practical interpretations of student feedback in the literature.

Gupta & Tripathy (2018) applied Clustering with NLP by creating a methodology that could be used in any domain. They tested it in a zoo dataset. The methodology consists of implementing a form of Clustering that takes a non-numeric data set and clusters it with the help of the word embeddings provided by the GloVe dataset by generating the vector representation for each sentence in the dataset of those words. Then a dimensionality reduction is performed on the data set using t-Distributed Stochastic Neighbor Embedding (t-SNE) to obtain the accurate number of dimensions for proper cluster formation. The data is then clustered using k-means++. The only issue with this technique is that it chooses the number of clusters based on minimum inertia and the least number of clusters in total. This difficulty was surpassed by using the Elbow method to decide the number of clusters formed by the algorithm.

Huang *et al.* (2018) applied Clustering with NLP in StackOverflow discussions to mine comparable technologies and opinions. They utilize tags in each discussion, considering the collection of technologies that a person would like to compare. To learn the tags, they compared two of the most used methods, the continuous skip-gram model and the CBOW model, where the first model outperforms the latter by a marginal difference. With this better model, they compared the difference between the number of dimensions and concluded that eight hundred was the one to use with the best accuracy. To obtain categorical knowledge, they run the tags against TagWiki to obtain its' definition and then extract the tag category with a POS tagger. To mine comparative opinions, they extracted comparative sentences using three steps for each pair of comparable technologies in the knowledge base. They first preprocessed the discussion considering only answers with a positive score and removing the punctuation and sentences that ended with question marks because they wanted to extract facts and not doubts. Finally, they lowercase everything to make tokens consistent with the technologies. Secondly, they locate candidate sentences using a large thesaurus of morphological forms of software-specific terms to match with tag names. In the last step, they select comparative sentences and develop a set of sentence patterns considering POS tags to obtain them. They use Word Mover's Distance to measure the similarity between sentences, which is helpful for short text comparison. This approach uses word embeddings to get a dense vector representation of each keyword from POS tags for comparisons like comparative adjectives and nouns, excluding the technologies under comparison. They then compute the minimal distance between keywords between sentences and use those distances in a similarity score. If the similarity is superior to the threshold, they are considered similar. Finally, to cluster representative comparison aspects, they build a graph where each node is a sentence. They use TF-IDF to extract keywords from a comparative sentence in one community to represent the comparison aspect of this community, removing stop words and choosing the top three with the highest scores to represent the community. Each community is regarded as a document.

Y. Liu et al. (2021) applied Clustering with NLP by collecting COVID-19-related data from Reddit in subreddits of North Carolina, utilizing data preprocessing techniques like POS tagging and stop word removal. After this step, word embedders were tested, GloVe and Word2Vec, with the Cosine Similarity measure used to calculate the similarity between words. Topic Modelling techniques and a BERT model were fine-tuned to find people's concerns and critical points from the sentences typed on Reddit's posts. With the results of the last step, K-Means were used to cluster the sentence vectors into three categories, concluding that reopening and spreading the virus were the most discussed topics during the time of the posts gathered.

Reimers et al. (2020) applied Clustering with NLP by testing it in an open-domain argument search context. They measure the quality of contextualized word embeddings, ELMo, and BERT to classify and cluster topic-dependent arguments. In terms of Argument Clustering, twenty-eight topics related to current issues about technology and society were picked. Since argument pairs addressing the same aspect should be assigned a high similarity score and arguments on various aspects a low score, they used a weak supervision approach to balance the selection of argument pairs regarding their similarity. After handling this issue, Agglomerative Hierarchical Clustering with average linkage was used to cluster arguments. They also tested K-means and DBSCAN, but Agglomerative Hierarchical Clustering provided the best results in preliminary experiments.

Färber & Steyer (2021) authors applied Clustering with NLP on the argument search domain to identify arguments in natural language texts. To present aggregated arguments to users based on topic-aware argument Clustering, they tried K-means and HDBSCAN, in addition to considering the argmax of the TF-IDF and LSA vectors to evaluate the results. Regarding word embeddings, TF-IDF, and BERT models, Bert-avg and Bert-cls were used as a pre-step for the Clustering task. Another interesting remark is that they evaluated whether calculating TF-IDF within each topic separated is superior to computing the overall arguments in the document corpus. The dimensionality reduction technique UMAP was tested before Clustering to verify its' performance related to not using it, in which HDBSCAN outperforms k-means on Bert-avg embeddings but using UMAP in combination with TF-IDF results in a slightly reduced performance. They found that Bert-avg embeddings result in slightly better scores than Bert-cls when using UMAP, concluding that this methodology can mine and search for arguments from an unstructured text on any given topic

Dumani & Schenkel (2020) applied Clustering with NLP by creating a quality-aware ranking framework for arguments extracted from texts and represented in graphs. To achieve that, they used a (claim, premise) dataset based on debates taken on online portals in which they used SBERT instead of BERT (Dumani et al., 2020) to obtain the embeddings of the claims and premises. With these embeddings, agglomerative Clustering using Euclidian distance metric and average linkage method was applied to achieve the Clustering task. Since the dataset was sizeable (400k) with many dimensions from the embedder (1024 dimensions), they clustered the dataset with the agglomerative technique. That reduced the time it would take to cluster with the agglomerative technique by using K-means for K=4 beforehand. Then they used agglomerative Clustering on the results of K-means.

Daxenberger et al. (2020) applied Clustering with NLP to the Argument Mining field by creating an argument classification and Clustering project for generalized search scenarios. For that, the technology mines, and clusters arguments from various textual sources for a broad range of topics and in multiple languages, generalizing to many different textual sources ranging from news to reviews. After fine-tuning a BERT base model, since it out-performs the pre-trained variant by a good margin, the embeddings obtained by this model are sent to the agglomerative hierarchical Clustering with a stopping threshold, aggregating all arguments retrieved for a topic into clusters of aspects.

2.5 Technologies for Data Science and Web service frameworks

R and Python are two of the top data science languages. Both are open-source and have a large user base that can help with problems faced by new or even experienced developers of those languages (Bansal, 2020). Both these languages are presented next, in addition to some of their respective web service frameworks.

2.5.1 R

Developed by Lucent Technologies in the Bell Laboratories, R is a language and environment that provides a wide variety of linear and nonlinear modeling, time-series analysis, classification, Clustering, and graphical techniques, becoming highly extensible. It provides ease in creating well-designed publication-quality plots that can include mathematical symbols and formulas when needed. The R environment is an integrated suite of software libraries for data manipulation, calculation, and graphical display. It includes efficient data handling and ease of storage, operators to carry out operations on full matrices or arrays, and a well-developed, simple, and effective programming language that has all the things the other languages offer, like conditional statements, loops, and recursive functions. R can be extended by using packages found in the R distribution or CRAN family of websites that cover a wide range of modern statistics, allowing it to be used for text mining operations and the implementation of web services with the scripts created (Johnson, 2022; The R Foundation, 2021). Some of the existing web service implementation solutions are explained next.

2.5.1.1 Shiny

Shiny is an R package developed by RStudio that makes it easy to build interactive web apps straight from R. It can host standalone apps, embed them in R Markdown documents, or build dashboards. It also allows you to extend it with CSS themes, HTML widgets, and JavaScript actions. It promises an easy way to write apps with no web development skills required (Shiny, 2017). Its' features are an intuitive and extensible reactive programming model that makes it easy to transform existing R scripts into web apps. The output automatically reacts to new user inputs, and tools for improving and monitoring performance are available. Native support for async programming, caching, load testing, unit testing, input validation, and more while bookmarking the application state or generating code to reproduce output(s). It is open source and constantly updates new features and bug fixing (RStudio/Shiny, 2021).

2.5.1.2 Phoenix Server

Phoenix Server is a web development framework for R developed by Revolution Analytics that can be hosted locally or in the cloud. With scalable R Servers with load balancing, it offers a RESTful API with a management console, standardized XML/JSON interfaces and object encoding, and Stateful and Stateless code execution while separating the R/statistical programming from the web development (Ooms, 2010).

2.5.1.3 RServe

RServe is a binary server developed by Simon Urbanek (2020) that allows other programs to use R facilities without having to initialize R or link against the R library. Every connection is separate from the rest, having its' own workspace and directory. Client-side implementations that use RServe are available for other popular programming languages like JavaScript, Java, or PHP. It supports remote connection, authentication, and file transfer commonly used to integrate the R backend for the computation of statistical models, plots, and more in other applications.

2.5.2 Python

Created by the company that named this programming language the Python Software Foundation, Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It is built in high-level data structures, combined with dynamic typing and binding, making it viable for rapid application development and an intermediary language to connect existing components. With simple and easy-to-learn syntax that shines in readability and reduces maintenance costs, Python supports modules and packages to allow for program modularity and code reuse (Python Software Foundation, 2022). Like R, Python has an environment built around it, with many software libraries for tasks like automation of tasks, data analysis and visualization, creation of websites and software, and more (Coursera, 2021). Some of the existing web service implementation solutions are explained next.

2.5.2.1 Django

Django is a full-stack, high-level Python web framework that encourages rapid development and clean, pragmatic design. Since it is full stack, it includes dozens of extra tools for everyday web development tasks like user authentication, content administration, and others. It is scalable and can quickly and flexibly deal with heavy traffic. It also offers insurance in terms of helping developers to avoid common security mistakes like leaving the website exposed to SQL injection, cross-site scripting, and request forgery (Django Software Foundation, 2018).

2.5.2.2 Flask

Flask is a Python micro web framework based on the WSGI library that offers the same flexibility as Python and a straightforward way to do web development. It is designed to be easy and quick to start, with the possibility of scaling it up to a complex application. Flask does not enforce any dependencies or project layouts and only offers suggestions, allowing the developers to decide which tools and libraries they want to use. Since it is a microframework, it does not offer graphical interface tools. However, since it is easily expandable, the community provides many extensions to make adding needed functionalities easy (Flask, 2010).

2.5.2.3 CherryPY

CherryPy is an object-oriented Python micro web-framework that allows developers to build web applications the same way they would build any object-oriented Python program resulting in minor source code done in less time. With more than 10 years of experience, CherryPY has proven to be fast and reliable, getting used in producing many website backends, from the simple ones to the most demanding (The CherryPy Team, 2016).

2.5.2.4 Bottle

The Bottle is a fast, simple, and lightweight WSGI micro web-framework for Python. It is distributed as a single file module without any dependencies on libraries other than the standard one for Python (Hellkamp, 2018).

2.6 Final Considerations

With this state-of-the-art, we first dove into the GDM context, understanding what it is and the benefits and drawbacks of making decisions in a group. After this, the GDSS concept definition was presented, as well as all its history. They were limited and synchronous until the ones nowadays, based on the web and ubiquitous. From this evolution through time, we got into the methods that a GDSS can use to solve their problems: mathematical aggregation of preferences or automatic negotiation. From this study of the available methods, an understanding was made that argumentation is the most used nowadays. Multi-agent systems play a crucial role in this method, where agents can be modeled based on the participant it represents, considering the participant's personality, mood, and behavior styles. Apart from this, argumentation modules started using Machine Learning to improve their capabilities, and Argument Mining is slowly getting adapted into these systems to take them to the next level. Next, we dove into the Data Management field, where initially Data Extraction was presented, showing three platforms that allow discussions with argumentation to be taken on: Reddit, Twitter, and Facebook. It was exposed to how to obtain data from those platforms via official API or other methods. Data Labelling was the next point of discussion, where some types of data labeling were exposed, and manual and automatic labeling was addressed. To end the Data Management part, Data Preprocessing was exposed with multiple tasks that can be performed in it.

Intra-sentence segmentation is the next topic which explains what that concept is and the three tools that allow it to do such a task. ClauCy was the first tool described, which made it look promising due to the GitHub example of the tool in action. However, a quick test with the test phrase "Mark likes apples, Marie likes oranges, and John likes pears" that should result in "Mark likes apples" + "Marie likes orange" + "John likes pears" instead resulted in "Mark liked apples" + "Marie liked liked" + "oranges liked pears" + "John liked pears" which is different from the expected results. Split and Rephrase was the next one to be described, where the concept was described and what it used to achieve that task. Lastly, Shallow Parsing was presented, which needed to present the concept of part-of-speech tagging, which this type of parsing uses. After obtaining the part-of-speech tags, multiple approaches to reach the objective of this task can be used.

After this, Clustering was approached, explaining the theoretical aspect and some of its types and techniques addressed. Finally, Clustering in Natural Language Processing was addressed, where the need for Vectorization was found since it deals with unstructured text, leading to the study of multiple Word Embedding technologies. Afterward, literature approaches were studied to understand better how Clustering is performed with NLP. Kim et al. (2018) approach show an excellent combination of both fields, but it is not what it is intended with this project since the goal is to cluster unstructured text and not structured data, that was, in their case, the microarray gene expression data. Sarkar et al. (2018) approach show one usage of Clustering to categorize unstructured data, finding hidden connections between them. Hema & David's (2018) approach uses many steps that could be utilized in this project. However, having an intra-sentence segmentation step in our project, the usage of fuzzy c-means becomes less needed. Most of the data is treated so it can only be part of one cluster, removing the need for soft Clustering techniques. Dragos & Schmeelk's (2020) approach was applied to education and student surveys, and it seems like it can be adapted to any other field. Both techniques used are not domain specific and the categorization done afterward was manually according to the top keywords, meeting the objectives of this project in terms of Clustering.

Gupta & Tripathy's (2018) methodology sounds interesting on paper, and the possibility of using it in any domain allows it to be adapted to this project to achieve its goals. Huang et al. (2018) approach starts to be more in line with the objectives of our project, especially on comparative opinions mining, which is like our goal of reaching a consensus on the best alternative in a specific problem using criteria to describe the available alternatives. Some aspects of Y. Liu et al. (2021) approach were considered in our work, such as data preprocessing options and word embedders. Reimers et al. (2020), and Färber & Steyer (2021), applied to the field of argument searching, which is close to but not in the same context as our work. Their approaches were used as an example for our project, using context-aware word embedding models (ELMo, BERT, and TF-IDF), the Clustering techniques used, and their tested hyperparameters. The dimensionality reduction aspect brought by Färber & Steyer (2021) is also interesting as it helped obtain better results by reducing the number of features passed to the Clustering techniques. Dumani & Schenkel's (2020) approach brought to attention some interesting points, like the size of the dataset used and which measures could be taken to overcome that. Instead of dimensionality reduction, they used K-means as a Clustering step to reduce the computational time. Daxenberger et al. (2020) project, in terms of argument Clustering, seems promising. They used a fine-tuned BERT model for the word embeddings and utilized agglomerative hierarchical Clustering to obtain arguments divided by aspects, like the one presented in our project.

To conclude, Technologies for Data Science and their respective Web service frameworks were studied. Python was the one decided to be used of the two primary technologies for Data Science since it is one of the most used. Also, the ease of transforming the source code into a web service comes from the existing web service frameworks in Python.

3 Value Analysis

Value Analysis is a collection of techniques, knowledge, and skills for increasing the value of a product by removing unwanted costs or enhancing its' functionalities while maintaining its quality, reliability, and performance. It also entails comprehending the product's components and associated costs (SendPulse, 2021).

This section presents a value analysis of the project to elucidate the value of this project to the client. For this, the author performed the innovation process using the New Concept Development (NCD) model, in which the opportunity is identified and analyzed, followed by the value of the solution to the client and the perceived value resulting in a value proposition. In the end, the author did a functionalities analysis using FAST and an alternative analysis using TOPSIS.

3.1 Innovation Process

Technological evolution keeps bringing the consistent development and introduction of new products and services to the market. Customers value products and services are essential for an organization's growth and prosperity since they bridge the company with the customers' needs (Gupta & Wilemon, 1990). Ensuring the development of a valuable product with constant innovation, a need to create a process is composed of three main steps that characterize the innovation process, as we can see in Fig. 11.

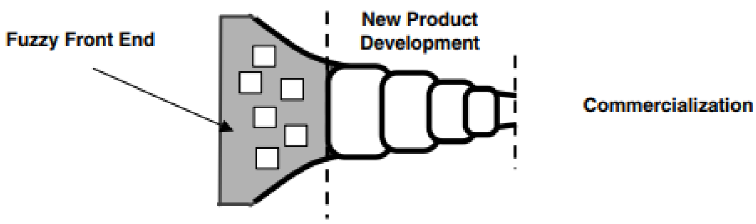


Fig. 11 – Innovation Process (Koen et al., 2002)

The Fuzzy Front End (FFE) is a process of conducting new and experimental research where many ideas come up. The New Concept Development (NCD) model is typically used to describe these steps of FFE. It provides a common ground regarding the definition of crucial components of the Front End of Innovation since the division between both is often less sharp because technology developments may need to be pursued at the intersection of both (Koen et al., 2001, 2002). The final phase of the process is Commercialization, in which the product developed in an early phase is ready for market release enabling the customer's needs to be satisfied (Koen et al., 2002).

The NCD model (Fig. 12) is composed of three main components:

- The engine is considered the central aspect of the business and entails things like leadership, business culture, and strategy that drive the five critical aspects of the company's strategy
- An internal area that defines the five key aspects of a company's strategy (opportunity identification, opportunity analysis, idea genesis, idea selection, concept, and technology development) from FFE
- Influencing factors which consist of organizational capabilities, the external world (law, government policies, customers), and enabling sciences (internal and external), affect the innovation process and are relatively uncontrollable by an organization

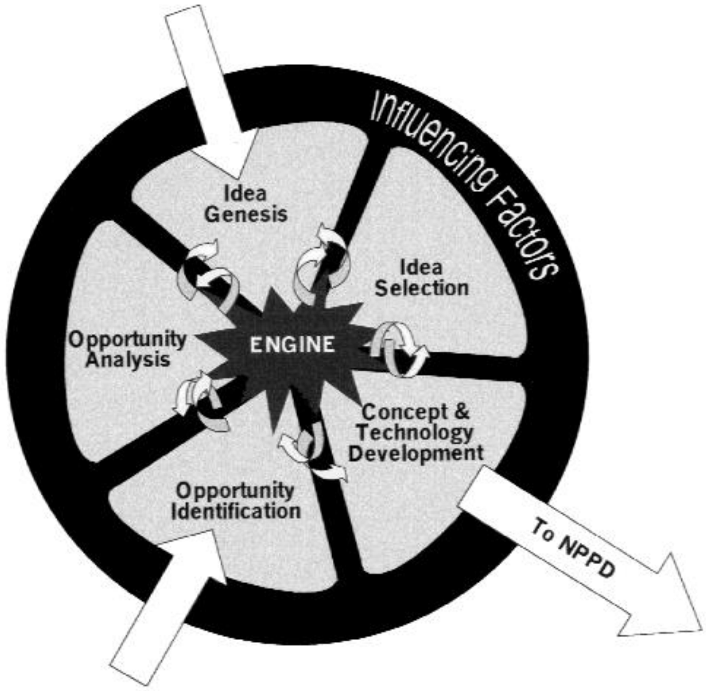


Fig. 12 – NCD model (Koen et al., 2002)

In this project, there is a robust exploratory component associated with it. Even if the final product is a web service, it englobes Machine Learning (ML) aspects which can lead to some uncertainty about the service. Since the development of the web service that contains ML imbued into it, it requires a process of investigation of approaches and algorithms as well as data management and exploration to understand the value of the models and their experimentations. These activities portray the FFE phase and the project inserts into it.

3.1.1 Opportunity Identification

The opportunity identification can be made in two ways, from a GDSS and a Clustering point of view. That is because this project is inserted in a fully intelligent and automatic GDSS that is being developed and because the project's main objective is the application of Clustering techniques in the GDM context.

From a GDSS point of view, we live in a fully connected world because of the advancements of the internet. Multinational companies started to spread worldwide, making traveling a necessity for higher-ups, and decision-making in groups is the most used technique to make decisions nowadays. GDSS is a type of system that wants to help these people daily. It obligated these systems to keep up with the advancements and to evolve into the web, allowing discussion participants to make decisions anytime and anywhere. Still, web-based GDSS is not widely accepted in organizations. Limitations on most web-based GDSS propositions still exist compared to face-to-face meetings, like the necessity to set up their preferences in the GDSS and time that higher-ups do not have because of their busy schedules. Since most participants do not have time to properly set themselves in the GDSS, they will not fully utilize the advantages of group decision-making. This lack of time makes it so the GDSS and its participants lose out on the variety of participants in terms of personality, argumentation style, and ease of communication inherent to face-to-face meetings, which can be more challenging with GDSS if they have complex interfaces.

There is a need for a fully intelligent GDSS that can remove this setup time by analyzing what is being said and obtaining meaningful knowledge from it, working passively in the background while decision-makers discuss the problem. This system should be able to analyze unstructured text or voice to help with the GDM process without the participants' need for any setups, turning this system into a mediator and helper in the GDM problems. This change might make it so that GDSS can be widely accepted in organizations.

Clustering is getting adapted to NLP tasks using Word Embeddings with meaningful results in many fields. However, no approach focused on the GDM context uses unstructured text to cluster the data. Therefore, there is a need to study the usage of this machine learning technique in the GDM context using unstructured text to see the results it can bring, helping to create the fully intelligent GDSS mentioned earlier.

The opportunity identification can be evaluated using the following practical methods, tools, and techniques (Koen et al., 2002):

- Create more opportunities by envisioning the future through
    - Roadmapping
    - Technology trend analysis

- Customer trend analysis
- Competitive intelligence analysis
- Market research
- Scenario planning

3.1.2 Opportunity Analysis

Following identifying an opportunity, an analysis is needed to understand the opportunity better. This analysis can be done in two ways, like the previous step, from a GDSS and a Clustering point of view.

In terms of GDSS, not many utilize unstructured discussions found on the web for their benefit. Initial approaches start to use natural language text in them, converting a decision matrix into natural language based on the alternative, the criteria used to evaluate said criteria. Giving a numerical rating was implemented in car evaluations (Chen et al., 2014) or the opposite route where decision-makers gave their opinion in natural language. It is easier in terms of usability for a participant to use natural language terms like “Agree” or “Disagree” than using numbers to define their opinion. These terms were then converted into numerical values to use in the decision-making system being implemented in a company in the ICT sector that had to prioritize products in terms of investment for the next six months (Cid-López et al., 2017). A more advanced approach started to use Sentiment Analysis models in the systems to improve decision-making, analyzing hotel text reviews in Trip Advisor in combination with ratings to produce results (Bueno et al., 2021). A complete approach was made by Zuheros *et al.* (2021), utilizing Trip Advisor reviews. They created a Sentiment Analysis-based Multi-Person Multi-criteria Decision Making (SA-MpMcDM) methodology to create a more intelligent decision aid. This system could build expert evaluations based on natural language reviews and even numerical ratings.

Similarly to the last approach, this one also used restaurant reviews in Trip Advisor from Tarragona as their use case to show their methodology (Jabreel et al., 2021). While the models that both approaches built with their methodologies could detect aspects, categories, and criteria to subsequently perform sentiment analysis or understand the polarity of the review better so it could achieve the objectives of that system, it is not enough for the GDM context. Since the base of it are Trip Advisor reviews, those lack the discussion and argumentation about a topic, in this case, restaurants. Those reviews are quick, and there is no argumentation to dispute other reviews, which is a crucial aspect of the GDM context to make better decisions.

Another aspect of GDSS that want to achieve intelligence passively without the need for people to input their information is the necessity of specific annotated datasets in multiple domains so that artificial intelligent models are not dependent on a domain. This need for structured and annotated datasets with good quality is present not only in GDSS but the entirety of the Artificial Intelligence area.

In terms of Clustering, as it was analyzed earlier in 2.4.3.3, Clustering of unstructured text with NLP is starting to gather steam in all the fields it can be utilized. Like in education to obtain meaningful information from student surveys or in the StackOverflow platform to compare and rank technologies based on users' opinions on the matter, an approach to the GDM context using this ideology does not exist yet. Existing approaches of Clustering in the GDM context are based on preferences defined by the participants, using either FCM or some variant of it (B. Liu et al., 2019; Palomares et al., 2014; T. Wu & Liu, 2016). Others on a simple k-means due to its' simplicity to implement (Alonso et al., 2009; Q. Liu et al., 2021; Z. Wu & Xu, 2018) leading to a lack of approaches for GDM with unstructured text.

The opportunity analysis can be evaluated using the following practical methods, tools, and techniques (Koen et al., 2002):

- The same methods, tools, and techniques are used to identify future opportunities, but the effort would be expanded in considerably more detail
- Assignment of a full-time specific multifunctional team of three to five people for large projects
    - Creating a charter for the team that points them in the right direction

3.1.3 Idea Generation

An idea is the first stage of a new service (Koen et al., 2014), so this idea considers that many discussions are done daily on the web. Participants in said discussions use text as a means to expose their opinion. This text is naturally unstructured, so a new system can automatically convert that unstructured text into a more structured one. That system is what the GDSS, where this solution is inserted, does. In a GDM problem, participants take sides defending and attacking the possible Alternatives to the problem, so the solution implemented intends to make this division, creating clusters of participants, facilitating the consensus reaching in a further step of the GDSS that is being implemented at the organization.

