De-duplication for product master data records using machine learning techniques

Elaborado Por: Julio Xavier Hallo Larrea

Nota de Aceptación

Certificamos que el presente Trabajo de Grado
Satisface, en alcances y calidad, todos los requisitos
Que demanda un Trabajo de Grado de Maestría.

Gloria Inés Álvarez Vargas Ph. D.
Directora

Hernán Camilo Rocha Niño Ph. D.
Jurado

María Constanza Pabón Burbano Ph. D.
Jurado

Aprobado en cumplimiento de los requisitos exigidos
por la Pontificia Universidad Javeriana Cali, para
optar el título de Magister en Ingeniería Industrial.

Hernán Camilo Rocha Niño Ph. D.
Decano Facultad de Ingeniería y Ciencias

Juan Carlos Martínez Arias
Director Posgrados de Ingeniería y Ciencias

Santiago de Cali, 1 de Septiembre de 2021

De-duplication for product master data records using machine learning techniques

By

Julio Xavier Hallo Larrea

A Thesis Presented as Fulfillment Requirement

for the Degree of

Master of Science of Engineering

Pontificia Universidad Javeriana Cali

Engineering Faculty

Postgraduate Department

July 2021

Santiago de Cali, 23 de Julio de 2021

Ingeniero:
Juan Carlos Martínez Arias
Director Posgrados de Ingeniería
Facultad de Ingeniería
Pontificia Universidad Javeriana Cali

Cumplido los requisitos establecidos en los artículos 5.6 y 5.7 de las Directrices para Trabajo de Grado de Maestría, solicitamos se autorice la sustentación del Trabajo de Grado denominado ***De-duplication for product master data records using machine learning techniques***", realizado por el estudiante ***Julio Xavier hallo Larrea*** con código ***201020022065*** perteneciente al énfasis en Ingeniería Industrial, bajo la dirección de la profesora ***Ing. Gloria Inés Álvarez Vargas PhD*** .

La suscrita directora del Trabajo de Grado autoriza para que se proceda a hacer su sustentación ante el Tribunal que para el efecto se designe, toda vez que ha revisado meticulosamente el documento y avala que el Trabajo de Grado ya se encuentra listo para ser evaluado oficialmente.

Atentamente,

Ing. Julio Xavier Hallo Larrea
C.C. 94.064.710 de Cali

Ing. Gloria Inés Álvarez Vargas PhD.
C.C. 30.306.105 de Manizales

Documentación anexa:
Copia digital del documento de Trabajo de Grado, con paginación completa.

Student Information

Full Name: Julio Xavier Hallo Larrea

Address: Avenida 5 Norte # 21N-35. Cali – Valle.

Mobile Telephone: +57 313 759 2455

Email Address: jxhallo@javerianacali.edu.co

Career: Industrial Engineer

University: Pontificia Universidad Javeriana Cali

Employer Company: Stibo Systems Colombia S.A.S

Job Title: Solution Consultancy Services Manager - LATAM

STIBO SYSTEMS
MASTER DATA MANAGEMENT

Stibo Systems A/S
Axel Kiers Vej 11
8270 Højbjerg
Denmark
Company no. 35822690

www.stibosystems.com

July 23, 2021

Engineer
Juan Carlos Martinez Arias
Post-graduate Program Director
Engineering and Science Faculty
Pontificia Universidad Javeriana – Cali

Re: Grant to Julio Xavier Hallo Larrea and the Pontificia Universidad Javeriana

Dear Juan Carlos Martinez Arias,

We at Stibo Systems are very pleased to support and provide aid towards the master thesis investigation project that our employee, Julio Xavier Hallo Larrea, is currently presenting as part of his master's degree in engineering. Furthermore, as the project objective is strictly tied with our core product offering, we are very interested in the result of this investigation. Therefore, we grant permission to Julio and the Pontificia Universidad Javeriana - Cali to use and publish Stibo Systems' public data according to the regulations enforced by law locally and internationally.

Kind regards,

DocuSigned by:
Ernesto Serrano
11B4757CB884496...
Ernesto Serrano
Executive Vice-President – Americas

DocuSigned by:
Eryn Carroll
92DD2866D8D1474...
Eryn Zaworski Carroll
Global Contracts/Legal Manager

STIBO ACCELERATOR

Aarhus, May 25th 2021

To:
Engineer, Juan Carlos Martinez Arias
Post-graduate Program Director,
Engineering and Science Faculty
Pontificia Universidad Javeriana – Cali

From:
Stibo Accelerator

Regarding Master Thesis, Julio Xavier Hallo Larrea

Stibo is passionate about what's around "the next corner", and we know that in the realm of tomorrow's technology, the rules are written as we go. Therefore, Stibo Accelerator was created in 2014 with a clear goal of investing in inviting in the best of the best students and nurturing them so they can focus on their projects in combining newest theory with practice and inspire the world. We provide 12-20 teams of students and startups every year with the best possible setup for exploring new trends and technologies in close relationship with the industry partners in our vast global network.

The Stibo Accelerator applies the unique strength of the Stibo network to bringing research and entrepreneurship closer to the industries we operate in. As of now the Stibo Accelerator have had 178 students from 16 different academic institutions involved in 82 innovation projects. Students and startups are some of the most innovative and out-of-the-box-thinking folks around, and we love to be inspired and surprised by the new ideas they bring out.

Stibo Accelerator have had the pleasure of working closely together with our Stibo Systems' colleague Julio Xavier Hallo Larrea regarding his promising Master Thesis investigation project. We are therefore more than happy to grant permission to both Julio and the Pontificia Universidad Javeriana - Cali to use and publish Stibo Accelerator's public data and logo according to the regulations enforced by locally and internationally law.

We in the Stibo Accelerator look very much forward to learn more about the investigation topics and the exiting findings and results Julio have learned with his Master Thesis. We would also like to offer our support for interesting future Master Thesis projects that have a potential within digitalization.

We are curious and we love to challenge conventional thinking. And we gladly share our projects and findings with everyone who shares our passion for innovation.

Kind Regards,

Karsten Dehler
Director, Stibo Accelerator

Stibo Accelerator | T. +45 89 39 88 72 | M. +45 22 80 77 55 | E. kade@stibo.com
Stibo Accelerator | Axel Kiers Vej 11 | DK-8270 Højbjerg | Denmark | www.stiboaccelerator.com

Pontificia Universidad JAVERIANA Cali

Maestría en Ingeniería

Facultad de Ingeniería y Ciencias

FICHA RESUMEN

ANTEPROYECTO DE TRABAJO DE GRADO DE MAESTRÍA

TITULO: ***"De-duplication for product master data records using machine learning techniques"***

1. ÉNFASIS: Ingeniería Industrial

2. ÁREA DE INVESTIGACIÓN: Aprendizaje Automático para Resolución de de-duplicación de entidades maestras

3. ESTUDIANTE: Julio Xavier Hallo Larrea

4. CORREO ELECTRÓNICO: jxhallo@javerianacali.edu.co

5. DIRECTOR: Ing. Gloria Inés Álvarez Vargas PhD.

6. CO-DIRECTOR(ES): N/A

7. GRUPO QUE LO AVALA: N/A

8. OTROS GRUPOS: N/A

9. PALABRAS CLAVE:

- Gestión de datos maestros
- Calidad de datos
- Resolución de de-duplicación de entidades
- Aprendizaje automático
- Redes neuronales profundas
- LSTM redes de memoria corto plazo a largo plazo
- Perceptrón Multicapa MLP

10. CÓDIGOS UNESCO CIENCIA Y TECNOLOGÍA: 1203.04, 1203.06

11. FECHA DE INICIO: 23 de Abril de 2020

12. DURACIÓN ESTIMADA: 5.5 Meses

Resumen

Con la transformación digital de las organizaciones, específicamente en grandes empresas como plataformas de comercio electrónico y marketplaces, los datos de productos han crecido exponencialmente para alcanzar los objetivos y necesidades comerciales. Para respaldar esto, tanto los profesionales como los académicos han reconocido la importancia de los datos maestros como recurso fundamental de la organización, y a su vez han identificado que la administración de datos maestros es un proceso independiente de la aplicación que lo describe, posee y administra. Con el fin de medir si este recurso es "apto para el uso", se han desarrollado metodologías, técnicas y artefactos de calidad de datos, definiendo los cuatro KPI clave: "completitud, exactitud, unicidad y oportunidad". Actualmente, las plataformas de software MDM proporcionan medios para lograr la medición y gestión correctas de los KPI descritos anteriormente. Por lo tanto, en el proceso de gestión, la interacción humana siempre es necesaria, específicamente cuando los algoritmos de de-duplicación actuales deben ajustarse en función de los datos etiquetados que muestran si dos o más entidades son o no duplicados. Esta investigación aborda este problema específico utilizando técnicas de aprendizaje automático, en las cuales diseñamos, construimos y probamos un modelo que de-duplica los registros de datos maestros de productos dentro de un corpus de datos de productos públicos.

Como resultado de la investigación, se han propuesto cinco (5) modelos de de-duplicación. Los modelos utilizan dos (2) tipos diferentes en arquitecturas de redes neuronales, Perceptrón Multicapa y LSTM, con dos (2) técnicas de pre-procesamiento de datos diferentes. Luego, todos los modelos han sido entrenados y probados utilizando los registros de pares de datos maestros de producto del corpus de datos seleccionado como parte de los objetivos de la investigación. Para evaluar el desempeño de cada modelo se han propuesto KPI's cuantitativos como F1 Score, entre otros, y KPI's cualitativos para clasificar la eficiencia de cada uno. Asimismo, se ha propuesto un árbol de decisión para seleccionar el modelo más adecuado según los objetivos de negocio y los recursos disponibles. Por último, se presentan las conclusiones y posible ampliación de la propuesta de investigación.

Abstract

With digital transformation of organizations, specifically in companies as large enterprises as eCommerce and marketplaces platforms, product data has grown exponentially in order to achieve the business goals and needs. To support this, both practitioners and academics have shed light on the importance of master data as an enterprise resource and master data management as an application-independent process which describes, owns and manages it. In order to measure its “fit for use”, data quality methodologies, technics and artifacts have been developed, defining the four key KPI’s: “accuracy, completeness, uniqueness and timeliness”. Currently, MDM software platforms provide means to achieve the correct measurement and management of the KPI’s described above. Thus, in the process human interaction is always necessary, specifically when current deduplication algorithms need to be adjusted and fine-tuned based on labeled data that shows if two or more entities are or are not duplicates. This investigation approaches this specific problem using machine learning techniques, in which we design, build and test a model that de-duplicates product master data records within a public product data corpus.

As result of the investigation, five (5) de-duplication models have been proposed. The models use two (2) different types on neural network architectures, Multilayer Perceptron and LSTM, with two (2) different data pre-processing techniques. Then all the models have been trained and tested using the data corpus product master data pair records selected as part of the investigation objectives. To evaluate each model performance quantitative KPI’s as F1 Score, among others, and qualitative KPI’s have been proposed to rank the efficiency of each one. Also, as decision tree to select the most suited model according to the business objectives and resources available has been proposed. Last, the conclusions and possible investigation proposal extension are presented.

Key Words

- Master Data Management
- Data Quality
- Entity deduplication resolution
- Machine Learning
- Deep Neural Networks
- Long Short Term Memory LSTM
- Multilayer Perceptron MLP

Dedication

To our Lord All Mighty, the principle of all things. To my parents Julio Hallo Granja and Miryam Larrea de Hallo that have always supported me to whom I owed everything. To my beloved wife Aura María Abadía. To the two engineers that changed my life forever, my grandfather Oswaldo Larrea†, and my uncle Raul Leon Palencia PhD.

Acknowledgments

I want to acknowledge and thank Ernesto Serrano, EVP Americas, German Escobar, PS Manager Director, and all Stibo Systems for supporting this investigation in addition to providing all the resources to accomplish this milestone. I also want to acknowledge and thank the Stibo Accelerator directives Kim Svendsen and Karsten Dehler for believing in the investigation project, for supporting it, and for providing the expert guidance needed to achieve it. In addition, I want to thank my advisor Engineer Gloria Inés Álvarez Vargas PhD. for her expert guidance, patience, and motivation to fulfill this project. Last but not least, I want to thank all the Pontificia Universidad Javeriana Cali, the Engineering Faculty, and more specially the Postgraduate Engineering Department, in head of Engineer Juan Carlos Martinez PhD. who provided all the comprehension and means to finalize this project and this chapter of my life.

Table of Contents

Table Index

Chart Index

Figure Index

INTRODUCTION

The global digital transformation, the forth revolution, has presented a broad amount of challenges both for academics and for practitioners. One of these is data. Today there is a huge need for all size companies and enterprises to manage and govern correctly its data, as per experts now it is considered “the new oil”, and it is starting to be addressed as a commodity. We have data everywhere, in social media, in web sites, in publicity, in traditional and non-traditional communication channels, and therefore various initiatives to harvest this “commodity” have upraised like AI, IoT, Chatbots, etc. In addition, governs have had the challenge to stablish regulations and laws to enforce the correct use of data, how it is presented, stored, used and even which are the individual’s rights upon their own data.

Product master data currently presents a huge challenge as consumers are now transitioning from a classic single channel approach to a omnichannel product master data ingestion like social media, web sites, eCommerce and marketplaces, just to point a few. This is due to the fact that now sourcing of products can be done from multiple places and vendors. Multiple academic studies like [1] and [2] have noticed this behavior where upon multiple web sites the information of the same product has been presented or on boarded slightly different, creating several variations of a “single version of the truth”. In addition the explosion of digital marketplaces and eCommerce platforms like amazon.com or e-bay.com, where multiple vendors can provide the exact same product and again each will have to on board it in the platform, creates duplicates of this product record. This from the consumer perspective provides an awful experience as there are multiple versions of a product creating confusion and at the end probably a low conversion rate. In order to minimize this issue and master correctly product master data, both software companies and academics have dedicated resources and efforts in order to provide product de-duplication mechanisms in order to create what is called a “Product Golden Record”, a single unified version of the product master data record.

In this investigation we address this specific problem. How can we create and train a mechanism to de-duplicate product master data records. Even though this problem has also been addressed by academics in the past, proposing excellent mechanisms and artifacts to actually achieve this goal using various methodologies independently, this investigation we will merge two approaches which will fill in the gaps of real world situations. This investigation addresses two common issues that have not been studied in conjunction in the past, but they reflect the most common real life scenario. The first one is the scarcity of homogenous data models where it becomes a challenge to define the product binary identification vectors, and the second one that product master data records often will just provide a “product description” or “product title” string as input for de-duplication. The novelty in this proposal is to address these two situations in conjunction, first applying a standardization method to extract key data from the product title, and second applying the result to actually de-duplicate products records. Both parts of the proposed solution use machine learning techniques. We have decided to use the “WDC Product Data Corpus and Gold Standard for Large-Scale Product Matching - Version 2.0” data set [3]. This data set has twenty six million product offer records from seventy nine websites, with four thousand four hundred pairs that were manually labeled as matches or non-matches. In the following sections you will find the details of the problem definition, the research objectives, the investigation boundaries and scope, the justification, followed by the proposed theorical framework, the methodology, the investigation experimental process development, the result analysis, and the research conclusions.

1 PROBLEM DEFINITION

1.1 PROBLEM STATEMENT

The eCommerce Observatory[1], an initiative funded and supported by MinTIC[2] CCCE[3] and RENATA[4], has identified the six key pillars for the eCommerce growth in Colombia [4]. These pillars are:

- Legal and Institutional
- Human Resources
- Supply
- Demand
- Logistics
- Technology

In addition, this investigation points out that all the breakthroughs in the Technology pillar rely in data, recognizing it as the central element for digital transformation and emerging technology adoption for eCommerce leverage. This supports that problems regarding data, i.e. master data, must be tackle in order to achieve the desired level of competence in terms of eCommerce for Colombia and the Latin American region.

The authors also point out the following macroeconomic facts and predictions which demonstrate that eCommerce is very important as a value chain in modern commerce. The first prediction is that eCommerce will reach USD $ 4.9 billions in 2021, with a growth of 256% since 2014, and a constant year by year 20.8% augmentation in contrast of a 3.5% GDP growth. In addition, the study states that the ratio of eCommerce growth versus GDP growth will be five to one (5:1) in the upcoming years.

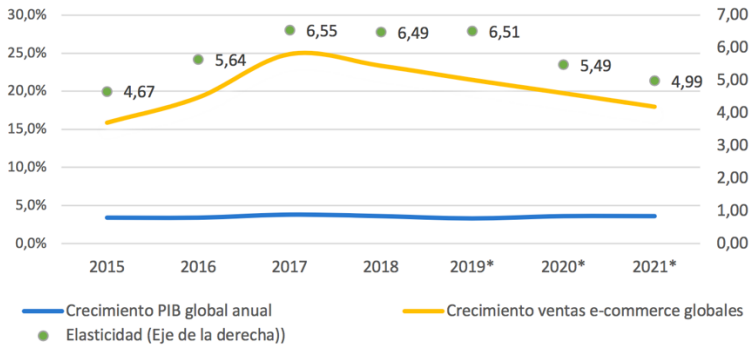


Figure 1- Ratio Global GDP and eCommerce. Source: [4]

Also [4] presented the same facts just for the Colombian market. As you can see from the figure bellow, in Colombia eCommerce grows in average 13 times more than the national GDP.

[1] Official source that characterizes and monitors the ecosystem of electronic commerce in Colombia. Founded in 2017, through the public-private strategic alliance between MinTIC, CCCE, and RENATA.
[2] Ministry of Information Technology and Communications (MinTIC).
[3] Colombian Chamber of Electronic Commerce (CCCE).
[4] National Academic Network of Advanced Technology (RENATA).

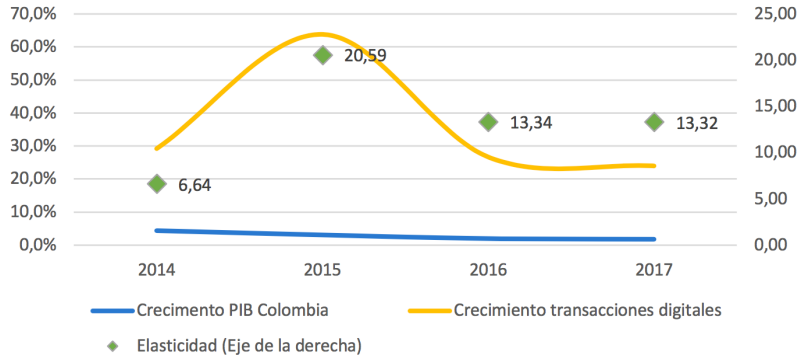


Figure 2- Ratio Colombian GDP and Colombian eCommerce. Source: [4]

The second macroeconomic fact and forecast analyzed by [4] is that the eCommerce value chain share keeps growing year after year and it will represent 15% of total retail sales in 2020 worldwide. For Colombia eCommerce grew with a ratio of five to one (5:1) at an average of 38.2% whereas retail commerce grew 7.8%. Moreover, as eCommerce is a retail channel, this behavior indicates that regular retail sales are experimenting a digital transformation and adoption.

From the above we can extrapolate that there is a real market need worldwide, in the Latin American region, and in Colombia, that needs attention in order to enhance the digital transformation technology pillar, supporting the eCommerce adoption.

As part of eCommerce exponential growth, a new business model has evolved simultaneously and rapidly as a variation of traditional eCommerce. The marketplaces. These multivendor platforms are able to provide to end users worldwide the same product from an abroad sources. The best example is amazon.com. Therefore we encounter a simple but worthily challenge that every multivendor eCommerce platform needs to address and it is how identify the same product source by different vendors and provide a single master data version of it to the end user. More details on the marketplace business model and how this challenge backs up this investigation will be addressed in section 4.3.1.

1.2 Problem Definition

How can we help de-duplicate product master data records, minimizing entity duplication resolution manual tasks, in order to provide an unique, complete, accurate and timeliness product master data record for eCommerce, marketplace and omnichannel applications?

1.3 Problem Systematization

- Which are the most common applications of machine learning techniques implemented to solve product master data record duplication for e-Commerce's and Marketplaces?
- Is there a labeled trained data set that can help build, train, and test a product master data record matching and de-duplication algorithm?
- Which machine learning technique could be applied to solve the investigation problem?
- How can machine learning techniques solve a product master data de-duplication problem in order to reduce mismatches and manual review tasks, specifically extracting "product characteristics" from product offer titles and descriptions?
- Which is the effectiveness of the proposed model?

2 RESEARCH OBJECTIVES

2.1 MAIN RESEARCH OBJECTIVE

Propose a model for product master data records de-duplication using machine learning techniques, in order to minimize manual entity duplication resolution tasks, learning from a subset of labeled matched data, measuring its effectiveness in order to provide a product master "golden record" for e-Commerce, marketplace, and omnichannel applications.

2.2 SPECIFIC OBJECTIVES

- Search, study, select and pre-process a public data set / data corpus in order to train and test the proposed solution using machine learning techniques.
- Select one machine learning technique as basis to propose a new model.
- Build a model of machine learning that will de-duplicate product records based on data extracted from the "product title" and "product significant characteristics", that will reduce de-duplication mismatches and manual review tasks.
- Evaluate the proposed model measuring the effectiveness against the labeled data set.

2.3 EXPECTED RESULTS

The following are the expected results and deliverables of the investigation:

- The data corpus to train, test and validate the proposed model. This data corpus will be preprocessed to in order to generate best results.
- The solution model based on machine learning.
- The evaluation results of the solution model applied the product a de-duplication problem.
- The investigation document that compiles all of the above deliverables and register the expected results.

3 RESEARCH BOUNDARIES AND SCOPE

This investigation will only address the duplicate entity resolution for product master data records based on a public data set provided by WDC Product Data Corpus and Gold Standard for Large-Scale Product Matching - Version 2.0[5]. It will propose an algorithm to extract "product titles" and "product characteristics" in order to feed the matching algorithm based on the findings of [5], [6], [2], and [1]. The de-duplication solution model proposal will applied to test implementation for Stibo Systems, but no real life validation will be expected within the scope of this investigation.

We will measure the results of the proposed artifact effectiveness using quantitative kpi's like *F_1-measure*[6]*, PC*[7]*, and PQ*[8] as done in previous researches.

In addition we will proffer new research topics based on the conclusions and findings of this investigation.

[5] http://webdatacommons.org/largescaleproductcorpus/v2/index.html
[6] Harmonic mean, or weighted average of the precision and recall scores [6] [5]
[7] Pair Completeness [6] [5]
[8] Pair Quality [6] [5]

4 RESEARCH JUSTIFICATION

4.1 BIBLIOGRAPHY ANALYSIS

For these investigation fifty eight (58) bibliographic resources have been studied. From these first set of sources, forty one (41) have been selected as part of the literature review included as foundation for the investigation. The figures bellow presents the distribution according to publication type and the publication year distribution.

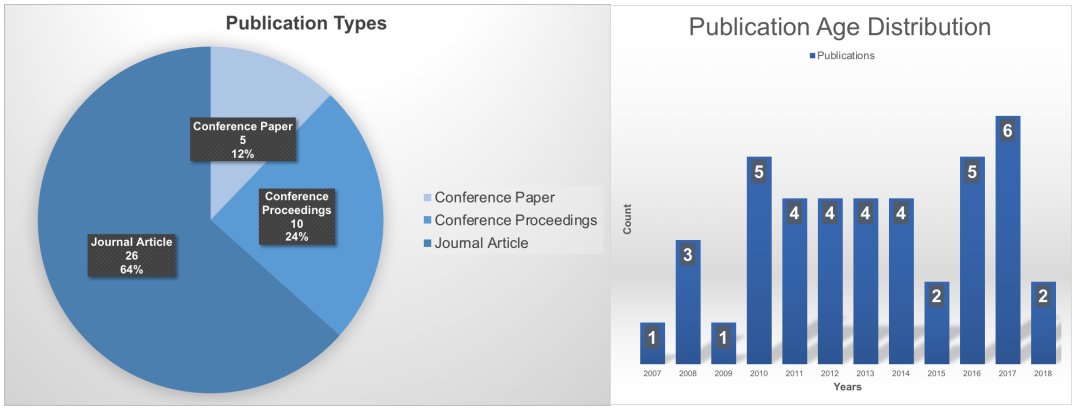


Figure 3- Bibliography Publication Types and Age Distribution. Source: Author.

Now in order to measure the literature review sources quality, the basic academic key performance indicators have been gathered from two different sources. The first one is "Web of Science – Journal Citation Reports" from which we will review the impact factor[9], the impact factor 5 years, the journal category quartile and the number of times the article has been cited. The second one is "Elsevier-SCOPUS" from which we will review the Field-Weighted Citation Impact[10], the citation benchmarking[11]. Taking into account that not all the bibliographic resources necessarily appear on both sources, the figures bellow present the detail of the average of these kpi's, but in summary we can conclude the following:

- Average IF 2018 = 3.39
- Average IF 5 Years = 3.79
- Average Cites in Web of Science = 30.82
- Average FWCI = 8.42
- Average cb = 70.55

[9] The **impact factor (IF)** is a measure of the frequency with which the average article in a journal has been cited in a particular year. It is used to measure the importance or rank of a journal by calculating the times it's articles are cited.
[10] The **Field-Weighted Citation Impact(FWCI)** is the ratio of the total citations actually received by the denominator's output, and the total citations that would be expected based on the average of the subject field.
[11] The **citation benchmarking (cb)** shows how citations received by this document compare with the average for similar documents.

- Average Cites in Scopus = 36.25

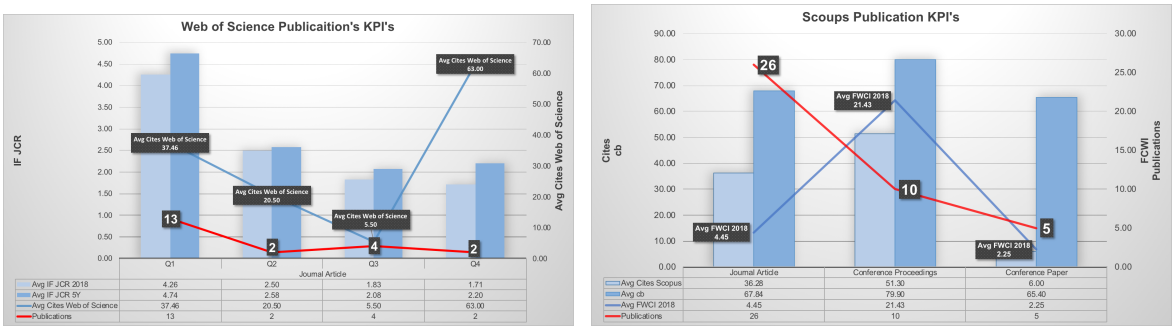


Figure 4- Bibliography KPI's. Source: Author.

From the above figures we can imply that the bibliographic resources are relevant, have good quality KPI's in order to support the investigation subject, and academics have researched on related and complementary topics related to this investigation.

4.2 JUSTIFICATION ON BIBLIOGRAPHY DATA SOURCES

One of the results presented by the authors [7] in this research, pointed out that over 80% of the relevant articles came from two sources. Gartner with 52.7% and Socups with 33.4%. As we can see, in this investigation we have focused or bibliography research in this two sources, in congruence with the findings and expert contribution made by academic researches previously. In addition, from the academic and scientific literature contribution point of view, this investigation aims to close the gap between industry investigation and publication of knowledge in the MDM field versus the one produced by scholars. From this literature review we can see that 54.18% of the investigation is conducted and published by Gartner, and on the other hand 42.07% are academic publications lead by conference proceeding publications with 23.05% and 19.02% with journal article publications. Due to this fact academia is highly recommended to increase its active participation in industrial practice through strategic collaboration or an industrial attachment [7].

4.3 ACADEMIC RESEARCHERS AROUND THE WORLD

As part of the result of the literature review, it was confirmed that there are other academics around the world investigating on master data management, master data quality, and it applications in the industry. In addition, we see that there is a great interest in practitioners to invest and promote formal academic investigation in order to address, study, and solve real life problems related to master data management. One of this examples is the Competence Center Corporate Data Quality (CC CDQ)[12] which is a research consortium and expert community in the field of data management addressing the challenges resulting from digitalization and data-driven strategies. Researchers come from leading academic institutions – among them the Faculty of Business and Economics (HEC – University of Lausanne), the Institute of Information Management (IWI – University of St. Gallen), and the Institute of Accounting, Control and Auditing (ACA – University of St. Gallen). The CC CDQ is headed by Prof. Dr. Christine Legner (HEC – University of Lausanne).

[12] Competence Center Corporate Data Quality (CC CDQ). https://www.cc-cdq.ch/

4.4 MARKET AND PRACTITIONER JUSTIFICATION

According to Gartner[13], a research and advisory firm, the Master Data Management infrastructure software market will grow in average from 2019 to 2022 by 4.81% achieving a revenue of US 1,847 million dollars worldwide. In addition to the above, Gartner points out that specifically for Latin America this market will grow in average 5.47% reaching a revenue of US 60 million dollars. Furthermore, if we narrow down these figures just for Colombia, Gartner forecasts a market growth in average of 5.62% reaching a revenue of US 3.2 million dollars in 2022[14] [8]. Therefore we see that there is market, worldwide, in our region and specifically in our country, where the need of master data management solutions that include software is rapidly growing. In other words also for practitioners there is a need to solve in terms of master data management, implying that the deduplication resolution and application of machine learning techniques to these issue is a current industry need.

In addition to the above Gartner[15], a research and advisory firm, points out that professionals need to focus on the foundational components needed to support artificial intelligence and machine learning in their enterprises [9]. Sapp points out that data management, in which master data management is included, is one of the foundational components needed to enable the use of AI/ML[16]. Without preparing theses foundations before implementing AI/ML solutions will not deliver the expected enterprise results as these rely heavily in data.

Last but not least Gartner[17], a research and advisory firm, also points out in its report "Six Pitfalls to Avoid When Planning Data Science and Machine Learning Projects" that a "pitfall" in building the correct use cases by linking business understandings and quality data to specific business needs is "Underestimating importance of data management" [10][18].

4.5 BACKGROUND EFFECTS ON THE MARKET

One of the biggest barriers found in eCommerce is the conversion rate. The CCCE through the eCommerce Observatory has identified that only 19% of internet users in Colombia between ages of 15 to 75 years perform an eCommerce transaction [11], even though 80% of the surveyed answered that they look up product and services details online. Specifically, the top 4 picks to look up for products and services information are digital channels with search engines at the top with 74% of favorability, followed by social media with 50%, then marketplaces 37%, and last retail websites/eCommerce's with 23%. This was a multiple choice question where surveyed could choose more than one option. In addition the survey presented that consumers visit in average two different digital channels, where they compare and or complement the product and service research in order to take a purchase decision.

From the above we can infer that master data duplication, as it has not been master in a timeliness, unique, accurate, and complete procedure across all channels is promoting consumers to abandon an eCommerce purchase. Also, we see that there is a need to have a unique "version of the truth" in terms of master data that is persistent across all channels. Furthermore, we can conclude that as marketplaces[19] are a big source of consultation source, and by definition marketplaces can offer the same product or service from multiple third parties, there is a tremendous need to minimize possible

[13] GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally, and is used herein with permission. All rights reserved.
[14] Forecast: Infrastructure Software Markets, Worldwide, 2017-2023, 1Q19 Update
[15] GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally, and is used herein with permission. All rights reserved.
[16] Laying the Foundation for Artificial Intelligence and Machine Learning: A Gartner Trend Insight Report (ID:G00373110)
[17] GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally, and is used herein with permission. All rights reserved.
[18] Six Pitfalls to Avoid When Planning Data Science and Machine Learning Projects (ID:G00325649)
[19] Online Marketplace (or online e-commerce marketplace) is a type of e-commerce site where product or service information is provided by multiple third parties, whereas transactions are processed by the marketplace operator. Online marketplaces are the primary type of multichannel ecommerce and can be a way to streamline the production process.

duplicate master data records in order to provide truthfully information to consumers. This problem has also been identified and addressed by [6] and [5] where they point out the fact there are multiple web sites that present the same product or service information but with differences in their content, causing consumers inability to make proper comparisons and take decisions. This supports the survey result made by the eCommerce Observatory where Colombian consumers search and compare products and services in two digital channels in average [11]. Also, in section 03 of the "BlackIndex: eCommerce Report in Colombia" [12] the authors conclude that the first step for retail purchase is the on line research.

Additionally, there is a need to ensure that the product and service information across all sales channels, including digital ones, is accurate. From the consumer statute law 1480 of 2011, one of the established causes for an eCommerce transaction reversal is "when the requested product or service does not correspond to the one solicited, or it does not meet the inherent characteristics or the ones attributed to it in the supplied information" [4]. According to the investigation presented by the eCommerce Observatory "Prospective for eCommerce in Colombia", this issue represents the 29% of total issues reported by consumers when they purchased products or services online.

4.6 MARKET POSSIBLE CAUSES

In terms of the possible causes related to the problem intended to solve in this investigation, the following segmentation can be done in coherence with the investigation done by the eCommerce Observatory "Prospective for eCommerce in Colombia".

4.6.1 DEMAND

The three main causes are (1)Low access, connectivity and digital culture, (2) Limitations in financial inclusion and (3) Deficient shopping experience [4]. In the third topic we find that one probable source is the absence of digital omnichannel experience presenting a "single version of the truth" in terms of master data for product and services. This is one of the probable causes by which just the 6.76% of the population in Colombia performs an eCommerce transaction, reducing the level of confidence of consumers.

4.6.2 SUPPLY

Linked to the above affirmation of deficient shopping experience, the suppliers of digital platforms need to work on facilitating the process of search, validation, purchase and order tracking [4]. Therefore the authors propose that the absence of visual/digital content, poor detail product and services information (i.e. product and services specs) and lack of comparison tools between products and services are causes of low eCommerce adoption. These probable causes can be linked to the low data quality KPI's of master data, specifically because there is no unique version of the record across all consumer's touch points.

4.6.3 MARKET FORECAST

The CCCE has stablished three main challenges to achieve by 2025. One of these is "Expand the penetration of supply and demand of eCommerce in Colombia" [4]. In the investigation approach they propose three scenarios in which the most probable and the optimistic imply a significant growth in eCommerce transactions. For the first scenario this KPI needs to duplicate its current level reaching 38%, and the optimistic scenario places this KPI at 80%. In addition, the investigation proposes that the number of companies that have an eCommerce platform must be over 32% and at least 50%, respectively for each scenario.

Another challenge to tackle by 2025 is to "Increase the value and number of digital transactions in Colombia" [4].Where in the most probable scenario the number of eCommerce transactions need to grow 25% year by year reaching at least the 290.4 millions, and in the optimistic scenario both KPI's need to be above the most probable ones.

Therefore, the confidence of the supply and demand of eCommerce, omnichannel, needs to rely on products and services master data records which are “fit for use”.

4.7 SYNTHESIS ON INVESTIGATION JUSTIFICATION

From the above sections we can conclude that the problem addressed in this investigation has a broad amount of high quality scientific literature as a starting point. Also, other academics around the world are investigating about this topic and have shown interest in the results of the project. In addition, we observe from market analysts experts that there is a need to solve this issues for practitioners and that there is a growing profitable market willing to spend resources in its applications worldwide, regionally, and locally. Furthermore, we can infer that if the problem in this investigation is not addressed, the market forecast opportunities for the region will not be used as advantages in order to develop the new growing e-Commerce / Marketplace channels for retail. And last but not least, we can see a gap between formal academic investigation and industry practitioner research that needs to be reduced by addressing real world issues with scientific research in order to bring these two knowledge pillars close together.

5 THEORICAL FRAMEWORK

5.1 BACKGROUND

With digital transformation of organizations, specifically in companies as large enterprises, data has grown exponentially in order to achieve the business goals and needs. Therefore since late 1980’s researches and practitioners have identified data as a key resource asset that needs to be assess and managed according to business objectives [13]. Subsequently, data as an asset has been compared and valued with other business resources as financial resources, human resources, and technological resources, among others [14]. In addition, the quality of this resource has also been a field of investigation, in order to establish how valuable it is and in which degree. Even though extensive investigation has failed to determine the exact connection of value data and its quality [13] due to the fact that existing literature examines data and its quality just in some of its data lifecycle stages [15], [16]; practitioners and academic researchers have agree that data is an important asset in the degree it can contribute to provide value. This investigation relays in the principle that data is only a valuable asset when it is recognized as the raw material that can produce information [13] after a business process[17]. An example of this understanding approach is an invoice. An invoice represents a transaction between a seller and a buyer, for which data from various domains like inventory quantity, lot number and expiration dates, product unique identifier, and customer billing and delivery records, are used to define the transaction goal and value. Therefore, these pieces of data have now produce adequate information and value for instance. Now that the understanding of how data produces value to organizations and its been recognized as a resource, it becomes necessary to manage its throughput level on which it can produce information. To achieve this, total quality management approach applied to this resource has coined the term ***“Total Data Quality Management” which is defined as “the application of total quality management (TQM) concepts and principles to improve data and information quality, including setting data quality policies and guidelines, data quality measurement (including data quality auditing and certification), data quality analysis, data cleansing and correction, data quality process improvement, and data quality education*** [18]***.”***

5.2 DATA RESOURCE

Now, to understand data as a resource, data by itself needs to be defined. Logical views represent data segments as the building blocks in which at the lowest level of aggregation there are data items which are instantiation of attributes of that entities, like the GTIN number in a product record. Then the union of data items represent a data record, for instance a product master data record. Then, the aggregation of data records form data base tables or files and these are then aggregated in data bases. For example an inventory data base can be made up of the product master data base table and the lot number and quantity data base table. At the top level an enterprise collection of data bases represent their data resource [14], [19], [20]. Figure 5 presents an illustration of this data resource definition as an subsequent aggregation of building blocks.

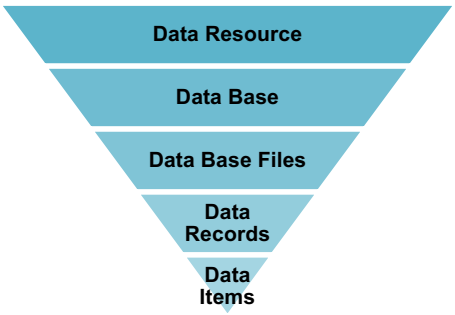


Figure 5- Data resource building blocks. Source: Author.

Accordingly scholars have define three pillars that compose data resource, which need to be understood in order to achieve value generation. These pillars are data lifecycle, data quality and data value [13]. Figure 6 presented bellow illustrates this definition. In the first pillar, data lifecycle, presents four stages that data requires to accomplish in order to produce information. The first one is data procurement, process by which data is acquired, defines the sources and methodologies to acquire it, both internally or externally. The second one is data storage and maintenance, which includes the actual storage in a system of record and maintenance tasks as editing a particular record like a product unique identifier. The third one is the process of information production, like explained above, it the usage of data to provide value to the organization. The fourth stage comprehends the deactivation and disposal of the data. The second pillar relates to data quality measurements and data quality assessment approaches [21] which can be described by the measurement points during the data lifecycle, the frequency of measurements, and the responsibility of taken and comparing those measurements to reference values. Last but not least, the third pillar encompasses the definition of how to determine data value and determine the strategy around how data produces value to the enterprise. This investigation agrees with Otto's affirmation that data resource produces value when it generates information, then data has value by itself [13]; but in addition this investigation implies that the value of data has a direct relation with its quality, that can be measured directly with data quality measurements.

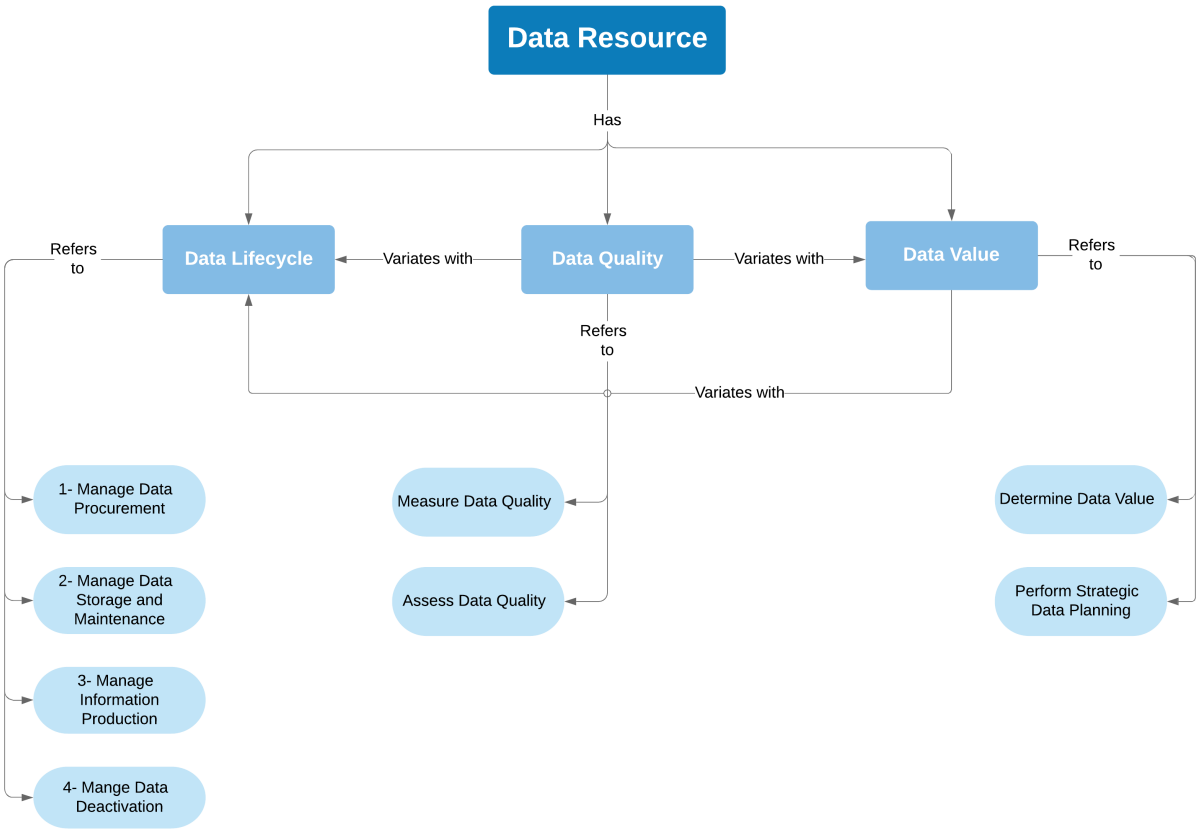


Figure 6- Lifecyle, quality, and value of the data resource. Source:[13]

5.3 Data Governance

Understanding the importance of data as a resource, how to manage it has become a strategic concern both for scholars and enterprises. To achieve this, data governance concept has evolved to orchestrate this resource. Several authors, which this investigation agrees with, define data governance as the strategic framework that determines the decision making privileges with the regards of use and management of data, and the goal of maximizing its value [13], [22], [23].

5.4 Master Data Governance

This investigation will focus its attention on master data, defined as key business objects used across all the enterprise which need to be defined uniquely, accurately, timeliness and completely [24] like suppliers, products, customers, locations or employees [25], [26]. As data domain, master data has also been addressed form the data governance dimension. Expanding the above affirmation about data governance, data governance for master data determines and assigns the wide-enterprise responsibility on data quality management [27] for this data domain. In detail, this investigation aims to contribute on how to deliver more value to organizations by increasing their data quality dimensions as part of their data resource management and governance. In addition, in 2006 a survey conducted on more than 350 organizations in North America that had implemented analytics and business intelligence systems, pointed out that one of the five success “practices” for delivering business value from data was a program of data governance [23]. It is also very important to distinguish the difference between governance and management. According to Weill and Ross [28], governance refers to what

decisions must be made to ensure the effective management and use of decision domains and who makes the decisions; whereas management involves making and implementing these decisions. This will allow us to draw a clear line between master data governance and master data management in which in alignment with the above this investigation propose the following:
“Master data governance refers to the framework on which the principles and objectives of how to manage master data as an asset in order to produce business value needs to be acquired, stored, managed to produce value, and dispose; ensuring data quality standards across the four data lifecycle stages presented above.”
“Master data management refers to the actual decision making and implementation of techniques, technology aids, and business process that will provide business value transforming data as a resource to business valuable information, ensuring the compliance of the master data governance framework.”
Khatri and Brown define a data governance framework in which five decision domains are clearly defined [23]. These are:

- Data Principles: Clarifying the role of data as an asset.
- Data Quality: Establishing the requirements of intended use of data. Specifically defining the standards with respect to accuracy, timeliness, completeness and uniqueness.
- Metadata: Establishing the semantics or “content” of data so that it is interpretable by the users.
- Data Access: Specifying access requirements of data.
- Data Lifecycle: Determining the definition, production, retention and retirement of data.

5.5 DATA QUALITY

Data quality is the measurement by which a data governance policy determines if a data record has “fitness for use” and therefore produces value when used to generate information [22][23]. If there is low data quality, data assets decrease their intrinsic value and the ability to produce information, in conclusion their utility for the enterprise is low [16]. As enterprises seek data quality maximization, DAMA International defines data quality management as a function for measuring, evaluating, improving, and ensuring data’s “fitness for use” [29]. Nowadays, in cohesive supply chains business models also known as “Business Networking Systems” relay on high quality data sharing and exchange, including master data [30]. Also, scholars have identified the need to assess master data quality due to growing regulatory and legal provisions companies need to comply with; in addition to the increased importance and use of information systems that support decision-making which require high quality master data as input [25]. Master data quality management also aims at carrying out preventive initiatives during data procurement lifecycle stage in order to be able to ensure its “fitness of use” [25].
Several researchers have identified four mail data quality dimensions, that also apply to master data, as it can be considered a subset of data within organizations. These data quality dimensions respond to the need of establishing metrics to determine the “fit for use” of data in order to produce information. Bellow the four dimensions are defined [23]:

- Accuracy: Refers to the correctness of data. This means. If the recorded value is in conformity to the actual value in respect to the intended use.
- Timeliness: Indicates if the recorded value is up-to-date for the task and in hand.
- Completeness: Suggests that the minimum required values are recorded and that that minimum requirement is of adequate depth/breadth.
- Uniqueness/Credibility: Indicates the trustworthiness of the source as its content.

5.6 MASTER DATA

Master data specifies the essential business entities on which business activities are based on. For example customers, suppliers, products, employees or assets [26]. Master data can be divided in three general concepts master data class, master data attribute, and master data object [25]. The

first one is a master data object or entity which represents a business object like a product that has certain attributions like color, length, width, height. These selected set of attributes for a master data object is called a master data class commonly defined in a data model specification. Therefore, master data management comprises all the related activities for creating, modifying or deleting a master data class, master data attribute or master data object entity. Vilminko-Heikkinen and Pekkola define master data as the key business objects in a company that are unambiguously defined and uniquely identified across the organization. [31].

5.7 MASTER DATA MANAGEMENT

Master data management is defined by Otto and Reichert [32] as an application-independent process which describes, owns and manages core business data entities. These master data entities are typically used across multiple business process, and it is often stored in and/or used by multiple application systems. An example presented by the former authors is the supplier master record, which is use by procurement and by accounts payables departments. In order to distinguishing master data from other data, these criteria can be examined to draw the delimiting line between these data sets.

- Time reference
- Modification frequency
- Volume stability
- Existential independence

In addition, master data management must tackle data issues by concentrating on the business processes, the data quality and the standardization and integration of master data in information systems used across all the organization [31]. Researches have stated that master data can be identified using simple criteria as reuse, stability and complexity and that by itself if not used to produce information it has little value to the organization [31]. But when used to produce information, consumed by other organization applications or systems, produces value to the company as it has effect on transaction data which describes relevant events in a company. If master data is has not “fit for use”, transactions that use these data will be of no value to the organization. An example is a customer shipping address that has an error, which will result in returns in a shipment that cannot be delivered. This kind of issues arose with the exponential growth of master data in companies, often stored in different information systems creating silos of information, lacking a single “version of the truth”. Both practitioners and researchers agree that Master Data Management (MDM) must be treated both in an organizational aspect as in a technical aspect [31]. Vilminko-Heikkinen and Pekkola identified the following steps in order to stablish a proper MDM function in a company:

1- Identifying the need and objectives
2- Identifying the organization’s core data and process that use it
3- Defining the governance
4- Defining maintenance process
5- Defining data standards
6- Metrics for MDM
7- Planning a MDM architecture
8- Planning training and communication
9- Forming a roadmap for MDM development
10- Defining MDM applications’ functional and operation characteristics

The biggest contribution of the article sited above is that MDM development is not just about the technology or data. Also the organization, its objectives and strategies must be considered in order to produce proper value.

According to Haneem, Ali, Kama & Basri [7] MDM is not just about technology, it is an approach through a combination of business processes, data governance policies, and technical implementation to resolve data quality issue in multiple scattered sources that lead to duplication, inaccuracy and inconsistency of master data.

5.8 WHY IS MASTER DATA MANAGEMENT AND GOVERNANCE IMPORTANT?

This investigation has pointed out the importance of data as an enterprise resource, how it produces value when used to produced information, how to measure its "fitness of use", and how to govern it and manage it. But we have not demonstrate it with a practical use case. Therefore, this investigation has payed attention to the new trends in organizations where data is key for decision making. Some of these new technologies and process like business intelligence, analytics, big data, IoT's, AI, among others, are being implemented in order to provide means for strategic decision making and competitive advantages. These platforms relay heavily on data to produce information. This fact has driven attention on researchers and practitioners, where they have asked a simple question "What are the key success factors that we need to address in order to implement the above technologies adequately?". That is why for instance Yeoh and Koronios [33] researched and provide a framework of critical success factors (CSF's) to seek in order to implement a business intelligence platform. They have depicted, among others, that poor data quality derived from source systems is one of the challenges that a BI[20] has to overcome in order to achieve a successful implementation. Moreover, they state that the identification of poor data quality materializes just when cross systems data analysis is conducted. An example of this is when you merge two source system records to find create a single enriched master data record, like a customer, but you cannot corelate both source systems because there is no key to identify them across the two systems. In this research, the authors ping point the need to have a cross-system analysis to help profile an uniform master data set that complies with business rules and standards, hence data quality standards. As a conclusion, from both stages of their investigation Yeoh and Koronios [33] present "Sustainable data quality and integrity" as a CSF for a BI platform implementation.

5.9 MACHINE LEARNING TECHNIQUES

Machine learning techniques allow computers to act without being explicitly programmed, constructing algorithms that can learn from data and make data driven decisions [34]. One of this machine learning techniques that has stand out rapidly in the recent scientific literature is deep learning. This technique consists of a hierarchical architecture with many layers of none linear information processing unit [34]. In their paper, the authors have pointed out that four types of deep learning architectures are nowadays investigated and used in computer vision, pattern recognition and speech recognition [34]. These deep learning architecture are the Boltzmann machine, the deep belief networks, the autoencoder and the convolutional neural network. One of their biggest contribution to the scientific literature in this research is that based on deep learning techniques, unsupervised learning algorithms can now be used to process unlabeled data. In addition, the trade-off between computational complexity and accuracy can be adjusted with flexibility in most cases. Goodfellow et al. in its publication "Deep Learning" [35] present several application areas that are of great interest for this investigation, specifically Language tasks which require modeling a large number of possible values, like words in vocabulary. Specifically in our approach where we want to provide as inputs the product description and specific product characteristics in tuples, to decide if they are duplicates or not, we see that this is a great approach in order to propose a model to solve the investigation problem.

As part of machine learning techniques, deep learning have driven increasing interest in the academic research field. As explained by [34] because of their capability to overcome the dependency of hand-design features. In this research four deep learning architectures are studied in order to provide an up-to-date overview, and their current applications.

[20] Business Intelligence Platform.

Restricted Boltzman Machine, also known as RBM, has become prominent since the publication of Hinton in 2006 [36], as they have been used to generate stochastic models of artificial neural networks that can learn the probability distribution with respect to its inputs. This architecture is recognized as a special type of Markov random fields with stochastic visible units in one layer and stochastic observable units in the other layer.

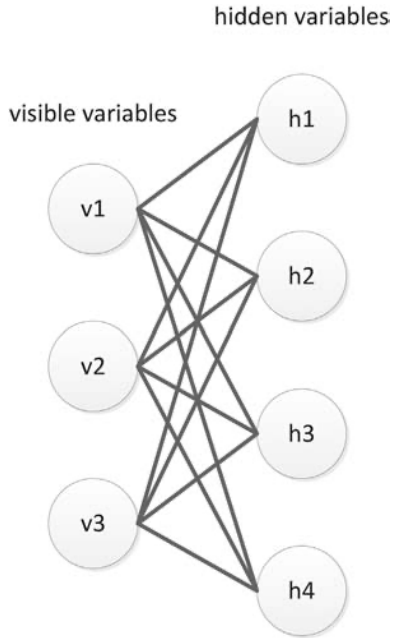


Figure 7- Schematic diagram of RBMs. Source:[34]

Deep Belief Networks (DBN) are an evolution proposed by Hinton in [36] of RBMs, where several RBMs are stacked together. This allows each visible layer of RBM to be linked to the hidden layer of the next RBM, taking into account that the two top layers are non-directional. Some of the applications of this architecture, where it has outstand among others, are image classification. In this specific application it outperforms others due to feature learning, based on a greedy layer-by-layer unsupervised training algorithm. Also, Liao et al. in [37] proposed novel image retrieval method which uses DBN and a Softmax Classifier. This algorithm is employed to retrieve images from a data base, presenting higher precision and better recall than shape-based algorithm and the perceptual hash algorithm.

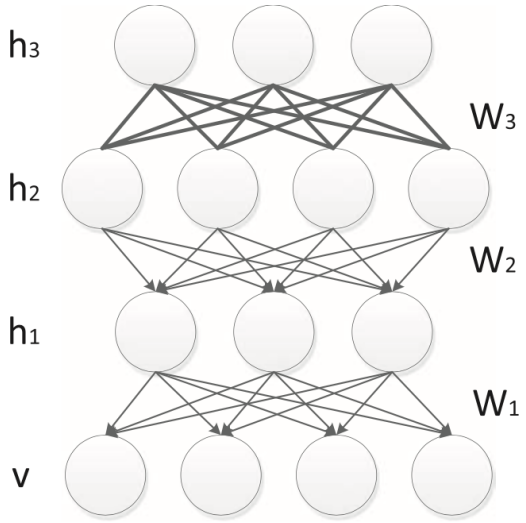


Figure 8- Schematic diagram of DBNs. Source:[34]

The autoencoder (AE) is an unsupervised learning algorithm used to efficiently code the dataset for the purpose of dimensionality reduction [34]. This architecture has been used lately to learn generative models of data, where it essentially tries to approximate the identity function in this process. A key advantage is that the model can extract useful features continuously during the propagation and filter useless information. One of the most novel applications of a variation of an (AE) is the total reconstruction error of the noisy speech spectrum, where this is minimized by adjusting the unknown parameters and the estimation of the clean speech spectrum.