The idea generation can be evaluated using the following practical methods, tools, and techniques (Koen et al., 2002):

- Methods for identifying unarticulated customer needs include ethnographic approaches and lead user methodology
- Early involvement of customer champion
- Discovering the archetype of the customer
- Market and business needs and issues continuously interspersing with the technological advances
- Identifying new technology solutions - Increasing technology flow through internal and external linkages, and partnering
- An organizational culture that encourages employees to spend free time testing and validating their own and others' ideas

- A variety of incentives to stimulate ideas
- A Web-enabled idea bank with easy access to product or service improvements, including linkages to customers and suppliers
- A formal role for someone (i.e., process owner) to coordinate ideas from generation through assessment
- A mechanism to handle ideas outside (or across) the scope of established business units
- A limited number of simple, measurable goals (or metrics) to track idea generation and enrichment
- Frequent job rotation to encourage knowledge sharing and extensive networking
- Mechanisms for communicating core competencies, core capabilities, and shared technologies broadly throughout the corporation u Inclusion of people with different cognitive styles on the idea enrichment team

From these, a series of questions were defined to understand and discuss the idea:

- How to fight the lack of specialized datasets for the GDM area?
- How to make use of existing discussions taken on the web?
- How to automatically segment the participants of a discussion based on what they typed?

3.1.4 Idea Selection

According to (Koen et al., 2001), in most companies, there are so many product/process ideas that deciding which ones to pursue to maximize corporate value is a vital job, where many ideas must be allowed to flourish with less assurance of success.

After the idea generation step, the next is to decide which ideas to pursue to obtain the most business value. The idea selection can be evaluated using the following practical methods, tools, and techniques (Koen et al., 2002):

- Portfolio methodologies are based on multiple factors (not just financial justification) using anchored scales (ordinal measures that utilize numeric indicators, each of which is associated with a set of words that help the respondent "anchor" their evaluation. The use of anchored scales removes much of the subjectivity when assigning a value to the project) - Technical success probability, Commercial success probability, Reward, Strategic fit, and Strategic leverage
- Formal idea selection process with prompt feedback to the idea submitters - Enhancement of methodology with electronic performance support systems, Web-enabling of the process
- Use of options theory to evaluate projects

However, it is crucial to select an idea to answer the questions formulated in the previous step (3.1.3), with this potential idea getting analyzed. It was verified that this idea could help the bigger picture of the GDSS system that the idea is supporting.

- How to fight the lack of specialized datasets for the GDM area? With the creation of a methodology to help create specialized datasets for the GDM area.
- How to make use of existing discussions taken on the web? Applying the methodology mentioned in the previous question to obtain a structured dataset that could be used in a GDM problem.
- How to automatically segment the participants of a discussion based on what they typed? ML techniques, mostly Clustering methods, can segment data based on mathematical approaches and therefore identify correlations between data that an average person could not easily detect.

3.1.5 Concept definition

The solution to be implemented consists of a web service based on Clustering techniques with some data management directed to a pipeline of GDSS being developed at the organization. With this solution, the GDSS can obtain the participants of a discussion segmented based on the Alternative that they support or go against or the most used Criterion by said participants, which in the end facilitates the consensus-reaching process of the GDSS.

The concept definition can be evaluated using the following practical methods, tools, and techniques (Koen et al., 2002):

- Goal deliberation approaches - Time spent on carefully defining the project goals and outcomes.
- Setting criteria for the corporation that describe what an attractive project looks like (in terms of financials, market growth, and market size).
- Rapid evaluation of high-potential innovations.
- Rigorous use of the technology stage-gate (TGS) process for high-risk projects
- Understanding and determining the performance capability limit of the technology
- Early involvement of the customer in actual product tests - Involvement of the customer even before the product is completed and staff up high-potential projects while still in FFE
- A partner outside of areas of core competence.
- Focus (in contrast to spreading too thin)
- Pursue alternative scientific approaches.
- Employ product champions if adequate funds are unavailable

3.2 Value

Value depends on a person's perception but has been defined in different theoretical contexts as “need, desire, interest, standard/ criteria, beliefs, attitudes, and preferences” (Nicola et al., 2012a). Other authors defined value as “the combination of quality, service, and price that reflects the perceived tangible and intangible benefits and costs” (Kotler & Kelly, 2006). To a business, the creation of value is the key to success. In contrast, any commercial activities are all about trading some tangible or intangible product or service and having its worth recognized and rewarded by the consumers or clients, whether inside the organization or collaborative network or outside the organization (Nicola et al., 2012b). In this section, the value for the customer and the perceived value are presented and detailed.

3.2.1 Value for the Customer

Value for the Customer is any demand-side, personal perception of advantage that results from a customer's association with what an organization is offering. It can occur as a reduction in sacrifice, the presence of benefit, the result of any weighted combination of sacrifice and benefit, or an aggregation of any or all of these over time (Woodall, 2003). Other authors defined value for the customer as "value generated by a company's product or service as perceived by the customer or the fulfillment of customer goals and desires by company products or services" (Graf & Maas, 2008). According to (Shanker, 2012), value for the customer is divided into two aspects, the desired value and the perceived value. The first refers to what a customer desires in a product or service, while the latter refers to the benefit a customer believes he has received from obtaining such a product or service.

3.2.2 Customer Perceived Value

Customer Perceived Value refers to the "consumer's overall assessment of the utility of a product based on perceptions of what is received and what is given" (Zeithaml, 1988). (Ulaga & Eggert, 2006) adds to the previous definition that "different customer segments perceive different values within the same product". Another definition is that the customer perceived value is an assessment that the customer does to a product or service, considering the benefits and sacrifices that the product or service brings to the said customer (Gutierrez et al., 2013). Therefore, if the benefits outweigh the sacrifices, the product or service has value to the customer.

The value of this solution to the customer is a web service that allows the customer to obtain meaningful information about a discussion, like the segment of participants by the Alternative they support or Criteria used to evaluate the Alternatives. Other than that, it also possesses the ability to process large amounts of unstructured data, like opinions taken in the form of text, and turn them into a structured form. Therefore, to evaluate the value for the solution's customer, the author created Tab 2 to highlight the benefits and sacrifices associated with this solution.

Benefits	Sacrifices
Ability to process large amounts of unstructured data	Costs of hosting the web service
Ability to segment the participants of a discussion based on the Alternative that they	Costs of developing the solution

mention and the Criteria used to evaluate the Alternatives	
Ability to obtain information that could be not easily visible	The necessity of manually annotating discussions on the web to train and test the solution because of the lack of datasets of this type
Ability to improve the GDSS that this solution integrates so that, in the end, the final GDSS results in a helper to a discussion taken on the web	

Tab 2 – Benefits and Sacrifices for the customer

The customer can take advantage of this system to obtain meaningful information about a discussion without having to do it manually. While also, being able to take structured and unstructured data makes it a significant benefit that should outweigh the sacrifices that are mainly costs associated with the web service like developing and hosting it.

3.3 Value Proposition

According to (Osterwalder, 2004), Value Proposition (VP) "is an overall view of a company's bundle of products and services that are of value to the customer" (Jalili & Rezaie, 2010). Defined VP as a tool to "specify a strategy to compete for new customers or increased share of existing customer businesses".

This project's value proposition consists of a web service based on Clustering techniques with some data management directed to a pipeline of GDSS being developed at the organization. With this solution, the GDSS can obtain the participants of a discussion segmented based on the Alternative that they support or go against or the most used Criterion by said participants, which in the end facilitates the consensus-reaching process of the GDSS.

3.3.1 Value Proposition Canvas

The Value Proposition Canvas, as shown in Fig. 13, is a tool developed by Alexander Osterwalder. It is the reason customers decide which company to choose since it solves a customer problem or satisfies a customer's need. It consists in a bundle of products or services that caters to the requirements of a customer segment. It identifies the benefits a company can offer and the customer values (Osterwalder & Pigneur, 2010), fitting each other (Fig. 14). The Value Proposition Canvas is divided into two segments, the product, and the customer. Hereafter, both these segments of the Value Proposition Canvas are detailed, identifying aspects like gain creators and pain relievers of the product with the gains and pains of the customer.

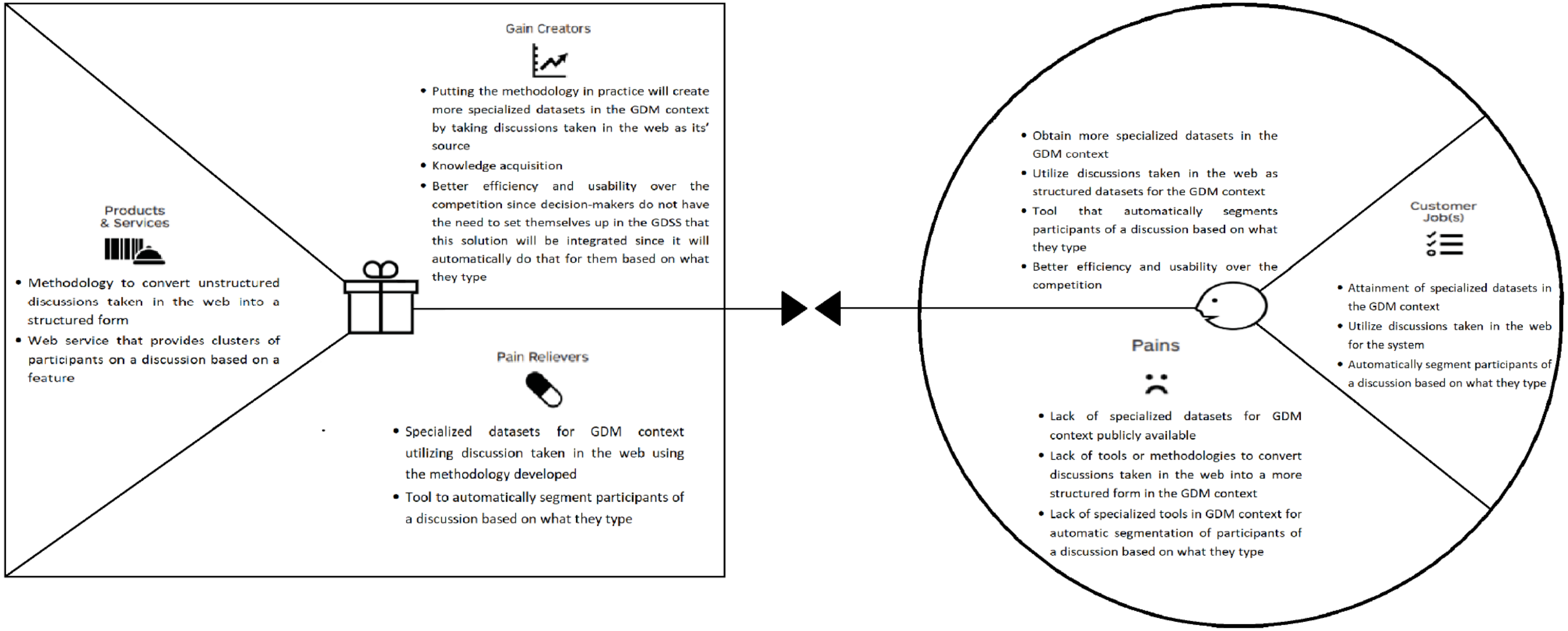


Fig. 13 – Value Proposition Canvas adapted from Osterwalder (2014)

Client

- **Gains** – Outcomes and benefits the business' customer wants. Some gains are required, expected, or desired by customers, and some would surprise them. They include functional utility, social gains, positive emotions, and cost savings (Osterwalder et al., 2014).
- **Pains** – Anything that annoys a customer before, during, and after trying to get a job done or prevents them from getting a job done. It also describes risks, that is, potential bad outcomes, related to getting a job done poorly or not at all (Osterwalder et al., 2014).
- **Customer Job(s)** – Task that a customer is trying to perform and complete, problems they are trying to solve, or needs they are trying to satisfy (Osterwalder et al., 2014).

Product

- **Products & Services** – A bundle of products and services that helps the customers complete functional, social, or emotional jobs or helps them satisfy basic needs (Osterwalder et al., 2014).
- **Gain Creators** – How the products or/and services create customer gains. They explicitly outline how the business intends to produce outcomes and benefits their customer expects, desire, or would be surprised by, including functional utility, social gains, positive emotions, and cost savings (Osterwalder et al., 2014).
- **Pain Relievers** – How exactly do the products or/and services alleviate specific customer pains? They explicitly outline how the business intends to eliminate or reduce some things that annoy customers before, during, or after trying to complete a job or prevent them from doing so (Osterwalder et al., 2014).

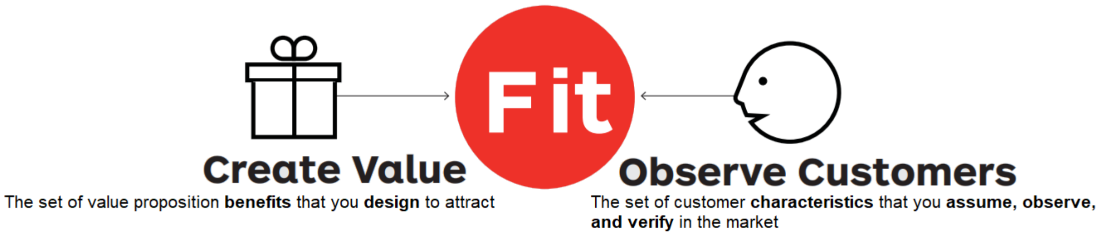


Fig. 14 – Customer and Business Fit (Osterwalder et al., 2014)

3.4 Functionalities Analysis (FAST)

The function Analysis System Technique (FAST) was developed by Charles Bytheway that provided a graphical representation of the function-based approach to the analysis of products and processes. Known as FAST Diagram, it organizes the functions performed by the product, process, or system being studied. It also studies the "How?" and "Why?" relationships (Borza, 2011).

This diagram defines a function by what a product should do to work and sell. It is restricted to a two-word format, an active verb, and a measurable noun, like Visualize Order. Forcing the users of this methodology to clearly and concisely capture what task needs to be performed, not how it should be performed. This methodology allows for exploring alternatives more easily (Borza, 2011).

In Fig. 15, it is possible to visualize the FAST methodology and the following diagram for the project. This project mainly involved investigating Clustering techniques that could fit the project's needs, considering discussions taken on the web for the GDSS on which this Clustering service is inserted. Therefore, the functions are described in tasks and steps that culminate into a web service.

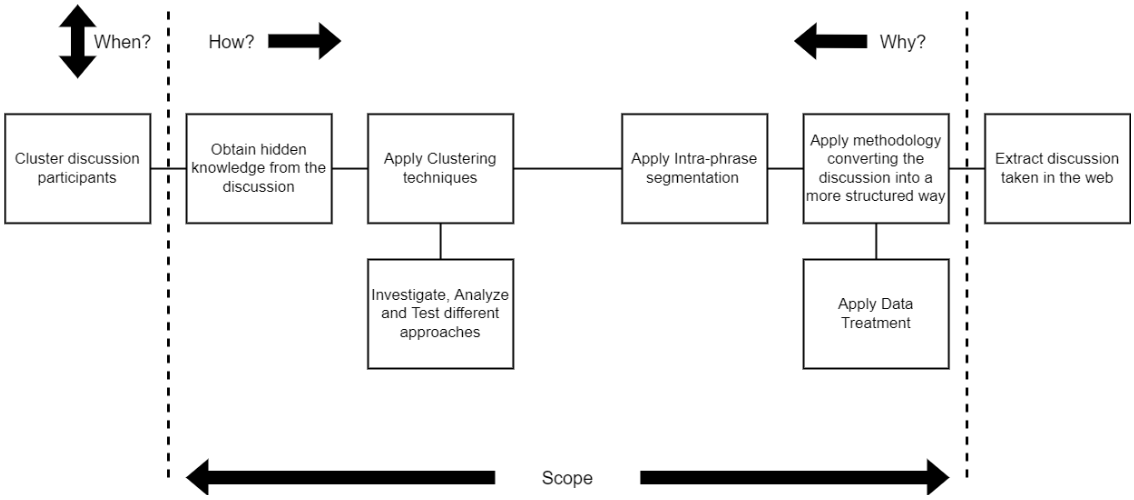


Fig. 15 – FAST Diagram

3.5 Alternatives Analysis (TOPSIS)

Since multiple options exist to solve a problem with objectives to achieve, deciding which alternative to choose becomes complex. Decision support approaches, such as Multiple Criteria Decision Making (MCDM), are used to improve the legitimacy and trustworthiness of the chosen solution. They aid in the decision-making process to reduce the responsibility of the final decision-maker and provide a solution that meets the criteria in consideration (Frazão et al., 2018).

The multi-criteria decision-making method used in this dissertation to decide which Python web service framework is used for developing the solution was TOPSIS. According to (Saragih et al., 2014) and (Made et al., 2013), the Technique for Order Preference by Similarity to Ideal Solution, mostly known as TOPSIS, is a method that was developed based on the concept of searching for the best alternative. This best alternative has the shortest distance from the positive ideal solution and the longest distance from the negative ideal solution from a geometric point of view using Euclidean distance. TOPSIS method ranks the alternatives based on the relative nearest score priority of the alternative to the positive ideal solution, making it able to measure the relative performance of any decision alternatives. The method was introduced initially by Hwang & Yoon (1981).

Since this dissertation results in a web service, the alternative analysis is performed on the web service frameworks that are the sub-sections of 2.5.2 since the solution is done in Python. Criteria and respective weights must be decided to evaluate the best alternative. Chakhar et al. (2015) compiled a list of commonly used criteria in Web services evaluation. For each criterion, they provide a brief description, if it is quantitative or qualitative, and the preference direction where “max” means “the higher, the better” and “min” means “the lower, the better”, which helps build the problem with TOPSIS. Balani (2016) came up with a less specific list of criteria that can be used to evaluate a web service framework. From both lists that these authors compiled, the criteria used in this dissertation to decide which alternative is the best can be seen in Tab 3.

Name	Description	Preference	Weight
Security	It captures the level and kind of security service that provides	max	0.3
Scalability	It defines whether the service capacities can be increased as needed	max	0.2
Reputation	It is a measure of trustworthiness. It mainly depends on the end user’s experience of using a service	max	0.15
Response Time	The lapse of time from request sending to response reception while Throughput is	min	0.1
Throughput	The rate at which a service can process requests	max	0.1
Ease of Development	Expose existing functionality as a web service. Influenced by existing documentation, learning curve, and community support	max	0.15

Tab 3 – Criteria to be used in TOPSIS

After the author defined the criteria to evaluate the alternatives and respective weights, values on each criterion for each alternative need to be given, these values are based on articles read online[43,44,45,46,47,48,49]. They discuss the web service framework alternatives mentioned in 2.5.2 sub-sections regarding pros and cons and key features, resulting in Tab 4.

Weight	0,3	0,2	0,15	0,1	0,1	0,15
	Security	Scalability	Reputation	Response Time	Throughput	Ease of Development
Django	9	9	9	9	6	5
Flask	7	9	9	5	9	9
CherryPY	5	7	7	7	7	4
Bottle	5	3	5	7	7	4

Tab 4 – TOPSIS Decision Matrix

After creating the initial decision matrix, a normalization step is needed, and the normalized decision matrix can be seen in Tab 5. With r_{ij} the normalized value in each table cell and x_{ij} the value given to each alternative/criterion pair, the following formula is used:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} {x_{ij}}^2}}$$

Weight	0,3	0,2	0,15	0,1	0,1	0,15
	Security	Scalability	Reputation	Response Time	Throughput	Ease of Development
Django	0,67082	0,60678	0,58585	0,630126	0,409197	0,425628
Flask	0,521749	0,60678	0,58585	0,35007	0,613795	0,766131
CherryPY	0,372678	0,47194	0,455661	0,490098	0,477396	0,340503
Bottle	0,372678	0,20226	0,325472	0,490098	0,477396	0,340503

Tab 5 – TOPSIS Normalized Decision Matrix

Afterward, from the normalized decision matrix, we create the weighted normalized decision matrix that multiplies each table cell value with each criterion's weight, reaching Tab 6.

[43] wiki.python.org/moin/WebFrameworks
[44] hackr.io/blog/python-frameworks
[45] steelkiwi.com/blog/top-10-python-web-frameworks-to-learn/
[46] monocubed.com/top-python-frameworks/
[47] netsolutions.com/insights/top-10-python-frameworks-for-web-development-in-2019/
[48] productcoalition.com/10-top-python-frameworks-for-web-development-in-2021-9264d709594d
[49] enlear.academy/top-python-web-development-frameworks-for-2022-782f50c89b26

Weight	0,3	0,2	0,15	0,1	0,1	0,15
	Security	Scalability	Reputation	Response Time	Throughput	Ease of Development
Django	0,201246	0,121356	0,087878	0,063013	0,04092	0,063844
Flask	0,156525	0,121356	0,087878	0,035007	0,06138	0,11492
CherryPY	0,111803	0,094388	0,068349	0,04901	0,04774	0,051075
Bottle	0,111803	0,040452	0,048821	0,04901	0,04774	0,051075

Tab 6 - TOPSIS Normalized Weighted Decision Matrix

We can do the next step from the normalized weighted decision matrix, identifying the positive and negative ideal solutions. A^+ being the positive ideal solution and A^- being the negative ideal solution, taking into consideration the preference column in Tab 3 and v_n^+ if criterion is a benefit or max preference, the max value found in that criterion. Suppose the Criterion is a cost or min preference, the min value in that Criterion and v_n^- . In that case, if the Criterion is a benefit or max preference, the min value is found in that Criterion. If the criterion is cost or min preference, the max value in that criterion, the vectors are given by the following formula:

$$A^+ = \{v_1^+, v_2^+, \dots, v_n^+\}$$

$$A^- = \{v_1^-, v_2^-, \dots, v_n^-\}$$

Obtaining the following vectors:

A^+ = 0,201246; 0,121356; 0,087878; 0,035007; 0,06138; 0,11492

A^- = 0,111803; 0,040452; 0,048821; 0,0630126; 0,04092; 0,051075

Considering S_i^+ gives the distance to the positive ideal solution, and S_i^- gives the distance to the negative ideal solution. These distances in x and x are calculated given the following formulas:

$$S_i^+ = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^+\right)^2}$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^-\right)^2}$$

Alternative	S_i^+
Django	0,061739
Flask	0,044721
CherryPY	0,116478
Bottle	0,124981

Tab 7 – TOPSIS Distance to Positive Ideal Solution

Alternative	S_i^-
Django	0,127413
Flask	0,123896
CherryPY	0,059439
Bottle	0,015576

Tab 8 – TOPSIS Distance to Negative Ideal Solution

Lastly, the Relative Closeness metric value can be seen in Tab 9 and was calculated by using the following formula:

$$V_i^+ = \frac{S_i^-}{S_i^- + S_i^+}$$

Alternative	V_i^+
Django	0,673601
Flask	0,734777
CherryPY	0,337883
Bottle	0,110813

Tab 9 – TOPSIS Relative Closeness to Ideal Solution

As we can see, through the values obtained in Tab 9, Flask has a relative closeness of 0.73 for the problem in hands, becoming the elected alternative through the TOPSIS technique. Flask is more advantageous than the rest for this problem, which can be confirmed by the dimension of the project, in which Django would be overkill for the size and features it offers that would not be taken advantage of with this project.

4 Proposed Solution

Before diving into the design of the solution, it is essential to explain the conceptual architecture of the GDSS to which the web service is added. The conceptual architecture was designed by Carneiro et al. (2021) and is based on micro-services architecture since it benefits these systems. It is better at fault isolation, continuous integration, and delivery, and easier to scatter the components through multiple servers and others. It allows for automatic deployment tactics and the possibility of writing each component in different programming languages and others (Carneiro et al., 2021). Fig. 16 represents the conceptual architecture of the GDSS, which uses an API Gateway as the single-entry point into the system, allowing the internal system architecture to be encapsulated and tailor-made for each client's needs. The internal system architecture is a set of possible services to handle the organization's business needs and microservices to support the decision-making process. These services can be like the usage of a Multi-Agent System in the Agents' Service and strategies to automatically propose solutions and other functionalities in the Decision-Making Service (Carneiro et al., 2021).

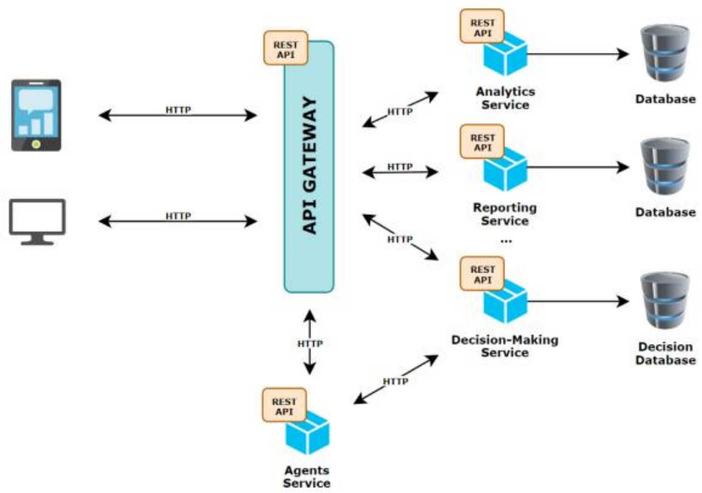


Fig. 16 – GDSS conceptual architecture

After a brief explanation of the conceptual micro-services-based architecture, requirements engineering is performed to understand the functional and non-functional requirements of the solution. Afterward, the architecture of this solution is presented, contemplating two alternatives to the design of this solution.

4.1 Requirements engineering

Requirement engineering is the process of defining, documenting, and maintaining a project's requirements, gathering and defining the functionalities provided by a system (GeeksforGeeks, 2020). Therefore, this section follows this process, resulting in functional requirements through functionalities and non-functional requirements using FURPS+.

4.1.1 Functional Requirements

Functional Requirements are a description of the service that the software must offer, describing a software system or its' components through the usage of functions that describe the inputs, behavior, and outputs of a component or system (Martin, 2021). Use cases are typically used to describe functional requirements since they describe how a user interacts with the system or component to reach a particular goal. There are multiple ways to describe a use case with different levels of granularity, ranging from simplified use case diagrams to the specification of success and failure scenarios, critical variations, and exceptions. For this dissertation, a use case diagram is used in addition to informal descriptions of each use case, allowing for a better understanding of them, which is the sole purpose of it, and why it is not a formal way of describing them. Fig. 17 presents the UML-based use case diagram relative to the system.

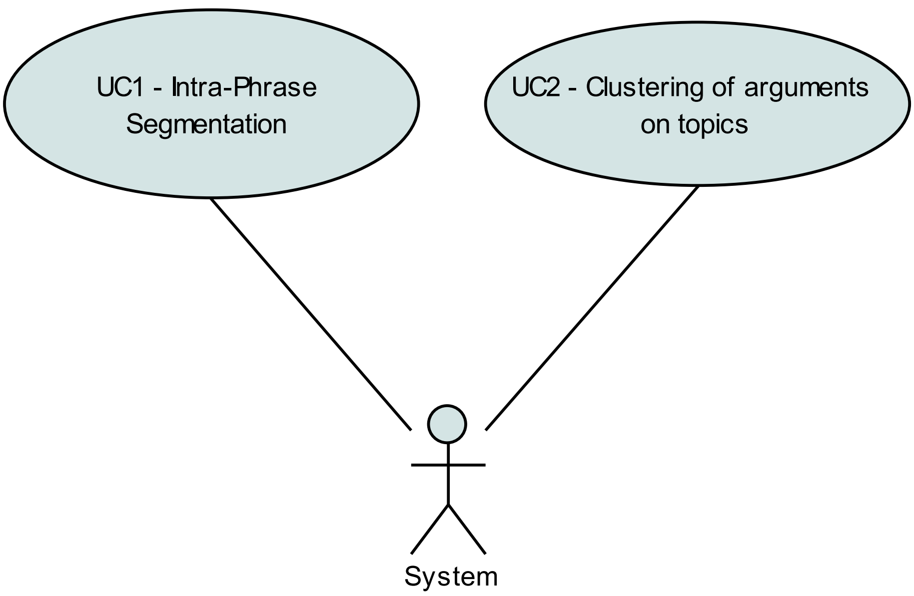


Fig. 17 – Use Case Diagram

UC1 – Intra-Phrase Segmentation. This use case revolves around the intra-phrase segmentation task mentioned in 2.3. It should allow consistently dividing a written argumentative opinion of a participant into significant bits that can portray what Criteria or Alternatives a participant uses in their opinion.

UC2 – Clustering of arguments on topics. This use case represents the objective with the same name, which is essential since it intuitively brings much knowledge to GDSS. This knowledge could be used by the Decision-Making service to suggest solutions with more quality based on the direction of the discussion.

4.1.2 Non-functional Requirements

After identifying the functional requirements, the non-functional requirements are identified. Even if they are not correlated directly to the system, they are correlated to how a system should function, describing the solution's operation capabilities and constraints that enhance its functional requirements (AltexSoft, 2019). The FURPS+ technique is one of the most used for non-functional requirement identification. It has developed by Hewlett-Packard (HP) as an evolution of the FURPS model that added the "+" into it. FURPS+ defines non-functional requirements by grouping them into categories: Functionality, Usability, Reliability, Performance, Supportability, Design Constraints, Implementation Requirements, Interface Requirements, and Physical Requirements (Langr & Ottinger, 2011).

Functionality.

- Clustering models with high accuracy (>70%)
- Intra-Segmentation model with high accuracy (>70%)

Usability.

- Accurate documentation since it can be used for future papers

Reliability.

- Responsive web service with a high availability rate

Performance.

- The Clustering model should be fast in its' task
- The intra-segmentation model should be fast in its' task

Supportability.

-

+.

Design Constraints.

- The micro-service architecture of the GDSS

Implementation Requirements.

-

Interface Requirements.

-

Physical Requirements.

-

4.2 Pipeline

Several techniques were tested, and the impact on the results was analyzed. Different embedders were tested considering the context (or not), different Clustering techniques (partitional and hierarchical-based), and dimension reduction techniques. Their impact on metrics was analyzed. Fig. 18 illustrates the pipeline we developed to run these experiments.

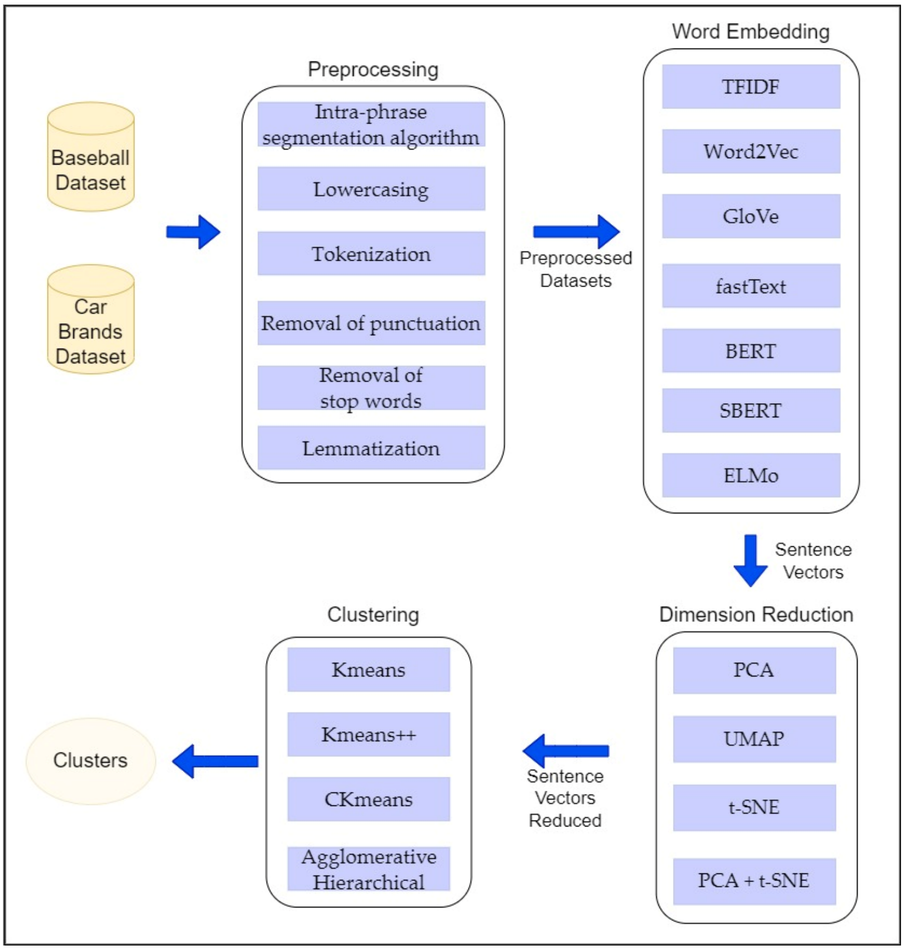


Fig. 18 – Methodology Overview

Every step of this pipeline is detailed next. The implementations and parameters used on which technique of each step.

4.2.1 Preprocessing steps

The same preprocessing steps were used in every approach testing. They applied the intra-phrase segmentation algorithm, followed by Lowercasing, Tokenization, Lemmatization, punctuation removal, and stop words. Whenever additional input data was sent to the Clustering method, the min-max method was used to normalize the categorical classes.

4.2.2 Word Embedders

In terms of Word Embedders, all the mentioned embedders are in 2.4.3.1. For TFIDF, the TfidfVectorizer was used. The setting tested for this word embedder that works best for this work was ngram_range=(1,1), where the first value indicates the minimum amount of grams to take into consideration and the second value the maximum, in which, by making them (1,1), the author is solely utilizing unigrams. For Word2Vec, it was used the word2vec-google-news-300[50] pre-trained vectors with binary mode activated and max length equaling 200. For GloVe, it used glove.6B.200d[51] with 6B tokens, 400K vocab, uncased, and 200 dimensions, in which the author maintained that dimensions preset (max length equaling 200). For fastText, it used crawl-300d-2M[52], which consists of 2-million-word vectors trained on Common Crawl with 600B tokens, and we used a max length equaling 200. In terms of BERT, a BERT-Base pre-trained model was utilized with 12 layers, 768 hidden states, 12 heads, and 110M parameters found on their GitHub page[53], while for SBERT, it utilized an all-MiniLM-L6-v2[54] pre-trained model with six layers and 384 hidden states totaling 1 billion training pairs. For ELMo, the author utilized this model's third version (v3)[55].

4.2.3 Dimensionality Reduction Techniques

For Dimensionality Reduction Techniques, the three techniques mentioned in 2.4.3.2 were used. In addition to these three techniques, a hybrid approach applies PCA and t-SNE. PCA to reduce to fifty dimensions, followed by t-SNE, suppresses some noise and speeds up the computation of pairwise distances between samples[56].

The models' outputs are not going to be changed in terms of dimension reduction. Then, use that as a baseline: from 200 to 100 dimensions in steps of 50, 100 to 25 in steps of 25, 25 to 5 in steps of 5, and 5 to 1 in steps of 1.

---

[50] huggingface.co/fse/word2vec-google-news-300
[51] nlp.stanford.edu/projects/glove/
[52] fasttext.cc/docs/en/english-vectors
[53] github.com/google-research/bert
[54] sbert.net/docs/pretrained_models
[55] tfhub.dev/google/elmo/3
[56] scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE

4.2.4 Clustering Techniques

For Clustering Techniques, from the one mentioned in 2.4.1, partitional and hierarchical-based techniques were delved into in this work. Regarding partitional techniques, the Kmeans technique was used with a 100-run experiment, with different centroid seeds, as a baseline technique. Kmeans++ utilized a 100-run experiment as well with different centroid seeds. Ckmeans utilized a 100-run experiment with different centroid seeds drawing 92% of the samples and 92% of features for each run. These two last values were obtained by preliminary testing performed. In terms of hierarchical-based techniques, the Agglomerative Hierarchical Clustering technique was used. The linkage equaling average parameter was utilized since it was the best linkage method in terms of performance in preliminary tests and was backed by state-of-the-art research.

4.2.5 Final Overview

These sections showed different techniques for each pipeline step, with information on their implementation and the parameters used. Tab 10 briefly details the utilized techniques, implementations, and parameters used.

	Technique	Implementation	Parameters
Word Embedding	TF-IDF	scikit-learn	ngram_range=(1,1)
	Word2Vec	word2vec-google-news-300	Binary mode = True max length = 200
	GloVe	glove.6B.200d	max length = 200
	fastText	crawl-300d-2M	max length = 200
	BERT	12/768 (BERT-Base)	-
	SBERT	all-MiniLM-L6-v2	-
	ELMo	v3	-
Dimensionality Reduction	PCA	scikit-learn	No change 200 to 100 in steps of 50 100 to 25 in steps of 25 25 to 5 in steps of 5 5 to 1 in steps of 1
	t-SNE	scikit-learn	
	PCA + t-SNE	PCA to 50 dimensions and then t-SNE application	
	UMAP	umap-learn	
Clustering	Kmeans	scikit-learn	n_init = 100
	Kmeans++	scikit-learn	n_init = 100
	Agglomerative Hierarchical	scikit-learn	linkage = average
	CKmeans	pyckmeans	n_rep = 100 p_samp = 0.92 p_feat = 0.92

Tab 10 – Quick visualization of the approach's definition

4.3 Architecture

Software architecture is the organization of the system, which details all the components, how they interact with each other, the environment in which they operate, and the principles used to design the system. It exposes the system's structure while hiding some implementation details (CAST, 2021).

This section presents two alternatives of architecture that highly influence how the web service that results from this dissertation functions. A UML Components diagram shows how the components of the system interact and its' interactions with the other services of the GDSS. After selecting an alternative, the alternative is detailed using two more UML diagrams that preview other system levels. Another thing to consider is that both alternatives aim to consider the functional and non-functional requirements exposed earlier.

4.3.1 Web-Service as a Tasker Architecture Alternative

This first architecture alternative comes from a fundamental standpoint of the web service acting as a tasker. In this case, a tasker would mean that the service would obtain the data from the decision-making service through the API Gateway, apply the functionalities wanted in terms of Clustering and send the results. It would not do anything else with the data that it receives. The results it obtains other than sending the results back to the requested service, as shown in Fig. 19. The Clustering Service would apply Data Preprocessing and Intra-sentence Segmentation to the data received and then apply the Clustering model appropriate for the functionality that was requested.

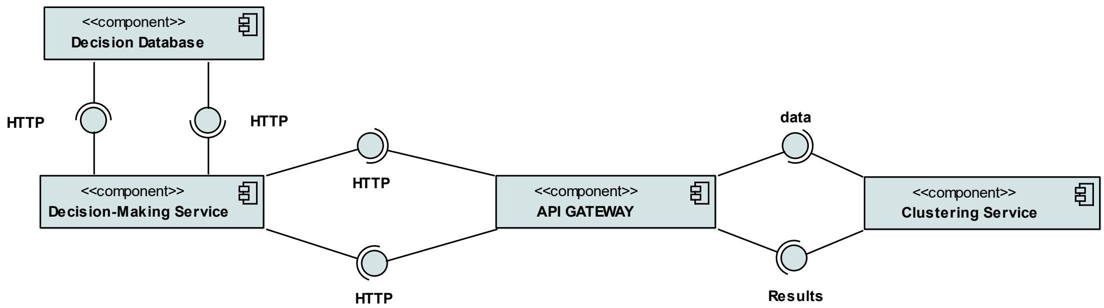


Fig. 19 – Web-Service as a Tasker Architecture Alternative

4.3.2 Web-Service as an Independent Service Architecture Alternative

This second architecture alternative picks up on where the first alternative was and takes a step further. Turning the Clustering service into an independent one where it has its' database that stores the data it obtains, the results it gets, and other important information allows it to be independent on its own. The service would obtain the data from the decision-making service through the API Gateway and store said data in its database, apply Data Preprocessing and

Intra-sentence Segmentation to the data, and apply the appropriate Clustering model for the functionality that was requested. The results can then be stored in the service's database and sent to the service that requested the Clustering service. This database allows other services like Analytics to obtain past Clustering results and produce additional knowledge to the GDSS based on multiple Clustering results or perform Clustering with historical data since the decision is not stale and can change over time. It also allows the Clustering service to scale and even have a data extraction package to obtain data from external sources if required. This proposed architecture can be seen in Fig. 20.

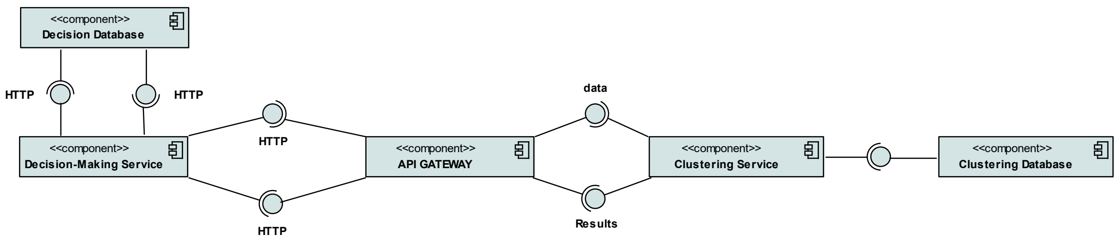


Fig. 20 – Web-Service as an Independent Service Architecture Alternative

The second proposed architecture alternative's benefit is that it makes the Clustering web service more independent. It allows it to grow if needed and not be stale as the first proposed architecture alternative. The second alternative was the selected one to be implemented as the architecture of the solution of this dissertation.

From a different perspective, Fig. 21 presents the package diagram of the selected architecture. This diagram focuses on presenting a logical view of the selected architecture. It allows us to perceive that five packages constitute the Clustering Service, the Communication Package, the Clustering Models Package, the Data Preprocessing Package, the Intra-sentence Segmentation Package, and the Database Package.

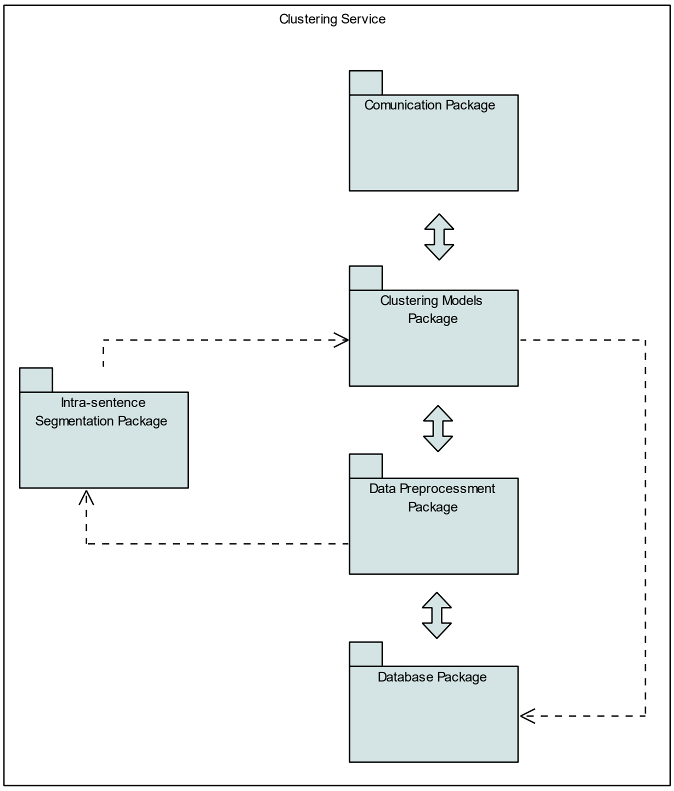


Fig. 21 – Package Diagram

The Communication Package oversees the Clustering service's requests, handling all the inputs and outputs the service receives and produces. The Clustering Models Package contains all the needed Clustering models to reach the use cases defined earlier. The Data Preprocessing and Intra-sentence Segmentation Packages contain the tools needed to perform the tasks that name said packages. The database Package handles all the operations related to the Database, like inserting, updating, deleting, and getting data.

To finalize, the Deployment Diagram is going to be presented. Fig. 22 exposes one way to deploy the components of the system. If the rest of the services that compose the GDSS is in GECAD's server, it is only natural for the Clustering service to go the same route. The database implantation depends on the usage of the service. It could either be deployed in GECAD's server if there is a low volume of data or in the cloud if there is a large volume of data. In the last case, Google Cloud was chosen platform. The storage in GECAD's server is limited, which could impose a limitation on other services running on it.

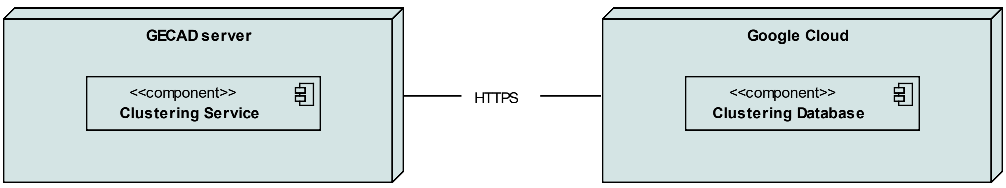


Fig. 22 – Deployment Diagram

5 Dynamic Organization of Conversations using Clustering

This section describes all the implementation steps for the Clustering Service defined in 4.3.2.

5.1 Dataset Exploration and Analysis

A careful search for datasets that can be applied to a decision-making context was made, focusing on places like Kaggle and NLP Index. It concluded that most of those available, for the most part, had quick reviews, which is not enough for the GDM context, which needs discussion and argumentation about a topic. This way, a methodology to create datasets for the GDM context based on Reddit discussions was created (Cardoso et al., 2022). Two datasets from such methodology are analyzed next since the Baseball dataset is more recent than the Car Brands dataset, the first used as the primary dataset to test approaches. In contrast, the latter dataset is used as a validation dataset, with both datasets having a detailed data exploration and analysis next.

5.1.1 Baseball Dataset

The Baseball Dataset was a dataset explicitly created for this context, where on “r/baseball”, a subreddit dedicated to the sport of Baseball. From there, two discussions were chosen: “Who do you think is the greatest baseball player of all-time?”[57] which had 50 comments and was designated as “Discussion A” and “Question on the greatest baseball player ever.”[58] which had 139 comments and was designated as “Discussion B”. Since both discussions' themes were similar, a decision was made to merge both into 1 dataset. After proper data handling treatment, separation into sentence level, and feature extraction, it came up with 488 lines with 20

[57] reddit.com/r/baseball/comments/2r5imp/who_do_you_think_is_the_greatest_baseball_player
[58] reddit.com/r/baseball/comments/av3gcy/question_on_the_greatest_baseball_player_ever

features ranging from the author of the message to the votes a particular message received and features specifically for the GDM context that were manually annotated and verified thoroughly.

Of the 20 features this dataset provides, only six are used in this work. These six features are the following:

- Sentence Text – message typed by a user divided into the sentence level.
- Alternative - list of values that contains the identifier of the Alternative
- Criterion - list of Criteria present in the text
- Aspect - indicates if a specific Entity is indicated explicitly or not in a particular opinion, taking Explicit or Implicit values
- Polarity - polarity of an opinion towards an Entity-Attribute pair in a phrase. It can be Positive, Negative, or Neutral
- OTE - an apparent reference to the Entity present in an opinion

Analyzing the distribution of the Alternatives available on this dataset (Fig. 23) as well as the distribution in the Aspect (Fig. 24), Polarity (Fig. 25), Entity (Fig. 26) with the usage of Criterion (Fig. 27), and the how the Player Entity Criteria were spread in the discussion when using the Player Entity features (Fig. 28), we can see that the dataset is not balanced, having some features that are present but rarely used by the decision-makers of the conversation. Analyzing the previously mentioned figures allows us to perceive that in the dataset, most of the arguments used by the participants are optimistic and utilize Criteria in their explanations of choosing such Alternative that is explicitly referred on the argument, with the Player Entity being the most used Entity to evaluate the available Alternatives.