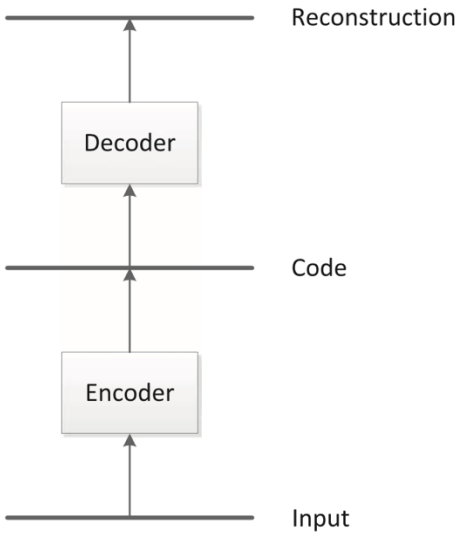


Figure 9- Schematic diagram of AEs. Source:[34]

Another very important architecture for deep learning is the Deep Convolutional Neural Network (CNN). This architecture have shown satisfactory performance in processing two dimensional data in grid-like topology like videos and images [34]. CNN have been tremendously successful due to the fact that one of its properties, parameter sharing allows to lower the number of unique model parameters and to significantly increase network sizes without requiring a corresponding increase in training data. Goodfellow et al. in [35] state that it is one of the greatest examples of how to incorporate domain knowledge into the network architecture.

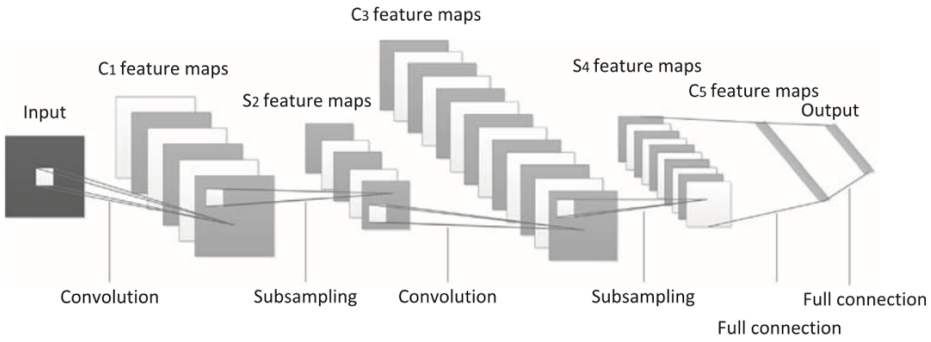


Figure 10- Schematic diagram of CNNs. Source:[34]

Long-short term memory (LSTM) is one type of artificial recurrent neural network (RNN). The main difference with standard feed-forward neural networks is that LSTM units have feedback connections[38]. The figured presented below illustrates a block diagram of a LSTM recurrent network "cell". This approach presented by Ian Goodfellow et. al. in [35] divides the block into input unit, state unit and output unit. The input feature is computed by a regular artificial neuron unit. Then its value can be into a state if the input gate controlled by a sigmoidal function allows it. After, the state unite has a self-loop which is also controlled by a gate called the forget gate. Last, the output gate is also controlled by a nonlinear sigmoidal function, that can activate or inactivate the output unit. Do to the fact that LSTM RNN can store input data across time, they are very precise in predicting what :comes next". This is very much well suited for applications that work with sequential data as speech recognition, time series, weather, etc. LSTM have a proven solution to two of RNN's major challenges, exploding gradients, and vanishing gradients. Exploding gradients are when the algorithm assigns high importance to weights. Vanishing gradients happen when the gradient values are too small causing the model to stop learning, or to take too much time to achieve a result. In short, LSTM units have the ability to remember data from inputs among long periods of time.

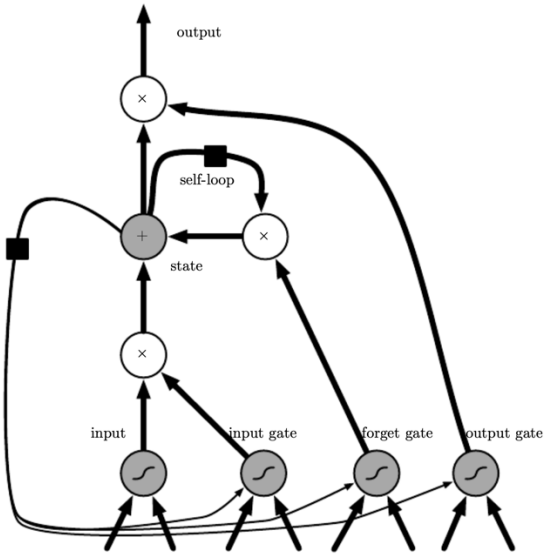


Figure 11- Block Diagram of LSTM "Cell". Source:[35]

Bidirectional LSTM neural networks (BiLSTM) are a set of two hidden traditional Long Short Term Memory neural network layers stacked together in order to process the input timesteps in to directions. The first layer will process the sequence as-is (forward), and the second one will process it on a reverse copy (backward). This architecture applies when all the timesteps of the input sequence are available. The concept of bidirectional recurrent neural networks (BRNN) was introduced by Schuster and Paliwal [39], where they stated that proposed method gives better results in classification and regression experiments. Comparing eight different neural networks on the TIMIT Phoneme Classification, the BRNN gave the best results both in training and test data sets. In conclusion, Schuster and Paliwal [39] where able to train a neural network in both directions simultaneously, without worrying about merging the outputs of two separate networks as both concentrate in minimizing the objective function for both time directions at the same time. Therefore, the BRNN can provide faster development of real applications with better results, as it does not need to search for the optimal delay parameter.

Then in 2005, Graves, Fernandez and Schmidhuber, addressed the same problem but using Bidirectional Long Short Term Memory (BiLSTM) networks [40]. In the experiments they proof that the BiLSTM outperforms single layer/direction LSTM networks for phoneme classification. In addition, in [41] an application of Deep Bidirectional Long Short Term Memory (DBiLSTM) network showed that it outperformed the other deep neural networks on the TIMIT classification problem. Therefore, this architecture is a fitted candidate to evaluate in this investigation.

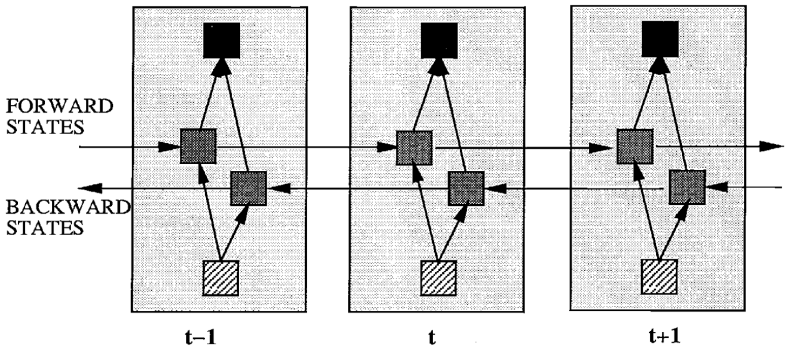


Figure 12- General Structure of BRNN. Source: [39]

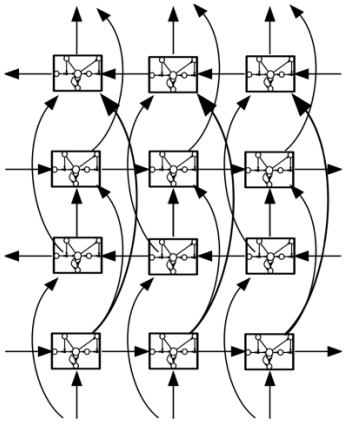


Figure 13- Deep Bidirectional Long Short-Term Memory (DBiLSTM) network. Source: [41]

5.10 Applications for Product Deduplication Resolution

From rapid and constant e-Commerce and marketplace applications for retail has grown tremendously. We can typically observe that their offer's catalogs are comprised of specifications of millions of products [42], and these must be matched the right product master data record. This task is time and resource consuming due to several reasons like unique identifiers are "not that unique" or are absent in the offer's data. Also, many offers are described using free text attributes like the "description", "title", or "name", in which product characteristics are embedded with their corresponding names, or simply none of these values are present in the free text, and there is no standard to name an offer which results in presence of words that are not related to the product characteristics or specifications [42]. Also, when several unmatched product records, including its offers, are found; there is a huge challenge for the end consumer to compare them [5]. The resolution of this issue is highly time and resource consuming, resulting in lost sales and low adoption of e-Commerce and Marketplace channels in emerging markets.

In the investigation "An LSH-Based Model-words-Driven Product Duplicate Detection Method" [6] proposed a solution that will comprise two stages of comparison in order to reduce computation time. The first part of the algorithm uses Locally Sensitive Hashing (LSH) to find candidate pairs. Then, the Multi-component Similarity Method (MSM) will be used to find product duplicate records. In addition key-value pairs and advance data cleansing extensions to the proposed model were implemented resulting in a reduction of 6% in the F_1-measuere reducing 95% the number of required computations. The following figure presents the method proposed in the investigation.

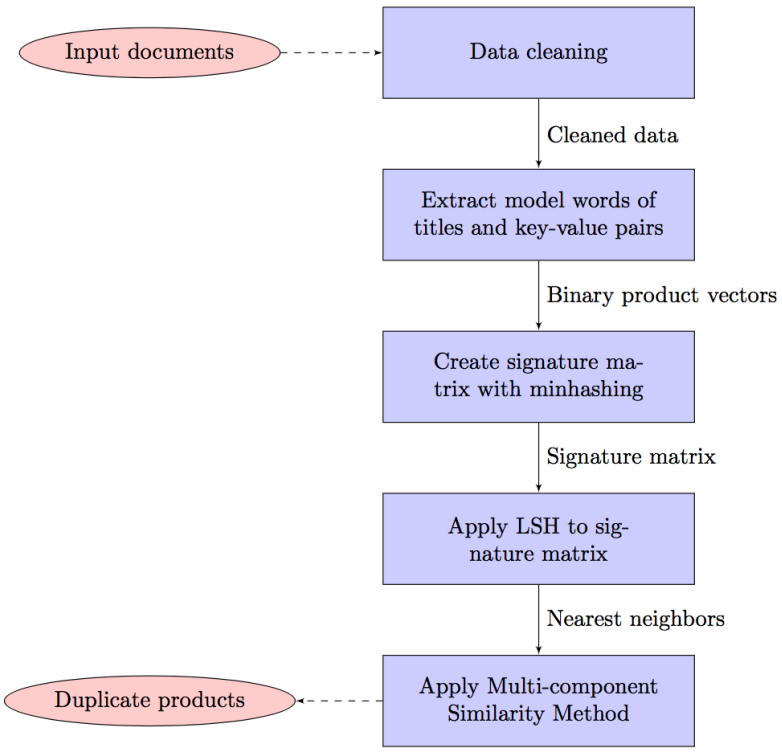


Figure 14- General Overview of MSMP+ Method. Source: [6]

One contribution of this investigation is the ability to extract model words (aka product characteristics) from the product record title creating a signature matrix with min-hashing to then apply LSH and MSM in order to find duplicate records. This investigation found that the pair completeness and pair quality improved 12.2% and 9.2% respectively, and the F_1-measure also meliorate in 9.3%.

Also, [5] in the research "Duplicate Detection in Web Shops using LSH to Reduce the Number of Computations" addresses the product master data record duplication issue. In this specific investigation the authors present method where first they build an uniform vector representation of the product master data record. Then the vectors are used to pre-select potential duplicate candidates using LSH algorithm. Then the MSM algorithm is applied to find duplicates. The result of the investigation presents that a 95% reduction in computations can be achieved with only a minor decrease of 9% in the F_1-measure. In concordance with the investigation carried out by [6], the method implies the extraction of model words to build the binary product vectors, then apply the LSH algorithm and last the MSM algorithm. From these two investigations we can infer that this approach is a good method to reproduce and enhance in order to achieve the desired results.

Another application of an unsupervised framework used as part of a product de-duplication solution is the research made by [1] presented in "An unsupervised framework for extracting and normalizing product attributes from multiple websites" article. Specifically in this investigation the authors focus on how to extract and normalize product attributes based on a *"probabilistic graphical model that can model the page-independent content information and the page-dependent layout information of the text fragments in Web pages".* In this investigation the authors present an extension of the "Dirichlet mixture model" that does not need supervision training. As part of the results we can see that the proposed model outperforms the benchmark in all the quantitative kpi's measurements across three product domains.

Also, [2] addresses the challenge of extracting product attributes/characteristics from product offer's descriptions using regular expressions, in which they propose a learning algorithm that seeks the best regular expressions in order to extract these product attributes. One of the contributions of this investigation is that *"regular expressions for information extraction rely on more complex constructs".* As a result of the investigation experiment conducted on the seven "most popular" product categories, presents that the proposed algorithm for learning regular expressions provides good results when the attribute value is numeric or semi-numeric. The findings of the experiment shows that F_1-measure in the top five product characteristics "Model, Storage, Display, Processor, Dimension" presents great results as its presented in the figure bellow, where precision, recall and F_1-measure are plotted. These results prove to have great value in extracting the embedded attributes in a product offer title or description. Specifically these results for the studied product categories are very useful, but the application cannot be extrapolated to another language for instance.

6 RESEARCH METHODOLOGY

6.1 MAIN RESEARCH METHODOLOGY

This research will use the “Engineering Method”, which is a direct descendant of the “Scientific Method”. This method includes the following stages:

- Analysis
- Design
- Development
- Evaluation

6.2 SPECIFIC RESEARCH METHODOLOGIES

Using as a guide the “Engineering Method” presented in section 6.1, this investigation will have the following tasks per specific objective. This approach will ensure the correct focus is applied in order to achieve each objective.

Investigation Specific Objectives	Associated Tasks to Achieve Investigation Specific Objectives
Process a set of data in order to build, train and test the proposed solution using machine learning techniques	Select a public data corpus that contains product master data records and they are labeled as duplicates or non-duplicates.
	Analyze the content of the corpus and provide high level analysis of the records in order to decide if it is fit for use in the investigation.
	Cleanse, and standardize, the product data corpus as part of the preprocessing task prior to use it in the model.
	Divide the product data corpus in Trian, Test and Validation subsets in order to use them accordingly during the investigation.
Select one machine learning technique as basis to propose a new model	Review the state of the art of current machine learning models that have been applied to the resolution of de-duplicated product records.
	Select from the reviewed models one technique to based the model design and development.
Build a model of machine learning solution / application that will de-duplicate product records based on data extracted from the "product title" and "product significant characteristics", that will reduce de-duplication mismatches and manual review tasks	Design a the machine learning model that aims the resolution of the de-duplicated product records.
	Build a the solution model using python programming language.
	Train the solution model using the product data corpus subset records
	Make initial adjustments to the solution model
Evaluate the proposed model measuring the effectiveness against the labeled data set	Propose a measuring standard to measure the model effectiveness
	Test the solution model with the product data corpus subset records
	Validate and register the effectiveness of the solution model
	Make final adjustments to the solution model
	Record the findings and deliver conclusions

Table 1- Specific Research Methodology Approach. Source: Author

7 INVESTIGATION EXPERIMENTAL PROCESS

7.1 INVESTIGATION PROCESS SUMMARY

This investigation proposes the following experimental process, which has been based upon the scientific method. This method encompasses the following pillars:

- Observation / Question
- Research topic area
- Hypothesis
- Test with experiment
- Analysis
- Conclusions

First the data corpus analysis will be presented, in order to achieve the first specific objective of this investigation. Then the data corpus will be revised and analyzed in order to measure and decide on its applicability to the investigation objectives. After, data corpus preprocessing development will be presented, as a fundamental pilar of this investigation. Two different approaches will be delivered. Next, the architecture analysis and tests will be performed in order to achieve the analyzing, building and training of the proposed solution as another specific objective of this investigation. Finally the evaluation of the proposed solutions, which are five different models, will be evaluated in order to select the best model and concluded on pros and cons. Also, the conclusions and recommendations of the investigation will be presented and the end of this document.

The diagram presented below summarizes the experimental process used in this investigation. The detail of each step in the process is described in the subsequent sections.

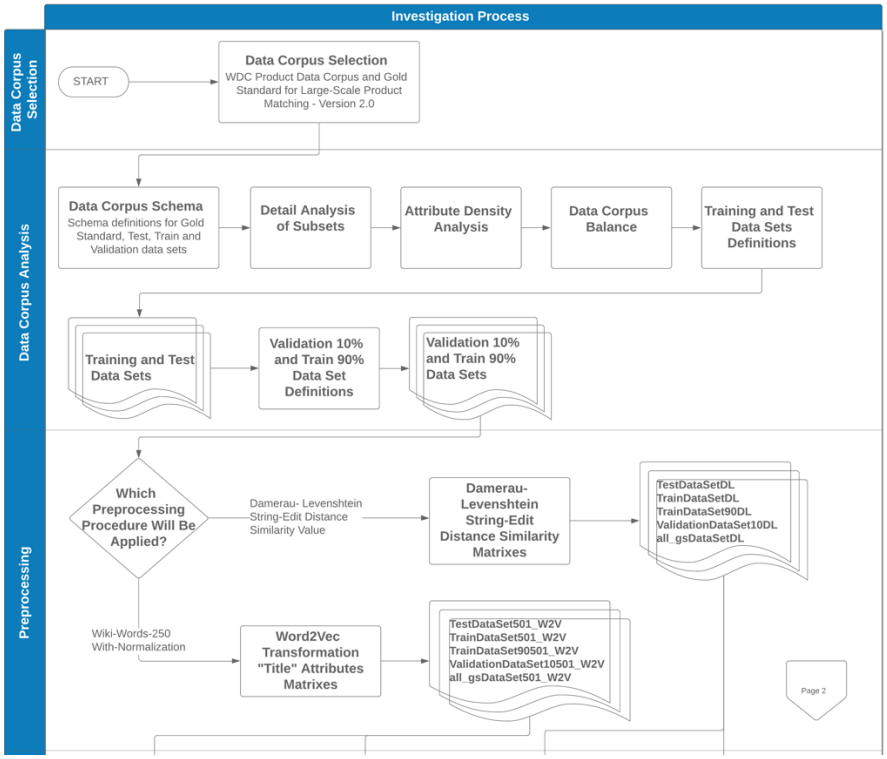


Figure 15- Investigation Process Summary Diagram – Page 1. Source: Author.

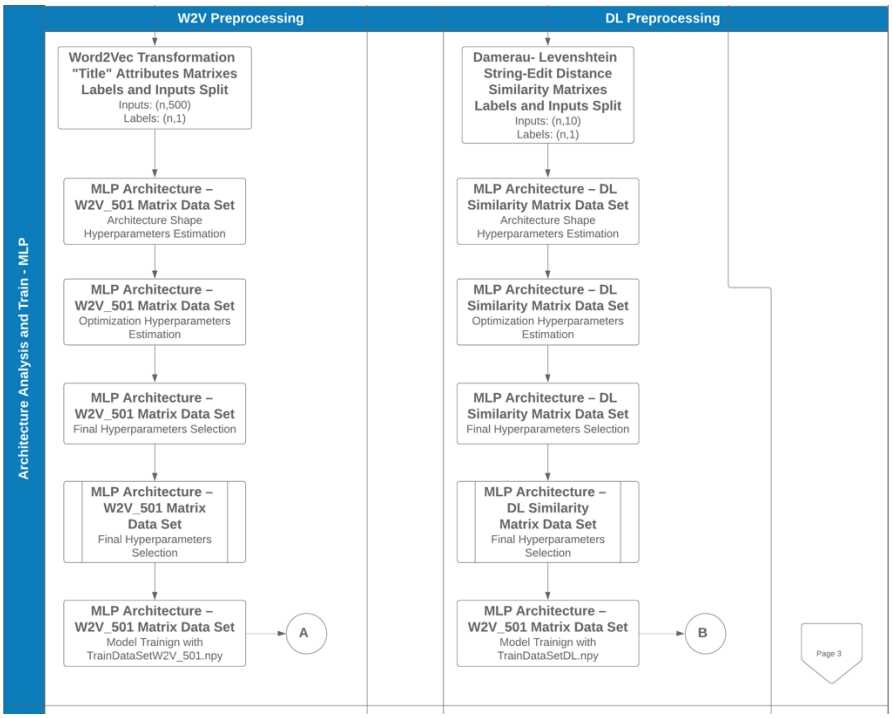


Figure 16- Investigation Process Summary Diagram – Page 2. Source: Author.

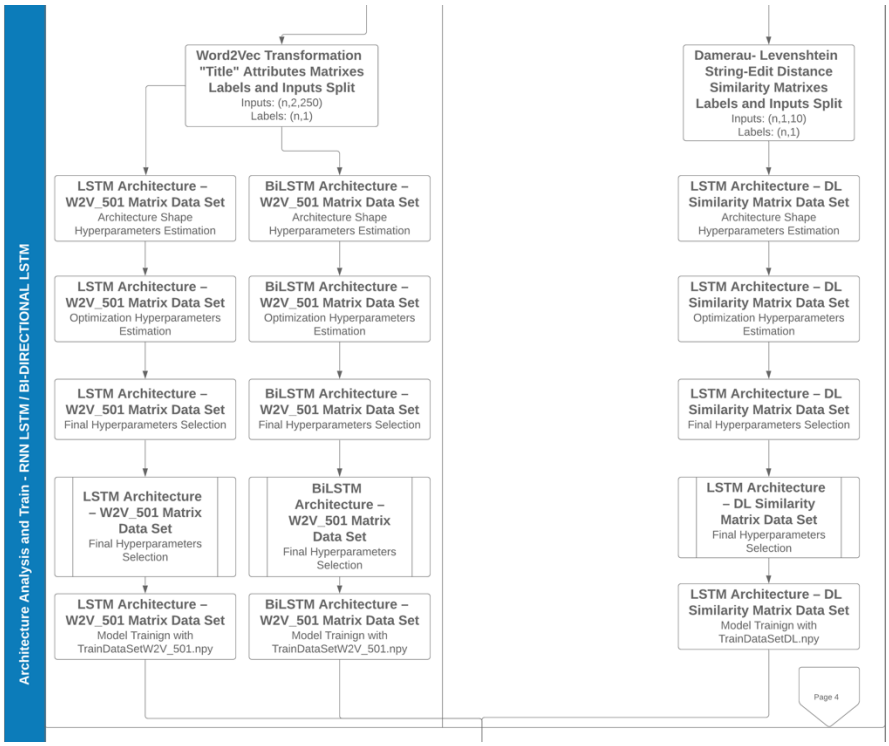


Figure 17- Investigation Process Summary Diagram – Page 3. Source: Author.

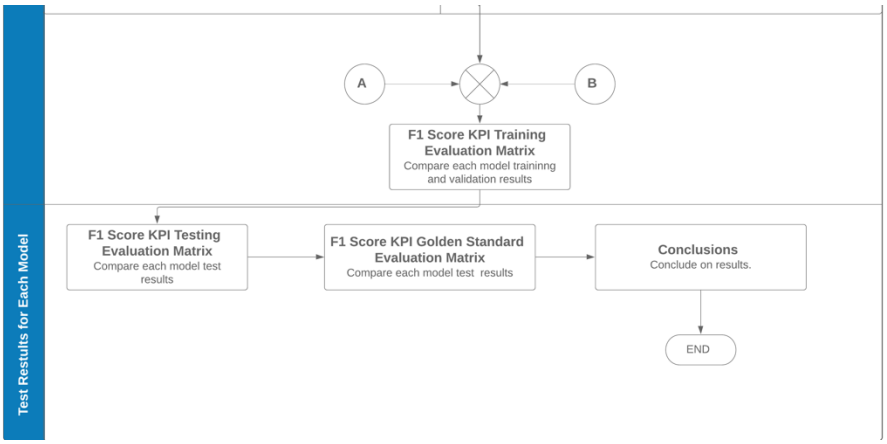


Figure 18- Investigation Process Summary Diagram – Page 4. Source: Author.

8 DATA CORPUS ANALYSIS

8.1 DATA CORPUS SELECTION

For this investigation, in order to test the investigation hypothesis, we need to acquire, analyze and describe a proper data corpus. First of all, as this investigation focuses in the solving the de-duplication of product master data records, we will need to find a data corpus with this kind of data. In addition, the selected corpus has to have the following pre-requisites:

- It has to be a product and or marketplace product offers data corpus that includes several product data record specification attributes that describe the product record.
- It has to have a large number of records in order to perform test, train and validation phases.
- It has to have labeled data in order to apply supervised training.
- It needs to be public.
- It must be from a reliable source.
- It needs to be recent.

Taking into account the above criteria, the selected data corpus has met all of them and will be of great value to this investigation.

- Selected Data Corpus: WDC Product Data Corpus and Gold Standard for Large-Scale Product Matching - Version 2.0[3]
- Data Corpus Source: School of Business Informatics and Mathematics. Data and Web Science Group – University of Mannheim[21]
- Data Corpus Basic Description:
    - This page provides Version 2.0 of the WDC Product Data Corpus and Gold Standard for Large-scale Product Matching for public download. The product data corpus consists of 26 million product offers originating from 79 thousand websites. The offers are grouped into 16 million clusters of offers referring to the same product using product identifiers, such as GTINs or MPNs. The gold standard consists of 4,400 pairs of offers that were manually verified as matches or non-matches. For easing the comparison of supervised matching methods, we also provide several pre-assembled training and validation sets for download (ranging from 9,000 and 214,000 pairs of offers).
- Date of release: 2019-10-23
- Labeled Data: This data corpus has label data set consisting of 4,400 pairs of offers that were manually verified and marked as matches or non-matches.
    - Training sets: In addition to the labeled data of 4,400 pairs of offers, the data corpus has created three training data sets in four sizes.
        - Small: 9038 pairs
        - Medium: 25567 pairs
        - Large: 103411 pairs
        - Extra-large: 214736 pairs

    These data sets can be used for the testing and training phases of the investigation. Letting the gold standard data set be used for validation purposes only.

8.2 DATA CORPUS SCHEMA

The data corpus has the following structure, which is inherited to all the data sets mention above. The data corpus format is json. Bellow the schema is presented:

[21] https://www.uni-mannheim.de/dws/

- **id**: Unique integer identifier of an offer
- **cluster_id**: The integer ID of the cluster to which an offer belongs.
- **identifiers**: A list of all identifier values that were assigned to an offer together with the schema.org terms that were used to annotate the values.
- **category**: One of 25 product categories the product was assigned to, NaN if not part of the English subset
- **title**: The product title
- **description**: The product description
- **brand**: The product brand
- **price**: The product price
- **specTableContent**: The specification table content of the products website as one string
- **keyValuePairs**: The key-value pairs that were extracted from the specification tables

8.2.1 Data Corpus Schema for Gold Standard and Training Sets

The Gold Standard and Training Subset JSON files contain product offer pairs where each attribute exists twice, once with the ***suffix _left*** for the left offer of a pair and once with the ***suffix _right*** for the right offer. Additionally there is the attribute ***pair_id*** which uniquely identifies each pair by concatenating their individual ids in the form ***id_left + '#' + id_right***. Finally the ***binary attribute label*** signifies if two product offers refer to the same product or not. The table below presents the detail of this schema.

DATA CORPUS SCHEMA FOR GOLD STANDARD AND TRAINING SETS	
Data Corpus Attribute	Attribute Type
id_left	int64
title_left	object
description_left	object
brand_left	object
price_left	object
specTableContent_left	object
keyValuePairs_left	object
category_left	object
cluster_id_left	int64
identifiers_left	object
id_right	int64
title_right	object
description_right	object
brand_right	object

DATA CORPUS SCHEMA FOR GOLD STANDARD AND TRAINING SETS	
price_right	object
specTableContent_right	object
keyValuePairs_right	object
category_right	object
cluster_id_right	int64
identifiers_right	object
label	int64
pair_id	object

Table 2- Data Corpus Schema for Gold Standard and Training Sets. Source: [3]

In addition to the definition above the following attributes of the data corpus that are defined as "object", consist of arrays that contain several attributes. For example the attribute "identifiers" contains an array of with the following attributes:

Identifiers Attribute Composition		
Identifier Key	dtype	multivalued
mpn	<U11	yes
sku	<U11	yes
gtin8	<U11	yes
gtin12	<U11	yes
gtin13	<U11	yes
gtin14	<U11	yes
identifier	<U11	yes
productID	<U11	yes

Table 3- Identifiers Attribute Composition for Gold Standard and Training Sets. Source: [3]

Another peculiarity is that these arrays can hold multiple values for the same key within the attribute. This is something that is worth analyzing as they might help during the de-duplication execution. Also the attributes ***specTableContent*** and ***keyValuePairs*** that are defined as "object", contain multivalued data for each record. The difference is that the ***keyValuePairs*** attribute contains an array with the data divided into different sections to retrieve specific data using a comma separator, whereas the attribute ***specTableContent*** has all the data in a single string. Bellow an example:

specTableContent vs keyValuePais Attribute Comparison	
specTableContent	***keyValuePairs***
'mfg model amd yd1400bbaebox mfg part yd1400bbaebox upc 730143308427 description ryzen 5 1400 3 2g 8mb 65w with wraith stealth cooler price 169 80 essential information blt item b8q6824 manufacturer part yd1400bbaebox manufacturer amd description ryzen 5 1400 3 2g 8mb 65w with wraith stealth cooler weight 1 1 lbs manufacturer s website http www amd com dimensions 5 4 x 5 4 x 5 3 upc 730143308427 return policy standard blt return policy'	*'{'mfg model': 'amd yd1400bbaebox', 'mfg part': 'yd1400bbaebox', 'upc': '730143308427', 'description': 'ryzen 5 1400 3 2g 8mb 65w with wraith stealth cooler', 'price': '169 80', 'blt item': 'b8q6824', 'manufacturer part': 'yd1400bbaebox', 'manufacturer': 'amd', 'weight': '1 1 lbs', 'manufacturer s website': 'http www amd com', 'dimensions': '5 4 x 5 4 x 5 3', 'return policy': 'standard blt return policy'}'*

Table 4- specTableContent vs keyValuePais Attribute Comparison. Source: [3]

8.3 Data Corpus Detail Analysis of Subsets

The next step is to analyze each provided subsets provided in the data corpus.

8.3.1 Golden Data Set

First, as the data corpus provides a data set of 4,400 pairs, which has been called the Golden Standard, and these pairs where manually reviewed in order to provide two matching pairs offers (positive) and 5 or 6 non-matching (negative) pairs per product, we have reserved these data set for validation purposes only. In addition this manual validation of the pairs has been doe only for four product categories within the data corpus with 1,100 pairs for each of the listed categories, only for the English data. The table below presents the detail of this data set.

Gold Standard Data Set Detail Table			
Product Category	**Positive Pairs**	**Negative Pairs**	**Percentage of Positive Pairs**
Computers	300	800	27.27%
Cameras	300	800	27.27%
Watches	300	800	27.27%
Shoes	300	800	27.27%
Total	**1,200**	**3,200**	27.27%

Table 5- Gold Standard Data Set Detail Table. Source: [3]

8.3.2 Data Subsets For Training and Testing

Taking into account the above, the data corpus also provides four (4) different data sets of four (4) different sizes (small, medium, large, extra-large), with positive and negative pairs for the same product categories included in the Gold Standard. Table 5 presents the detail provided in the data

corpus if each one of these subsets. In the following sections we will transform these sets four the investigation best convenience.

Test and Training Data Sets Detail Table					
Data Set Size Name	**Product Category**	**Positive Pairs**	**Negative Pairs**	**Total Pairs Per Data Set**	**Percentage of Positive Pairs**
Small	Computers	722	2,112	2,834	25.48%
Small	Cameras	486	1,400	1,886	25.77%
Small	Watches	580	1,675	2,255	25.72%
Small	Shoes	530	1,533	2,063	25.69%
Small	**Total**	**2,318**	**6,720**	**9,038**	**25.65%**
Medium	Computers	1,762	6,332	8,094	21.77%
Medium	Cameras	1,108	4,147	5,255	21.08%
Medium	Watches	1,418	4,995	6,413	22.11%
Medium	Shoes	1,214	4,591	5,805	20.91%
Medium	**Total**	**5,502**	**20,065**	**25,567**	**21.52%**
Large	Computers	6,146	27,213	33,359	18.42%
Large	Cameras	3,843	16,193	20,036	19.18%
Large	Watches	5,163	21,864	27,027	19.10%
Large	Shoes	3,482	19,507	22,989	15.15%
Large	**Total**	**18,634**	**84,777**	**103,411**	**18.02%**
Extra-Large	Computers	9,690	58,771	68,461	14.15%
Extra-Large	Cameras	7,178	35,099	42,277	16.98%
Extra-Large	Watches	9,264	52,305	61,569	15.05%
Extra-Large	Shoes	4,141	38,288	42,429	9.76%
Extra-Large	**Total**	**30,273**	**184,463**	**214,736**	**14.10%**

Table 6- Train and Test Data Sets Detail Table. Source: [3]

One interesting fact to annotate is that while the data set size increases, the percentage of positive pairs decreases. The chart below presents this relation.

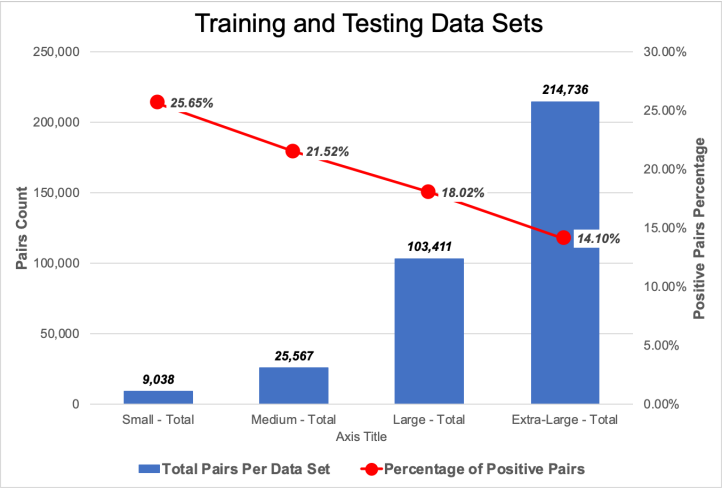


Chart 1- Training and Testing Data Sets Size vs Positive Pairs Percentage. Source: Author

8.3.3 *DATA CORPUS ATTRIBUTE DENSITY ANALYSIS*

For each of the data sets, we need to understand its attribute density, in order to make decisions about which attributes to include, or not to include. Specify if the data corpus array objects will be divided into individual attributes or treated as a whole.

The first facts that needs to be noticed is that all the data sets have 100% of density in the **"Title"** and **"Category"** attributes. This will ensure that at least we have a valid key to provide to the model that will limit and/or enforce comparison just between records of the same category. It can be used as a pre-processing hint for the model. Also, as it has been stated in the state of the art literature review, products and offers on online websites, e-commerce's, and marketplaces, will at least have the record **"Title"** which includes a representative amount of the record data that describes it accurately. Therefore, these two attributes will be of mandatory use in the proposed solution.

Then, from the data corpus schema it has been selected the following additional attributes for detail study.

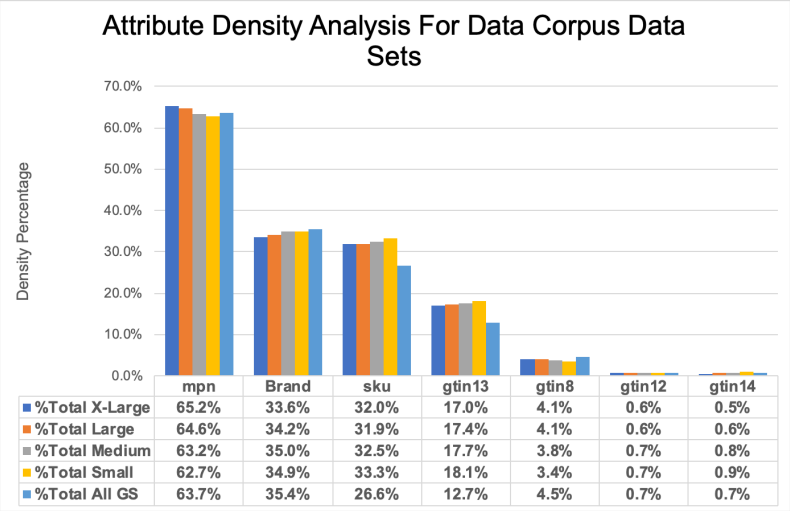
- Brand
- Identifiers Array Key Values: These attribute values and key values have been selected as they represent unique identifiers for the record. Even though that the exact same product can have completely different unique identifiers, theory states that they are a good starting point to de-duplicate.
    - MPN: Material Part Number provided by the original manufacturer or supplier.
    - SKU[22]: Stock Keeping Unit identification. It can be the one provided by the original manufacturer or supplier, or it can be one assigned by the retailer
    - GTINs[23]: Global Trade Identification Number which is an international standard to identify uniquely products with bar code structures[43].
        - GTIN-8 (EAN/UCC-8): An 8-digit number used predominately outside of North America

[22] SKU: Stock Keeping Unit acronym. Used commonly to identify uniquely a product with in a ERP or WMS system. Used as unique key of the record within the boundries of the organization.

[23] GTIN: GTIN describes a family of GS1 (EAN.UCC) global data structures that employ 14 digits and can be encoded into various types of data carriers. Currently, GTIN is used exclusively within bar codes, but it could also be used in other data carriers such as radio frequency identification (RFID). The GTIN is only a term and does not impact any existing standards, nor does it place any additional requirements on scanning hardware. For North American companies, the UPC is an existing form of the GTIN.

- GTIN-12 (UPC-A): An 12-digit number used primarily in North America
- GTIN-13 (EAN/UCC-13): An 13-digit number used predominately outside of North America.
- GTIN-14 (EAN/UCC-14 or ITF-14): this is a 14-digit number used to identify trade items at various packaging levels.

Chart 2 presents the attribute density of the selected attributes in each data set, including the validation, test and training data sets. As the **"Title"** and **"Category"** attributes have a 100% density in all data sets, they have been removed of this chart, in order to shed attention in the remaining attributes.



	mpn	Brand	sku	gtin13	gtin8	gtin12	gtin14
%Total X-Large	65.2%	33.6%	32.0%	17.0%	4.1%	0.6%	0.5%
%Total Large	64.6%	34.2%	31.9%	17.4%	4.1%	0.6%	0.6%
%Total Medium	63.2%	35.0%	32.5%	17.7%	3.8%	0.7%	0.8%
%Total Small	62.7%	34.9%	33.3%	18.1%	3.4%	0.7%	0.9%
%Total All GS	63.7%	35.4%	26.6%	12.7%	4.5%	0.7%	0.7%

Chart 2- Attribute Density Analysis for Data Sets. Source: Author

From the chart above we have concluded that unique identifiers, a priori, are not a good source of de-duplication identification. First of all, the **"MPN Identifier"** key value is only present in in average in 63.9% of the records across all data sets. It can be implied that despite the fact that all manufacturers and suppliers uniquely identify each of their products with a specific identifier, roughly just 64% of the websites publish this data. Moreover, we can imply if the record is supplied by a second or third echelon in the supply chain, like a retailer or distributor, this data might not be present or simply it is not published. Second, the **"Brand"** attribute is only present in 34.6% of the records. Although it has been noticed as a very important attribute to publish and describe products, there is no standard that enforces its publication. Moreover, it has been stated that in more than 80% of the regular offers in market place and products in an e-commerce website, have embedded the product brand within the **"Title".** Third, with an average of 31.3% of density across all the data sets, the **"SKU Identifier"** key value, contributes regular or near to non-contribution to the attribute density analysis and probably to the proposed solution. This is due to the fact that while it represents a common unique identifier for products, it is more likely that every echelon in the supply chain provides its own value, as it is just valid for internal purposes within a company and not across supply chain partners. Last, all of the **"GTIN Identifiers"** key values, with very low density in each data set will not provide the intended amount of unique identification that the theory implies. There are several factors that contribute to this behavior in the data sets. The first one is that GTIN codes are heavily used in logistics, therefore if presented to the public in websites, it does not provide high value, resulting in very low presence of the data in websites, e-commerce, and marketplaces. In addition, as this standard implies a suffix which represents the country of origin of manufacturing, resulting in a different value if the exact same product is manufacture in two different countries. This has been

addressed by Karpischek et al. in their publication ***"The not so unique global trade identification number: Product master data quality in publicly available sources"***[44]***.***

As preliminary conclusions, we can imply that the model will need to rely heavily in the most dense attributes, being these the **"Title"** and **"Category"**. The analysis also supports the fact that there is not a standard in the data model definition across all the information supply chain, which starts with the product manufacturer, and then it mutates when the data passes down to the next echelon in the chain like distributors, retailers, marketplaces. So when product master data is presented to end consumers, it is quite different, resembling the "bull whip effect". Lastly, this is also a great representation of real life scenarios, where there is no perfect data sets, and these are all heterogeneously composed. Therefore, this proofs the value within the objective of this investigation, where we aim to identify these duplicates in order to provide better value to product master data, composed and related of product offers.

8.3.4 DATA CORPUS BALANCE

For each of the data sets, we need to understand the distribution of true negatives versus true positives and how this balance might affect the result of the implementation. The char bellow presents a detail of the percentage of positive pairs in each data set.

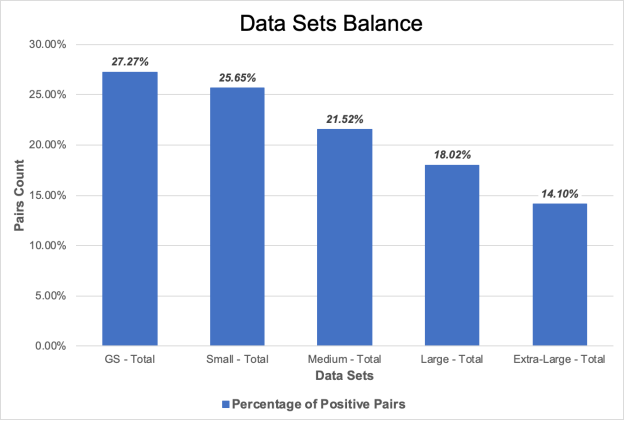


Chart 3- Data Sets Percentage of Positive Pairs. Source: Author

From the char above, we can conclude that data corpus, in all the data sets is unbalanced. Also, as the data sets grow in size, the percentage of positive pairs decreases. These two facts need to be taken into account during the development of the solution model, as for instance the KPI's to measure the effectiveness need to be chosen accordingly, and that probably, using the bigger data sets may result in an overfitting behavior.

As preliminary conclusion, this also implies that in real life scenarios, there is in average 1:5 ratio of duplicate pairs versus non-duplicate pairs. This is also a great insight to take into account during the next phase of this investigation.

8.4 Training and Test Data Sets Definitions

8.4.1 Independent Data Sets

In order to use the bigger amount of data of the provided test and train data sets, we have decided to build an independent data set appending the four data subsets together, and removing duplicate pairs in order to provide a single data set which we will later divide into a 70% / 30% for training and testing purposes. The first step is to analyze if there are duplicated pairs across all training data subsets. To achieve this we created the following matrix, where we compared each data subset in order to find if they all or some pair records where included in the other data subset. After the four subset date merge the total pair count is 352,752. Then we removed the duplicated data and finished with a single data set of 268,228 pair records that are totally independent. The total repeated pairs found after the data set merge were 84,524. We also found that the 4,400 pair records of the Golden Standard data set are completely independent. We will reserve this data set for further evaluation. The matrixes below present the details of our findings.

Independent Pairs Analysis Matrix						
Data Set Name	**Data Set Size**	**GS - Total**	**Small - Total**	**Medium - Total**	**Large - Total**	**Extra-Large - Total**
GS - Total	**4,400**	**0**	**0**	**0**	**0**	**0**
Small - Total	**9,038**	**0**	**0**	**2,066**	**4,250**	**5,691**
Medium - Total	**25,567**	**0**	**2,066**	**0**	**11,218**	**15,281**
Large - Total	**103,411**	**0**	**4,250**	**11,218**	**0**	**60,034**
Extra-Large - Total	**214,736**	**0**	**5,691**	**15,281**	**60,034**	**0**

Table 7- Independent Pairs Analysis. Source: [3]

New Data Set Training / Testing / Validation	
Item Description	**Number of Paris**
Total Records	**352,752**
Total Removed	**84,524**
New Data Set Size	**268,228**

Table 8- New Data Set Table. Source: [3]

Now, we found that the balance of this new data set has the following distribution:

Total Records	Duplicates (label=1)	Non Duplicates (label=0)
268,228	33,527	234,701
Percentage	12.50%	87.50%

Table 9- New Data Set Balance Distribution. Source: [3]

In addition to the above, and in order to work with only the four main product categories included in the "Golden Standard" data set, we selected just the following pair records. This includes 96.57% of the selected records. This decision will allow us to compare in our final validation exercise the exact same product categories as the ones matched and review manually included in the "Golden Standard" data set. The following table shows the count and balance analysis:

New Data Set Training / Testing Four Main Product Categories					
Product Category	Pair Record Count	Duplicates (label=1)	Non Duplicates (label=0)	Percentage Duplicates (label=1)	Precentage Non Duplicates (label=0)
Computers_and_Accessories	82,713	10,244	72,469	12.38%	87.62%
Jewelry	73,626	10,027	63,599	13.62%	86.38%
Camera_and_Photo	51,411	7,530	43,881	14.65%	85.35%
Shoes	51,295	4,527	46,768	8.83%	91.17%
Total	259,045	32,328	226,717	12.48%	87.52%

Table 10- New Data Set Training / Testing Four Main Product Categories. Source: [3]

8.4.2 Adjusting Data Set Duplicates Balance 27.27%

To improve the balance of the data set between duplicate records and non-duplicate records, in order to bring it close the "Golden Standard" data set, the following transformation was made. For each product category, all the duplicate pair records with label = 1 are going to be included in the final data set. For the non-duplicate pair records per each product category, only a random subset of records was selected in order to maintain the balance distribution of 27.27% duplicates in the data set. The subset was built with the corresponding size that each product category should have in order to maintain this duplicate percentage balance. The table below presents the details:

New Data Set Training / Testing Four Main Product Categories with 27.27% Duplicates Balance					
Product Category	Pair Record Count	Duplicates (label=1)	New Random Non Duplicates Pairs (label=0)	Percentage Duplicates (label=1)	Precentage New Random Non Duplicates Pairs (label=0)
Computers_and_Accessories	37,565	10,244	27,321	27.27%	72.73%
Jewelry	36,769	10,027	26,742	27.27%	72.73%
Camera_and_Photo	27,613	7,530	20,083	27.27%	72.73%
Shoes	16,601	4,527	12,074	27.27%	72.73%
Total	118,548	32,328	86,220	27.27%	72.73%

Table 11- New Data Set Training / Testing Four Main Product Categories with 27.29% Duplicates Balance. Source: Author

8.4.3 Creating Training and Test Final Data Sets

As part of this investigation methodology, we will divide the final and cleanse data set with a 70% / 30% distribution for training and testing purposes. This division was made independently within each product category in order to maintain in each of these the duplicate pair record balance. Table 11 below presents the details for the training data set and Table 12 presents the detail for testing data set.

TrainDataSet: New Data Set For Training Four Main Product Categories 70% of Records					
Product Category	Pair Record Count	Duplicates (label=1)	Non Duplicates Pairs (label=0)	Percentage Duplicates (label=1)	Precentage Non Duplicates Pairs (label=0)
Computers_and_Accessories	26,296	7,171	19,125	27.27%	72.73%
Jewelry	25,738	7,019	18,719	27.27%	72.73%
Camera_and_Photo	19,329	5,271	14,058	27.27%	72.73%
Shoes	11,621	3,169	8,452	27.27%	72.73%
Total	82,984	22,630	60,354	27.27%	72.73%

Table 12- New Data Set For Training Four Main Product Categories 70% of Records. Source: Author

TestDataSet: New Data Set For Testing Four Main Product Categories 30% of Records					
Product Category	Pair Record Count	Duplicates (label=1)	Non Duplicates Pairs (label=0)	Percentage Duplicates (label=1)	Precentage Non Duplicates Pairs (label=0)
Computers_and_Accessories	11,269	3,073	8,196	27.27%	72.73%
Jewelry	11,031	3,008	8,023	27.27%	72.73%
Camera_and_Photo	8,284	2,259	6,025	27.27%	72.73%
Shoes	4,980	1,358	3,622	27.27%	72.73%
Total	35,564	9,698	25,866	27.27%	72.73%

Table 13- New Data Set For Testing Four Main Product Categories 30% of Records. Source: Author

8.4.4 CREATING VALIDATIONDATASET10 WITH 10% RECORDS OF TRAIN DATA SET

For hyperparameter estimation, we have extracted 10% of the records, maintaining the percentage duplicates (label = 1), from the Train Data Set. This new data set is called Validation Data Set 10% [3]. When final tests are executed the data set depicted in section 8.4.3 will be used with the total amount of records. This data set contains 8,242 total records. It will be used for validating the best hyperparameters in each proposed architecture experiment.

8.4.5 CREATING TRAINDATASET90 WITH 90% RECORDS OF TRAIN DATA SET

For initial training of the models, we have extracted 90% of the records, maintaining the percentage duplicates (label = 1), from the Train Data Set. With a droop function we have removed the 8,242 records in the Validation Data Set 10%. This new data set is called Train Data Set 90% [3]. When final tests are executed the data set depicted in section 8.4.3 will be used with the total amount of records. This data set contains 74,232 total records.

9 PREPROCESSING

9.1 NORMALIZED DATA

The original data corpus provides two versions of the data. One without any transformation which includes all the raw textual properties, and a normalized version which presents all the data in lowercase and all non-alphanumeric characters have been removed. We will work with the normalized data corpus.

9.2 ADDITIONAL NORMALIZATION OR STANDARDIZATION

Taking into account that the machine learning algorithms need numeric inputs, all the selected data needs to be transformed to arrays of numeric data. As presented in section 8.2, a tuple represents a pair of two product records, where we will call to the first record “left” and to the second record “right” for further understanding. This investigation will test two different approaches. The first one is to transform each pair record “left” and “right” attribute, calculating what has been defined as the “Damerau-Levenshtein String Edit Distance Similarity Value”. For this approach the string-edit

algorithm proposed by Frederick J. Damerau in [45], extending Vladimir I. Levenshtein first approach, will be used to calculate how similar each attribute value string on the "left" side of the tuple is from its corresponding value in the "right" side of the tuple. What this means is that por each record (pair_id) in each data set, we will calculate the similarity for each pair of corresponding attributes, and append the "pair_id" and "label" attributes. In the second approach, the 20 attributes will be pre-process using a token based text embedding trained with a word2vec skip-gram based on the algorithm proposed by Mikolov et al. in [46]. The pre-trained algorithm has been extracted from TensorFlow HUB[24]. Below each pre-processing approach will be described in detail. The goal of the performing the two different pre-processing transformations is to measure the effectiveness vs. complexity of each pre-processing technique, and understand if the final result is affected significantly in each case.

9.2.1 DAMERAU- LEVENSHTEIN STRING-EDIT DISTANCE SIMILARITY VALUE

For each equivalent pair of attributes within each record, label with the same attribute name and the suffix "_left" and "_right" respectively, the following function will be applied returning a single value between 0 and 1 [0,1]. This value represents the ratio of similarity of each value string in terms of the "Damerau- Levenshtein" string comparison algorithm. The only two attributes that won't be parsed are the "pair_id" and the "label", as per definition the first one identifies uniquely the record and the later one presents the label of duplicate (1) or non-duplicate record (0). Hence the data sets matrix size will reduce from 22 columns to 11 columns. Here the transformation function:

First each string value has been parse using the regular expression presented in Figure 1, in order to remove non-alphanumeric characters, white spaces, and transform all to lower case.

$$norm(x) = RegExp("\backslash s|[\hat{}a-z0-9]")$$

Figure 19- Normalize String Regular Expression Function. Source: [3]

Then the length of each compared string is measured and the maximum length will be used as denominator in order to retrieve the ratio between the Damerau-Levenshtein distance and the longest string. This will ensure that the ratio is a number between zero and one [0,1].

$$d_{a,b}(i,j) = min \begin{cases} 0, & if\ i = j = 0 \\ d_{a,b}(i-1,j) + 1, & if\ i > 0 \\ d_{a,b}(i,j-1) + 1, & if\ j > 0 \\ d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)}, & if\ i,j > 0 \\ d_{a,b}(i-2,j-2) + 1, & if\ i,j > 1\ \wedge a[i] = b[j-1] \wedge a[i-1] = b[j] \end{cases}$$

Figure 20- Damerau–Levenshtein distance between two strings a and b[25]. Source: [47]

$$DL(x,y) = \begin{cases} 0, & if\ x = 0\ \vee\ y = 0 \\ |1 - \frac{d(x,y)}{\max(len(x), len(y))}|, & if\ x \neq 0\ \wedge\ y \neq 0 \end{cases}$$

Figure 21- Damerau- Levenshtein String-Edit Distance Similarity Value Function. Source: [3]

[24] TensorFlow HUB: TensorFlow Hub is a library for reusable machine learning modules. https://www.tensorflow.org/hub

[25] The Damerau–Levenshtein distance between *a* and *b* is then given by the function value for full strings: $d_{a,b}(|a|,|b|)$ where $i = |a|$ denotes the length of string *a* and $j = |b|$ is the length of *b*.

9.2.2 *Damerau- Levenshtein string-edit distance Similarity Matrixes*

For each data set, this preprocessing method has been applied. Each resulting matrix has been saved for further use during the model experimentation phase. The table below presents the details of this exercise.