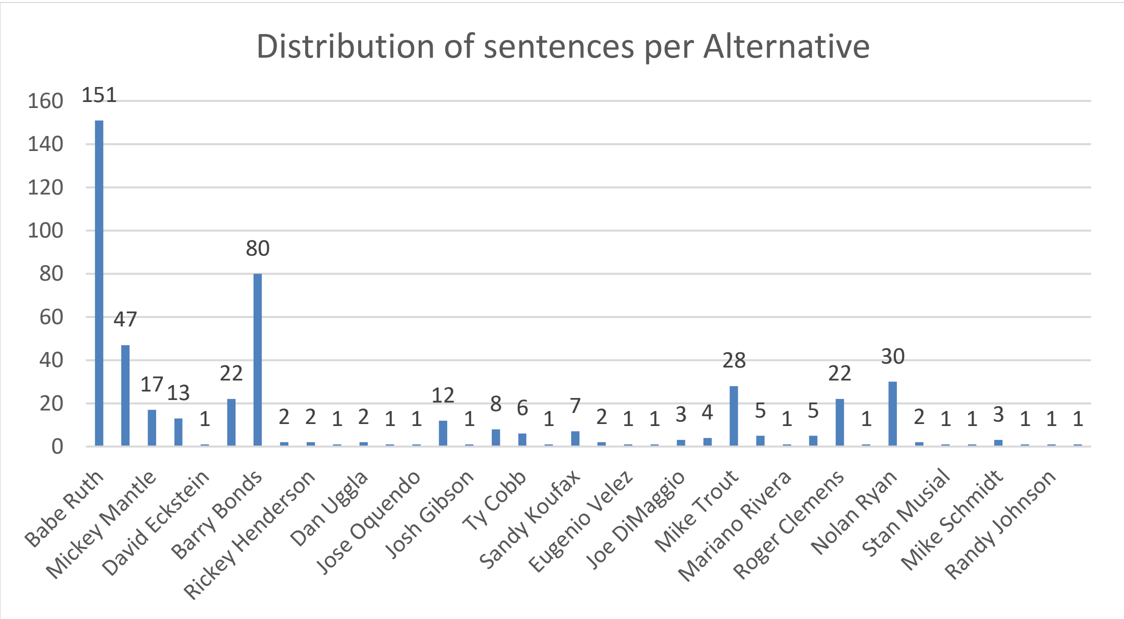


Fig. 23 – Distribution of sentences per Alternative in the Baseball Dataset

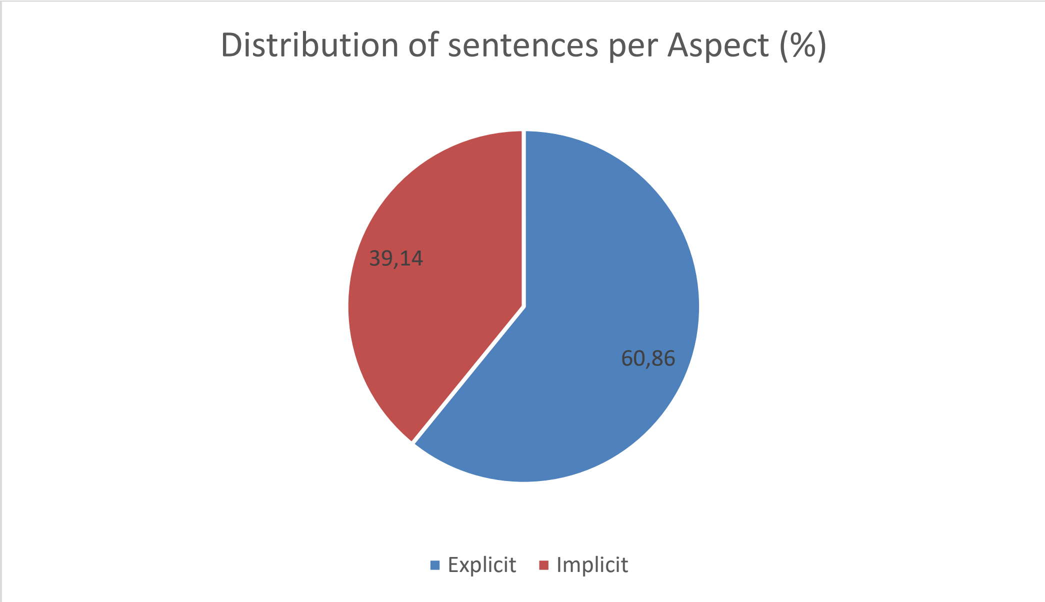


Fig. 24 – Distribution of sentences per Aspect (%) in the Baseball Dataset

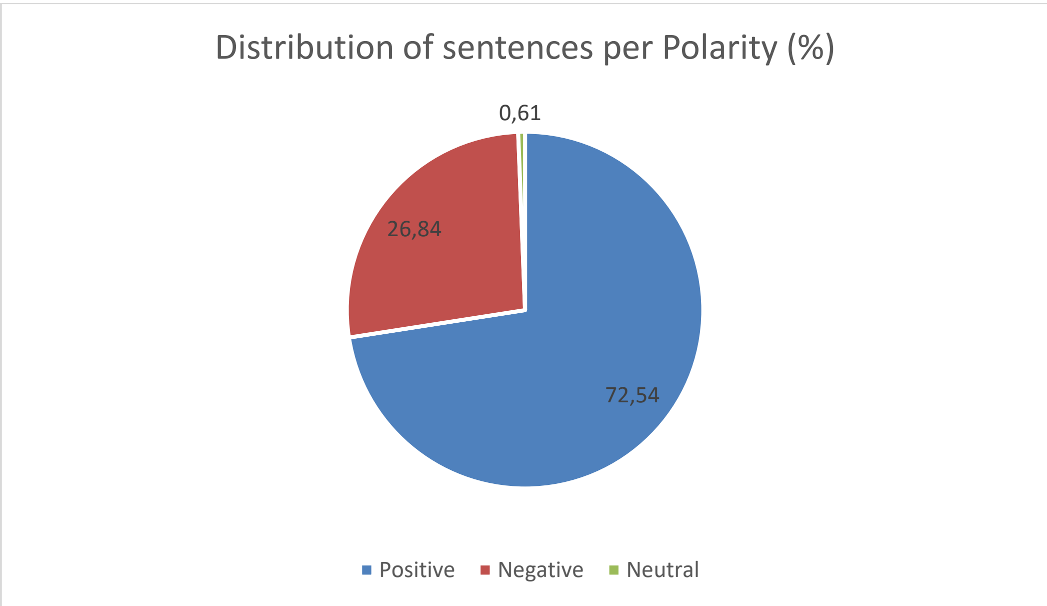


Fig. 25 – Distribution of sentences per Polarity (%) in the Baseball Dataset

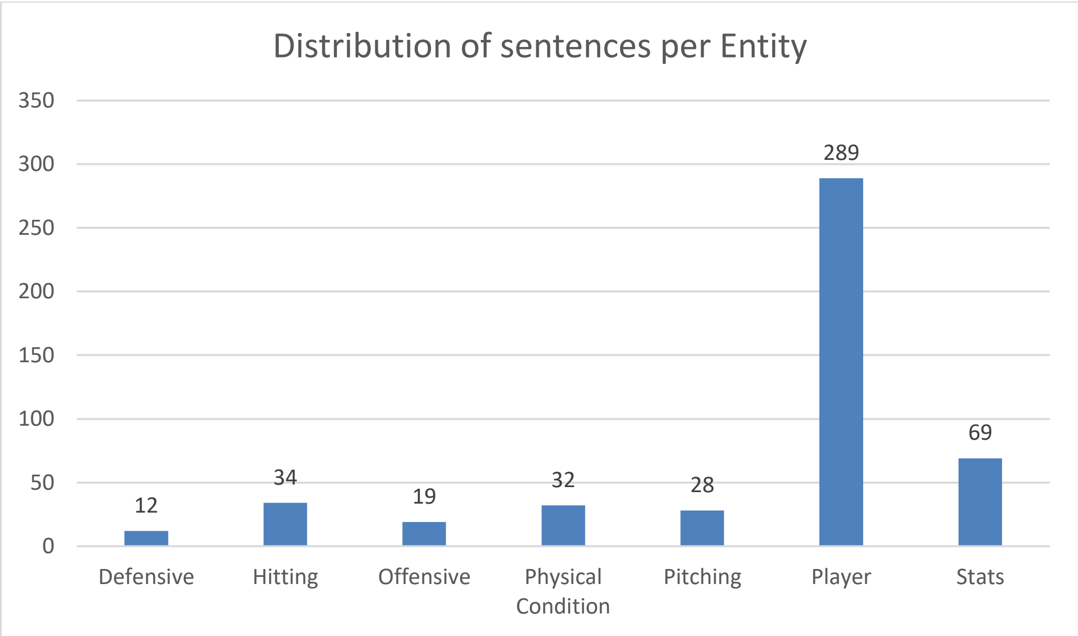


Fig. 26 – Distribution of sentences per Entity in the Baseball Dataset

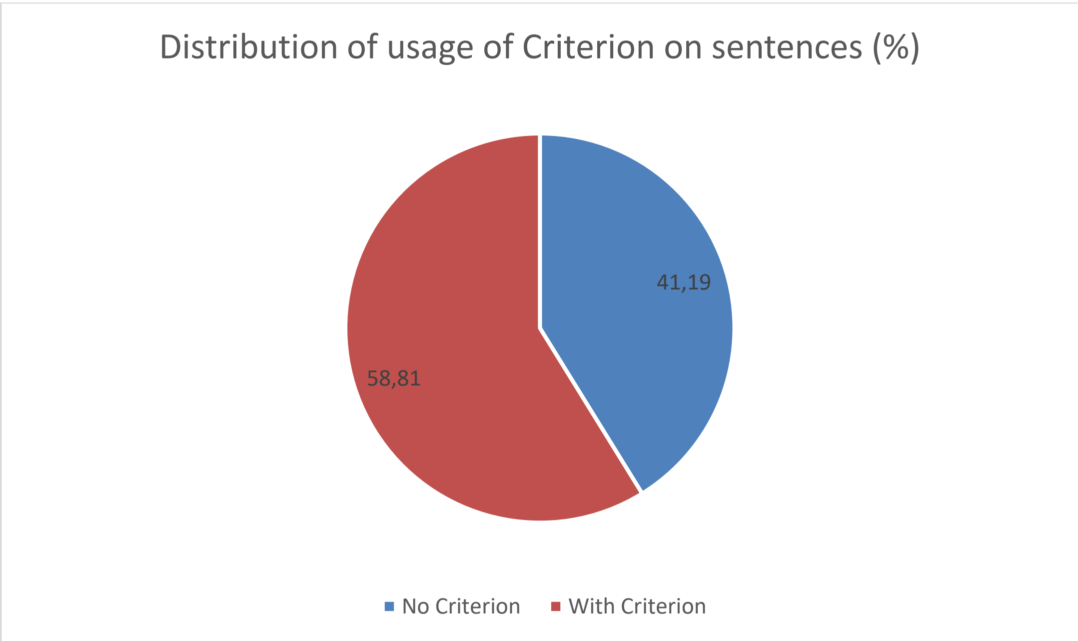


Fig. 27 – Distribution of usage of Criterion on sentences (%) in Baseball Dataset

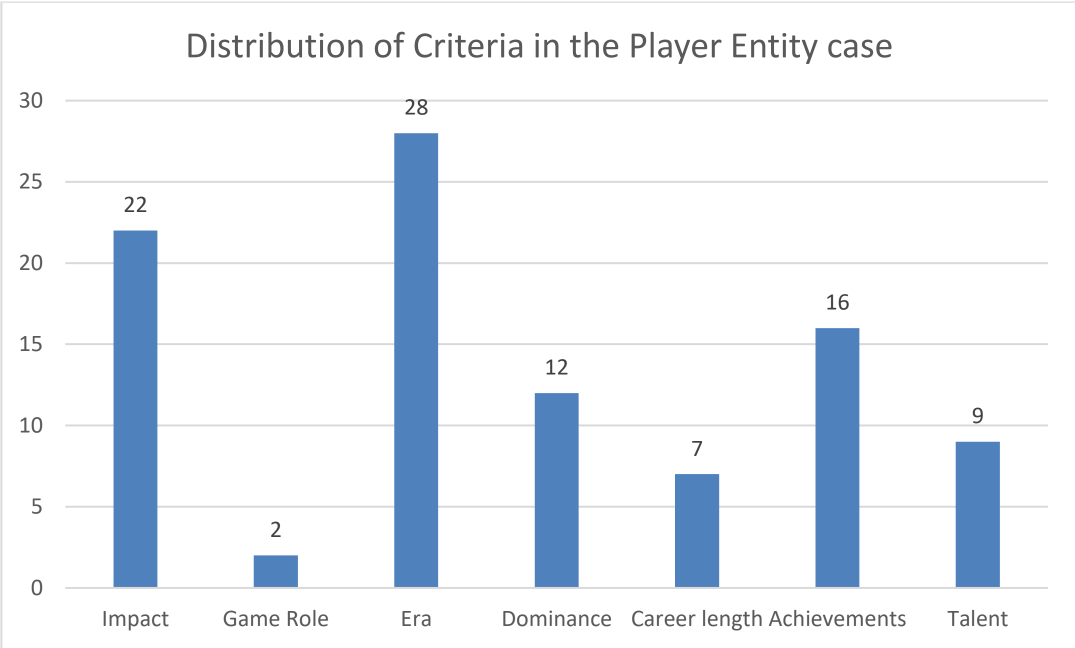


Fig. 28 – Distribution of Criteria in the Player Entity case in Baseball Dataset

5.1.2 Car Brands Dataset

The Car Brands dataset is a Dataset that followed an early version of the methodology mentioned above, working as the base of that methodology. Obtained from Reddit, related to the topic "What is your favorite car brand and why?"[59] has 187 responses. Each participant expresses their opinion of their favorite car brand, presenting the reasons that support their opinion. After handling and transforming the discussion to a dataset, it resulted in 388 lines with 17 features, like the previously mentioned dataset but without the votes gotten in each response, using the same features mentioned in the previous section to solve the problem in question for this work.

Analyzing the distribution of the Alternatives available on this dataset (Fig. 29) as well as the distribution in the Aspect (Fig. 30Fig. 24), Polarity (Fig. 31), Entity (Fig. 32) with the usage of Criterion (Fig. 33) and the how the Brand and Model Entities and related Criteria were spread in the discussion when using the Brand and Model Entities features (Fig. 34), we can see that the dataset is not balanced, having some features that are present but rarely used by the decision-makers of the conversation. Analyzing the previously mentioned figures allows us to perceive that in the dataset, most of the arguments used by the participants are optimistic and utilize Criteria in their explanations of choosing such an Alternative that is explicitly referred on the argument with the Brand and Model Entities being the most used Entities to evaluate the available Alternatives.

[59] reddit.com/r/cars/comments/386hc4/rcars_what_is_your_favourite_car_brand_and_why/

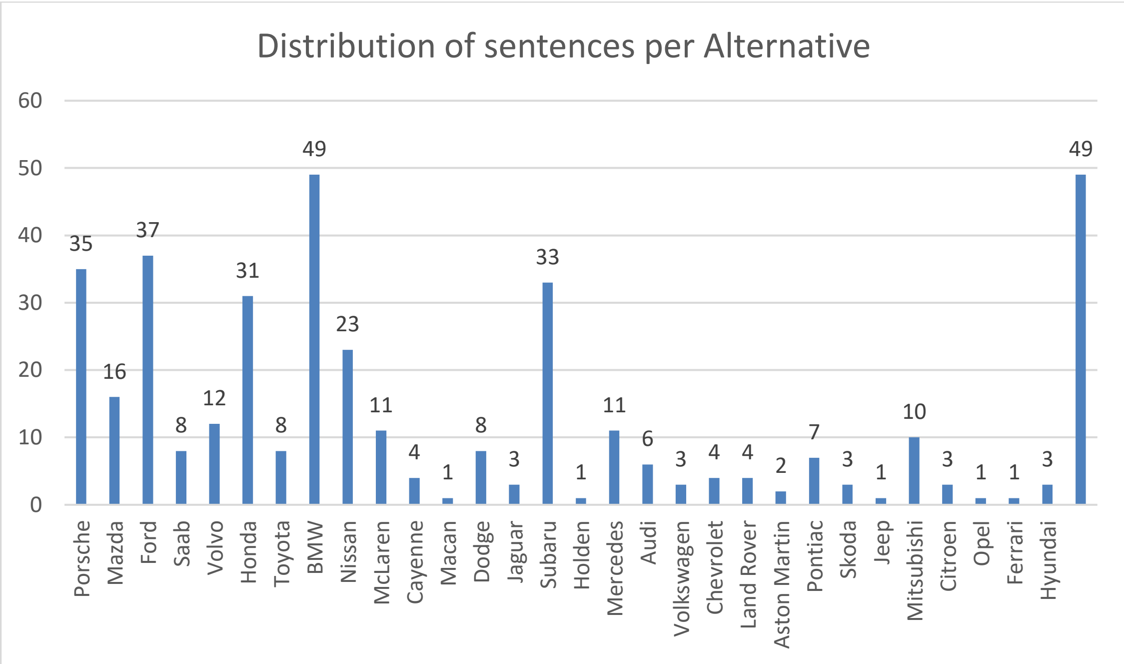


Fig. 29 – Distribution of sentences per Alternative in Car Brands Dataset

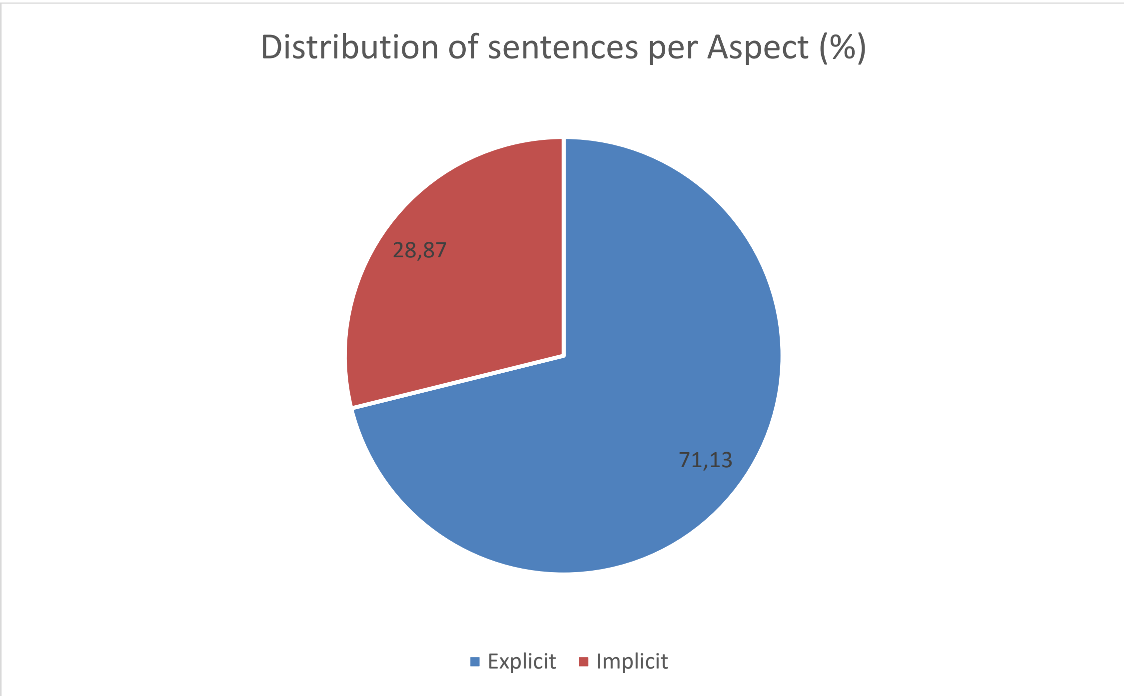


Fig. 30 – Distribution of sentences per Aspect (%) in the Car Brands Dataset

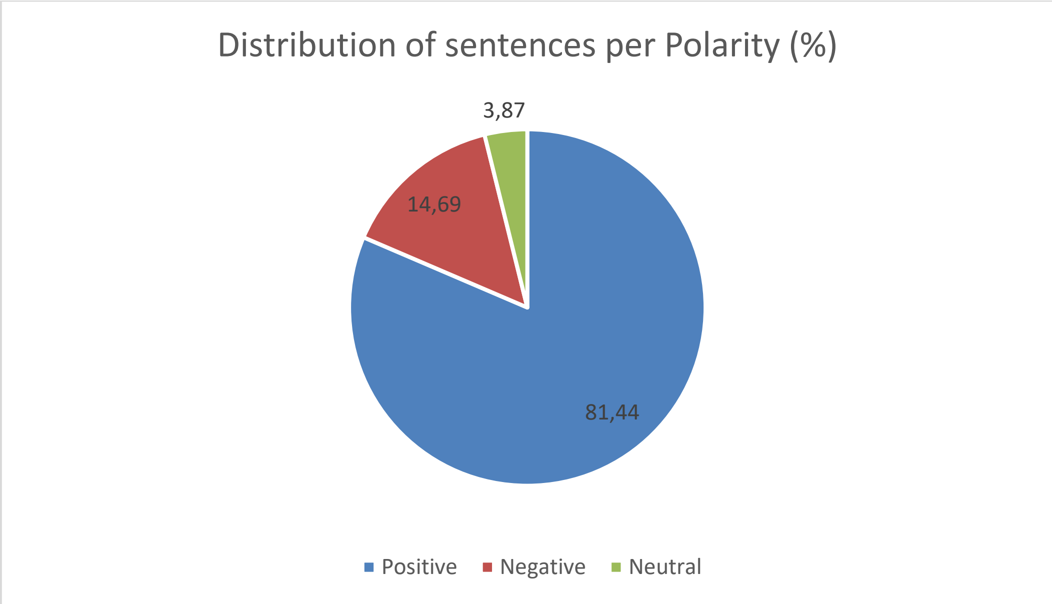


Fig. 31 – Distribution of sentences per Polarity (%) in the Car Brands Dataset

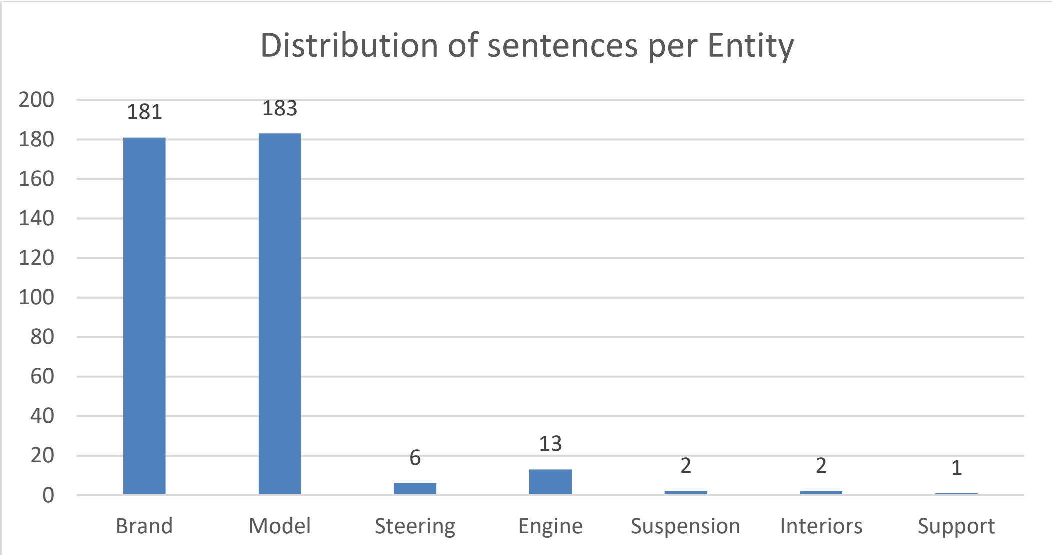


Fig. 32 – Distribution of sentences per Entity in Car Brands Dataset

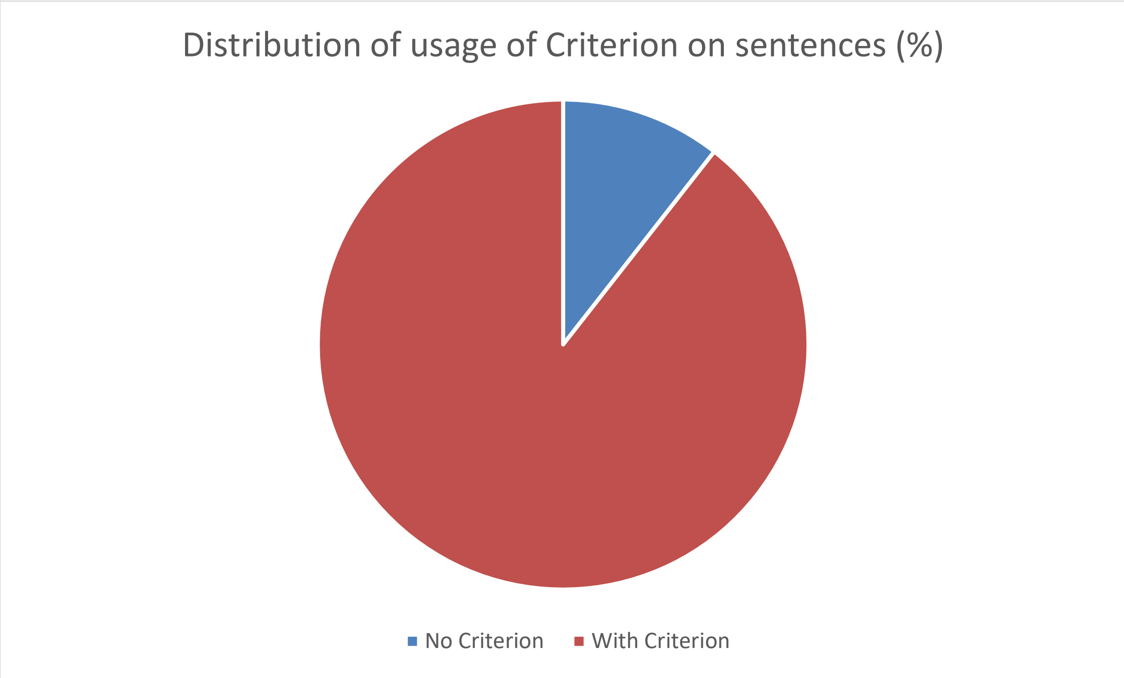


Fig. 33 – Distribution of usage of Criterion on sentences (%) in Car Brands Dataset

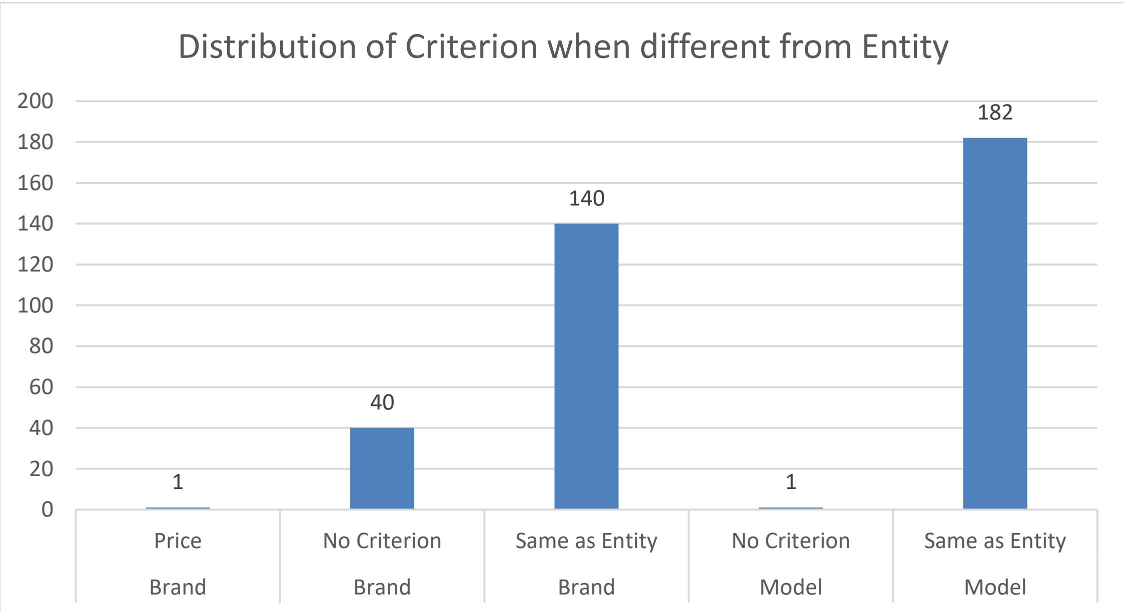


Fig. 34 – Distribution of Criterion when different from Entity in Car Brands Dataset

5.2 Intra-sentence Segmentation

From the available ways described in 2.3.3, the RegexParser was the chosen way to approach this task for this problem. As mentioned in that sub-section, the author believes there are two ways to use this method: the first is to extract meaningful information that constitutes the chunks, and the second would be to remove non-meaningful information from the sentences, breaking the sentences into chunks naturally.