Damerau- Levenshtein String-Edit Distance Similarity Matrix			
Matrix Position	**Position Content Definition: DL Function**	**Atttribute Input**	**dtype**
0	DL	brand_left,brand_right	float64
1	DL	category_left,category_right	float64
2	DL	description_left,description_right	float64
3	DL	title_left,title_right	float64
4	DL	gtin8_left,gtin8_right	float64
5	DL	gtin12_left,gtin12_right	float64
6	DL	gtin13_left,gtin13_right	float64
7	DL	gtin14_left,gtin14_right	float64
8	DL	mpn_left,mpn_right	float64
9	DL	sku_left,sku_right	float64
10	Value	label	float64

Table 14- Damerau- Levenshtein String-Edit Distance Similarity Matrix Content Definition. Source: Author

Damerau- Levenshtein String-Edit Distance Similarity Matrixes. Preprocessed Data Sets			
Data Set	**Data Set Definition**	**Matrix Numpy Array**	**Array Shape**
TestDataSet	New Data Set For Testing Four Main Product Categories 30% of Records	TestDataSetDL.npy	(35564,11)
TrainDataSet	New Data Set For Training Four Main Product Categories 70% of Records	TrainDataSetDL.npy	(82984,11)
TrainDataSet90	New Data Set For Training Hyperparameter Estimation. 90% of Records of TrainDataSet	TrainDataSet90DL.npy	(74232,11)
ValidationDataSet10	New Data Set For Validating Hyperparameter Estimation. 10% of Records of TrainDataSet	ValidationDataSet10DL.npy	(8242,11)
GS Total	Manual Validated Data Set Provided In Original Data Corpus[3]. 4,400 Golden Standard Records.	all_gsDataSetDL.npy	(4400,11)

Table 15- Damerau- Levenshtein String-Edit Distance Similarity Matrixes. Preprocessed Data Sets. Source: Author

9.2.3 WIKI-WORDS-250-WITH-NORMALIZATION

TensorFlow Hub is a library of reusable machine learning modules, and it contains a specific section dedicated to text embedding. This investigation will use the word2vec skipgram architecture proposed by Mikolov et al. in [46]. The selected module is called "wiki-words-250-with-normalization" [48]. This text embedding algorithm module is token based and it was trained on English Wikipedia corpus. This model is based on skipgram version of word2vec with 1 out-of-vocabulary bucket. The text input returns a 250-dimensional embedding vector. Some details of the skipgram model is that it uses a hierarchical softmax function with sub-sampling 1e-5. The text input is delivered in a batch of sentences in a one dimension tensor[26] of strings. The module preprocess the input by removing punctuation and splitting on spaces. It maps all out-of-vocabulary tokens into one bucket which is initialized with zeros. Finally, word embeddings are combined into sentence embedding using the square root (sqrtn) combiner depicted in tf.nn.embedding_lookup_sparse[49].

The resulting preprocessed matrix will have a shape of [n,501], representing n the number of records in each data set. The first two hundred fifty (250) columns, from position zero to two hundred forty nine [0,249], represents the attribute transformation using the method described in this section applied to the "title_left" attribute of each pair record. The second two hundred fifty (250) columns, from position two hundred fifty to four hundred ninety nine [250,449], represents the attribute transformation using the method described in this section applied to the "title_right" attribute of each pair record. Then the "label" value, which can only have values of "0" and "1" representing non-duplicate and duplicate respectively, for the pair record is stored in the five hundredth first column [500].

9.2.4 WORD2VEC TRANSFORMATION FOR "TITLE" ATTRIBUTES MATRIXES

For each data set, this preprocessing method has been applied. Each resulting matrix has been saved for further use during the model experimentation phase. The table below presents the details of this exercise.

Word2Vec Transformation for "Title" Attributes Matrix			
Matrix Position	**Position Content Definition**	**Atttribute Input**	**dtype**
[0,249]	**Word2Vec wiki-words-250-with-normalization**	**title_left**	**int64**
[250,500]	**Word2Vec wiki-words-250-with-normalization**	**title_right**	**int64**
[500]	**Value**	**label**	**int64**

Table 16- Word2Vec Transformation for "Title" Attributes Matrix. Source: Author

Word2Vec Transformation for "Title" Attributes Matrixes. Preprocessed Data Sets			
Data Set	**Data Set Definition**	**Matrix Numpy Array**	**Array Shape**
TestDataSet	**New Data Set For Testing Four Main Product Categories 30% of Records**	**TestDataSetW2V_501.npy**	**(35564,501)**

[26] A tensor is a vector or matrix of n-dimensions that represents all types of data. All values in a tensor hold identical data type with a known (or partially known) shape. The shape of the data is the dimensionality of the matrix or array.

Word2Vec Transformation for "Title" Attributes Matrixes. Preprocessed Data Sets			
Data Set	**Data Set Definition**	**Matrix Numpy Array**	**Array Shape**
TrainDataSet	**New Data Set For Training Four Main Product Categories 70% of Records**	**TrainDataSetW2V_501.npy**	**(82984,501)**
TrainDataSet90	**New Data Set For Training Hyperparameter Estimation. 90% of Records of TrainDataSet**	**TrainDataSet90W2V_501.npy**	**(74232,501)**
ValidationDataSet10	**New Data Set For Validating Hyperparameter Estimation. 10% of Records of TrainDataSet**	**ValidationDataSet10W2V_501.npy**	**(8242,501)**
GS Total	**Manual Validated Data Set Provided In Original Data Corpus**[3]**. 4,400 Golden Standard Records.**	**all_gsDataSetW2V_501.npy**	**(4400,501)**

Table 17- Word2Vec Transformation for "Title" Attributes Matrixes. Source: Author

10 MODEL CONSTRUCTION

10.1 ARCHITECTURE ANALYSIS

This investigation will use two machine learning architectures to include in the analysis and experimental tests. In conjunction with the two different preprocessing approaches we will be able to setup a robust test matrix in order to identify which combination of preprocessed data technique and analysis, architecture, and hyperparameters will return the most accurate solution in terms of performance. As we saw with [34] results where deep neural networks were used to solve problems in speech recognition, and pattern recognition, among others, suites very well this investigation purpose. Moreover, Mikolov et al. in [46] has proved that neural networks are very efficient measuring word similarity using continues word representation vectors. In addition it was proven that this architecture is both efficient achieving great results in the word similarity task, but also it was computationally very efficient. We can also see that Zong, Wu, Chu, in [50] uses machine learning techniques in an application that searches for duplicate material records within ERP system.

The proposed deep neural network architectures to be tested are a feed forward neural network, also known as the multi-layer perceptron, with backpropagation learning adjustments. We called this our base of naïve architecture. The second architecture is a deep neural network architecture called Recurrent Neural Network (RNN), specifically we will use the Long Short Term Memory (LSTM) cells, which is an evolution of our first approach as identified in our literature review. As stated by Palangi et al. in [51] LSTM RNN model sequentially takes each word in a sentence, extracts its information, and embeds it into a semantic vector. Due to its ability to capture long term memory, the LSTM-RNN accumulates increasingly richer information as it goes through the sentence, and when it reaches the last word, the hidden layer of the network provides a semantic representation of the whole sentence. This is of great use when addressing product deduplication problem using a single input as the

“Product Title” or “Product Description”. In addition, this architecture has been used by Ebraheem, M. et al. in [52] for distributed entity resolution applying bi-directional RNN with LSTM cells.

10.1.1 Architecture Analysis and Test Process

The multi-layer perceptron neural network (MLP)

10.1.2 Multi-layer Perceptron Neural Network

The multi-layer perceptron neural network (MLP) is often used in classification tasks. Specifically as presented by Kooli et al. in [53] this architecture has very good performance in entity resolution with a large train data set. Moreover it is a good starting point to compare with more complex architectures as RNNs, as we are aiming in this investigation. Also, Vinayakumar et al. in [54] compares the performance of a MLP to other deep learning architectures in their investigation which is intended to identify “phishing emails”.

10.1.2.1.1 MLP Architecture

Our MPL architecture is designed to have an input layer equal to the feature matrix size. When using the “Damerau- Levenshtein String-Edit Distance Similarity Matrixes”, as the last position represents the label, the input size will be ten (10). When using the “Word2Vec Transformation for "Title" Attributes Matrixes” the input size will be five hundred (500), as the fifth hundred first (501) represents the label. The architecture will use dense layer connections. This means that each neuron will be connected to the following ones in the subsequent layer. The output layer will have a single neuron, to preform correctly the binary classification of duplicated or non-duplicate record. The details of the fine hyperparameter tuning will be addressed in section 10.2, as the number of hidden layers and neurons per layer. The following figure illustrates the proposed architecture.

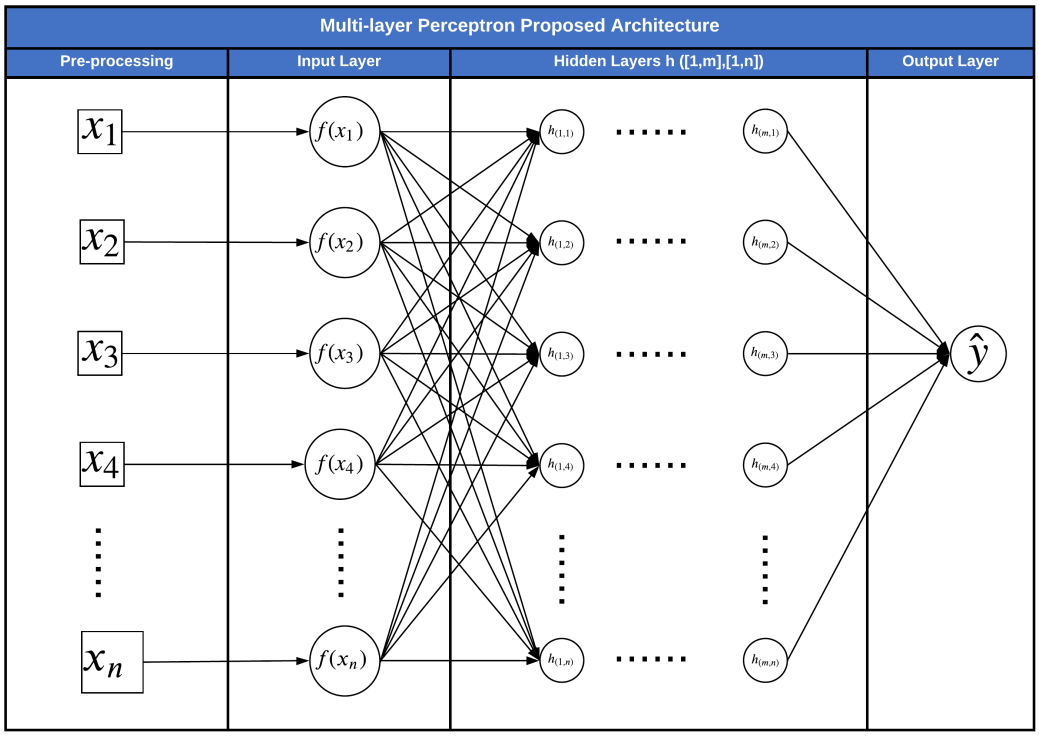


Figure 22- MLP Architecture Diagram. Source: Author.

10.1.3 RECURRENT NEURAL NETWORK

As we can see from [51], RNN using Long Short-Term Memory (LSTM) cells is widely used in Natural Language Processing (NPL) tasks. Specifically Palangi et al. in [51] presented that RNN using LSTM outperforms state of the art models in web document retrieval tasks. Also we can see from [52] that RNN with LSTM achieves good accuracy, high efficiency, and ease-of-use in entity resolution tasks. In addition, Agarwal et al. in their investigation *"A deep network model for paraphrase detection in short text messages"* [55] have proposed a deep network architecture using Convolutional Neural Network (CNN) in conjunction with RNN to identify sentences semantically identical. This approach has tremendous similarity with our investigation in order to achieve entity resolution using just a "title phrase" that describes a product. In addition to the above, we can see another great application of RNN architecture to measurement task of semantic similarity, within the field on NLP. This application proposed by Ye et al. in [56], presents the usage of this architecture to classify in three different classes a given pair of questions. Moreover, RNN's are used in Natural Language Sentence Matching (NLSM), specifically in the task of understanding if a sentence is a paraphrase of another sentence in a given pair. Hunt et al. in [57] presents a very valuable result where RNN architecture outperforms other models in paraphrase identification tasks. Also, we can see from [58], that RNN architecture outperform other state of the art models in "lexicon text analysis", more specifically using POS tagging.

10.1.3.1.1 RNN ARCHITECTURE LSTM

Our RNN architecture is designed to have an input layer equal to the feature matrix size. When using the "Damerau- Levenshtein String-Edit Distance Similarity Matrixes", as the last position represents the label and it will fed to the network as an individual label vector, and the input vector has the following shape (n,1,10), as we will be using a single timestep for each feature. This is due to the fact that a LSTM RNN requires a 3D input array, where the first dimension is the sample size (number of samples), the second dimension represents the timesteps, and last dimension the number of features. When using the "Word2Vec Transformation for "Title" Attributes Matrixes" the input vector has the following shape (n,2,250). We have decided to feed each normalized "Title" vector as a timestep with 250 features each for every sample. Each timestep represents the "Title" of each member of the compared pair. The fifth hundred first (501) position represents the label and it will be provided in a label vector. The following figure illustrates the proposed architecture.

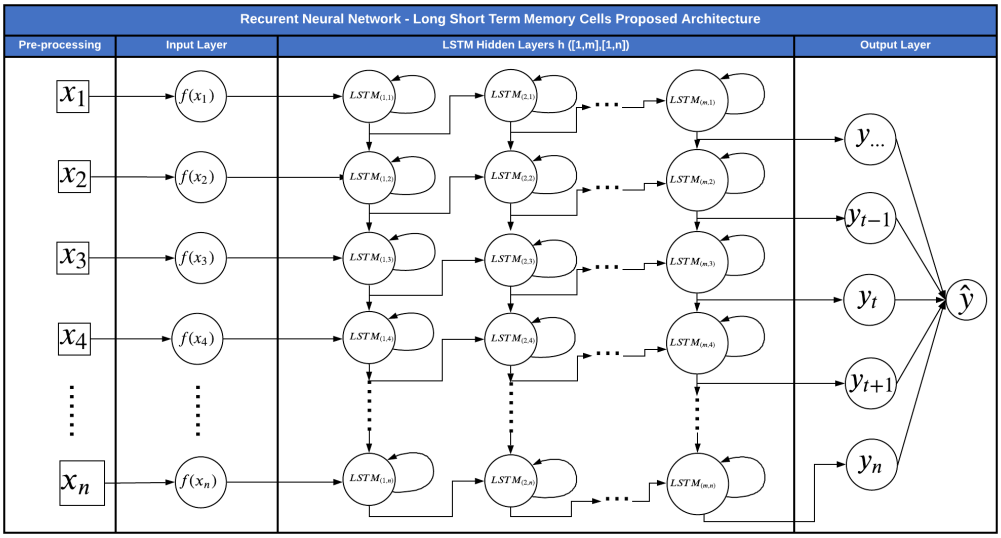


Figure 23- LSTM RNN Architecture Diagram. Source: Author.

10.1.3.1.2 RNN ARCHITECTURE BI-DIRECTIONAL LSTM

For the “Word2Vec Transformation for "Title" Attributes Matrixes”, we will propose a second variation of the LSTM RNN, using the Bi-Directional LSTM RNN approach. With this additional experiment fixture we want to test the applicability of a RNN with LSTM to solve entity resolution proposed by Ebraheem M. et al. in [52]. For this architecture the input vector has the same structure and definition a s the one presented in section 10.1.3.1.1.

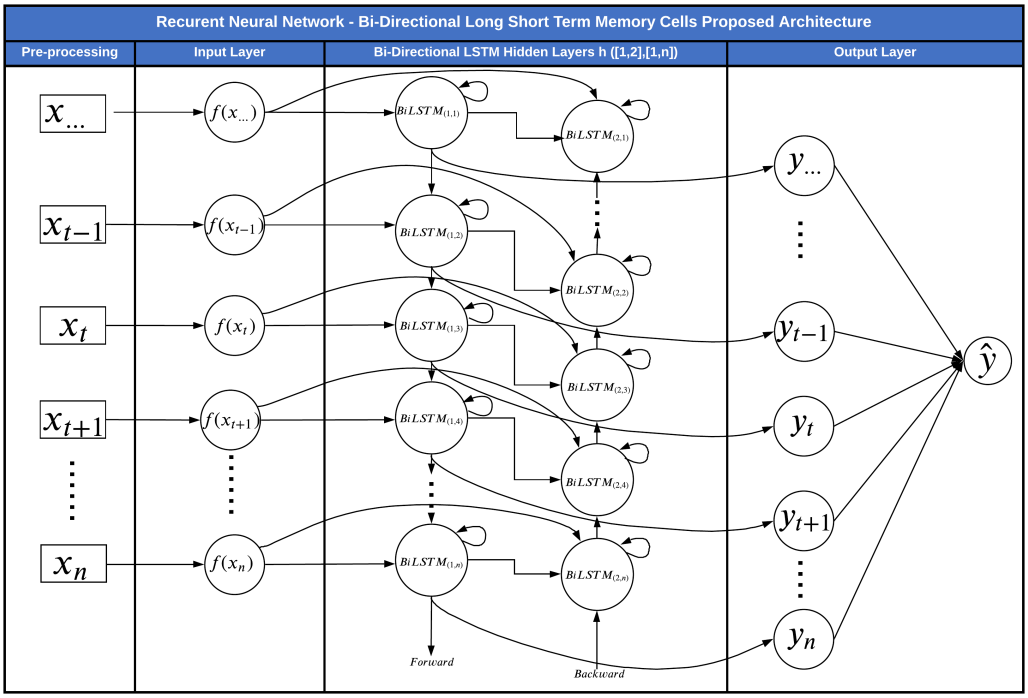


Figure 24- Bi-Directional LSTM RNN Architecture Diagram. Source: Author.

Taking the above into account the sets per each preprocessing strategy are presented in the following table.

Damerau- Levenshtein String-Edit Distance Similarity Matrixes. Preprocessed Data Sets for LSTM Architecture			
Data Set	**Data Set Definition**	**Matrix Numpy Array**	**Array Shape**
TestDataSet	**New Data Set For Testing Four Main Product Categories 30% of Records**	**TestDataSetDL.npy**	**(35564,1,10)**
TrainDataSet	**New Data Set For Training Four Main Product Categories 70% of Records**	**TrainDataSetDL.npy**	**(82984,1,10)**
TrainDataSet90	**New Data Set For Training Hyperparameter Estimation. 90% of Records of TrainDataSet**	**TrainDataSet90DL.npy**	**(74232,1,10)**
ValidationDataSet10	**New Data Set For Validating Hyperparameter Estimation. 10% of Records of TrainDataSet**	**ValidationDataSet10DL.npy**	**(8242,1,10)**

Damerau- Levenshtein String-Edit Distance Similarity Matrixes. Preprocessed Data Sets for LSTM Architecture			
Data Set	**Data Set Definition**	**Matrix Numpy Array**	**Array Shape**
GS Total	**Manual Validated Data Set Provided In Original Data Corpus**[3]**. 4,400 Golden Standard Records.**	**all_gsDataSetDL.npy**	**(4400,1,10)**

Table 18- Damerau- Levenshtein String-Edit Distance Similarity Matrixes. Preprocessed Data Sets for LSTM. Source: Author

Word2Vec Transformation for "Title" Attributes Matrixes. Preprocessed Data Sets			
Data Set	**Data Set Definition**	**Matrix Numpy Array**	**Array Shape**
TestDataSet	**New Data Set For Testing Four Main Product Categories 30% of Records**	**TestDataSetW2V_501.npy**	**(35564,2,250)**
TrainDataSet	**New Data Set For Training Four Main Product Categories 70% of Records**	**TrainDataSetW2V_501.npy**	**(82984,2,250)**
TrainDataSet90	**New Data Set For Training Hyperparameter Estimation. 90% of Records of TrainDataSet**	**TrainDataSet90W2V_501.npy**	**(74232,2,250)**
ValidationDataSet10	**New Data Set For Validating Hyperparameter Estimation. 10% of Records of TrainDataSet**	**ValidationDataSet10W2V_501.npy**	**(8242,2,250)**
GS Total	**Manual Validated Data Set Provided In Original Data Corpus**[3]**. 4,400 Golden Standard Records.**	**all_gsDataSetW2V_501.npy**	**(4400,2,250)**

Table 19- Word2Vec Transformation for "Title" Attributes Matrixes. . Preprocessed Data Sets for LSTM and Bi-Directional LSTM. Source: Author

10.2 HYPERPARAMETERS ESTIMATION

This section addresses the experimental propositions and results of hyperparameters estimation for each architecture setup with each type of preprocessed data matrixes. Therefore we will have two parameter estimations per proposed architecture. The first one will address the MLP architecture with the DL Matrix Data Sets, and the second one will use the W2V_501 Matrix Data Sets with the MLP architecture. The third and fourth hyperparameter estimation experiment will use the RNN architecture and the DL Matrix Data Set and the W2V_501 Matrix Data Sets respectively. As an additional variation of the RNN architecture, we will estimate hyperparameters for the bi-directional LSTM RNN just with the W2V_501 Matrix Data Sets. We will use the TrainDataSet90 data sets, both with the DL and W2V_501 preprocess data, to estimate the hyperparameters. Then the ValidationDataSet10 data sets will be used to measure the hyperparameter selection performance. Each set of experiments will have a define number of estimated parameters and the best performance parameter array will be selected to promote to the train phase of the investigation.

10.2.1 MLP HYPERPARAMETER ESTIMATION

For the MLP hyperparameter estimation we will use a predefined Keras[27] optimization library called Talos[28], which is built on top of TensorFlow[29]. This library has been provided by Mikko Kotila as a freeware with the corresponding MIT License[30]. This library will allow us to estimate up to thirty hyperparameters for our experiments. It will provide aid with the evaluation of each parameter permutation returning a data frame with the results. Then we will select the best model to promote as the selected hyperparameters. For each experiment we will record the performance KPI metrics F1 Score, Precision, Recall, Accuracy, True Negatives, True Positives, False Negatives, and False Positives. With these metrics also a confusion matrix will be presented. The best model will be selected using the highest F1 Score KPI, as it has been stated in the literature review that this metric is much more robust when working with unbalanced data in terms of classification.

10.2.2 MLP ARCHITECTURE – DL SIMILARITY MATRIX DATA SET

Our first experimental setup includes the selection of the MLP architecture and the DL Similarity Matrix Data Sets of preprocess data. What we aim to achieve with this setup is to prove a basic naïve model which uses both the simplest data preprocessing procedure and then the simplest neural network architecture. The results of these preliminary experiment have been already applied in real life applications by the author having good results.

For each estimation experiment a set of parameters dictionary has been configured based in the available estimation parameters provided in Talos Scan[31] library. The dictionary includes the following selected parameters:

Hyperparameters Estimation: Experiment permutations parameter dictionary		
Parameter	**Parameter ID**	**Definition**
Learning Rate	**lr**	**Learning rate input for the "lr" function for the optimizers according to TALOS function. Assuming a default learning rate 1, rescales the learning rate such that learning rates amongst different optimizers are more or less equivalent.**[32]
First Neuron	**first_neuron**	**Fix number of neurons assigned to the network hidden layers. It will**

[27] Keras: Keras is an open-source library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. https://keras.io/

[28] Talos: Hyperparameter Optimization for Keras. https://github.com/autonomio/talos/blob/master/README.md

[29] TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. https://www.tensorflow.org/

[30] Talos License: https://github.com/autonomio/talos/blob/master/LICENSE

[31] Talos Scan Library: https://github.com/autonomio/talos/blob/master/docs/Scan.md

[32] TALOS Library normalizers.py: Assuming a default learning rate 1, rescales the learning rate such that learning rates amongst different optimizers are more or less equivalent.
https://github.com/autonomio/talos/blob/524a9f0de725dc41d8ad47c42384264085ccf3d3/talos/model/normalizers.py

Hyperparameters Estimation: Experiment permutations parameter dictionary		
Parameter	**Parameter ID**	**Definition**
		change upon the 'shpae' definition of the network.
Hidden Layers	**hidden_layers**	**Number fo hidden layers of the network.**
Batch Size	**batch_size**	**Computation is done in batches. This method is designed for performance in large scale inputs. Number of samples per gradient update.**
Epochs	**epochs**	**Number of iterations to which the data set will be processed by the network.**
Dropout	**dropout**	**Parameter at which randomly selected neurons are ignored during training. It will use a rate between 0 and 1.**
Shapes	**shapes**	**Provides the ability to include network shape in experiments. If params dictionary for the round contains float value for params['shapes'] then a linear contraction towards the last_neuron value. The higher the value, the fewer layers it takes to reach lesser than last_neuron. Supports three inbuilt shapes 'brick', 'funnel', and 'triangle'.**
Optimizer	**optimizer**	**Loss function optimizers used in each permutation.**
Losses	**losses**	**The loss function used in each permutation.**
Activation	**activation**	**The neuron activation funtion of the hidden layers.**
Last Activation	**last_activation**	**The neuron activation funtion of the output layer of the network.**

Table 20- Experiment permutations parameter dictionary. Sources: Author, [59]

The first set of hyperparameters to estimate will be the ones related to the MLP network size and shape. This first set of estimations resulted in ninety (90) combinations in total. The combined parameters are the number of hidden layers, and the network shape. For all hyperparameter estimation experiments will combine with three different values of the epochs parameter. The intention is to always understand what is the result of the experiment as you increase in a factor of then the training epochs.

Hyperparameters Estimation: Network architecture shape experiment permutations parameter values		
Parameter	**Parameter ID**	**Permutation Values**
Learning Rate	**lr**	**[0.5]**
First Neuron	**first_neuron**	**[10]**
Hidden Layers	**hidden_layers**	**[1,2,3,4,5,6,7,8,9,10]**
Batch Size	**batch_size**	**[1000]**
Epochs	**epochs**	**[10,100,1000]**
Dropout	**dropout**	**[0.5]**
Shapes	**shapes**	**['funnel','brick',traingle']**
Optimizer	**optimizer**	**[Adam]**
Losses	**losses**	**[binary_crossentropy]**
Activation	**activation**	**[elu]**
Last Activation	**last_activation**	**[sigmoid]**

Table 21- Network architecture shape experiment permutations parameter values. DL Similarity Matrix Preprocessing. Source: Author

The selected metric to review the best fit of the trained and validated model is the F1 Score. In addition we will take into account the confusion matrix result for the top five models, as it is very important to understand for our implementation to minimize false positives, which represent pair records labeled as duplicates or “the same product”, when they are really non-duplicates or “different products”. This approach is taken into account because when selling product offers in a marketplace, merging different offers from different products, will result in sales returns, low performance qualifications, and overhead costs. Below the concepts and terminology of the confusion matrix is presented.