Both approaches were implemented initially, in which the second one proved to have better results early on, and it was the one that decided to evolve. The first regex consisted in using the CC (Coordinating conjunction) tag from NLTK's POS tags[60], which consists of terms like "and" and "but", in addition to the comma, dot, and semi-colon (<CC>|<,>|<.>|<;>).

The next upgrade to the method included a comparison detection mechanism. With some research[61] and trial and error, a regex was created that could detect when a comparison was being made (<RB|RBR>?<JJ|JJR><IN><NN|NNP|NNS>). This regex consists of initially detecting if an adverb or comparative adverb was used, followed by a comparative adjective, a subordinating preposition or conjunction, and ending on a noun, common, singular, or mass, or proper, singular, or common, plural.

The next step was the removal of noise from the results. This noise comes from using sentence connectors. Since we are working on the sentence level, we do not need these connectors. They usually followed by a comma, which made them noise on the results of the method so far. From this list[62] and analyzing both datasets, the following list of noise words was considered in following Tab 11:

Again	However	Also	Furthermore	Moreover
Likewise	Equally	However	Nevertheless	Otherwise
Meanwhile	Eventually	Currently	Hence	Therefore
Thereupon	Clearly	Anyways	Especially	Specifically
etc	though	Edit	Basically	Sure
Additionally	Besides	Similarly	Finally	Summarising
Though	Instead	Nonetheless	Equally	Thus
Consequently	Accordingly			

Tab 11 – List of Noise Words considered

One significant problem still stands, even though we are correctly chunking the sentences and assigning the suitable chunks to the annotations. The OTE annotation consisted of an apparent reference to the Entity present in an opinion. This way, matching the OTE annotation to the resulting chunks led to a jump in the quality of the results.

The final change to the method was removing the automatic association of chunks with the annotations when they matched in length (three annotations and three resulting chunks leading to direct association), but this did not improve the results, so the automatic association stayed.

[60] ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos
[61] stackoverflow.com/questions/15388831/what-are-all-possible-pos-tags-of-nltk
[62] medium.com/@danhduy9/list-of-sentence-connectors-in-english-with-examples-1952c02fa374

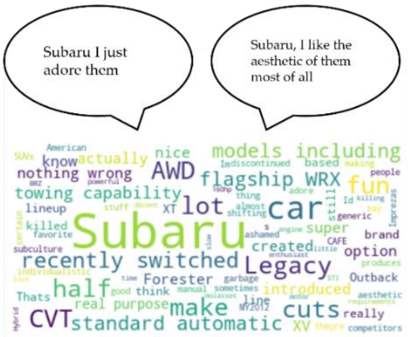
5.3 Clustering

After multiple experiments with different input data and configurations of approaches, it was concluded that the best approach to solve the objective of this work was the Kmeans++ Clustering technique with SBERT word embedder and UMAP reducing to 2 dimensions. Utilizing this approach on the baseball dataset wielded a Silhouette Score of 0.63 with 8 clusters. Analyzing these clusters manually and through word clouds (Fig. 35 A to H) made us accept this approach because of the variety and the excellent division between them, even though it did not have the highest Silhouette Score of all approaches.

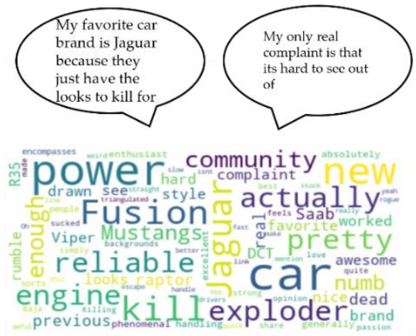
(A) Ted Williams cluster
(B) Mike Trout cluster
(C) Barry Bonds cluster
(D) Babe Ruth cluster
(E) Others cluster
(F) Willie Mays cluster
(G) Nolan Ryan and Roger Clemens cluster
(H) Mickey Mantle cluster

Fig. 35 – Wordclouds with the Clustering results of the baseball dataset

Validating this approach on the other dataset the author had, the car brands dataset, wielded good results but not as good as the baseball dataset results, with 16 clusters and 0.59 Silhouette Score. The word clouds (Fig. 36 A to H and Fig. 37 A to H) show some variety, but some clusters could have been better divided.

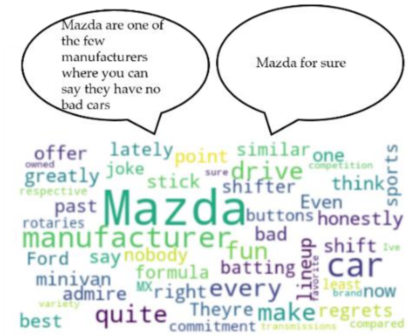


(A) Subaru cluster

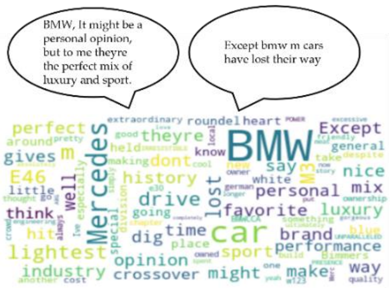


(B) Jaguar and Mustang cluster

(C) Porsche cluster

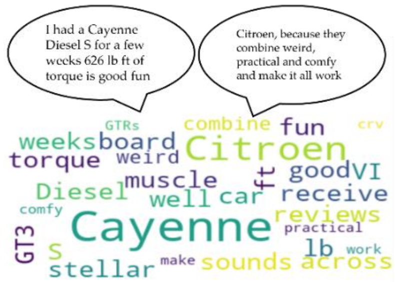


(D) Mazda cluster

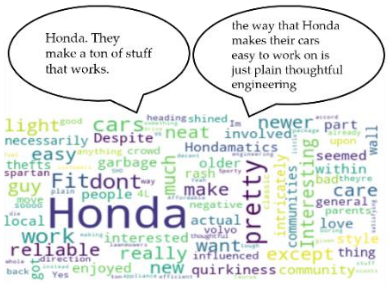


(E) Mercedes and BMW cluster

(F) Hyundai cluster

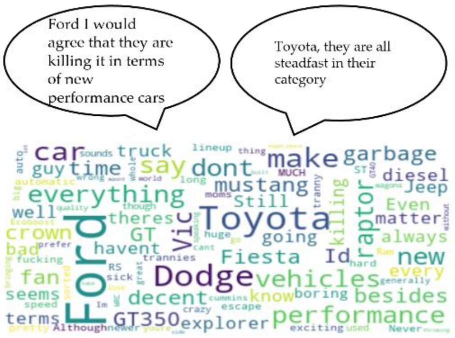


(G) Mercedes and BMW cluster

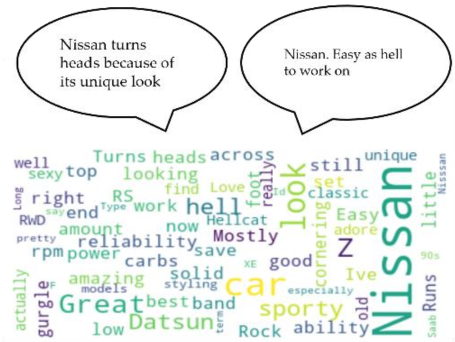


(H) Honda cluster

Fig. 36 – Wordclouds with some clusters results of the car brands dataset

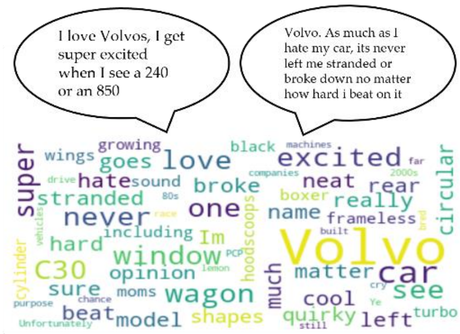


(A) Ford, Toyota and Dodge cluster



(B) Nissan cluster

(C) Audi and Volkswagen cluster



(D) Volvo cluster

(E) Others cluster

(F) Mclaren cluster

(G) Pontiac cluster

(H) Mitsubishi cluster

Fig. 37 – Wordclouds with the remaining clusters results of the car brands dataset

This discrepancy can be attributed to multiple factors like the annotation quality of the datasets, the size since the baseball dataset is more significant than the car brands dataset, the distribution of Alternatives in each dataset, the quality of the discussion, and the arguments used. Nevertheless, the author believes this approach can dynamically organize the conversation based on the arguments used. In a natural setting, not fixating on the value of clusters and with even more data, this approach picks the correct number of clusters for the input data and groups that data into perceivable clusters that can then be utilized in intelligent reports for the decision-makers.

6 Experimentation and Evaluation

This section describes the process of analysis, experimentation, and evaluation of the solution. Initially, the author identifies the hypothesis and description of the evaluation indicators. Afterward, the author explains the evaluation methodology and, in the end, the evaluation of the experiments and results.

6.1 Hypothesis

A research hypothesis is a claim about the relation of two or more variables. It needs to be a specific, quantifiable, verifiable, and testable, predictive statement about a possible outcome of scientific research where metrics and tests previously identified are used to authenticate the veracity of the results (Lavrakas, 2012; Wolverton, 2009).

The main goal of this project is to provide a web service that provides clusters of participants on a discussion based on the feature decided, segmenting the participants by the Alternative they support and by the most important criterion used by each participant to evaluate the set of Alternatives available. Therefore, the hypothesis focuses on the effectiveness of the web service and the quality of the resulting segments.

6.2 Evaluation Indicators

This section analyzes and theorizes some Clustering metrics on how the web service is evaluated.

6.2.1 Clustering Metrics

The results of a Clustering technique can be evaluated through metrics that might consider the ground truth labels if they are available. Ground truth labels are humanly provided classifications of the data on which the algorithms are trained or evaluated against them (Urumov, 2021). Some metrics are addressed ahead.

6.2.1.1 Intrinsic Metrics

When ground truth labels are unavailable, only a few metrics are available to evaluate the performance of a Clustering technique (Han et al., 2012). The available metrics for this case are explained ahead.

Silhouette Coefficient. Silhouette is a method that provides a concise measure of how similar an object is to its' cluster compared to other clusters through the usage of distance metrics. Typically, Euclidian is the one used to calculate the Silhouette Coefficient. The results range from -1 to 1, where a high value means the clusters are well separated, minimizing the intra-cluster and maximizing the distance inter-cluster (Rousseeuw, 1987).

Calinski-Harabasz Index. Calinski-Harabasz Index (CH), also known as the Variance Ratio Criterion, is a measure of how similar an object is to its cluster (cohesion) compared to other clusters (separation) (Caliñski & Harabasz, 1974). The higher value of the CH index means the clusters are dense and well separated, although there is no "acceptable" cut-off value. We need to choose a solution that gives a peak or a sharp elbow on the line plot of CH indices. On the other hand, if the line is smooth (horizontal or ascending or descending), then there is no reason to prefer one solution over others (Dey, 2021).

Davies–Bouldin Index. Davies–Bouldin Index (DBI) is a measure that indicates the similarity of clusters and assumes that the data has density and uses a decreasing function of distance from a vector characteristic of the cluster. The measure can be used to infer the appropriateness of data partitions and can therefore be used to compare the relative appropriateness of various divisions of the data (Davies & Bouldin, 1979).

6.2.1.2 Extrinsic Metrics

When ground truth labels are available, some metrics exist to evaluate the performance of a Clustering technique (Han et al., 2012). The available metrics for this case are explained ahead.

Mutual Information. Mutual Information functions based on Entropy, and Entropy decreases as the uncertainty decreases. This way, Mutual Information reduces the Entropy of class labels when we are given the cluster labels, allowing us to know how much the uncertainty about class labels decreases when we know the cluster labels, similar to information gain in decision trees (Yıldırım, 2021).

Homogeneity. Homogeneity is a measure where a Clustering technique must assign only those data points members of a single class to a cluster. The class distribution within each cluster should be skewed to a single class, that is, zero Entropy. To determine how close a given Clustering is to this ideal solution, an examination of the conditional Entropy of the class distribution given the proposed Clustering results (Rosenberg & Hirschberg, 2007).

Completeness. Completeness is symmetrical to homogeneity. A Clustering technique must assign all those data points members of a single class to a cluster. The distribution of cluster assignments within each class is examined. These distributions are skewed entirely to a single cluster in a complete Clustering solution. This degree of skew can be evaluated by calculating the conditional Entropy of the proposed cluster distribution given the class of the component data points (Rosenberg & Hirschberg, 2007).

V-measure. V-measure is an Entropy-based measure that explicitly measures how successfully satisfied the criteria of homogeneity and completeness are. V-measure is computed as the harmonic mean of distinct homogeneity and completeness scores, just as precision and recall are commonly combined into F-measure. As F-measure scores can be weighted, V-measure can be weighted to favor the contributions of homogeneity or completeness (Rosenberg & Hirschberg, 2007).

Rand index. Rand index or Rand measure is a similarity measure between two different Clustering results of the same data set. The measure considers how each pair of data points is assigned in each Clustering (Rand, 1971).

6.2.2 Web Service Evaluation

This web service is a module in a fully intelligent GDSS that allows the complete segmentation of participants based on their opinion on the discussion. This web service's success depends on the other modules that compose the GDSS, like how the unstructured text is processed and the accuracy of other modules that extract features from the text, like the Alternatives it supports or the Criteria used.

The evaluation of the web service is divided into three parts in which the metrics were adapted from the usability factors (Shafinah et al., 2010):

1. Evaluation of the intra-sentence segmentation algorithm with the following metrics in Tab 12:

Metric	Definition
Efficiency	The time needed to obtain the results
Accuracy	Correctness of the output information
Effectiveness	The capability of the algorithm to achieve the intended goals

Tab 12 – Intra-sentence segmentation evaluation

2. Evaluation of the Clustering algorithm with the following metrics in Tab 13:

Metric	Definition
Efficiency	The time needed to obtain the results
Accuracy	Correctness of the output information
Effectiveness	The capability of the algorithm to achieve the intended goals

Tab 13 – Clustering evaluation

3. Evaluation of the web service with the following metrics in Tab 14:

Metric	Definition
Efficiency	The time needed to obtain the results
Effectiveness	The capability of the web service to achieve the intended goals

Tab 14 – Web service evaluation

6.3 Evaluation Methodology

As described previously in the hypothesis section (6.1), the hypothesis focuses on the effectiveness of the web service and the quality of the resulting segments of users. For that, the author created evaluation indicators, as presented in the evaluation methodology.

6.3.1 Intra-sentence segmentation evaluation

For intra-sentence segmentation evaluation, the plan is the conception and simulation of scenarios and case studies with data found in discussions taken on the web. Those are accessible and appropriate for the system, varying their context to demonstrate the algorithm's applicability to different domains where a discussion could occur.

6.3.2 Clustering evaluation

For Clustering evaluation, the plan is the conception and simulation of scenarios and case studies with data found in discussions taken on the web that are accessible and appropriate for the system, varying the context of them to demonstrate the applicability of the algorithm to different domains in which a discussion could take place.

6.3.3 Web service evaluation

For web service evaluation, the author performs load tests on the web service with the help of Postman. Unit tests to a web service can be allocated together, becoming a collection, and Postman’s Collection Runner enables the creation of load tests in which the user can configure specific parameters like:

- Iteration – number of requests to do
- Delay – Ramp up time, how often the users use the web-service
- API collection and listing – choose which collection and which requests are done during the load test
- Environment selection

6.4 Evaluation of Intra-sentence segmentation

Following the road that the intra-sentence segmentation tool described in 5.2, in Fig. 38, we can see how the tool's accuracy evolved as more features were added to it when tested in the baseball dataset. Adding the OTE helper to assist the assignment of the resulting chunks back to the annotation gave the most significant bump to the tool, followed by removing the noise words.

Removing the automatic association had an insignificant increase in the tool's accuracy, in which the author used the car brands dataset to decide whether to keep the automatic association. By the result of this test, as we can see in Fig. 39, the automatic association led to better results in the car brands dataset. It was decided to keep it since it was much better in the car brands dataset and insignificantly worse in the baseball dataset.

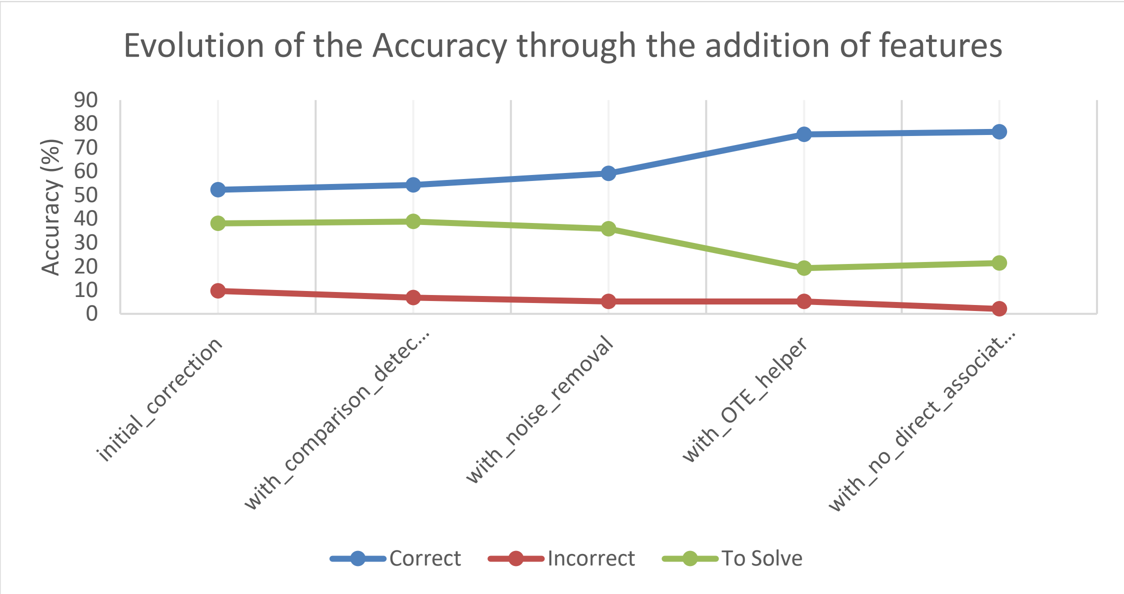


Fig. 38 – Evolution of the Accuracy of the Intra-sentence Segmentation method in the Baseball Dataset

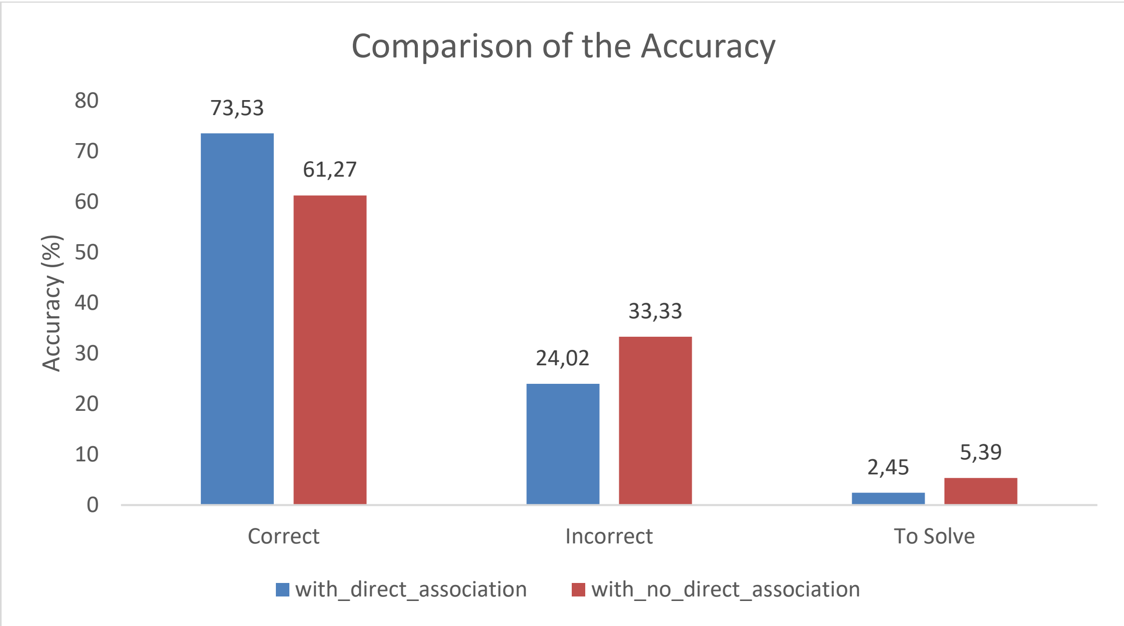


Fig. 39 – Comparison of the results when direct association was disabled in the Car Brands Dataset

6.5 Evaluation of Clustering

This section describes how the Clustering is evaluated, encompassing the proposed approach and the comparison and evaluation methodology. It utilizes the pipeline described in 4.2.

6.5.1 Sentences as only input data

In this subset of experiments, only the sentences typed by the participants of the Reddit discussion were used as input data for each Clustering technique. Initially, to evaluate which word embedders we should use moving forward, we decided to fixate the k (number of clusters hyperparameter) to the number of Alternatives on the baseball dataset. Since the Baseball dataset was manually annotated, the ground truth la-bels were available, allowing us to use the Mutual Information (MI) metric to evaluate the performance of the approaches. The author decided to use MI since other metrics that use ground truth labels, such as Homogeneity and Completeness, have MI as part of their calculations. Furthermore, MI compares the ideal Clustering results through the ground truth labels and the obtained Clustering results and determines how similar both are.

6.5.1.1 Word Embedders variation.

As shown in Fig. 40, using K-means as a baseline Clustering technique, Word2Vec, fastText, and GloVe all had similar results. Since all of them are static word vectors, GloVe was decided to be used from those three since it was the fastest computationally wise. SBERT performed better than BERT and was much better computational-wise, hence why BERT embeddings were dropped moving forward. Furthermore, ELMo is outperformed by SBERT as well, and since both embedders consider the context, SBERT was chosen to move forward as that type of embedder. This way, from these preliminary experiments, TF-IDF, GloVe, and SBERT were the chosen embedders to be kept using in the following experiments.