Confusion Matrix Terminology Table			
Symbol	**Name**	**Definition**	**Investigation Definition**
P	**Condition Positive**	**The number of positive cases in the data**	**The number of labeled pairs marked as duplicate products.**
N	**Condition Negative**	**The number of negative cases in the data**	**The number of labeled pairs marked as non-duplicate products.**
TP	**True Positive**	**The number of predicted positive cases that match the actual positive cases**	**The number of predicted duplicated products, that were labled as duplicated products.**
TN	**True Negative**	**The number of predicted negative cases that match the actual negative cases**	**The number of predicted non-duplicated products, that were labled as non-duplicated products.**
FP	**False Positives**	**The number of predicted positive cases that are actual negative cases. Type I Error.**	**The number of predicted duplicated products, that were labled as non-duplicated products.**

Confusion Matrix Terminology Table			
Symbol	**Name**	**Definition**	**Investigation Definition**
FN	**False Negatives**	**The number of predicted negatives cases that are actual positive cases. Type II Error.**	**The number of predicted non-duplicated products, that were labled as duplicated products.**

Table 22- Confusion Matrix Terminology Table. Includes Investigation Localization. Sources: Author, [60], [61], [62]

Confusion Matrix Terminology Performance KPI's Table		
Symbol	**Name**	**Definition**
TPR	**Recall, Sensitivity, Hit Rate, True Positive Rate**	**Ratio between the true positives and the predicted results**
PPV	**Precision, Positive Predicted Value**	**Ratio between the true positives and the actual results**
ACC	**Accuracy**	**Ratio between the sum of the true positive and trunegatives, and the total cases**
F1	**F1 Socre**	**The harmonic mean of precision and recall**

Table 23- Confusion Matrix Terminology Performance KPI's Table. Sources: [60], [61], [62]

$$TPR = \frac{TP}{TP + FN}$$

Figure 25- Recall equation. Sources: [60], [61], [62]

$$PPV = \frac{TP}{TP + FP}$$

Figure 26- Precision equation. Sources: [60], [61], [62]

$$ACC = \frac{TP + TN}{P + N}$$

Figure 27- Accuracy equation. Sources: [60], [61], [62]

$$F1 = 2 \times \frac{PPV \times TPR}{PPV + TPR}$$

Figure 28- Accuracy equation. Sources: [60], [61], [62]

The following chart presents the result of the 90 experimental iterations, grouped by the number of hidden layers. Each color of the bars represent the number of epochs of the experiment. The highest F1 Score is 0.73, present in 5 records.

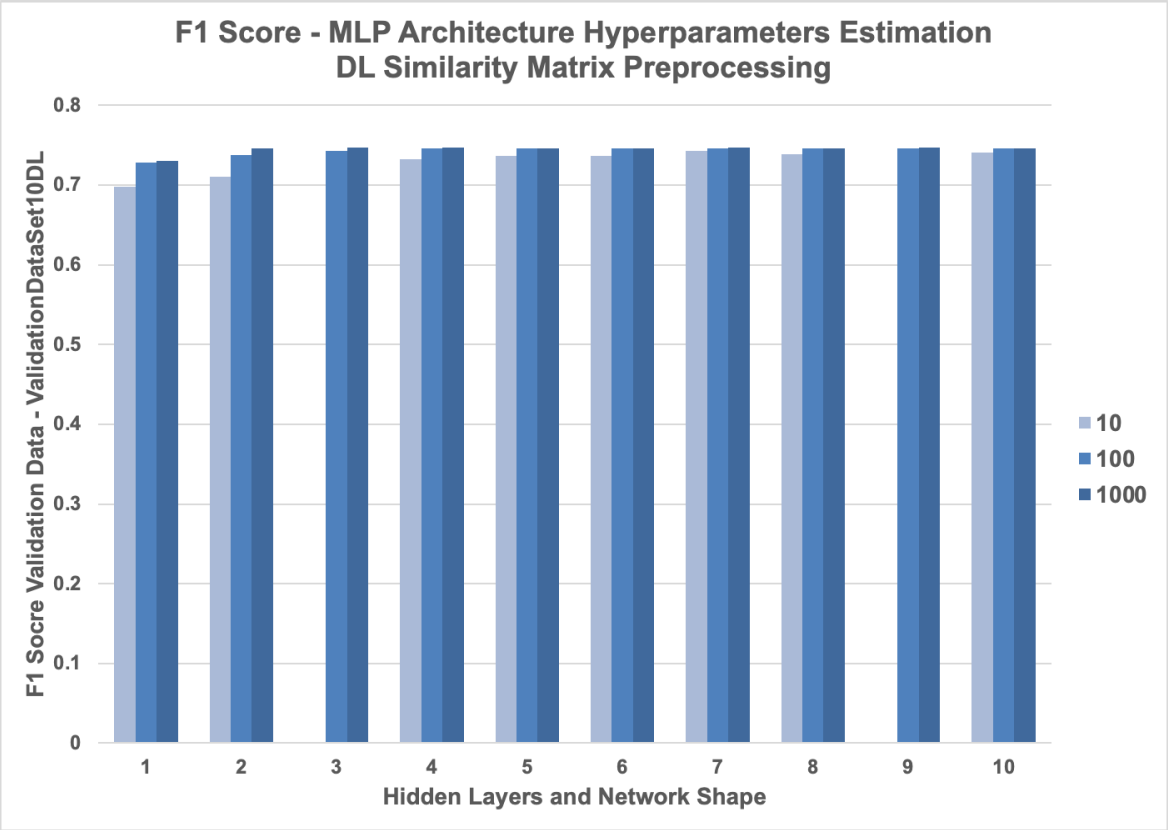


Chart 4- F1 Score per Hidden Layers – MLP Architecture. DL Similarity Matrix Preprocessing. Source: Author

The following chart presents the result of the 90 experimental iterations, grouped by the architecture shape parameter. Each color of the bars represent the number of epochs of the experiment. The highest F1 Score is 0.73.

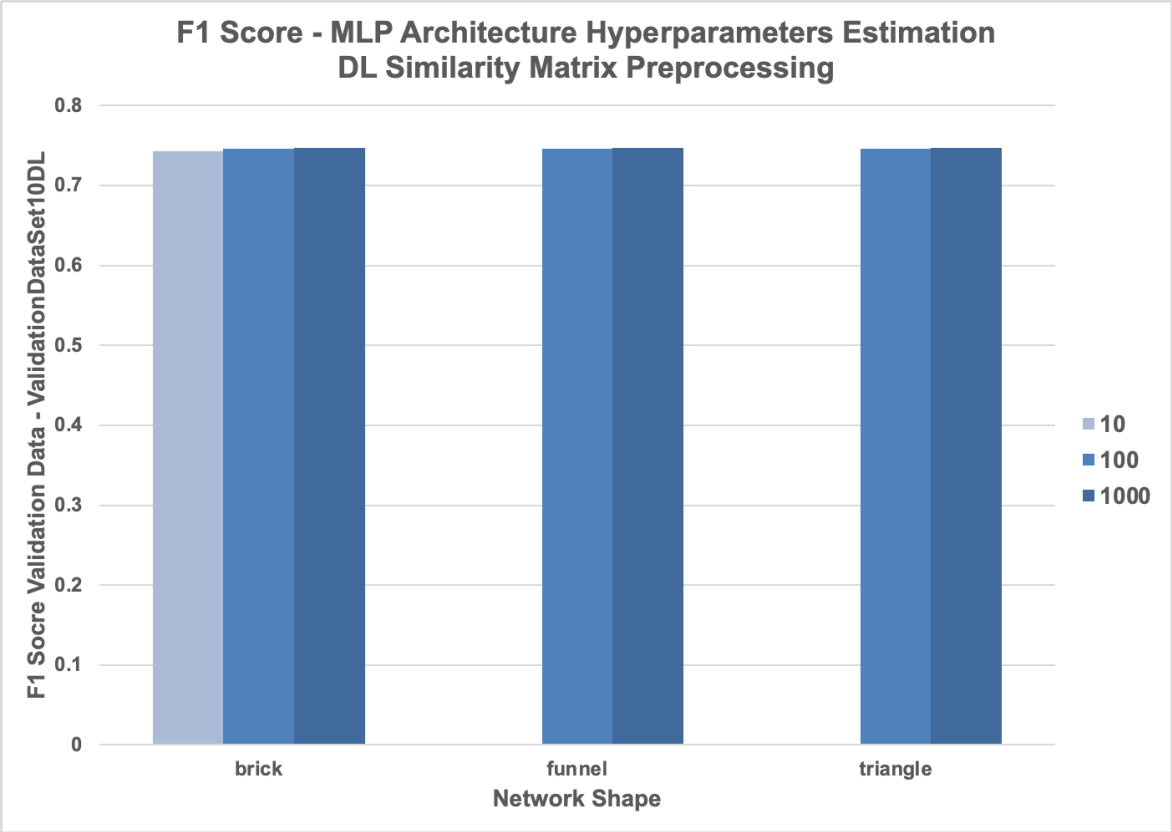


Chart 5- F1 Score per Network Shape – MLP Architecture. DL Similarity Matrix Preprocessing. Source: Author

The following chart presents the result of the 90 experimental iterations, taking into account the number of hidden layers and the network shape combined. Each color of the bars represent the number of epochs of the experiment. The highest F1 Score is 0.73, and it was presented in five different architecture setups. For these architecture setups, the confusion matrix with its KPI's will be presented in order to choose the best architecture hyperparameters.

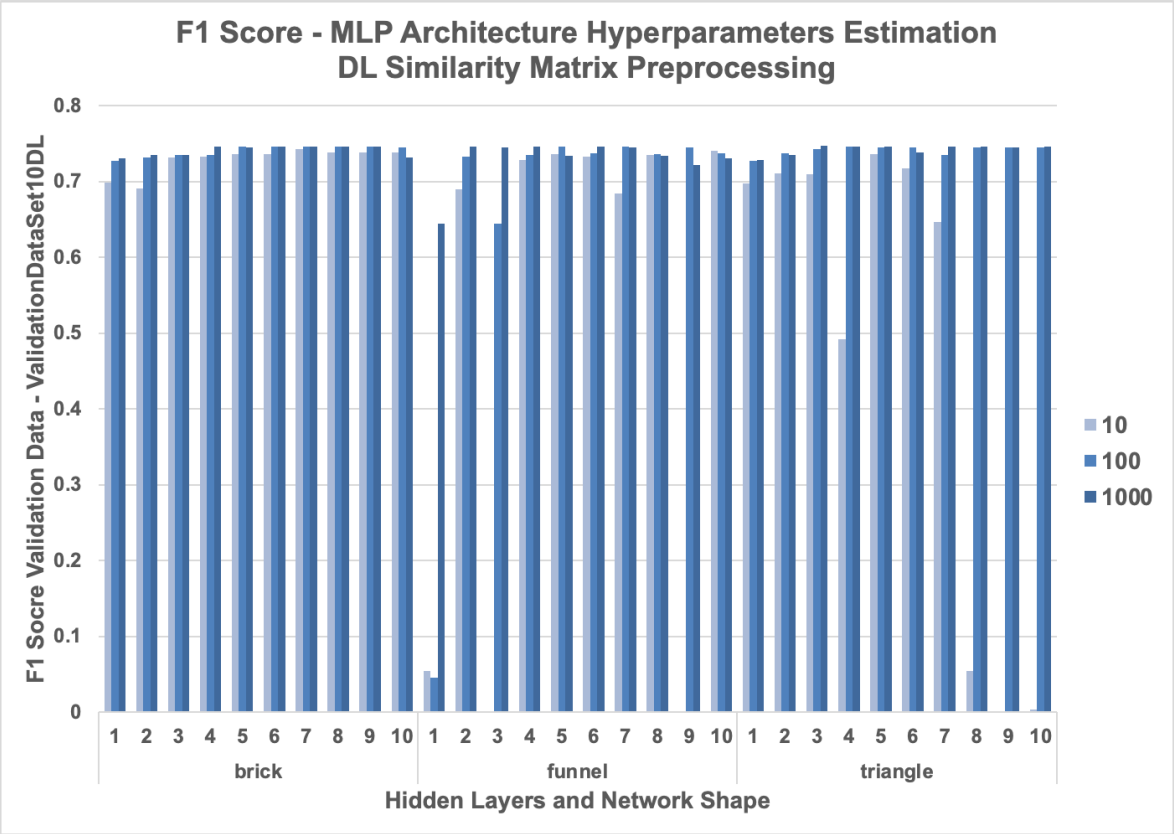


Chart 6- F1 Score per Hidden Layers and Network Shape – MLP Architecture. DL Similarity Matrix Preprocessing. Source: Author

	Experiment Iteration		68	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.58992487
	5946	33	PPV	0.97587717
	FN	TP	ACC	0.88340209
	928	1335	F1	**0.735334619**

Hidden Layers	Epochs	Network Shape
3	1000	brick

	Experiment Iteration		65	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.58992487
	5945	34	PPV	0.97516435
	FN	TP	ACC	0.88328076
	928	1335	F1	**0.735132159**

Hidden Layers	Epochs	Network Shape
2	1000	brick

Table 24- Confusion Matrix MLP Architecture Hyperparameters Estimation Results DL Similarity Matrix Preprocessing 68, 65. Source: Author

	Experiment Iteration		66	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.58992487
	5945	34	PPV	0.97516435
	FN	TP	ACC	0.88328076
	928	1335	F1	**0.735132159**
Hidden Layers	Epochs	Network Shape		
2	1000	triangle		

	Experiment Iteration		51	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.58992487
	5943	36	PPV	0.97374177
	FN	TP	ACC	0.8830381
	928	1335	F1	**0.734727573**
Hidden Layers	Epochs	Network Shape		
7	100	triangle		

Table 25- Confusion Matrix MLP Architecture Hyperparameters Estimation Results DL Similarity Matrix Preprocessing 66, 51. Source: Author

	Experiment Iteration		41	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.58992487
	5942	37	PPV	0.97303206
	FN	TP	ACC	0.88291677
	928	1335	F1	**0.734525447**
Hidden Layers	Epochs	Network Shape		
4	100	brick		

Table 26- Confusion Matrix MLP Architecture Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 41. Source: Author

As part of this analysis we can conclude that even though the F1 Score KPI is pretty much the same in all the hyperparameter setups, the false positives variate in just 3 points between experiment 68 and 41. The computational cost and the network complexity between each these two experiments is also significatively high. While experiment 68 took 986.09 seconds to iterate using 451 trainable parameters, experiment 41 took 120.34 seconds to iterate, using 561 trainable parameters.

Therefore the selected hyperparameters for the MLP Architecture are the ones in experiment 41. The table below shows the selected parameters, followed by the model summary.

Hyperparameters Estimation Selection Experiment 41: Network architecture shape experiment permutations parameter values		
Parameter	**Parameter ID**	**Permutation Values**
Learning Rate	**lr**	**[0.5]**
First Neuron	**first_neuron**	**[10]**
Hidden Layers	**hidden_layers**	**[4]**
Batch Size	**batch_size**	**[1000]**
Epochs	**epochs**	**[100]**
Dropout	**dropout**	**[0.5]**
Shapes	**shapes**	**['brick']**
Optimizer	**optimizer**	**[Adam]**
Losses	**losses**	**[binary_crossentropy]**
Activation	**activation**	**[elu]**
Last Activation	**last_activation**	**[sigmoid]**

Table 27- Hyperparameters Estimation Selection Experiment 41 MLP Architecture. DL Similarity Matrix Preprocessing. Source: Author

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_298 (Dense)            (None, 10)                110
_________________________________________________________________
dropout_248 (Dropout)        (None, 10)                0
_________________________________________________________________
dense_299 (Dense)            (None, 10)                110
_________________________________________________________________
dropout_249 (Dropout)        (None, 10)                0
_________________________________________________________________
dense_300 (Dense)            (None, 10)                110
_________________________________________________________________
dropout_250 (Dropout)        (None, 10)                0
_________________________________________________________________
dense_301 (Dense)            (None, 10)                110
_________________________________________________________________
dropout_251 (Dropout)        (None, 10)                0
_________________________________________________________________
dense_302 (Dense)            (None, 10)                110
_________________________________________________________________
dropout_252 (Dropout)        (None, 10)                0
_________________________________________________________________
dense_303 (Dense)            (None, 1)                 11
=================================================================
Total params: 561
Trainable params: 561
Non-trainable params: 0
```

Figure 29- MLP Architecture Sequential Model Experiment 41. DL Similarity Matrix Preprocessing. Source: Author

Now after selecting the MLP architecture hyperparameters, the optimization hyperparameters will be estimated. An experimental setup with 924 combinations has been constructed. The idea of this second round of hyperparameter estimations is to focus on the optimization hyperparameters. These are:

- Learning Rate (lr)
- Dropout (dropout)
- Optimizer (optimizer)
- Losses (losses)
- Activation Function (activation): This activation function is used in the hidden layers only. The output layer will always use the sigmoid activation function due to the nature of our experiment.

The details of the permutable hyperparameters are presented in the table below.

Hyperparameters Estimation: Network optimization experiment permutations hyperparameter values		
Parameter	**Parameter ID**	**Permutation Values**
Learning Rate	**lr**	**[0.5, 2.75]**
First Neuron	**first_neuron**	**[10]**
Hidden Layers	**hidden_layers**	**[4]**
Batch Size	**batch_size**	**[1000]**
Epochs	**epochs**	**[100]**
Dropout	**dropout**	**[0, 0.1666, 0.333]**
Shapes	**shapes**	**['brick']**
Optimizer	**optimizer**	**[SGD, RMSprop, Adam, Adadelta, Adagrad, Adamax, Nadam]**
Losses	**losses**	**['binary_crossentropy', 'logcosh']**
Activation	**activation**	**['relu', 'softmax', 'softplus', 'softsign', 'tanh', 'selu', 'elu']**
Last Activation	**last_activation**	**[sigmoid]**

Table 28- Network optimization experiment permutations parameter values. DL Similarity Matrix Preprocessing. Source: Author

After the experiment execution we marked the four top results using the F1 Score KPI. The following charts present the result analysis per each hyperparameter combination. As presented below, the F1 Score of the best model is just above 0.8. This means an increase of performance of 10.11% after setting the architecture hyperparameters.

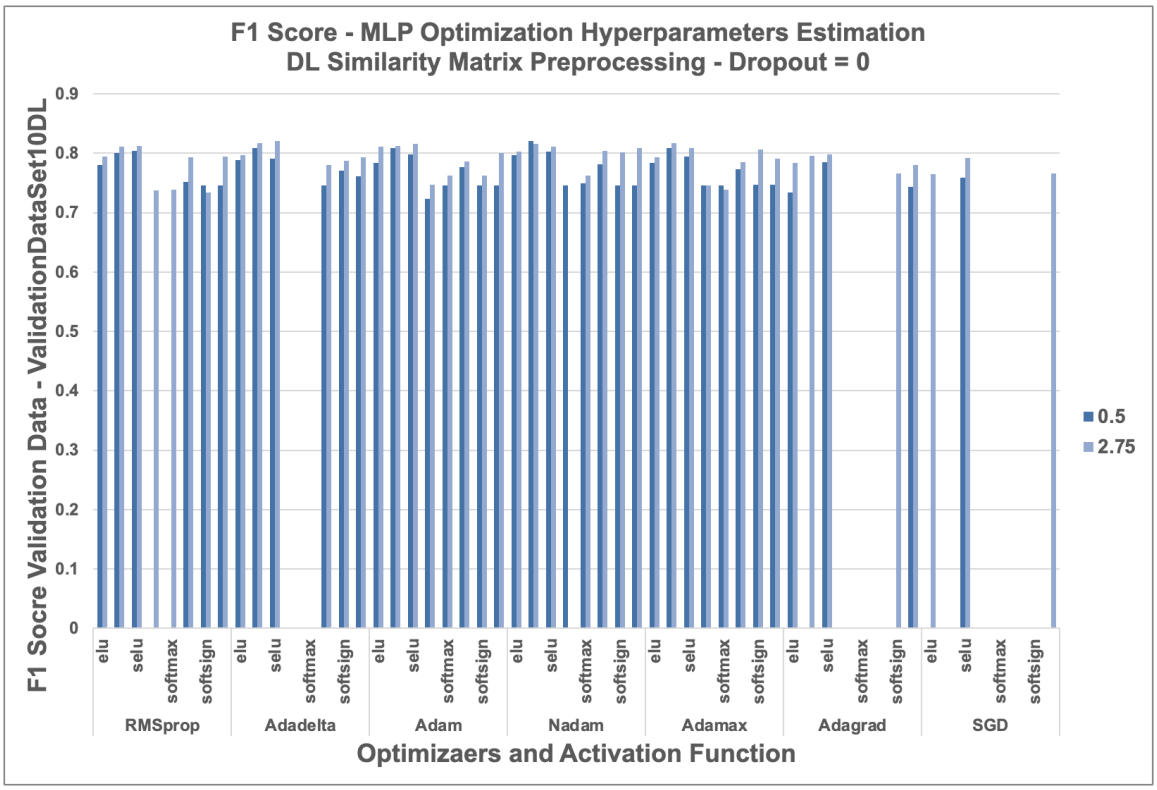


Chart 7- F1 Score per Optimizers and Activation Function. Dropout = 0 – MLP Optimizers. DL Similarity Matrix Preprocessing. Source: Author

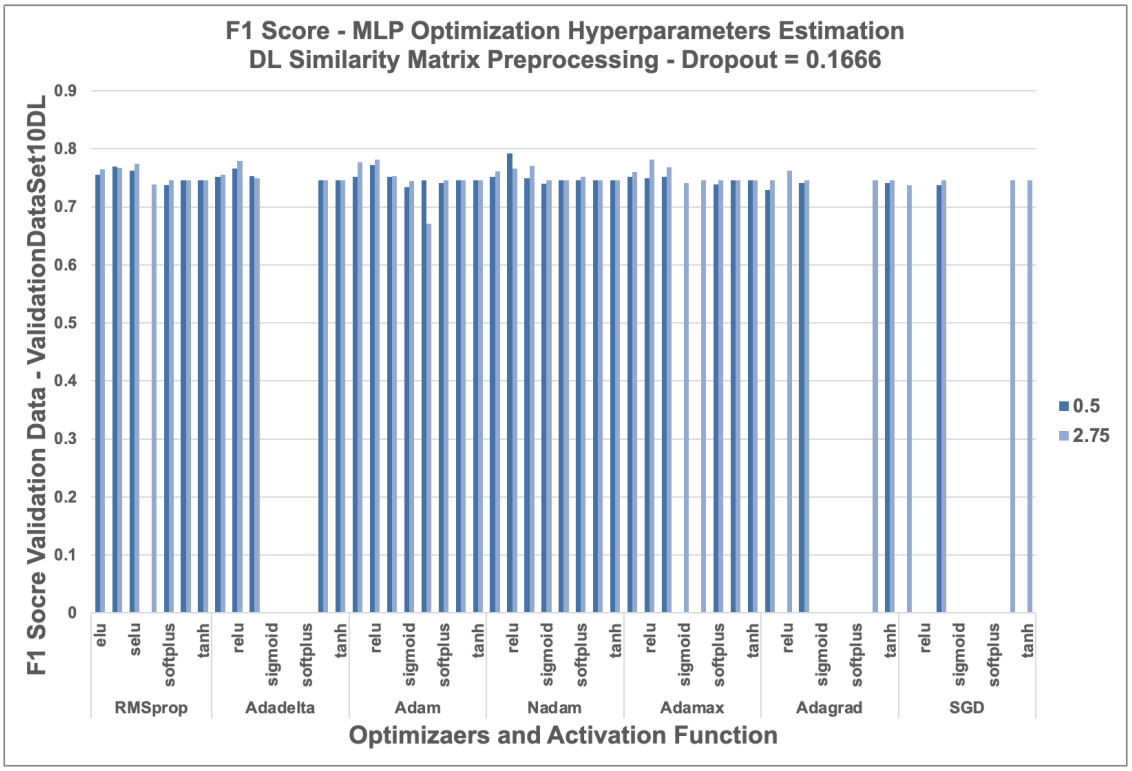


Chart 8- F1 Score per Optimizers and Activation Function. Dropout = 0.1666 – MLP Optimizers. DL Similarity Matrix Preprocessing. Source: Author

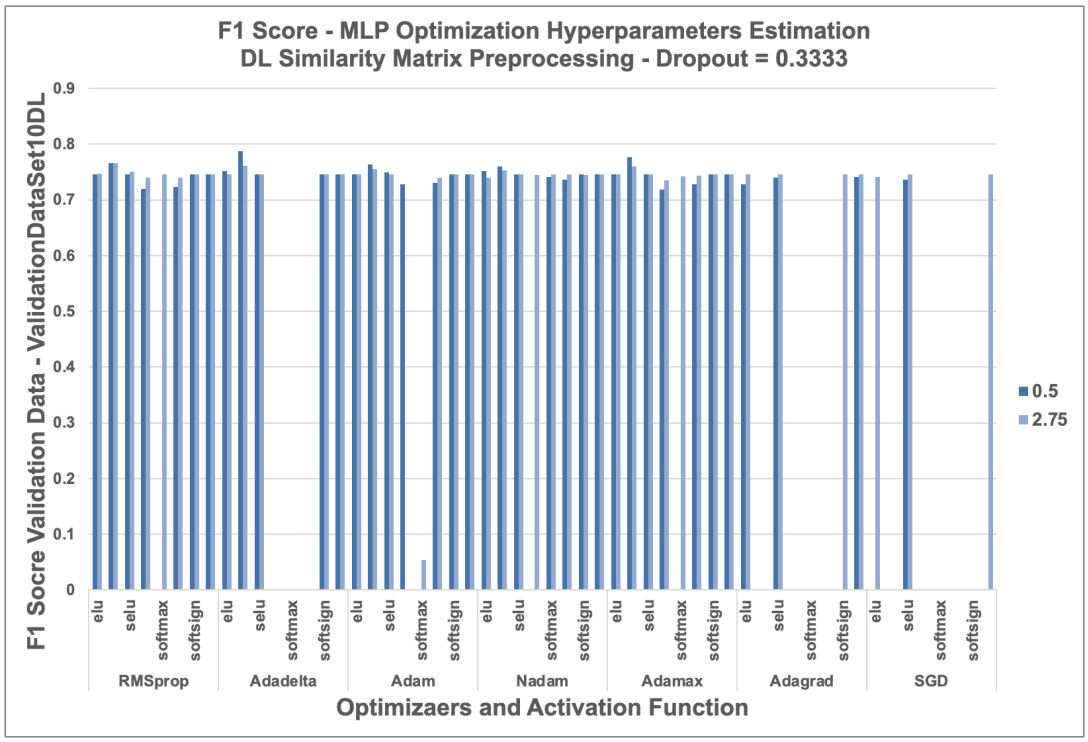


Chart 9- F1 Score per Optimizers and Activation Function. Dropout = 0.3333 – MLP Optimizers. DL Similarity Matrix Preprocessing. Source: Author

As presented previously, it is also very important to understand and analyze the confusion matrix and all the performance KPI's, specially the false positives, as we stated that it is also desired to minimize this KPI. The following tables present the confusion matrixes and KPI's for the top five (5) iteration results.

	Experiment Iteration		770	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.68979234
	5943	36	PPV	0.97745776
	FN	TP	ACC	0.91045862
	702	1561	F1	**0.80880829**
Dropout	**Optimizer**	**Learning Rate**	**Losses**	**Activation Function**
0	Nadam	2.75	binary_crossentropy	selu

Table 29- Confusion Matrix MLP Optimization Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 770. Source: Author

	Experiment Iteration		767	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.67476803
	5947	32	PPV	0.97947401
	FN	TP	ACC	0.90681875
	736	1527	F1	**0.79905808**
Dropout	Optimizer	Learning Rate	Losses	Activation Function
0	Adadelta	2.75	binary_ crossentropy	selu

	Experiment Iteration		2	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.68139637
	5944	35	PPV	0.97780597
	FN	TP	ACC	0.90827471
	721	1542	F1	**0.803125**
Dropout	Optimizer	Learning Rate	Losses	Activation Function
0	Adam	2.75	logcosh	selu

Table 30- Confusion Matrix MLP Optimization Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 767, 2. Source: Author

	Experiment Iteration		1	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.67432612
	5939	40	PPV	0.9744572
	FN	TP	ACC	0.90572679
	737	1526	F1	**0.79707495**
Dropout	Optimizer	Learning Rate	Losses	Activation Function
0	RMSprop	2.75	logcosh	selu

	Experiment Iteration		758	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.66195315
	5942	37	PPV	0.97589576
	FN	TP	ACC	0.90269351
	765	1498	F1	**0.78883623**
Dropout	Optimizer	Learning Rate	Losses	Activation Function
0	RMSprop	0.5	binary crossentropy	selu

Table 31- Confusion Matrix MLP Optimization Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 1, 758. Source: Author

From the above, the best model is 770. This one has provided the best KPI's and has minimized type I and type II error in the confusion matrix, increasing up to 0.8088 the F1 Score, and 10.11% the performance of the MLP model. As a result we have concluded the hyperparameter estimation for this preprocessing method. The table below summarize the selected hyperparameters to promote to the next phase of the investigation.

Hyperparameters Selection: DL Similarity Preprocessing - MLP Network		
Parameter	**Parameter ID**	**Combination Values**
Learning Rate	**lr**	**[2.75]**
First Neuron	**first_neuron**	**[10]**
Hidden Layers	**hidden_layers**	**[4]**
Batch Size	**batch_size**	**[1000]**
Epochs	**epochs**	**[100]**
Dropout	**dropout**	**[0]**
Shapes	**shapes**	**['brick']**
Optimizer	**optimizer**	**[Nadam]**
Losses	**losses**	**['binary_crossentropy']**
Activation	**activation**	**['selu']**
Last Activation	**last_activation**	**[sigmoid]**

Table 32- Hyperparameters Selection DL Similarity Matrix Preprocessing – MLP Network. Source: Author

10.2.3 MLP ARCHITECTURE – W2V_501 MATRIX DATA SET

Our second experimental setup includes the selection of the MLP architecture and the Word2Vec 501 Data Sets of preprocess data. What we aim to achieve with this setup is to prove a model which uses just the “title” preprocessed data as a single argument of the pair record. The intention is to test the hypothesis in which a product offer in a marketplace can be de-duplicated only by analyzing its title. To achieve this, we have used the more complex data processing methodology where we used the Word2Vec algorithm and transforming the pair record into a 250 position vector for each title attribute, resulting in a vector of size 500 per each pair record. Also, we will use the MLP architecture in order to see how does the preprocessing data procedure affects the final result.

For each estimation experiment a set of parameters dictionary has been configured based in the available estimation parameters provided in Talos Scan library. The dictionary includes the following selected parameters:

Hyperparameters Estimation: Experiment permutations parameter dictionary		
Parameter	**Parameter ID**	**Definition**
Learning Rate	**lr**	**Learning rate input for the "lr" function for the optimizers according to TALOS function. Assuming a default learning rate 1, rescales the learning rate such that learning rates amongst different optimizers are more or less equivalent.**[33]
First Neuron	**first_neuron**	**Fix number of neurons assigned to the network hidden layers. It will change upon the 'shpae' definition of the network.**
Hidden Layers	**hidden_layers**	**Number fo hidden layers of the network.**
Batch Size	**batch_size**	**Computation is done in batches. This method is designed for performance in large scale inputs. Number of samples per gradient update.**
Epochs	**epochs**	**Number of iterations to which the data set will be processed by the network.**
Dropout	**dropout**	**Parameter at which randomly selected neurons are ignored during training. It will use a rate between 0 and 1.**
Shapes	**shapes**	**Provides the ability to include network shape in experiments. If params dictionary for the round contains float value for params['shapes'] then a linear contraction towards the last_neuron value. The higher the value, the fewer layers it takes to reach lesser than last_neuron. Supports three inbuilt shapes 'brick', 'funnel', and 'triangle'.**
Optimizer	**optimizer**	**Loss function optimizers used in each permutation.**

[33] TALOS Library normalizers.py: Assuming a default learning rate 1, rescales the learning rate such that learning rates amongst different optimizers are more or less equivalent.
https://github.com/autonomio/talos/blob/524a9f0de725dc41d8ad47c42384264085ccf3d3/talos/model/normalizers.py

Hyperparameters Estimation: Experiment permutations parameter dictionary		
Parameter	**Parameter ID**	**Definition**
Losses	**losses**	**The loss function used in each permutation.**
Activation	**activation**	**The neuron activation funtion of the hidden layers.**
Last Activation	**last_activation**	**The neuron activation funtion of the output layer of the network.**

Table 33- Experiment permutations parameter dictionary. Sources: Author, [59]

The first set of hyperparameters to estimate will be the ones related to the MLP network size and shape. This first set of estimations resulted in thirty (30) combinations in total. The combined parameters are the number of hidden layers, and the network shape. For all hyperparameter estimation experiments will use with three different values of the epochs parameter. The intention is to always understand what is the result of the experiment as you increase in a factor of then the training epochs.

Hyperparameters Estimation: Network architecture shape experiment permutations parameter values		
Parameter	**Parameter ID**	**Permutation Values**
Learning Rate	**lr**	**[0.5]**
First Neuron	**first_neuron**	**[500]**
Hidden Layers	**hidden_layers**	**[1,2,3,4,5,8,10]**
Batch Size	**batch_size**	**[1000]**
Epochs	**epochs**	**[10,100,1000]**
Dropout	**dropout**	**[0.5]**
Shapes	**shapes**	**['funnel','brick',traingle']**
Optimizer	**optimizer**	**[Adam]**
Losses	**losses**	**[binary_crossentropy]**
Activation	**activation**	**[elu]**
Last Activation	**last_activation**	**[sigmoid]**

Table 34- Network architecture shape experiment permutations parameter values. W2V 501 Preprocessing. Source: Author

The selected metric to review the best fit of the trained and validated model is the F1 Score. In addition we will take into account the confusion matrix result for the top five models, as it is very important to understand for our implementation to minimize false positives, which represent pair records labeled as duplicates or “the same product”, when they are really non-duplicates or “different products”. This approach is taken into account because when selling product offers in a marketplace, merging different offers from different products, will result in sales returns, low performance qualifications, and overhead costs. To review details on confusion matrix definition and performance KPI’s, please review section 10.2.2.

The following chart presents the result of the 36 experimental iterations, grouped by the number of hidden layers. Each color of the bars represent the number of epochs of the experiment.

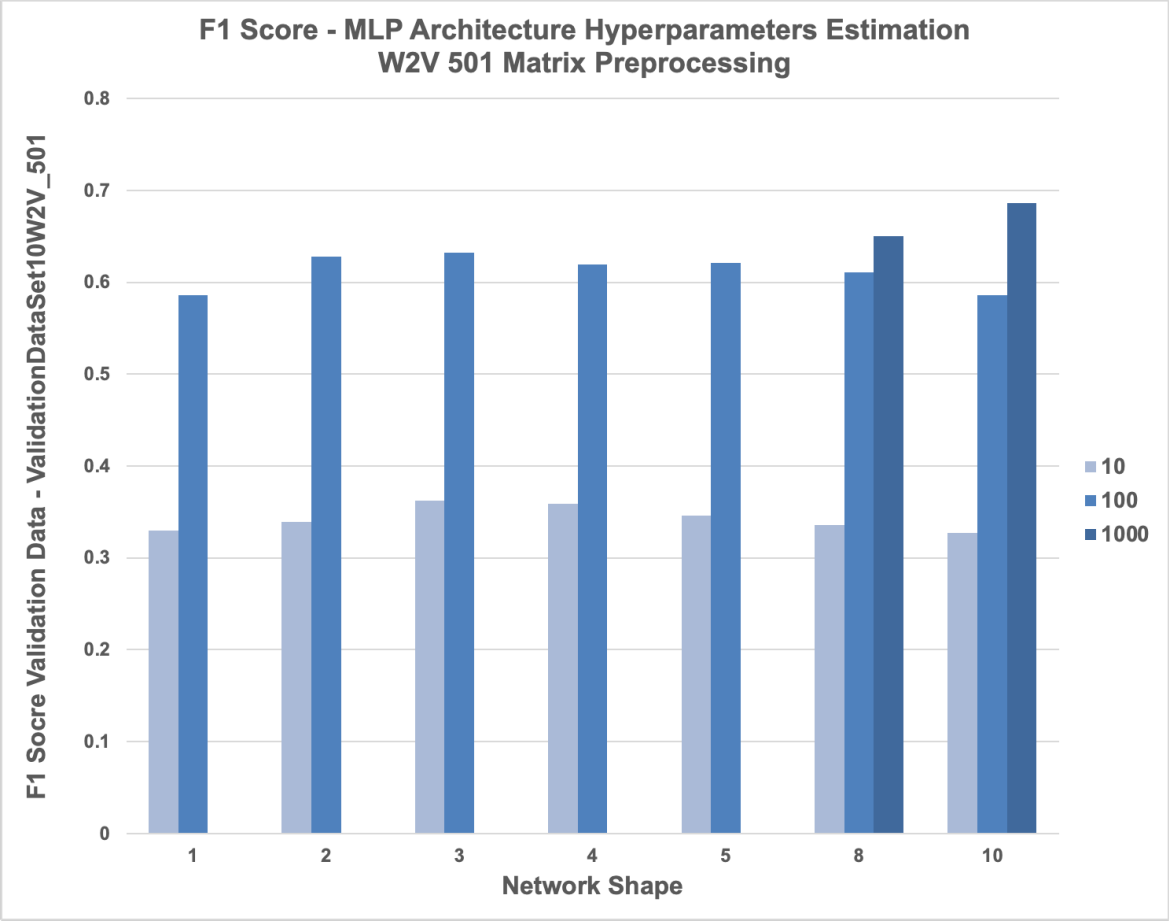


Chart 10- F1 Score per Hidden Layers – MLP Architecture. W2V 501 Preprocessing. Source: Author

The following chart presents the result of the 36 experimental iterations, grouped by the architecture shape parameter. Each color of the bars represent the number of epochs of the experiment.

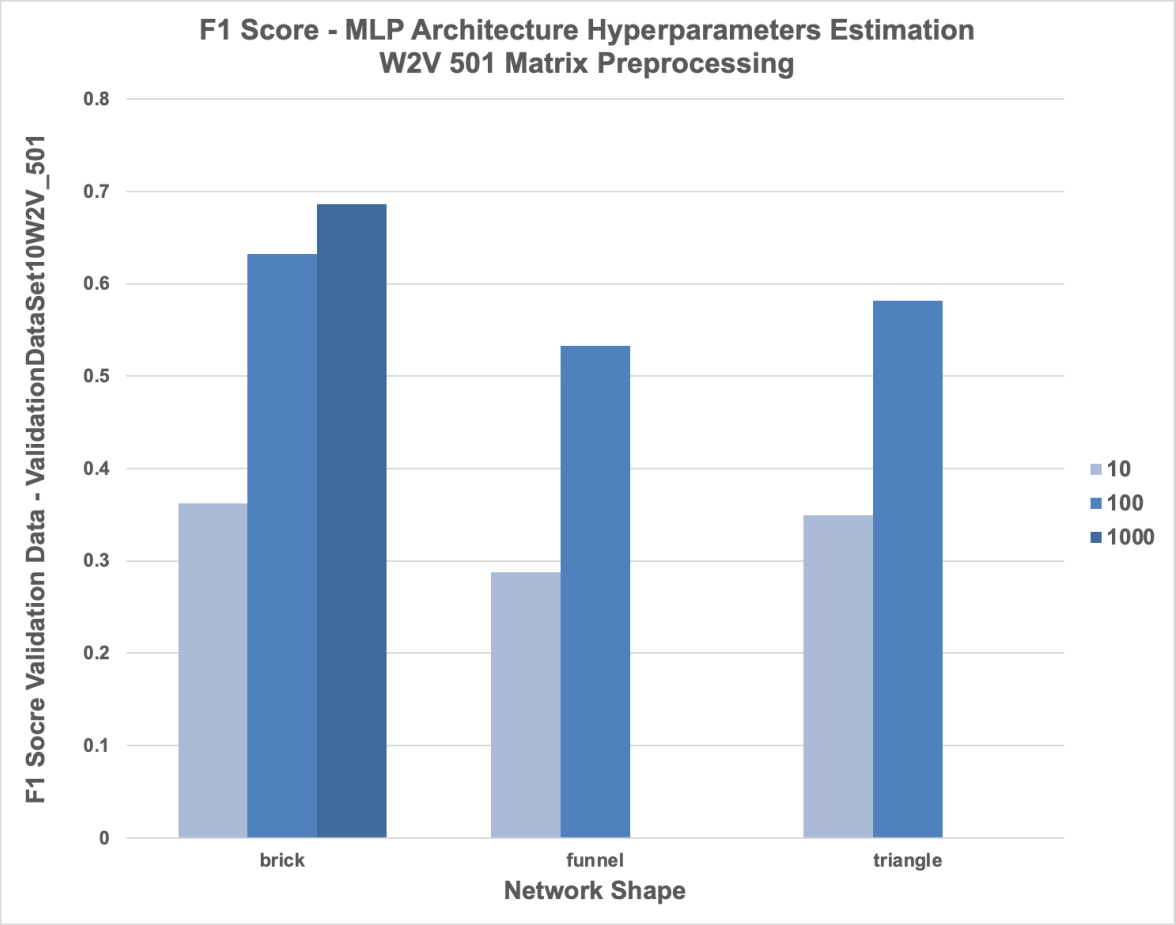


Chart 11- F1 Score per Network Shape – MLP Architecture. W2V 501 Preprocessing. Source: Author

The following chart presents the result of the 36 experimental iterations, taking into account the number of hidden layers and the network shape combined. Each color of the bars represent the number of epochs of the experiment. The highest F1 Score is 0.6820. For the top five (5) architecture setups, according to the F1 score KPI, the confusion matrix with its KPI's will be presented in order to choose the best architecture hyperparameters.

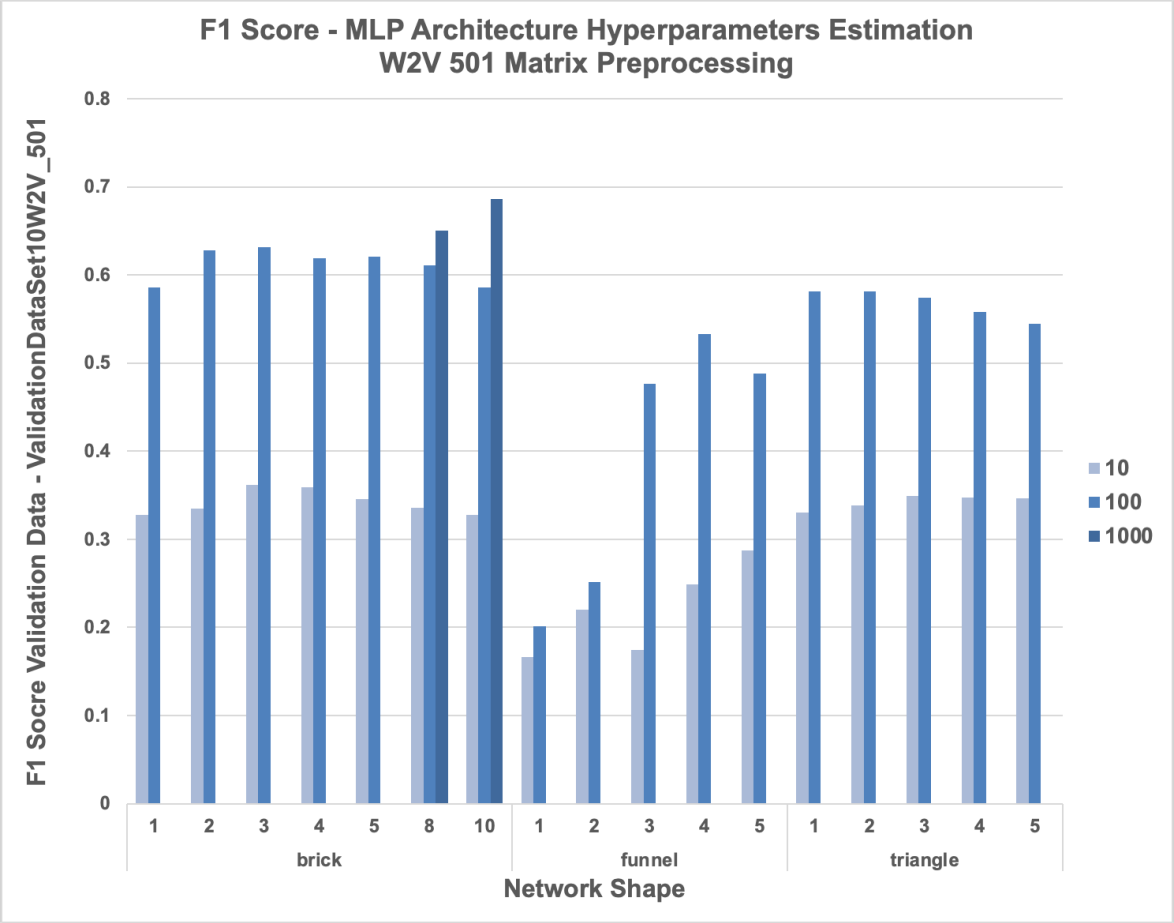


Chart 12- F1 Score per Network Shape – MLP Architecture. W2V 501Preprocessing. Source: Author

	Experiment Iteration		32	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.614670813
	5554	425	PPV	0.765969157
	FN	TP	ACC	0.842635274
	872	1391	F1	**0.682029909**
Hidden Layers	Epochs	Network Shape		
10	1000	brick		

	Experiment Iteration		31	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.529385746
	5693	286	PPV	0.80727762
	FN	TP	ACC	0.836083472
	1065	1198	F1	**0.639444889**
Hidden Layers	Epochs	Network Shape		
8	1000	brick		

	Experiment Iteration		26	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.517896593
	5626	353	PPV	0.768524587
	FN	TP	ACC	0.824799776
	1091	1172	F1	**0.618796199**
Hidden Layers	Epochs	Network Shape		
4	100	brick		

	Experiment Iteration		23	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.502430379
	5677	302	PPV	0.790132046
	FN	TP	ACC	0.826741099
	1126	1137	F1	**0.614262561**
Hidden Layers	Epochs	Network Shape		
3	100	brick		

	Experiment Iteration		20	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.521431744
	5577	402	PPV	0.745891273
	FN	TP	ACC	0.819825292
	1083	1180	F1	**0.613784135**
Hidden Layers	Epochs	Network Shape		
2	100	brick		

Table 35- Confusion Matrix MLP Architecture Hyperparameters Estimation Results W2V 501 32, 31, 26, 23, 20. Source: Author

As part of this analysis we can conclude that even though the F1 Score KPI oscillates between 0.61 and 0.68, even after increasing the training epochs and the network size in terms of hidden layers. To move forward we will select experimental array number 32 that has the following architecture parameters.

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_32 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_29 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_33 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_30 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_34 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_31 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_35 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_32 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_36 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_33 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_37 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_34 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_38 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_35 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_39 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_36 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_40 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_37 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_41 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_38 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_42 (Dense)             (None, 500)               250500
_________________________________________________________________
dropout_39 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_43 (Dense)             (None, 1)                 501
=================================================================
Total params: 2,756,001
Trainable params: 2,756,001
Non-trainable params: 0
```

Figure 30- MLP Architecture Sequential Model Experiment 32. Word2Vec Matrix Preprocessing. Source: Author

Hyperparameters Estimation Selection Experiment 32: Network architecture shape experiment permutations parameter values		
Parameter	**Parameter ID**	**Permutation Values**
Learning Rate	**lr**	**[0.5]**
First Neuron	**first_neuron**	**[500]**
Hidden Layers	**hidden_layers**	**[10]**
Batch Size	**batch_size**	**[1000]**
Epochs	**epochs**	**[1000]**
Dropout	**dropout**	**[0.5]**
Shapes	**shapes**	**['brick']**
Optimizer	**optimizer**	**[Adam]**
Losses	**losses**	**[binary_crossentropy]**
Activation	**activation**	**[elu]**
Last Activation	**last_activation**	**[sigmoid]**

Table 36- Hyperparameters Estimation Selection Experiment 41 MLP Architecture Word2Vec Matrix Preprocessing. Source: Author

Now after selecting the MLP architecture hyperparameters, the optimization hyperparameters will be estimated. An experimental setup with 63 combinations has been constructed. The idea of this second round of hyperparameter estimations is to focus on the optimization hyperparameters. These are:

- Learning Rate (lr)
- Dropout (dropout)
- Optimizer (optimizer)
- Losses (losses)
- Activation Function (activation): This activation function is used in the hidden layers only. The output layer will always use the sigmoid activation function due to the nature of our experiment.

The details of the combinations hyperparameters are presented in the table below.

Hyperparameters Estimation: Network optimization experiment permutations hyperparameter values		
Parameter	**Parameter ID**	**Permutation Values**
Learning Rate	**lr**	**[0.5, 2.75]**
First Neuron	**first_neuron**	**[500]**

Hyperparameters Estimation: Network optimization experiment permutations hyperparameter values		
Parameter	**Parameter ID**	**Permutation Values**
Hidden Layers	**hidden_layers**	**[10]**
Batch Size	**batch_size**	**[1000]**
Epochs	**epochs**	**[1000]**
Dropout	**dropout**	**[0, 0.1666, 0.333]**
Shapes	**shapes**	**['brick']**
Optimizer	**optimizer**	**[SGD, RMSprop, Adam, Adadelta, Adagrad, Adamax, Nadam]**
Losses	**losses**	**['binary_crossentropy', 'logcosh']**
Activation	**activation**	**['relu', 'softmax', 'softplus', 'softsign', 'tanh', 'selu', 'elu']**
Last Activation	**last_activation**	**[sigmoid]**

Table 37- Network optimization experiment permutations parameter values. Word2Vec 501 Matrix Preprocessing. Source: Author

As presented in the following charts, the best experimental result of the 63 experiments achieved a F1 Score of 0.8087. This means a 18.58% of improvement in the performance metric. Each chart presents the results for a specific Dropout value with the combination of Optimizers and Activation Functions, and each series represents the experiment Learning Rate.

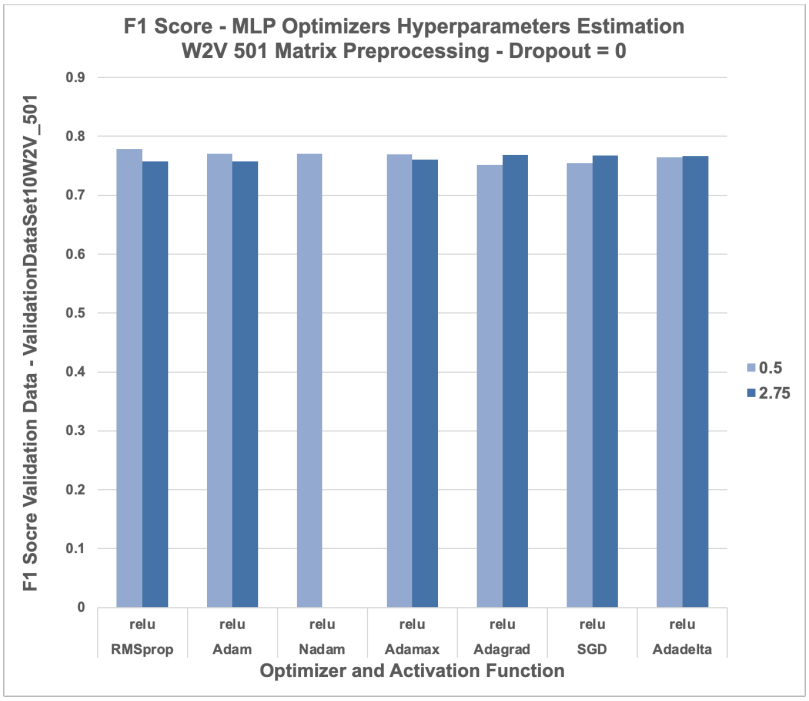


Chart 13- F1 Score per Optimizer and Activation Functions. Dropout = 0 – MLP Architecture. W2V 501Preprocessing. Source: Author

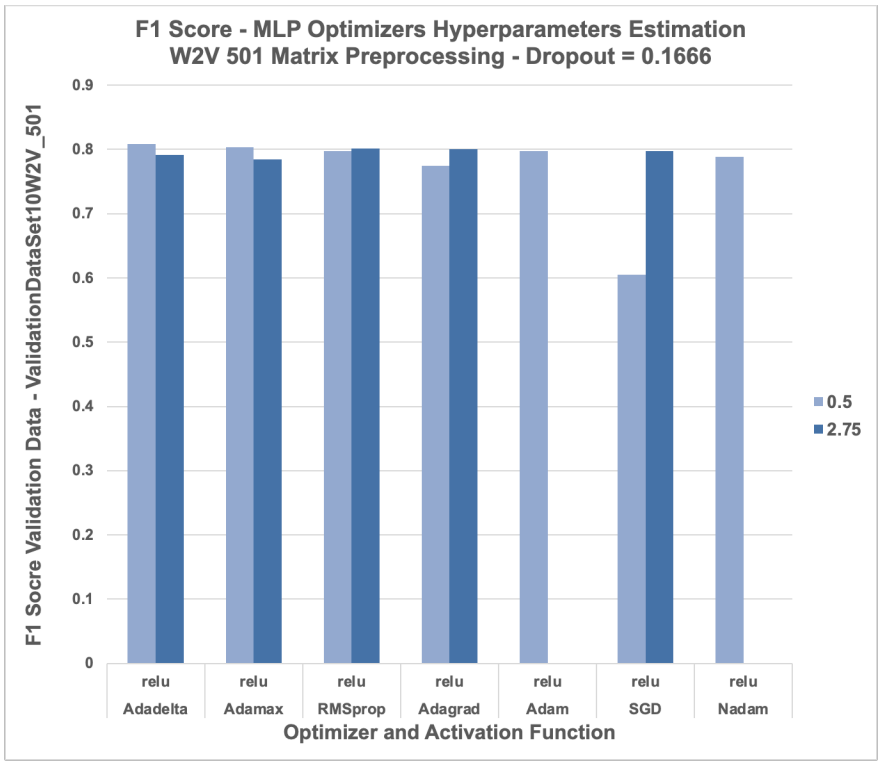


Chart 14- F1 Score per Optimizer and Activation Functions. Dropout = 0.1666 – MLP Architecture. W2V 501Preprocessing. Source: Author

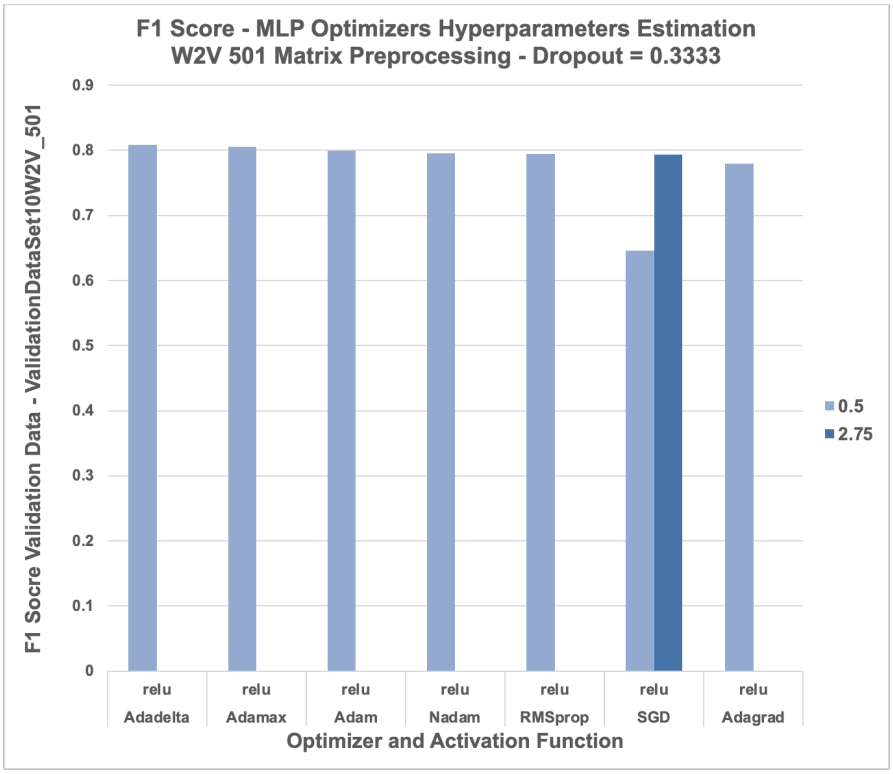


Chart 15- F1 Score per Optimizer and Activation Functions. Dropout = 0.3333 – MLP Architecture. W2V 501Preprocessing. Source: Author

	Experiment Iteration		96	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.83915156
	5445	534	PPV	0.78051787
	FN	TP	ACC	0.89104586
	364	1899	F1	**0.80877342**
Dropout	Optimizer	Learning Rate	Losses	Activation Function
0.3333	Adadelta	0.5	binary_crossentropy	relu

Table 38- Confusion Matrix MLP Optimization Hyperparameters Estimation Result W2V 501 Preprocessing 91. Source: Author

	Experiment Iteration		68	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.84843128
	5411	568	PPV	0.77170419
	FN	TP	ACC	0.88946855
	343	1920	F1	**0.80825089**
Dropout	Optimizer	Learning Rate	Losses	Activation Function
0.1666	Adadelta	0.5	binary_ crossentropy	relu

	Experiment Iteration		98	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.87229341
	5317	662	PPV	0.74886190
	FN	TP	ACC	0.88461536
	289	1974	F1	**0.80587875**
Dropout	Optimizer	Learning Rate	Losses	Activation Function
0.3333	Adamax	0.5	binary_ crossentropy	relu

Table 39- Confusion Matrix MLP Optimization Hyperparameters Estimation Result W2V 501 Preprocessing 68, 98. Source: Author

	Experiment Iteration		70	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.83738398
	5420	559	PPV	0.77220863
	FN	TP	ACC	0.88752728
	368	1895	F1	**0.80347678**
Dropout	Optimizer	Learning Rate	Losses	Activation Function
0.1666	Adamax	0.5	binary_ crossentropy	relu

	Experiment Iteration		73	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.84224480
	5391	588	PPV	0.76423418
	FN	TP	ACC	0.88534337
	357	1906	F1	**0.88534337**
Dropout	Optimizer	Learning Rate	Losses	Activation Function
0.1666	RMSprop	2.75	binary_ crossentropy	relu

Table 40- Confusion Matrix MLP Optimization Hyperparameters Estimation Result W2V 501 Preprocessing 70, 63. Source: Author

From the above, the best model is 96. This one has provided the best KPI's and has minimized type I and type II error in the confusion matrix, increasing up to 0.8087 the F1 Score, and 18.58% the performance of the MLP model. As a result we have concluded the hyperparameter estimation for this preprocessing method. The table below summarize the selected hyperparameters to promote to the next phase of the investigation.