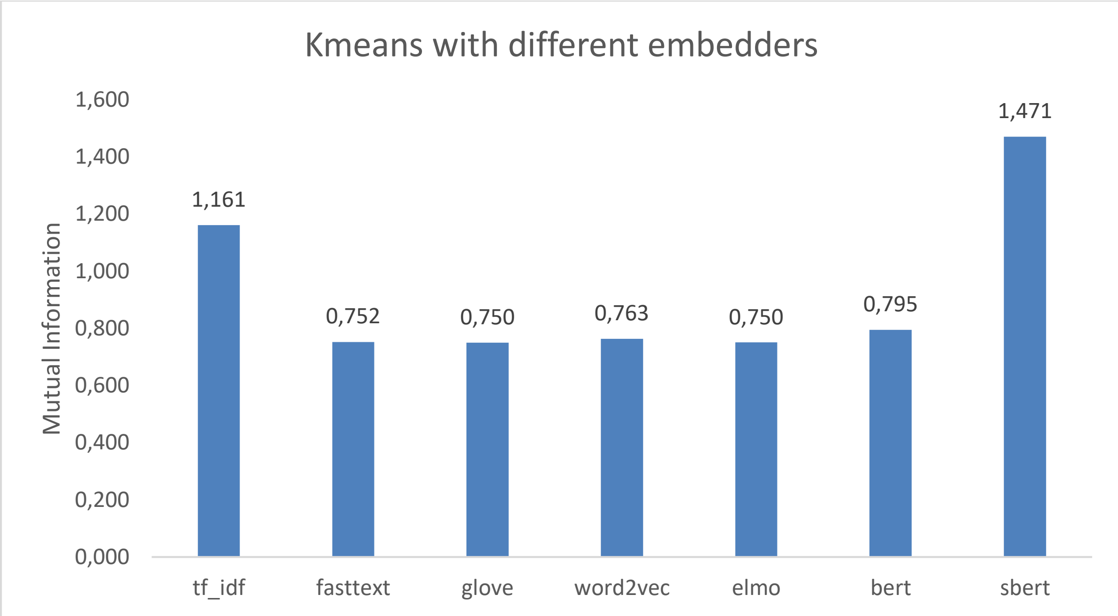


Fig. 40 – Kmeans performance with different word embedders on the baseball dataset

6.5.1.2 Dimensionality Reduction Techniques variation.

Having decided on word embeddings to be used in the experiments, we decided not to fixate the number of clusters (K) and test it out with multiple K ranging from 2 to 128 in exponentials of 2. With this change, our ground labels could not be used since the number of clusters might not be the same as the number of unique ground labels, leading to only the Silhouette Score getting used. The combination of the Silhouette Score and the K value it maxes out for each approach dictates the quality of the results.

Fig. 41 shows the best Silhouette Score for each technique without applying dimensionality reduction techniques. In contrast, Fig. 42 shows how the Clustering results improve when putting all embedders with the exact final dimensions 200 from the GloVe word embedder technique. We can see a tendency of UMAP to outperform PCA in this number of dimensions.

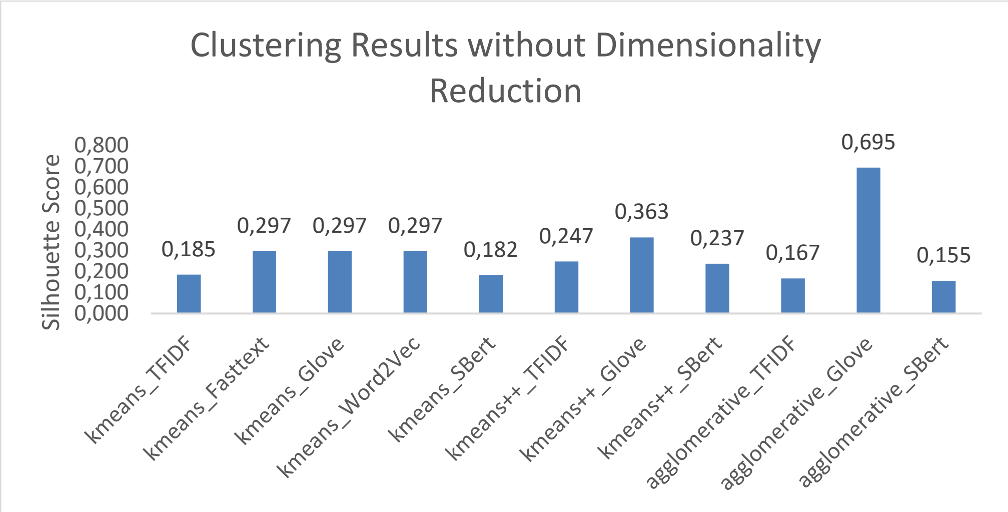


Fig. 41 – Clustering results in the new test settings on the baseball dataset

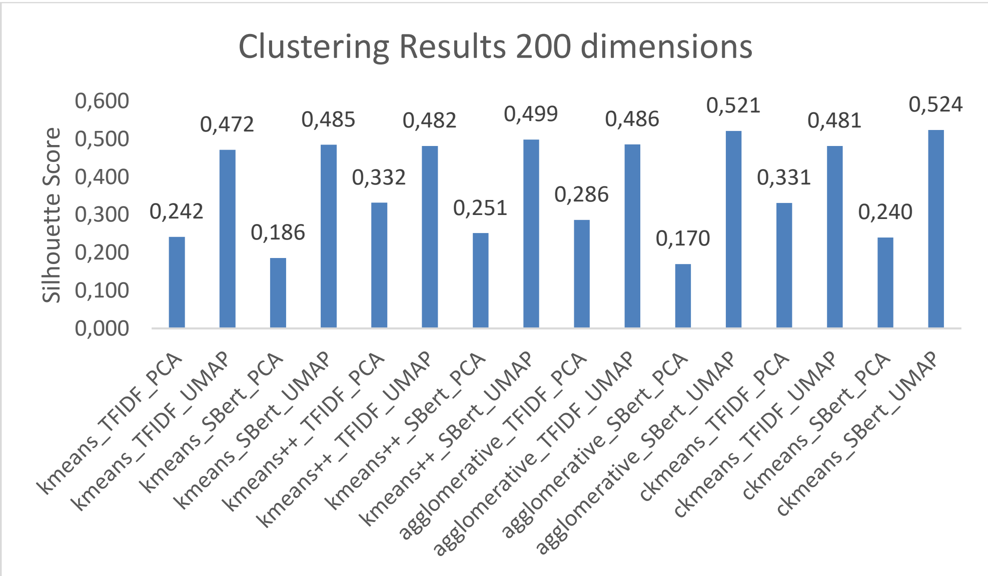


Fig. 42 – Clustering results with every approach at the 200 dimensions on the baseball dataset

Fig. 43 shows how the Silhouette Score varies when reducing the number of dimensions of the embeddings. Analyzing the tendencies of the approaches since multiple approaches are overlapping, making them less readable, we can see that most approaches presented there show a minimal increase in performance until ten dimensions. From there, it steadily increases as dimensions get reduced, except for the Agglomerative Hierarchical Clustering technique with GloVe word embedder and PCA dimensionality reduction. That approach shows a stabilization until 15 dimensions and then a decrease in performance, joining the tendency of the remaining approaches on that graph after ten dimensions. The same pattern can be seen in Fig. 44 with the TFIDF with PCA, which spiked in performance after ten dimensions. The Glove with UMAP does not have a perceivable pattern where it spikes at specific dimensions. Based on most approaches' patterns, a decision was made to experiment only with ten until one dimension.

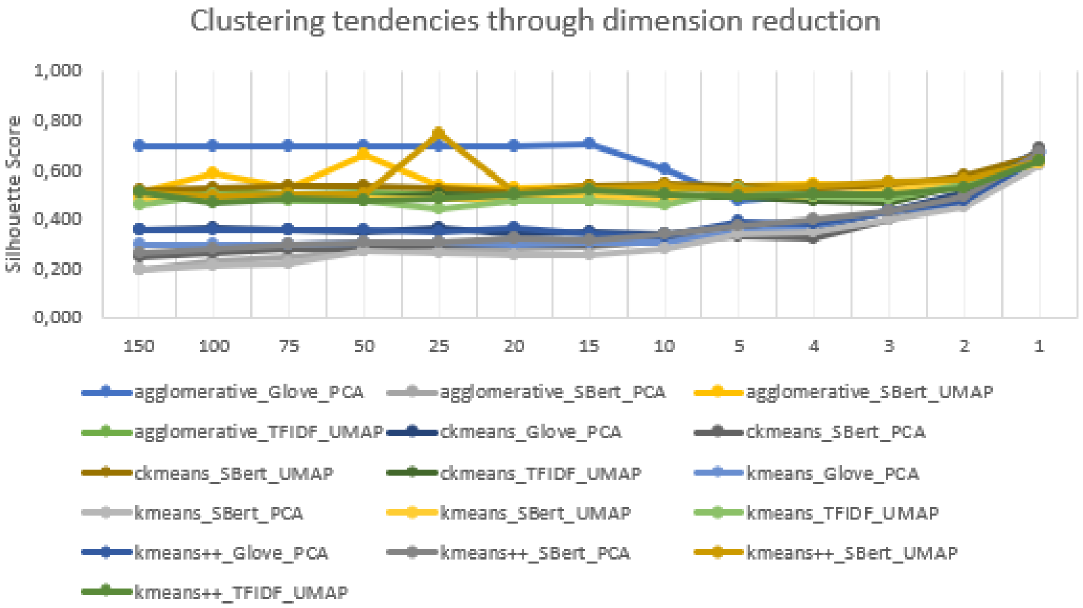


Fig. 43 – Clustering tendencies through dimension reduction on the baseball dataset

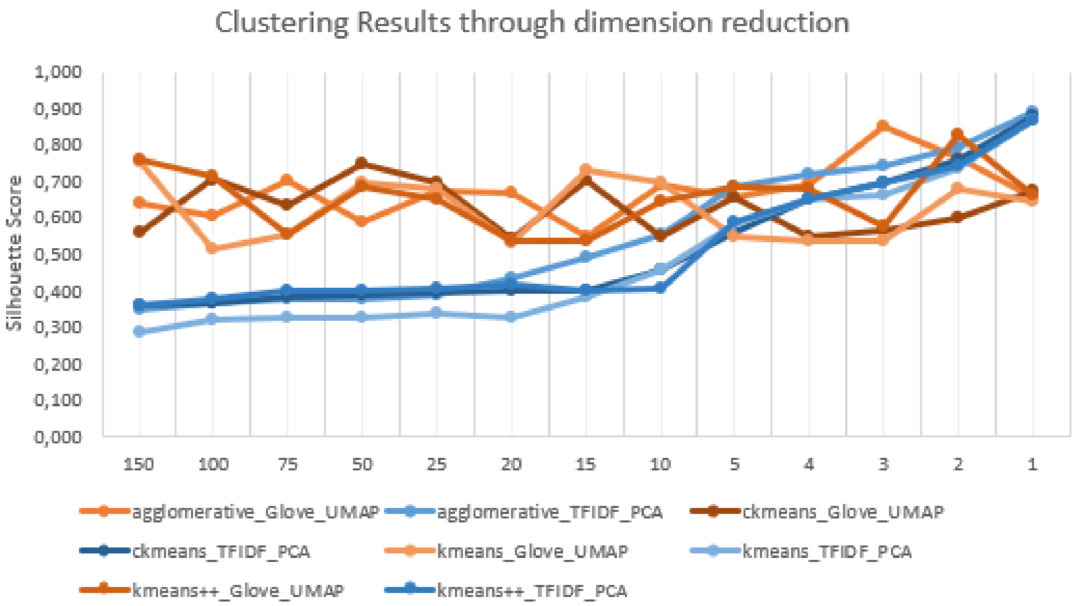
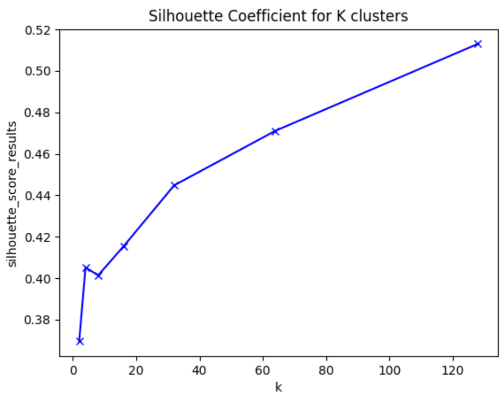
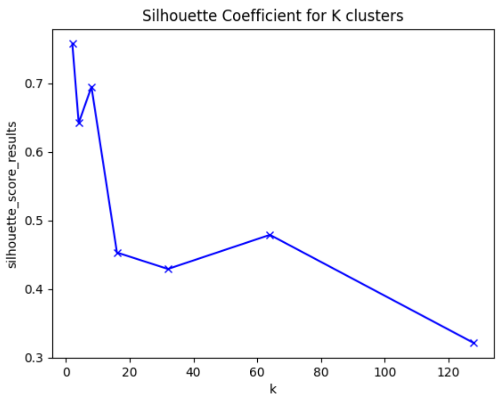


Fig. 44 – Clustering results through dimension reduction on the baseball dataset

Even though it displays promising results, Silhouette Score wise when analyzing the number of clusters used to reach those values, either maximizing at the 128 clusters (Fig. 45 (A)) or on 2 clusters (Fig. 45 (B)). This tendency is not the expected result for this task since 2 clusters groups the data with no perceivable differences, and 128 clusters are too many clusters with no interest in solving the problem that this work intends to solve. Therefore, a tradeoff between the Silhouette Score and the number of clusters a particular approach maxed its' Silhouette Score is considered when evaluating the obtained results.



(A) Maximizing in 128 clusters



(B) Maximizing in 2 clusters

Fig. 45 – Example of approaches performance through multiple Ks

6.5.2 Addition of Polarity to the input data

Adding Polarity to the input data did not change the results Silhouette Score-wise, nor did a brief analysis of the resulting clusters. Still, the number of clusters for the best Silhouette Score of each approach starts to show good values, with some approaches maximizing at 4 and 8 clusters. Each is more in line with the expected number of clusters when they are not predefined. Since one sentence can have multiple annotations, brief experiments were done to minimize the number of duplicated sentences, no duplicates, and two dupes max, with no interesting results to appoint.

6.5.3 Addition of Alternative and Criterion to the input data

Adding Alternative and Criteria to the input data (already with Sentence and Polarity) resulted in some interesting results, one approach that maximized at 4 clusters, kmeans++ Clustering technique with TFIDF word embedder, and PCA reducing to 2 dimensions. This approach divided the data into Polarity and Criterion, but with the Criterion part being if it existed or not, creating clusters with positive Polarity with Criterion and negative Polarity with Criterion. Unfortunately, this was not the objective of the work. However, it was interesting that an unsupervised technique made such a division, which led us to believe that the technique put too much emphasis on those two features and was not equally distributed. Only adding Alternative and Criteria to the sentence without Polarity did not achieve any compelling results.

6.5.4 Addition of Alternative to the Implicit Sentences Text

This way, the author believed that going entirely just for the sentence as the only input for the Clustering technique would bring more desirable results. With just the input sentence, the author adjusted some parameters, like the range of K to be from 2 to 20 in steps of 1, and decided to add the Alternative to the input sentence, resulting in high-quality clusters closely related to what was expected. The author found that two dimensions were the best number to reduce dimensions to since it is more in line with practices of the area where a reduction to 2 dimensions for visualization is made. Most approaches maximized their Silhouette Score at an adequate K (not in the lowest value of 2 or the highest value of 20) with 2 dimensions. The best approaches did not show any improvements between the 2 and 1 dimensions.

Adding the Alternative at the beginning or the end of the sentence did not wield any significant changes to the Silhouette Score.

6.5.5 Final Considerations

As shown in Fig. 46, the best approach was the agglomerative hierarchical Clustering technique with TFIDF word embedder and PCA reducing to 2 dimensions. However, this approach reached the best value at 2 clusters, which we previously discarded. Therefore, the accepted approach to solve the objective of this work was the kmeans++ Clustering technique with SBERT word embedder and UMAP reducing to 2 dimensions which resulted in 8 clusters.

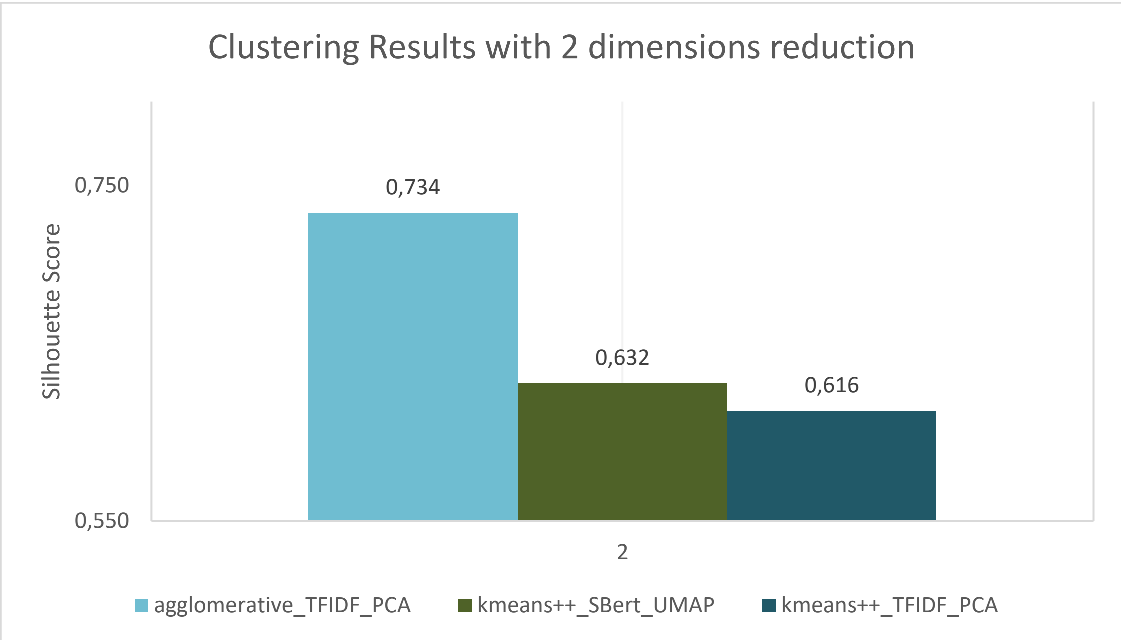


Fig. 46 – Best Clustering results with 2 dimensions reduction on the baseball dataset

7 Conclusion

In this section, the conclusions attained during the development of this work are presented. A summary of the thesis and an analysis of the achieved outcomes, project limitations, and possible future work are performed.

7.1 Summary

Group Decision Support Systems are constantly evolving to make them more suitable and accepted by organizations. The increasing utilization of Machine Learning in these systems is expanding the horizon of what they can do, therefore shooting up the value of these systems by giving them intelligent aspects. Through Argument Mining, more concretely, Aspect Based Sentiment Analysis, decision-makers improve their understanding of the conversations taking place by receiving organized and intelligent reports. This work intends to dynamically organize the conversations taking place by grouping them on the arguments used, allowing the decision-makers to perceive the conversation's route better.

To achieve our goals, we studied and experimented with multiple Clustering techniques, word embedders, and dimensionality reduction techniques to understand what configuration of these three techniques works best for the GDM context. From the tested experiments, the best approach consisted of applying the K-means++ Clustering technique with SBERT word embedder and UMAP dimensionality reduction technique, reducing to 2 dimensions which resulted in 8 clusters with 0.63 Silhouette Score. Using the same approach on the validation dataset (car brands dataset) wielded satisfactory results but not as good as in the baseball dataset. This difference in results can be attributed to the small dimension of the car brands dataset and its higher dispersion concerning Alternatives leading to a higher number of formed clusters containing few observations. However, the author believes this approach is a feasible solution for the problem we intend to tackle. In a natural setting, not fixating on the value of clusters and with even more data, this approach picks the correct number of clusters for the input data and groups that data into perceivable clusters that can then be utilized in intelligent reports for the decision-makers.

7.2 Achieved Outcomes

From the described objectives in 1.3, an evaluation of them is performed. Following the order shown in the section:

- The first objective was achieved successfully. It required additional research after the first delivery of this document to overcome some obstacles found on the project, which resulted in section 2 of the document.
- The second objective was achieved in a way that Clustering has its' use as a dynamic conversation organizer since it is an unsupervised method in which we supplemented it with the Alternative when the sentence was implicitly mentioning an Alternative.
- The third and fourth objectives were achieved successfully when the models were tested in the same conditions to understand which Clustering techniques and preprocessing tools were the best, which led to evaluating the best setting on the Car Brands dataset to validate the setting.
- The fifth objective was not achieved because of the obstacles faced during the execution of this work. It made it necessary to reformulate the hypothesis question and reconsider how to move forward to achieve the best results in line with the GDSS system that has this work as a functionality.

From the described Functional and Non-functional requirements in 4.1, in terms of functional requirements, all were achieved. In terms of non-functional requirements, the functionality, usability, and performance constraints were met, while the others cannot be evaluated since the fifth objective was not achieved.

7.3 Limitations

In terms of limitations of this work, one major limitation is the lack of datasets on the Group Decision Making context. This limitation made it only possible to validate the best setting with the older dataset we had, the Car Brands Dataset, which is more limited and of lesser quality than the Baseball Dataset made during this work and resulted in a paper (section 1.4).

Since Alternatives are abundantly present in sentences in a straightforward way, it made it impossible for the Clustering techniques to divide the sentences based on the Criteria used, even throughout multiple experiments changing the input data of the Clustering technique.

In the case of the Alternatives being implicitly mentioned in the text, which we supplemented with the Alternative that we annotated, the Machine Learning model performs that job in the future. The inaccuracy of the model and the inaccuracy of the intra-sentence segmentation method affect the accuracy of this work.

7.4 Future Work

As future work, more datasets could be created to validate this work and create a custom stopword list based on further research. Picking up this work and transforming it into a web service based on the design shown in section 4.3.2, the last objective of the work, is also planned to be achieved. Additionally, the writing of a scientific paper to be submitted to a journal or conference regarding the intra-sentence segmentation tool in the context of Group Decision Making is planned.

References

Agrawal, R., & Wimmers, E. (2000). A framework for expressing and combining preferences. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, *29*(2), 297–305. https://doi.org/10.1145/335191.335423

Akash Kumar. (2022, March 10). *Principal Component Analysis with Python*. GeeksforGeeks. https://www.geeksforgeeks.org/principal-component-analysis-with-python/

Allaoui, M., Kherfi, M., & Cheriet, A. (2020). Considerably improving clustering algorithms using umap dimensionality reduction technique: A comparative study. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12119 LNCS*, 317–325. https://doi.org/10.1007/978-3-030-51935-3_34

Alonso, S., Perez, I. J., Cabrerizo, F. J., & Herrera-Viedma, E. (2009). A fuzzy group decision making model for large groups of individuals. *2009 IEEE International Conference on Fuzzy Systems*, 643–648. https://doi.org/10.1109/FUZZY.2009.5277355

AltexSoft. (2019). *Non-functional Requirements: Examples, Types, Approaches*. AltexSoft. https://www.altexsoft.com/blog/non-functional-requirements/#usability

Apidianaki, M., Tannier, X., & Richart, C. (2016). Datasets for aspect-based sentiment analysis in French. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 1122–1126. https://hal.archives-ouvertes.fr/hal-01838536

Appen. (2021). *Data Annotation: What Is It? Annotated Datasets, Tools, Services Defined*. Appen. https://appen.com/blog/data-annotation/

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, *07-09-Janu*, 1027–1035. https://doi.org/10.13140/2.1.4197.9845

Attardi, G., & Dell'Orletta, F. (2008). Chunking and Dependency Parsing. *Proceedings of LREC 2008 Workshop on Partial Parsing*, *21*, 3–8.

Bachani, N. (2020). *Chunking in NLP: decoded*. Towards Data Science. https://towardsdatascience.com/chunking-in-nlp-decoded-b4a71b2b4e24

Balani, N. (2016). *Evaluating Web service frameworks - Current and Future Technology Trends*. https://navveenbalani.dev/index.php/articles/evaluating-web-service-frameworks/

Bansal, N. (2020). *R vs Python for Data Science: The Best Data Science Language*. Sisense. https://www.sisense.com/blog/r-vs-python-whats-the-best-language-for-natural-language-processing/

Barbaresi, A. (2015). Collection, Description, and Visualization of the German Reddit Corpus. *Proceedings of the 2nd Workshop on Natural Language Processing for Computer-Mediated Communication / Social Media*, 7–11.

Becker, K., & Bacelo, A. (2000). The evaluation of GRADD: A GDSS supporting asynchronous and distributed meetings. *Proceedings - 6th International Workshop on Groupware, CRIWG 2000*, 19–26. https://doi.org/10.1109/CRIWG.2000.885151

Bezdek, J. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. In *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer US. https://doi.org/10.1007/978-1-4757-0450-1

Bhatia, S. (2004). Adaptive K-Means Clustering. *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA*. https://doi.org/10.13140/2.1.4197.9845

Boe, B. (2014). *PRAW: The Python Reddit Api Wrapper* . PRAW 3.0.0 Documentation. https://praw.readthedocs.io/en/v3.0.0/

Bogdani. (2016). *Text Chunking with NLTK*. NLP-FOR-HACKERS. https://nlpforhackers.io/text-chunking/

Bogomolov, T., Korolkiewicz, M., & Bogomolova, S. (2019). Identifying Patterns in Fresh Produce Purchases. In *Handbook of research on big data clustering and machine learning* (pp. 378–408). IGI Global. https://doi.org/10.4018/978-1-7998-0106-1.ch018

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Bordogna, G., & Pasi, G. (2011). Soft clustering for information retrieval applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(2), 138–146. https://doi.org/10.1002/widm.3

Borza, J. (2011). FAST Diagrames: The Foundation for Creating Effective Function Models. *Trizcon 2011*, 1–10.

Bramer, M. (2007). Clustering. In *Principles of Data Mining* (pp. 221–238). Springer London. https://doi.org/10.1007/978-1-84628-766-4_14

Brans, J., & Vincke, P. (1985). Note—A Preference Ranking Organisation Method. *Management Science*, *31*(6), 647–656. https://doi.org/10.1287/mnsc.31.6.647

Brauers, W., Zavadskas, E., Peldschus, F., & Turskis, Z. (2008). MULTI-OBJECTIVE DECISION-MAKING FOR ROAD DESIGN. *TRANSPORT*, *23*(3), 183–193. https://doi.org/10.3846/1648-4142.2008.23.183-193

Brown, D. (1975). Aggregation of Preferences. *The Quarterly Journal of Economics*, *89*(3), 456. https://doi.org/10.2307/1885263

Brownlee, J. (2019). *Machine Learning Mastery: What Are Word Embeddings for Text?* Machine Learning Mastery. https://machinelearningmastery.com/what-are-word-embeddings/

Bryant, C., & White, L. (2019). Administrative Behavior. In *Managing Development in the Third World*. Simon and Schuster. https://doi.org/10.4324/9780429052125-6

Bueno, I., Carrasco, R., Porcel, C., Kou, G., & Herrera-Viedma, E. (2021). A linguistic multi-criteria decision making methodology for the evaluation of tourist services considering customer opinion value. *Applied Soft Computing*, *101*, 107045. https://doi.org/10.1016/j.asoc.2020.107045

Caliñski, T., & Harabasz, J. (1974). A Dendrite Method Foe Cluster Analysis. *Communications in Statistics*, *3*(1), 1–27. https://doi.org/10.1080/03610927408827101

Campello, R., Kröger, P., Sander, J., & Zimek, A. (2020). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(2), 1–15. https://doi.org/10.1002/widm.1343

Cardoso, T., Rodrigues, V., Conceição, L., Carneiro, J., Marreiros, G., & Novais, P. (2022). Aspect Based Sentiment Analysis Annotation Methodology for Group Decision Making Problems: An Insight on the Baseball Domain. In A. Rocha, H. Adeli, G. Dzemyda, & F. Moreira (Eds.), *Information Systems and Technologies* (pp. 25–36). Springer International Publishing. https://doi.org/10.1007/978-3-031-04819-7_3

Carneiro, J., Alves, P., Marreiros, G., & Novais, P. (2020). A conceptual group decision support system for current times: Dispersed group decision-making. *Advances in Intelligent Systems and Computing*, *1003*(July 2019), 150–159. https://doi.org/10.1007/978-3-030-23887-2_18

Carneiro, J., Alves, P., Marreiros, G., & Novais, P. (2021). Group decision support systems for current times: Overcoming the challenges of dispersed group decision-making. *Neurocomputing*, *423*, 735–746. https://doi.org/10.1016/j.neucom.2020.04.100

Carneiro, J., Andrade, R., Alves, P., Conceição, L., Novais, P., & Marreiros, G. (2020). A consensus-based group decision support system using a multi-agent microservices approach. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, *2020-May*, 2098–2100.

Carneiro, J., Laranjeira, J., Marreiros, G., Freitas, C., & Santos, R. (2012). A Context-aware Model to Support Ubiquitous Group Decision Making. *Journal of Internet Services and Information Security*, *2*(1/2), 105–118. http://isyou.info/jisis/vol2/no12/jisis-2012-vol2-no12-07.pdf

Carneiro, J., Martinho, D., Marreiros, G., Jimenez, A., & Novais, P. (2018). Dynamic argumentation in UbiGDSS. *Knowledge and Information Systems*, *55*(3), 633–669. https://doi.org/10.1007/s10115-017-1093-6

Carneiro, J., Martinho, D., Marreiros, G., & Novais, P. (2016). Introducing dynamic argumentation to UbiGDSS. *Advances in Intelligent Systems and Computing*, *474*, 471–479. https://doi.org/10.1007/978-3-319-40162-1_51

Carneiro, J., Martinho, D., Marreiros, G., & Novais, P. (2017). Including credibility and expertise in group decision-making process: An approach designed for UbiGDSS. *Advances in Intelligent Systems and Computing*, *570*, 416–425. https://doi.org/10.1007/978-3-319-56538-5_42

Carneiro, J., Martinho, D., Marreiros, G., & Novais, P. (2019). Arguing with Behavior Influence: A Model for Web-Based Group Decision Support Systems. *International Journal of Information Technology and Decision Making*, *18*(2), 517–553. https://doi.org/10.1142/S0219622018500542

Carneiro, J., Santos, R., Marreiros, G., & Novais, P. (2014). *Overcoming the Lack of Human-Interaction in Ubiquitous Group Decision Support Systems*. *49*(SoftTech), 116–124. https://doi.org/10.14257/astl.2014.49.24

Carneiro, J., Saraiva, P., Martinho, D., Marreiros, G., & Novais, P. (2018). Representing decision-makers using styles of behavior: An approach designed for group decision support systems. *Cognitive Systems Research*, *47*, 109–132. https://doi.org/10.1016/j.cogsys.2017.09.002

Carvalho, F., & Lechevallier, Y. (2009). Partitional clustering algorithms for symbolic interval data based on single adaptive distances. *Pattern Recognition*, *42*(7), 1223–1236. https://doi.org/10.1016/j.patcog.2008.11.016

CAST. (2021). *What Is Software Architecture - Examples, Tools, & Design*. CASTSoftware. https://www.castsoftware.com/glossary/what-is-software-architecture-tools-design-definition-explanation-best

Chakhar, S., Haddad, S., Mokdad, L., Mousseau, V., & Youcef, S. (2015). Multicriteria evaluation-based framework for composite web service selection. In *Evaluation and Decision Models with Multiple Criteria: Case Studies* (pp. 167–200). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-46816-6_6

Charnes, A., Cooper, W., Devoe, J., Learner, D., & Reinecke, W. (1968). A Goal Programming Model for Media Planning. *Management Science*, *14*(8), B-423-B-430. https://doi.org/10.1287/mnsc.14.8.b423

Chen, S., Liu, J., Wang, H., Xu, Y., & Augusto, J. (2014). A linguistic multi-criteria decision making approach based on logical reasoning. *Information Sciences*, *258*, 266–276. https://doi.org/10.1016/j.ins.2013.08.040

Chourdakis, E., & Thornton, C. (2021). *ClauCy*. GitHub. https://github.com/mmxgn/spacy-clausie

Chowdhury, G. (2003). Natural language processing. *Annual Review of Information Science and Technology*, *37*(1), 51–89. https://doi.org/10.1002/aris.1440370103

Čibej, J., Fišer, D., & Erjavec, T. (2016). Normalisation, Tokenisation and Sentence Segmentation of Slovene Tweets. *Normalisation and Analysis of Social Media Texts (NormSoMe)*, *2016*, 5–10. http://www.lrec-conf.org/proceedings/lrec2016/workshops/LREC2016Workshop-NormSoMe_Proceedings.pdf#page=10

Cid-López, A., Hornos, M., Carrasco, R., Herrera-Viedma, E., & Chiclana, F. (2017). Linguistic multi-criteria decision-making model with output variable expressive richness. *Expert Systems with Applications*, *83*, 350–362. https://doi.org/10.1016/j.eswa.2017.04.049

Clesham, M. (2020). *5 Stages of the Agile System Development Life Cycle*. BrightWork.Com. https://www.brightwork.com/blog/5-stages-of-the-agile-system-development-life-cycle

Conceição, L., Carneiro, J., Marreiros, G., & Novais, P. (2017). A SOA web-based group decision support system considering affective aspects. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10423 LNAI*, 65–74. https://doi.org/10.1007/978-3-319-65340-2_6

Conceição, L., Carneiro, J., Marreiros, G., & Novais, P. (2021). Applying machine learning classifiers in argumentation context. *Advances in Intelligent Systems and Computing*, *1237 AISC*, 314–320. https://doi.org/10.1007/978-3-030-53036-5_34

Costa, J., Melo, P., Godinho, P., & Dias, L. (2003). The AGAP system: A GDSS for project analysis and evaluation. *European Journal of Operational Research*, *145*(2), 287–303. https://doi.org/10.1016/S0377-2217(02)00535-0

Coursera. (2021). *What Is Python Used For? A Beginner's Guide*. Coursera. https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

Cunningham, P. (2008). Dimension reduction. *Cognitive Technologies*, 91–112. https://doi.org/10.1007/978-3-540-75171-7_4

Cycle, W. L. (2018). *History of Agile*. Planview LeanKit. https://www.planview.com/resources/guide/agile-methodologies-a-beginners-guide/history-of-agile/

D'Souza, S. (2017). Parser Extraction of Triples in Unstructured Text. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *5*(4), 143–148. https://doi.org/10.11591/ijai.v5.i4.pp143-148

Datagen. (2021). *The Danger of Hand Labeling Your Data*. Datagen. https://datagen.tech/the-danger-of-hand-labeling-your-data/

Davies, D., & Bouldin, D. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2), 224–227. https://doi.org/10.1109/TPAMI.1979.4766909

Daxenberger, J., Schiller, B., Stahlhut, C., Kaiser, E., & Gurevych, I. (2020). ArgumenText: Argument Classification and Clustering in a Generalized Search Scenario. *Datenbank-Spektrum*, *20*(2), 115–121. https://doi.org/10.1007/s13222-020-00347-7

Del Corro, L., & Gemulla, R. (2013). ClausIE: Clause-Based Open Information Extraction. *Proceedings of the 22nd International Conference on World Wide Web*, 355–366. https://doi.org/10.1145/2488388.2488420

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Denny, M., & Spirling, A. (2018). Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do about It. *Political Analysis*, *26*(2), 168–189. https://doi.org/10.1017/pan.2017.44

DeSanctis, G., & Gallupe, B. (1984). Group decision support systems: A new frontier. *ACM SIGMIS Database*, *16*(2), 3–10. https://doi.org/10.1145/1040688.1040689

DeSanctis, G., & Gallupe, R. (1987). Foundation for the Study of Group Decision Support Systems. *Management Science*, *33*(5), 589–609. https://doi.org/10.1287/mnsc.33.5.589

Deshpande, A., & Lobo, L. (2013). Text Summarization using Clustering Technique. *International Journal of Engineering Trends and Technology (IJETT)*, *4*(8).

http://julio.staff.ipb.ac.id/files/2011/12/IJETT-V4I8P120.pdf

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*(Mlm), 4171–4186. https://doi.org/10.48550/arxiv.1810.04805

Dey, D. (2019). *Affinity Propagation in ML | To find the number of clusters*. GeeksforGeeks. https://www.geeksforgeeks.org/affinity-propagation-in-ml-to-find-the-number-of-clusters/

Dey, D. (2021). *Calinski-Harabasz Index – Cluster Validity indices*. Geeks for Geeks. https://www.geeksforgeeks.org/calinski-harabasz-index-cluster-validity-indices-set-3/

Django Software Foundation. (2018). *The Web framework for perfectionists with deadlines | Django*. Django Software Foundation. https://www.djangoproject.com/

Dorigo, M., & Stutzle, T. (2004). Ant Colony. *Optimization*.

Dragos, D., & Schmeelk, S. (2020). What are they Reporting? Examining Student Cybersecurity Course Surveys through the Lens of Machine Learning. *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 961–964. https://doi.org/10.1109/ICMLA51294.2020.00156

Dumani, L., Neumann, P. J., & Schenkel, R. (2020). A Framework for Argument Retrieval. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12035 LNCS* (Issue 375342983, pp. 431–445). Springer International Publishing. https://doi.org/10.1007/978-3-030-45439-5_29

Dumani, L., & Schenkel, R. (2020). Quality-Aware Ranking of Arguments. *International Conference on Information and Knowledge Management, Proceedings*, 335–344. https://doi.org/10.1145/3340531.3411960

Dunn, J. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, *3*(3), 32–57. https://doi.org/10.1080/01969727308546046

Eilon, S. (1969). WHAT IS A DECISION. *Mgmt Science*, *16*(4), 172–189. https://doi.org/10.1287/mnsc.16.4.b172

Escudeiro, N. (2012). *Semi-automatic classification : using active learning for efficient class coverage* [Faculdade de Engenharia da Universidade do Porto]. https://repositorio-aberto.up.pt/bitstream/10216/73245/2/25912.pdf

Färber, M., & Steyer, A. (2021). *Towards Full-Fledged Argument Search: A Framework for Extracting and Clustering Arguments from Unstructured Text*. http://arxiv.org/abs/2112.00160

Flask. (2010). *Welcome to Flask -- Flask Documentation (2.0.x)*. Pallets Projects. https://flask.palletsprojects.com/en/2.0.x/%0Ahttps://flask.palletsprojects.com/en/2.0.x/#

Forgy, E. (1965). Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Biometrics*, *21*, 768–769.

Franzoni, V., Li, Y., & Mengoni, P. (2017). A path-based model for emotion abstraction on facebook using sentiment analysis and taxonomy knowledge. *Proceedings - 2017 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2017*, 947–952. https://doi.org/10.1145/3106426.3109420

Frazão, T., Camilo, D., Cabral, E., & Souza, R. (2018). Multicriteria decision analysis (MCDA) in health care: A systematic review of the main characteristics and methodological steps. *BMC Medical Informatics and Decision Making*, *18*(1), 1–16. https://doi.org/10.1186/s12911-018-0663-1

Freund, Y., Iyer, R., Schapire, R., & Singer, Y. (2004). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, *4*(6), 933–969. https://doi.org/10.1162/1532443041827916

Frey, B., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*(5814), 972–976. https://doi.org/10.1126/science.1136800

Gaertler, M. (2005). Clustering. In U. Brandes & T. Erlebach (Eds.), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 3418 LNCS* (pp. 178–215). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-31955-9_8

Ganesan, K. (2019). *All you need to know about Text Preprocessing for Machine Learning & NLP*. KDnuggets. https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html

GeeksforGeeks. (2020). *Software Engineering | Requirements Engineering Process*. GeeksforGeeks. https://www.geeksforgeeks.org/software-engineering-requirements-engineering-process/

Girvan, M., & Newman, M. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(12), 7821–7826. https://doi.org/10.1073/pnas.122653799

Goldberg, D., & Holland, J. (1988). Genetic Algorithms and Machine Learning. *Machine Learning*, *3*(2), 95–99. https://doi.org/10.1023/A:1022602019183

Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, *10*(1), 92. https://doi.org/10.2200/S00762ED1V01Y201703HLT037

Graf, A., & Maas, P. (2008). Customer value from a customer perspective: A comprehensive review. *Journal Fur Betriebswirtschaft*, *58*(1), 1–20. https://doi.org/10.1007/s11301-008-0032-8

Guo, Y., Ge, T., & Wei, F. (2020). Fact-aware sentence split and rephrase with permutation invariant training. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, *1*(2), 7855–7862. https://doi.org/10.1609/aaai.v34i05.6291

Gupta, A., & Tripathy, B. (2018). Implementing GloVe for context based k-means++ clustering. *Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2017*, 1041–1046. https://doi.org/10.1109/ISS1.2017.8389339

Gupta, A., & Wilemon, D. (1990). Accelerating the Development of Technology-Based New Products. *California Management Review*, *32*(2), 24–44. https://doi.org/10.2307/41166603

Gutierrez, C., Montero-Simo, M., Araque-Padilla, R., & Gutierrez-Gutierrez, L. (2013). Evaluation of perceived value in the consumption of coffee with ethical attributes. *Agrociencia (Montecillo)*, *47*(2), 195–207.

Hamdani, H., Wardoyo, R., & Mustofa, K. (2018). Weighting model for group decision support system: A review. *Indonesian Journal of Electrical Engineering and Computer Science*, *11*(3), 962–974. https://doi.org/10.11591/ijeecs.v11.i3.pp962-974

Hamdani, N., & Hamdadou, D. (2019). A multicriteria group decision support system: An approach based agents and web services. *International Journal of Information Technology and Web Engineering*, *14*(2), 1–26. https://doi.org/10.4018/IJITWE.2019040101

Han, J., Kamber, M., & Pei, J. (2012). Cluster Analysis. In *Data Mining* (pp. 443–495). Elsevier. https://doi.org/10.1016/B978-0-12-381479-1.00010-1

Hartigan, J. (1977). Clustering Algorithms. *Journal of Marketing Research*, *14*(1). https://doi.org/10.2307/3151073

He, W. (2021). *Using Natural Language Processing Techniques to Analyze the Impact of Covid-19 on Stock Market*.

Hellkamp, M. (2018). *Bottle: Python Web Framework*. Bottle. https://bottlepy.org/docs/dev/

Hema, D., & David, V. (2018). Fuzzy Clustering and Genetic Algorithm for Clinical Pratice Guideline Execution Engines. *2017 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2017*, 13–16. https://doi.org/10.1109/ICCIC.2017.8524335

Hinton, G., & Roweis, S. (2002). Stochastic Neighbor Embedding. *Advances in Neural Information Processing Systems*, *15*. https://papers.nips.cc/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf

Hirokawa, R., & Rost, K. (1992). Effective Group Decision Making in Organizations: Field Test of the Vigilant Interaction Theory. *Management Communication Quarterly*, *5*(3), 267–288. https://doi.org/10.1177/0893318992005003001

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417–441. https://doi.org/10.1037/H0071325

Hotelling, H. (1936). Relations Between Two Sets of Variates. *Biometrika*, *28*(3/4), 321. https://doi.org/10.2307/2333955

Huang, Y., Chen, C., Xing, Z., Lin, T., & Liu, Y. (2018). Tell them apart: distilling technology differences from crowd-scale comparison discussions. *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 214–224.

https://doi.org/10.1145/3238147.3238208

Huber, G. (1984). Issues in the design of group decision support sytems. *MIS Quarterly: Management Information Systems*, *8*(3), 195–204. https://doi.org/10.2307/248666

Husain, A. (2014). A multi-agent system for scalable group decision making. *Applied Mathematical Sciences*, *8*(109–112), 5543–5552. https://doi.org/10.12988/ams.2014.45367

Hwang, C.-L., & Yoon, K. (1981). *Multiple Attribute Decision Making: An Introduction*. Springer-Verlag.

IBM Cloud Education. (2021). *What is Data Labeling?* IBM Cloud Blog. https://aws.amazon.com/sagemaker/data-labeling/what-is-data-labeling/%0Ahttps://www.ibm.com/cloud/learn/data-labeling

Iliassich, L. (2016). *Clustering Algorithms: From Start To State Of The Art*. Toptal. https://www.toptal.com/machine-learning/clustering-algorithms

Iwai, A., & Sado, K. (2010). A design of Web-based GDSS that supports anonymous communication and the convergent process. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010*, *I*, 693–698.

Jabreel, M., Maaroof, N., Valls, A., & Moreno, A. (2021). Introducing sentiment analysis of textual reviews in a multi-criteria decision aid system. *Applied Sciences (Switzerland)*, *11*(1), 1–26. https://doi.org/10.3390/app11010216

Jalili, M., & Rezaie, K. (2010). Quality principles deployment to achieve strategic results. *International Journal of Business Excellence*, *3*(2), 226–259. https://doi.org/10.1504/IJBEX.2010.030730

Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Wooldridge, M., & Sierra, C. (2001). Automated Negotiation: Prospects, Methods and Challenges. *Group Decision and Negotiation*, *10*(2), 199–215. https://doi.org/10.1023/A:1008746126376

Johnson, D. (2022). *What is R Programming Language? Introduction & Basics of R*. Guru99. https://www.guru99.com/r-programming-introduction-basics.html

Jurafsky, D., & Martin, J. (2021). *Speech and Language Processing* (3rd ed.). https://web.stanford.edu/~jurafsky/slp3/

Kaewsanmua, K. (2021). *Data Labeling Software: Best Tools For Data Labeling in 2021*. NeptuneBlog. https://neptune.ai/blog/data-labeling-software

Kambhatla, N., & Leen, T. (1997). Dimension Reduction by Local Principal Component Analysis. *Neural Computation*, *9*(7), 1493–1516. https://doi.org/10.1162/neco.1997.9.7.1493

Karani, D. (2018). *Introduction to Word Embedding and Word2Vec*. Towards Data Science. https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

Kim, C., Yin, P., Soto, C., Blaby, I., & Yoo, S. (2018). Multimodal biological analysis using NLP and expression profile. *2018 New York Scientific Data Summit, NYSDS 2018 - Proceedings*.

https://doi.org/10.1109/NYSDS.2018.8538944

Kim, S. D., Zhang, B.-T., & Kim, Y. T. (2000). *Reducing parsing complexity by intra-sentence segmentation based on maximum entropy model*. 164–171. https://doi.org/10.3115/1117794.1117815

Kim, S. D., Zhang, B. T., & Kim, Y. T. (2001). Learning-based intrasentence segmentation for efficient translation of long sentences. *Machine Translation*, *16*(3), 151–174. https://doi.org/10.1023/A:1019896420277

Koen, P., Ajamian, G., Boyce, S., Clamen, A., Fisher, E., Fountoulakis, S., Johnson, A., Puri, P., & Seibert, R. (2002). Fuzzy Front End: Effective Methods, Tools, and Techniques. *Wiley, New York, NY*.

Koen, P., Ajamian, G., Burkart, R., Clamen, A., Davidson, J., D'Amore, R., Elkins, C., Herald, K., Incorvia, M., Johnson, A., Karol, R., Seibert, R., Slavejkov, A., & Wagner, K. (2001). Providing clarity and a common language to the "fuzzy front end." *Research Technology Management*, *44*(2), 46–55. https://doi.org/10.1080/08956308.2001.11671418

Koen, P., Bertels, H., & Kleinschmidt, E. (2014). Managing the front end of innovation-part II: Results from a three-year study. *Research Technology Management*, *57*(3), 25–35. https://doi.org/10.5437/08956308X5703199

Kotler, P., & Kelly, K. (2006). *Marketing Management* (12th ed.). Pearson. https://www.pearson.com/us/higher-education/product/Kotler-Marketing-Management-12th-Edition/9780131457577.html

Kraus, S., Hoz-Weiss, P., Wilkenfeld, J., Andersen, D., & Pate, A. (2008). Resolving crises through automated bilateral negotiations. *Artificial Intelligence*, *172*(1), 1–18. https://doi.org/10.1016/j.artint.2007.05.007

Kraus, S., Sycara, K., & Evenchik, A. (1998). Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, *104*(1–2), 1–69. https://doi.org/10.1016/s0004-3702(98)00078-2

Kumar, J., & Kumari, R. (2018). Finding Instantaneous Community of Ideologically Similar Users in Social Forums. *2017 9th International Conference on Advanced Computing, ICoAC 2017*, 102–108. https://doi.org/10.1109/ICoAC.2017.8441193

Kumar, S. (2019). *Getting started with Text Preprocessing*. Kaggle. https://www.kaggle.com/sudalairajkumar/getting-started-with-text-preprocessing/notebook

Kumar, S. (2021). *Fuzzy C-Means Clustering — Is it Better than K-Means Clustering?* Towards Data Science. https://towardsdatascience.com/fuzzy-c-means-clustering-is-it-better-than-k-means-clustering-448a0aba1ee7

Kumawat, D., & Jain, V. (2015). POS Tagging Approaches: A Comparison. *International Journal of Computer Applications*, *118*(6), 32–38. https://doi.org/10.5120/20752-3148

Lacroix, O. (2018). Investigating NP-Chunking with Universal Dependencies for English. *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, *UDW 2018*, 85–90. https://doi.org/10.18653/v1/W18-6010

Langr, J., & Ottinger, T. (2011). *FURPS+*. Agile in a Flash. https://agileinaflash.blogspot.com/2009/04/furps.html

Lavrakas, P. (2012). Encyclopedia of Survey Research Methods. In *Encyclopedia of Survey Research Methods*. SAGE Publications. https://doi.org/10.4135/9781412963947

Lawrence, J., & Reed, C. (2019). Argument mining: A survey. *Computational Linguistics*, *45*(4), 765–818. https://doi.org/10.1162/COLIa00364

Lee, H., Buckland, M., & Shepherdson, J. (2003). A multi-agent system to support location-based group decision making in mobile teams. In *BT Technology Journal* (Vol. 21, Issue 1, pp. 105–113). https://doi.org/10.1023/A:1022416613660

Lerner, J., Li, Y., Valdesolo, P., & Kassam, K. (2015). Emotion and decision making. *Annual Review of Psychology*, *66*, 799–823. https://doi.org/10.1146/annurev-psych-010213-115043

Leskovec, J., Rajaraman, A., & Ullman, J. (2014). Mining of Massive Datasets. In *Mining of Massive Datasets: Second Edition*. Cambridge University Press. https://doi.org/10.1017/CBO9781139924801

Lewicki, R., & Litterer, J. (1985). Negotiation. Homewood, IL: Richard D. Irwin. *Inc. LewickiNegotiation1985*.

Liao, W., Liu, Y., & Choudhary, A. (2004). A Grid-based Clustering Algorithm using Adaptive Mesh Refinement. *7th Workshop on Mining Scientific and Engineering Datasets of SIAM International Conference on Data Mining*, *3*(10), 61–69. http://users.eecs.northwestern.edu/~choudhar/Publications/LiaLiu04A.pdf

Liu, B., Zhou, Q., Ding, R., Ni, W., & Herrera, F. (2019). Defective alternatives detection-based multi-attribute intuitionistic fuzzy large-scale decision making model. *Knowledge-Based Systems*, *186*, 104962. https://doi.org/10.1016/j.knosys.2019.104962

Liu, Q., Wu, H., & Xu, Z. (2021). Consensus model based on probability K-means clustering algorithm for large scale group decision making. *International Journal of Machine Learning and Cybernetics*, *12*(6), 1609–1626. https://doi.org/10.1007/s13042-020-01258-5

Liu, W., Dong, Y., Chiclana, F., Cabrerizo, F., & Herrera-Viedma, E. (2017). Group decision-making based on heterogeneous preference relations with self-confidence. *Fuzzy Optimization and Decision Making*, *16*(4), 429–447. https://doi.org/10.1007/s10700-016-9254-8

Liu, Y., Whitfield, C., Zhang, T., Hauser, A., Reynolds, T., & Anwar, M. (2021). Monitoring COVID-19 pandemic through the lens of social media using natural language processing and machine learning. *Health Information Science and Systems*, *9*(1), 1–16. https://doi.org/10.1007/s13755-021-00158-4

Lyle, J. (2005). *Sports Coaching Concepts*. Sports Coaching Concepts. https://doi.org/10.4324/9780203994986

Made, G., Wimatsari, S., Gede, K., Putra, D., & Buana, P. (2013). Multi-Attribute Decision Making Scholarship Selection Using A Modified Fuzzy TOPSIS. In *International Journal of Computer Science Issues* (Vol. 10, Issue 1). www.IJCSI.org

Majumder, M. (2015). Multi Criteria Decision Making. In *Impact of Urbanization on Water Shortage in Face of Climatic Aberrations* (pp. 35–47). Springer, Singapore. https://doi.org/10.1007/978-981-4560-73-3_2

Marreiros, G. (2008). Agentes de Apoio à Argumentação e Decisão em Grupo [Universidade do Minho]. In *BUM - Teses de Doutoramento*. https://hdl.handle.net/1822/7643

Marreiros, G., Ramos, C., Machado, J., & Neves, J. (2009). Argument-based decision making in ambient intelligence environments. *International Journal of Reasoning-Based Intelligent Systems*, *1*(3/4), 182. https://doi.org/10.1504/IJRIS.2009.028017

Marreiros, G., Santos, R., Freitas, C., Ramos, C., Neves, J., & Bulas-Cruz, J. (2008). LAID - A smart decision room with ambient intelligence for group decision making and argumentation support considering emotional aspects. *International Journal of Smart Home*, *2*(2), 77–93.

Marreiros, G., Santos, R., Ramos, C., & Neves, J. (2010). Context-aware emotion-based model for group decision making. *IEEE Intelligent Systems*, *25*(2), 31–39. https://doi.org/10.1109/MIS.2010.46

Marreiros, G., Santos, R., Ramos, C., Neves, J., & Bulas-Cruz, J. (2008). ABS4GD: A multi-agent system that simulates group decision processes considering emotional and argumentative aspects. *AAAI Spring Symposium - Technical Report*, *SS-08-04*, 88–95.

Martin, M. (2021). *What is a Functional Requirement in Software Engineering? Specification, Types, Examples*. Guru99. https://www.guru99.com/functional-requirement-specification-example.html

Martinho, D., Carneiro, J., Marreiros, G., & Novais, P. (2017). Defining an architecture for a ubiquitous group decision support system. *Advances in Intelligent Systems and Computing*, *615*, 246–253. https://doi.org/10.1007/978-3-319-61118-1_30

Matsatsinis, N., & Tzoannopoulos, K.-D. (2008). Multiple criteria group decision support through the usage of argumentation-based multi-agent systems: an overview. *Operational Research*, *8*(2), 185–199. https://doi.org/10.1007/s12351-008-0008-4

Matteucci, M. (2022). *Clustering - Introduction*. Politecnico Di Milano. https://matteucci.faculty.polimi.it/Clustering/tutorial_html/index.html

McCormick, C., & Ryan, N. (2020, April 14). *BERT Word Embeddings Tutorial*. https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#why-bert-embeddings%0Ahttp://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/%0Ahttps://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/%0Ahttp://mccormickml.com/2019/05/1

McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, *3*(29), 861. https://doi.org/10.21105/JOSS.00861

Meta. (2017). *Graph API*. Https://Developers.Facebook.Com/Docs/Graph-Api/Reference/V2.8/Insights. https://developers.facebook.com/docs/graph-api/

Meta. (2021). *Facebook Meta*. Meta. https://about.facebook.com/meta/

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 1–9.

Miller, S. (1997). Implementing strategic decisions: Four key success factors. *Organization Studies*, *18*(4), 577–602. https://doi.org/10.1177/017084069701800402

Mohanty, S., & Bowne-Anderson, H. (2021). *Hand Labeling Considered Harmful*. O'Reilly. https://www.oreilly.com/radar/arguments-against-hand-labeling/

Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, *52*(1–2), 91–118. https://doi.org/10.1023/A:1023949509487

Moon, H., Conlon, D., Humphrey, S., Quigley, N., Devers, C., & Nowakowski, J. (2003). Group decision process and incrementalism in organizational decision making. *Organizational Behavior and Human Decision Processes*, *92*(1–2), 67–79. https://doi.org/10.1016/S0749-5978(03)00079-7

Morge, M., & Beaune, P. (2004). A Negotiation Support System based on a Multi-agent System specificity and preference relations on arguments. *Proceedings of the 2004 ACM Symposium on Applied Computing - SAC '04*. https://doi.org/10.1145/967900

Narayan, S., Gardent, C., Cohen, S. B., & Shimorina, A. (2017a). Split and rephrase. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 606–616. https://doi.org/10.18653/v1/d17-1064

Narayan, S., Gardent, C., Cohen, S., & Shimorina, A. (2017b). Split and rephrase. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 606–616. https://doi.org/10.18653/v1/d17-1064

Nguyen, H., Richards, R., Chan, C., & Liszka, K. (2016). RedTweet: recommendation engine for reddit. *Journal of Intelligent Information Systems*, *47*(2), 247–265. https://doi.org/10.1007/s10844-016-0410-y

Nicola, S., Ferreira, E., & Ferreira, J. (2012a). A novel framework for modeling value for the customer, an essay on negotiation. In *International Journal of Information Technology and Decision Making* (Vol. 11, Issue 3). https://doi.org/10.1142/S0219622012500162

Nicola, S., Ferreira, E., & Ferreira, J. (2012b). Conceptual Model for Decomposing the Value for the Customer. *3rd Industrial Engineering and Management Symposium*, 41–43.

Nithyashree, V. (2021). *What is Chunking in Natural Language processing?* AnalyticsVidhya. https://www.analyticsvidhya.com/blog/2021/10/what-is-chunking-in-natural-language-processing/

Ooms, J. (2010). *Introduction to Web Development with R moving to the cloud*. UseR! https://www.r-project.org/conferences/useR-2010/slides/Ooms.pdf

Oracle. (2021). *What Is Data Management*. https://www.oracle.com/database/what-is-data-management/

Orăsan, C. (2000). A hybrid method for clause splitting in unrestricted English texts. *Proceedings of ACIDCA '2000, Corpora and Natural Language Processing*, 129–134. https://dinel.org.uk/wp-content/uploads/2020/04/Orăsan-2000-A-hybrid-method-for-clause-splitting-in-unrestricted-English-texts.pdf

Osterwalder, A. (2004). *The business model ontology a proposition in a design science approach*.

Osterwalder, A., & Pigneur, Y. (2010). *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley. https://www.wiley.com/en-us/Business+Model+Generation:+A+Handbook+for+Visionaries,+Game+Changers,+and+Challengers-p-9780470876411

Osterwalder, A., Pigneur, Y., Bernarda, G., & Smith, A. (2014). *Value Proposition Design: How to Create Products and Services Customers Want*. Wiley (The Strategyzer Series). https://www.wiley.com/en-us/Value+Proposition+Design%3A+How+to+Create+Products+and+Services+Customers+Want-p-9781118968055

Palomares, I., Martínez, L., & Herrera, F. (2014). A consensus model to detect and manage noncooperative behaviors in large-scale group decision making. *IEEE Transactions on Fuzzy Systems*, *22*(3), 516–530. https://doi.org/10.1109/TFUZZ.2013.2262769

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 2227–2237. https://doi.org/10.18653/v1/n18-1202

Petrov, S., Das, D., & McDonald, R. (2012). A universal part-of-speech tagset. *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*, 2089–2096.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-smadi, M., Al-ayyoub, M., Qin, B., Clercq, O., Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., … Kotelnikov, E. (2016). SemEval-2016 Task 5: Aspect Based Sentiment Analysis. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 19–30. https://doi.org/http://dx.doi.org/10.18653/v1/S16-1002

Python Software Foundation. (2022). *What is Python? Executive Summary*. Python. https://www.python.org/doc/essays/blurb/

Qin, S., Zhou, L. S., & Yue, Y. X. (2010). A web-based distributed group decision support system for railway construction organization. *CAR 2010 - 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics*, *2*, 362–365. https://doi.org/10.1109/CAR.2010.5456524

Rahwan, I., & Simari, G. R. (2009). Argumentation in artificial intelligence. In I. Rahwan (Ed.), *Argumentation in Artificial Intelligence* (1st ed., Vol. 47). Springer. https://doi.org/10.1007/978-0-387-98197-0

Rajavela, R., Ravichandranb, S., Nagappana, P., & Gobichettipalayam, K. (2021). Cloud service negotiation framework for real-time E-commerce application using game theory decision system. *Journal of Intelligent and Fuzzy Systems*, *41*(5), 5617–5628. https://doi.org/10.3233/JIFS-189882

Ramasubramanian, C., & Ramya, R. (2013). Effective Pre-Processing Activities in Text Mining using Improved Porter's Stemming Algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, *2*(12), 4536–4538. www.ijarcce.com

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, *66*(336), 846–850. https://doi.org/10.1080/01621459.1971.10482356

Ray, B., Khatua, S., & Roy, S. (2014). Negotiation based service brokering using game theory. *Proceedings - International Conference on 2014 Applications and Innovations in Mobile Computing, AIMoC 2014*, 1–8. https://doi.org/10.1109/AIMOC.2014.6785511

Reddit. (2016). *API*. Reddit. https://www.reddit.com/wiki/api

Reddit. (2021). *Homepage*. Reddit. https://www.redditinc.com/

Rees, J., & Koehler, G. (2002). An Evolutionary Approach to Group Decision Making. *INFORMS Journal on Computing*, *14*(3), 278–292. https://doi.org/10.1287/ijoc.14.3.278.109

Regenwetter, M. (2009). Perspectives on Preference Aggregation. *Perspectives on Psychological Science*, *4*(4), 403–407. https://doi.org/10.1111/j.1745-6924.2009.01146.x

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3980–3990. https://doi.org/10.18653/v1/D19-1410

Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., & Gurevych, I. (2020). Classification and clustering of arguments with contextualized word embeddings. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 567–578. https://doi.org/10.18653/v1/p19-1054

Rosenberg, A., & Hirschberg, J. (2007). V-Measure: A conditional entropy-based external cluster evaluation measure. *EMNLP-CoNLL 2007 - Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, *June*, 410–420.

Rossi, S., Di Napoli, C., Barile, F., & Liguori, L. (2016a). A Multi-agent System for Group Decision

Support Based on Conflict Resolution Styles. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10238 LNCS*, 134–148. https://doi.org/10.1007/978-3-319-57285-7_9

Rossi, S., Di Napoli, C., Barile, F., & Liguori, L. (2016b). Conflict resolution profiles and agent negotiation for group recommendations. *CEUR Workshop Proceedings*, *1664*, 29–34.

Rothlauf, F. (2011). *Design of modern heuristics: principles and application*. Springer Science \& Business Media. https://doi.org/10.1007/978-3-540-72962-4

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*(C), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

RStudio/Shiny. (2021). *rstudio/shiny: Easy interactive web applications with R*. GitHub. https://github.com/rstudio/shiny

Saaty, T. (1988). What is the Analytic Hierarchy Process? *Mathematical Models for Decision Support*, 109–121. https://doi.org/10.1007/978-3-642-83555-1_5

Saaty, T. (1990). How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, *48*(1), 9–26. https://doi.org/10.1016/0377-2217(90)90057-I

Saaty, T. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, *1*(1), 83–98. https://doi.org/10.1504/IJSSci.2008.01759

SageMaker. (2019). *Automate Data Labeling - Amazon SageMaker*. Amazon AWS. https://docs.aws.amazon.com/sagemaker/latest/dg/sms-automated-labeling.html

Saltz, J. (2020). *CRISP-DM is Still the Most Popular Framework for Executing Data Science Projects*. Data Science Process Alliance. https://www.datascience-pm.com/crisp-dm-still-most-popular/

Santos, R., Marreiros, G., Ramos, C., Neves, J., & Bulas-Cruz, J. (2006). Multi-agent approach for ubiquitous group decision support involving emotions. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4159 LNCS*, 1174–1185. https://doi.org/10.1007/11833529_118

Santos, R., Marreiros, G., Ramos, C., Neves, J., & Bulas-Cruz, J. (2010). Using personality types to support argumentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6057 LNAI*, 292–304. https://doi.org/10.1007/978-3-642-12805-9_17

Santos, R., Marreiros, G., Ramos, C., Neves, J., & Bulas-Cruz, J. (2011). Personality, emotion, and mood in agent-based group decision making. *IEEE Intelligent Systems*, *26*(6), 58–66. https://doi.org/10.1109/MIS.2011.92

Saragih, H., Marbun, M., & Reza, B. (2014). Development of Decision Support System Determining the Student As Scholarship Recipients By Fuzzy Multi Attribute Decision Making (Fmadm). In *Jurnal Sistem Informasi* (Vol. 9, Issue 2). https://doi.org/10.21609/jsi.v9i2.354

Sarkar, D. (2018). *Understanding language syntax and structure: A practitioner's guide to NLP*.

KDnuggets. https://www.kdnuggets.com/2018/08/understanding-language-syntax-and-structure-practitioners-guide-nlp-3.html

Sarkar, S., Lodhi, V., & Maiti, J. (2018). Text-clustering based deep neural network for prediction of occupational accident risk: A case study. *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing, ISAI-NLP 2018 - Proceedings*. https://doi.org/10.1109/iSAI-NLP.2018.8692881

Schwarz, N. (2000). Emotion, cognition, and decision making. *Cognition and Emotion*, *14*(4), 433–440. https://doi.org/10.1080/026999300402745

SendPulse. (2021). *What is a Value Analysis: Definitive Guide - Definition | SendPulse*. https://sendpulse.com/support/glossary/value-analysis

Shafinah, K., Selamat, M., Abdullah, R., Muhamad, A., & Noor, A. (2010). System evaluation for a Decision Support System. *Information Technology Journal*, *9*(5), 889–898. https://doi.org/10.3923/itj.2010.889.898

Shanker, A. (2012). Q&A. What Is Customer Value and How Do You Deliver It? *Technology Innovation Management Review*, *2*(2), 32–33. https://doi.org/10.22215/timreview525

Sharma, R. (2020). *Cluster Analysis in Data Mining: Applications, Methods & Requirements*. UpGrad. https://www.upgrad.com/blog/cluster-analysis-data-mining/

Shiny. (2017). *Shiny*. RStudio. https://shiny.rstudio.com/

Siddiqui, A., Raza, S., & Tariq, Z. (2018). A web-based group decision support system for academic term preparation. *Decision Support Systems*, *114*(August), 1–17. https://doi.org/10.1016/j.dss.2018.08.005

Simon, H. (1986). Rationality in Psychology and Economics. *Journal of Business*, *59*(4), 209–224. https://doi.org/10.7551/mitpress/4711.003.0029

Sonagara, D., & Badheka, S. (2014). Comparison of Basic Clustering Algorithms. *International Journal of Computer Science and Mobile Computing*, *3*(10), 58–61. www.ijcsmc.com

SridHaran, M. (2018). *CRISP-DM: A framework for Data Mining & Analysis*. Think Insights. https://thinkinsights.net/digital/crisp-dm/#CRISPDM-Methodology

Stewart, G. (1993). On the early history of the singular value decomposition. *SIAM Review*, *35*(4), 551–566. https://doi.org/10.1137/1035134

Tanguiane, A. (2012). *Aggregation and Representation of Preferences: Introduction to Mathematical Theory of Democracy*. Springer Berlin Heidelberg. https://books.google.pt/books?id=E8bsCAAAQBAJ

Thatcher, A., & De la Cour, A. (2003). Small group decision-making in face-to-face and computer-mediated environments: The role of personality. *Behaviour and Information Technology*, *22*(3), 203–218. https://doi.org/10.1080/0144929031000117071

The CherryPy Team. (2016). *CherryPy — A Minimalist Python Web Framework*. CherryPy. https://cherrypy.org/

The R Foundation. (2021). *R: What is R?* R-Project. https://www.r-project.org/about.html

Theiler, S. (2019). *Basics of Using Pre-trained GloVe Vectors in Python*. Medium - Analytics Vidhya. https://medium.com/analytics-vidhya/basics-of-using-pre-trained-glove-vectors-in-python-d38905f356db#:~:text=Global Vectors for Word Representation,in a high-dimensional space.

Turban, E. (1995). *Decision Support and Expert Systems (4th Ed.): Management Support Systems*. Prentice-Hall, Inc.

Twitter. (2010). *Twitter API Wiki / Twitter API Documentation*. Twitter. http://apiwiki.twitter.com/w/page/22554679/Twitter-API-Documentation

Twitter. (2020). *New user FAQ*. Twitter. https://help.twitter.com/en/new-user-faq

Ulaga, W., & Eggert, A. (2006). Relationship value and relationship quality: Broadening the nomological network of business-to-business relationships. *European Journal of Marketing*, *40*(3–4), 311–327. https://doi.org/10.1108/03090560610648075

Urbanek, S. (2020). *Rserve - Binary R server*. RForge. http://www.rforge.net/Rserve/

Urumov, G. (2021). *The Solid Facts of Ground Truth Annotations*. UNDERSTAND.AI. https://understand.ai/blog/annotation/machine-learning/autonomous-driving/2021/05/31/ground-truth-annotations.html

US Bureau of Labor Statistics. (2019). *Psychologists*. Occupational Outlook Handbook. https://www.bls.gov/ooh/life-physical-and-social-science/hydrologists.htm%0Ahttps://www.bls.gov/ooh/life-physical-and-social-science/psychologists.htm%0Ahttps://www.bls.gov/ooh/life-physical-and-social-science/psychologists.htm#:~:targetText=Employment of

Van Der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

van Eemeren, F., Jackson, S., & Jacobs, S. (2015). Argumentation. In *Argumentation Library* (Vol. 27, pp. 3–25). Springer, Cham. https://doi.org/10.1007/978-3-319-20955-5_1

Villegas, W., Lujan, S., & Buenano, D. (2018). Towards the integration of business intelligence tools applied to educational data mining. *EDUNINE 2018 - 2nd IEEE World Engineering Education Conference: The Role of Professional Associations in Contemporaneous Engineer Careers, Proceedings*. https://doi.org/10.1109/EDUNINE.2018.8450954

Von Neumann, J., & Morgenstern, O. (1953). *Theory of games and economic behavior* (3rd ed.). Princeton university press.

Walton, D., & Krabbe, E. (1995). *Commitment in dialogue: Basic concepts of interpersonal reasoning*. SUNY press. https://books.google.pt/books?hl=pt-PT&lr=&id=QX93b7tufIEC&oi=fnd&pg=PR9&dq=walton+and+krabbe+&ots=FWwJbTgobN&sig=6FrZ-7sXeA3tX1PzkqimripOvpE&redir_esc=y#v=onepage&q=walton and krabbe&f=false

Weiss, S., Indurkhya, N., Zhang, T., & Damerau, F. (2004). Text mining: Predictive methods for analyzing unstructured information. *Text Mining: Predictive Methods for Analyzing*

Unstructured Information, 1–237. https://doi.org/10.1007/978-0-387-34555-0

Wicks, R., & Post, M. (2021). A unified approach to sentence segmentation of punctuated text in many languages. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3995–4007. https://doi.org/10.18653/v1/2021.acl-long.309

Wolverton, M. (2009). Research Design, Hypothesis Testing , and Sampling. *The Appraisal Journal*, 370–383.

Woodall, T. (2003). Conceptualising "value for the customer": an attributional, structural and dispositional analysis. *Academy of Marketing Science Review*, *2003*(12).

Wu, J. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, *11*(1), 95–103. https://doi.org/10.1214/aos/1176346060

Wu, T., & Liu, X. (2016). An interval type-2 fuzzy clustering solution for large-scale multiple-criteria group decision-making problems. *Knowledge-Based Systems*, *114*, 118–127. https://doi.org/10.1016/j.knosys.2016.10.004

Wu, Z., & Xu, J. (2018). A consensus model for large-scale group decision making with hesitant fuzzy information and changeable clusters. *Information Fusion*, *41*, 217–231. https://doi.org/10.1016/j.inffus.2017.09.011

XLeap. (2021). *XLeap Group Decision Support System*. XLeap. https://www.xleap.net/decision-support

Yan, P. (2021). *A Comprehensive Python Implementation of GloVe*. Towards Data Science. https://towardsdatascience.com/a-comprehensive-python-implementation-of-glove-c94257c2813d

Yıldırım, S. (2021). *Evaluation Metrics for Clustering Models*. Towards Data Science. https://towardsdatascience.com/evaluation-metrics-for-clustering-models-5dde821dd6cd

Zamfirescu, C.-B. (2003). An agent-oriented approach for supporting Self-facilitation for group decisions. *Studies in Informatics and Control*, *12*(2), 137–148.

Zanakis, S. H., Solomon, A., Wishart, N., & Dublish, S. (1998). Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research*, *107*(3), 507–529. https://doi.org/10.1016/S0377-2217(97)00147-1

Zeithaml, V. (1988). Consumer Perceptions of Price, Quality, and Value: A Means-End Model and Synthesis of Evidence. *Journal of Marketing*, *52*(3), 2–22.

Zhang, X.-D. (2020). Machine Learning. *A Matrix Algebra Approach to Artificial Intelligence*, 223–440. https://doi.org/10.1007/978-981-15-2770-8_6

Zheng, X., Martin, P., Powley, W., & Brohman, K. (2010). Applying bargaining game theory to Web services negotiation. *Proceedings - 2010 IEEE 7th International Conference on Services Computing, SCC 2010*, 218–225. https://doi.org/10.1109/SCC.2010.54

Zheng, Y. (2018). Identifying dominators and followers in group decision making based on the personality traits. *CEUR Workshop Proceedings*, *2068*.

Zhu, F., Guan, J., Wang, Y., Zhou, J., & Liao, B. (2005). An automatic negotiation method based on CBR and agent reasoning. *Proceedings - Fifth International Conference on Computer and Information Technology, CIT 2005*, *2005*, 1019–1023. https://doi.org/10.1109/CIT.2005.65

Zuheros, C., Martínez-Cámara, E., Herrera-Viedma, E., & Herrera, F. (2021). Sentiment Analysis based Multi-Person Multi-criteria Decision Making methodology using natural language processing and deep learning for smarter decision aid. Case study of restaurant choice using TripAdvisor reviews. *Information Fusion*, *68*(July 2020), 22–36. https://doi.org/10.1016/j.inffus.2020.10.019