Hyperparameters Selection: W2V 501 Preprocessing - MLP Network		
Parameter	**Parameter ID**	**Permutation Values**
Learning Rate	**lr**	**[0.5]**
First Neuron	**first_neuron**	**[500]**
Hidden Layers	**hidden_layers**	**[10]**
Batch Size	**batch_size**	**[1000]**
Epochs	**epochs**	**[1000]**
Dropout	**dropout**	**[0.3333]**
Shapes	**shapes**	**['brick']**
Optimizer	**optimizer**	**[Adadelta]**
Losses	**losses**	**['binary_crossentropy']**
Activation	**activation**	**['relu']**
Last Activation	**last_activation**	**[sigmoid]**

Table 41- Hyperparameters Selection W2V 501 Matrix Preprocessing – MLP Network. Source: Author

10.2.4 RNN LSTM ARCHITECTURE – DL SIMILARITY MATRIX DATA SET

Our third experimental setup includes the selection of the RNN - LSTM architecture and the DL Similarity Matrix Data Sets of preprocess data. The objective to achieve with this setup is to test a different architecture approach with the same preprocessing methodologies, in order to compare the neural network architecture performance. With this RNN LSTM architecture, we will introduce four (4) new hyperparameters, which are found inside the LSTM neuron[34].

- LSTM Internal Activation Function: Activation function in of the LSTM cell.
- LSTM Internal Recurrent Activation Function: Activation function to use in the recurrent step.
- LSTM Internal Dropout: Fraction of the units to drop from the for the linear transformation of the inputs.
- LSTM Internal Recurrent Dropout: Fraction of the units to drop for the linear transformation of the recurrent state.

Using the same approach presented in section 10.2.2 two sets of estimation rounds will be performed. First one to estimate architecture size and shape hyperparameters, and the second one to estimate optimization hyperparameters. For each estimation experiment a set of parameters dictionary has been configured based in the available estimation parameters provided in Talos Scan[35] library. The dictionary includes the following selected parameters:

Hyperparameters Selection: DL Similarity Matrix Preprocessing – RNN LSTM Network		
Parameter	**Parameter ID**	**Permutation Values**
LSTM Internal Activation Function	**LSTM_activation**	**['tanh']**
LSTM Internal Recurrent Activation Function	**LSTM_recurrent_activation**	**['sigmoid']**
LSTM Internal Dropout	**LSTM_dropout**	**[0]**
LSTM Internal Recurrent Dropout	**LSTM_recurrent_dropout**	**[0]**
Learning Rate	**lr**	**[0.5]**
First Neuron	**first_neuron**	**[500]**
LSTM Layers	**hidden_layers**	**[1,2,3]**
Batch Size	**batch_size**	**[32,64,128,1000]**
Epochs	**epochs**	**[10,100,1000]**
Dropout	**dropout**	**[0.5]**
Optimizer	**optimizer**	**[Adam]**
Losses	**losses**	**['binary_crossentropy']**

[34] Keras API reference / LSTM Layer: https://keras.io/api/layers/recurrent_layers/lstm/

[35] Talos Scan Library: https://github.com/autonomio/talos/blob/master/docs/Scan.md

Hyperparameters Selection: DL Similarity Matrix Preprocessing – RNN LSTM Network		
Parameter	**Parameter ID**	**Permutation Values**
Activation	**activation**	**['elu']**
Last Activation	**last_activation**	**[sigmoid]**

Table 42- Hyperparameters Selection DL Similarity Matrix Preprocessing – RNN LSTM Network. Source: Author

After executing the corresponding experiments in this round, that compiled 36 experiments, the following results were gathered in terms of F1 Score KPI.

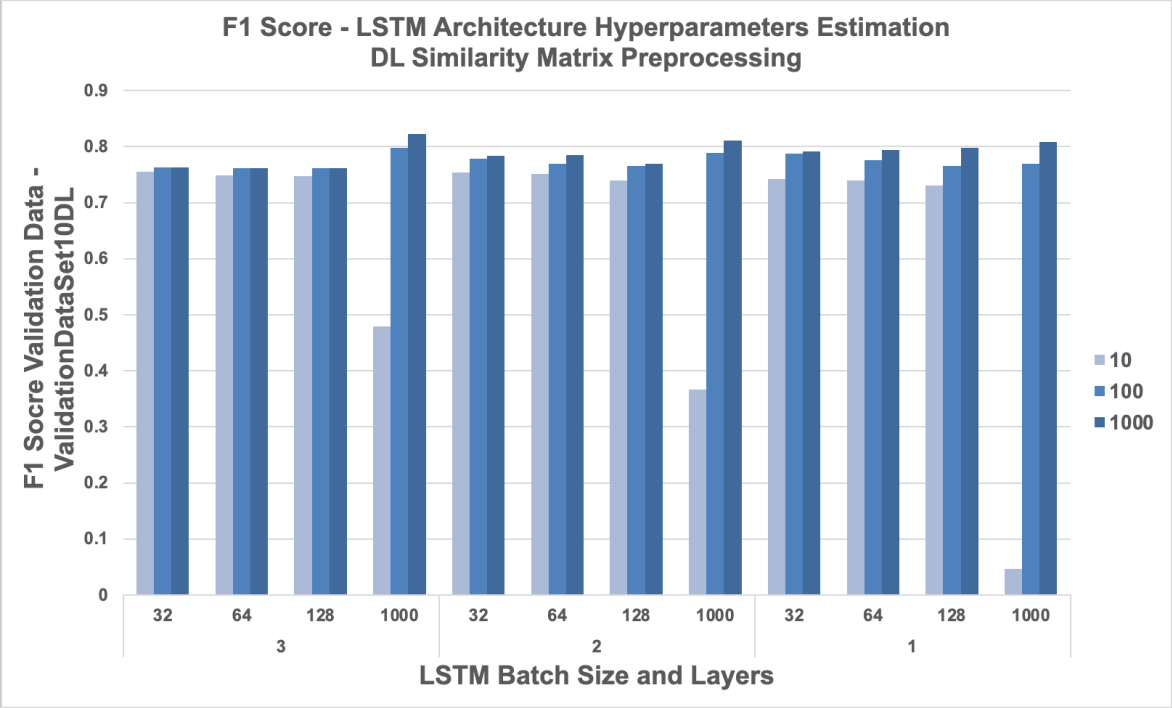


Chart 16- F1 Score per Batch Size and LSTM Layers. Each series represents training epochs – RNN LSTM Architecture. DL Similarity Matrix Preprocessing. Source: Author

	Experiment Iteration		**9**	
	Actual Class		**KPI's**	
Predicted Class	**TN**	**FP**	**TPR**	0.72691118
	5888	91	**PPV**	0.94758063
	FN	**TP**	**ACC**	0.91397720
	618	1645	**F1**	**0.82270567**
LSTM Layers	**Epochs**	**Batch Size**		
3	1000	1000		

Table 43- Confusion Matrix RNN LSTM Architecture Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 9. Source: Author

	Experiment Iteration		6	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.69730448
	5930	49	PPV	0.96988320
	FN	TP	ACC	0.91094392
	685	1578	F1	**0.81131105**
LSTM Layers	Epochs	Batch Size		
2	1000	1000		

	Experiment Iteration		3	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.69951391
	5905	74	PPV	0.95534098
	FN	TP	ACC	0.90851736
	680	1583	F1	**0.80765306**
LSTM Layers	Epochs	Batch Size		
1	1000	1000		

Table 44- Confusion Matrix RNN LSTM Architecture Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 6, 3. Source: Author

	Experiment Iteration		8	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.74326115
	5711	268	PPV	0.86256408
	FN	TP	ACC	0.89699101
	581	1682	F1	**0.79848089**
LSTM Layers	Epochs	Batch Size		
3	100	1000		

	Experiment Iteration		18	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.67874503
	5926	53	PPV	0.96664571
	FN	TP	ACC	0.90536278
	727	1536	F1	**0.79750778**
LSTM Layers	Epochs	Batch Size		
1	1000	128		

Table 45- Confusion Matrix RNN LSTM Architecture Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 8, 18. Source: Author

From the results presented above, the best performance model is experiment 9 with F1 Score 0.8227 and has the smallest false positives results. The figure below presents the model.

```
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 1, 10)             840
_________________________________________________________________
lstm_1 (LSTM)                (None, 1, 10)             840
_________________________________________________________________
lstm_2 (LSTM)                (None, 10)                840
_________________________________________________________________
dropout (Dropout)            (None, 10)                0
_________________________________________________________________
dense (Dense)                (None, 1)                 11
=================================================================
Total params: 2,531
Trainable params: 2,531
Non-trainable params: 0
```

Figure 31- RNN LSTM Architecture Model Experiment 9. DL Similarity Matrix Preprocessing. Source: Author

Now after selecting the LSTM architecture size and shape hyperparameters, the optimization hyperparameters will be estimated. An experimental setup with 42 combinations has been constructed. The idea of this second round of hyperparameter estimations is to focus on the optimization hyperparameters. These are:

- LSTM Activation (LSTM_activation)
- LSTM Dropout (LSTM_dropout)
- Batch Size (batch_size)
- Learning Rate (lr)
- Dropout (dropout)
- Optimizer (optimizer)

The details of the combinations hyperparameters are presented in the table below.

Hyperparameters Selection: DL Similarity Matrix Preprocessing – RNN LSTM Optimizers		
Parameter	**Parameter ID**	**Permutation Values**
LSTM Internal Activation Function	**LSTM_activation**	**['tanh','elu']**
LSTM Internal Recurrent Activation Function	**LSTM_recurrent_activation**	**['sigmoid']**
LSTM Internal Dropout	**LSTM_dropout**	**[0,0333,0.5,0.8333]**
LSTM Internal Recurrent Dropout	**LSTM_recurrent_dropout**	**[0]**
Learning Rate	**lr**	**[0.5,2.75]**
First Neuron	**first_neuron**	**[10]**
LSTM Layers	**hidden_layers**	**[3]**
Batch Size	**batch_size**	**[32,64,128,1000]**
Epochs	**epochs**	**[1000]**
Dropout	**dropout**	**[0.3333,0.5,0.8333]**
Optimizer	**optimizer**	**[Adam, SGD]**
Losses	**losses**	**['binary_crossentropy']**
Activation	**activation**	**['elu']**
Last Activation	**last_activation**	**[sigmoid]**

Table 46- Hyperparameters Selection DL Similarity Matrix Preprocessing – RNN LSTM Optimizers. Source: Author

As presented in the following charts, the best experimental result of the 42 experiments achieved a F1 Score of 0.8227. This means a 0.37% of improvement in the performance metric. Each chart presents the results for a specific Dropout value with the combination of Optimizers and Activation Functions, and each series represents the experiment Learning Rate.

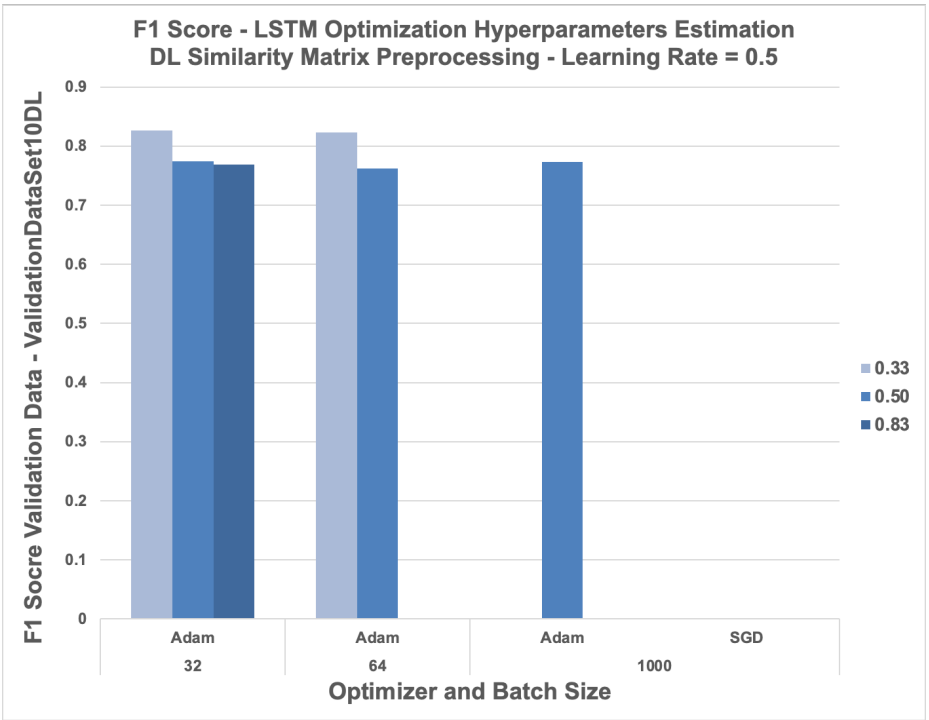


Chart 17- F1 Score per Optimizer and Batch Size. Learning Rate = 0.5 – LSTM Architecture. DL Similarity Matrix Preprocessing. Source: Author

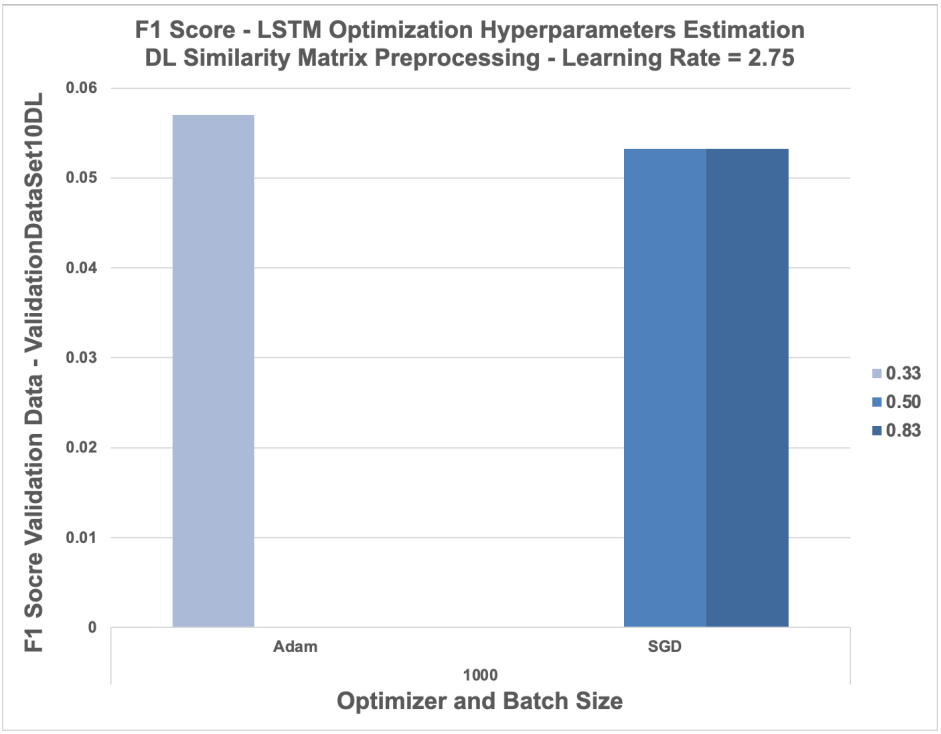


Chart 18- F1 Score per Optimizer and Batch Size. Learning Rate = 2.75 – LSTM Architecture. DL Similarity Matrix Preprocessing. Source: Author

	Experiment Iteration		38		
	Actual Class		KPI's		
Predicted Class	TN	FP	TPR		0.734423339
	5879	100	PPV		0.943246305
	FN	TP	ACC		0.914947808
	601	1662	F1		**0.825838509**
LSTM Activation	LSTM Dropout	Batch Size	Dropout	lr	Optimizer
tanh	0	32	0.3333	0.5	Adam

Table 47- Confusion Matrix RNN LSTM Optimization Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 38. Source: Author

	Experiment Iteration		41		
	Actual Class		KPI's		
Predicted Class	TN	FP	TPR		0.734423339
	5863	116	PPV		0.934758127
	FN	TP	ACC		0.913006544
	601	1662	F1		**0.822568671**
LSTM Activation	LSTM Dropout	Batch Size	Dropout	lr	Optimizer
tanh	0	64	0.3333	0.5	Adam

	Experiment Iteration		39		
	Actual Class		KPI's		
Predicted Class	TN	FP	TPR		0.635881603
	5966	13	PPV		0.991046846
	FN	TP	ACC		0.898446977
	824	1439	F1		**0.774697174**
LSTM Activation	LSTM Dropout	Batch Size	Dropout	lr	Optimizer
tanh	0	32	0.5	0.5	Adam

Table 48- Confusion Matrix RNN LSTM Optimization Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 41, 39. Source: Author

	Experiment Iteration		37		
	Actual Class		KPI's		
Predicted Class	TN	FP	TPR		0.632346451
	5970	9	PPV		0.993749976
	FN	TP	ACC		0.897961676
	832	1431	F1		**0.772886849**
LSTM Activation	LSTM Dropout	Batch Size	Dropout	lr	Optimizer
tanh	0	1000	0.5	0.5	Adam

	Experiment Iteration		40		
	Actual Class		KPI's		
Predicted Class	TN	FP	TPR		0.62439239
	5977	2	PPV		0.998586595
	FN	TP	ACC		0.896627009
	850	1413	F1		**0.768352365**
LSTM Activation	LSTM Dropout	Batch Size	Dropout	lr	Optimizer
tanh	0	32	0.8333	0.5	Adam

Table 49- Confusion Matrix RNN LSTM Optimization Hyperparameters Estimation Result DL Similarity Matrix Preprocessing 37, 40. Source: Author

From the above, the best model is 38. This one has provided the best KPI's and has minimized type I and type II error in the confusion matrix, increasing up to 0.8258 the F1 Score. As a result we have concluded the hyperparameter estimation for this preprocessing method and architecture. The table below summarize the selected hyperparameters to promote to the next phase of the investigation.

Hyperparameters Selection: DL Similarity Matrix Preprocessing – RNN LSTM		
Parameter	**Parameter ID**	**Combination Values**
LSTM Internal Activation Function	**LSTM_activation**	**['tanh']**
LSTM Internal Recurrent Activation Function	**LSTM_recurrent_activation**	**['sigmoid']**
LSTM Internal Dropout	**LSTM_dropout**	**[0]**
LSTM Internal Recurrent Dropout	**LSTM_recurrent_dropout**	**[0]**
Learning Rate	**lr**	**[0.5]**
First Neuron	**first_neuron**	**[10]**
LSTM Layers	**hidden_layers**	**[3]**
Batch Size	**batch_size**	**[32]**
Epochs	**epochs**	**[1000]**
Dropout	**dropout**	**[0.3333]**
Optimizer	**optimizer**	**[Adam]**
Losses	**losses**	**['binary_crossentropy']**
Activation	**activation**	**['elu']**
Last Activation	**last_activation**	**[sigmoid]**

Table 50- Hyperparameters Selection DL Similarity Matrix Preprocessing – RNN LSTM. Source: Author

This will be the model selected for this specific approach in order to promote to the test evaluation phase.

10.2.5 RNN LSTM Architecture – W2V 501 Matrix Data Set

Our fourth experimental setup includes the selection of the RNN - LSTM architecture and the Word2Vec 501 Matrix Data Sets of preprocess data. The objective to achieve with this setup is to test a different architecture approach with the same preprocessing methodologies, in order to compare the neural network architecture performance. As part of the hyperparameter estimation, the same hyperparameters described in section 10.2.4 are used in this experiment.

For each estimation experiment a set of parameters dictionary has been configured based in the available estimation parameters provided in Talos Scan[36] library. The dictionary includes the following selected parameters:

Hyperparameters Selection: W2V 501 Preprocessing – RNN LSTM Network		
Parameter	**Parameter ID**	**Permutation Values**
LSTM Internal Activation Function	**LSTM_activation**	**['tanh']**
LSTM Internal Recurrent Activation Function	**LSTM_recurrent_activation**	**['sigmoid']**
LSTM Internal Dropout	**LSTM_dropout**	**[0]**
LSTM Internal Recurrent Dropout	**LSTM_recurrent_dropout**	**[0]**
Learning Rate	**lr**	**[0.5]**
LSTM Units	**first_neuron**	**[256,512]**
LSTM Layers	**hidden_layers**	**[1,2]**
Batch Size	**batch_size**	**[32,64,128]**
Epochs	**epochs**	**[10,100,1000]**
Dropout	**dropout**	**[0]**
Optimizer	**optimizer**	**[Adam]**
Losses	**losses**	**['binary_crossentropy']**
Activation	**activation**	**['relu']**
Last Activation	**last_activation**	**[sigmoid, softmax]**

Table 51- Hyperparameters Selection W2V 501 Matrix Preprocessing – RNN LSTM Network. Source: Author

With this experimental array, we present 94 experiments for architecture hyperparameter estimation. We will test the performance iterating with number of LSTM Layers, number of LSTM Units, different batch sizes, last activation function, and training epochs.

[36] Talos Scan Library: https://github.com/autonomio/talos/blob/master/docs/Scan.md

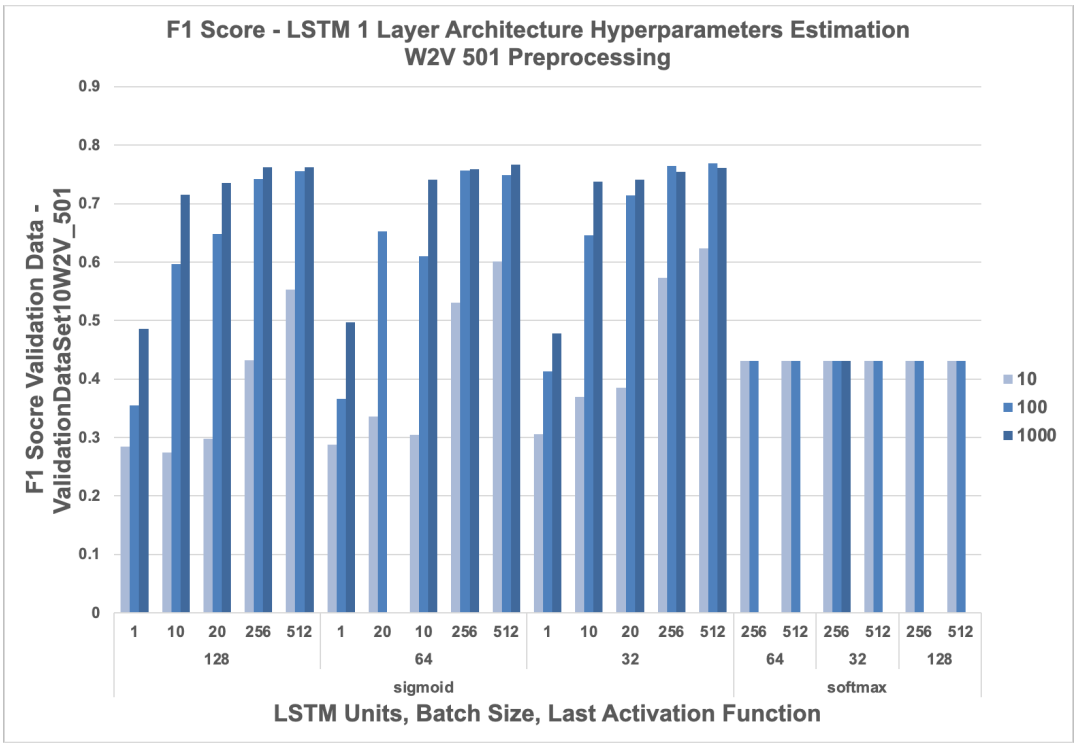


Chart 19- F1 Score per LSTM Units, Batch Size, Last Activation Function for LSTM 1 Layer. Each series represents training epochs – RNN LSTM Architecture. W2V 501 Matrix Preprocessing. Source: Author

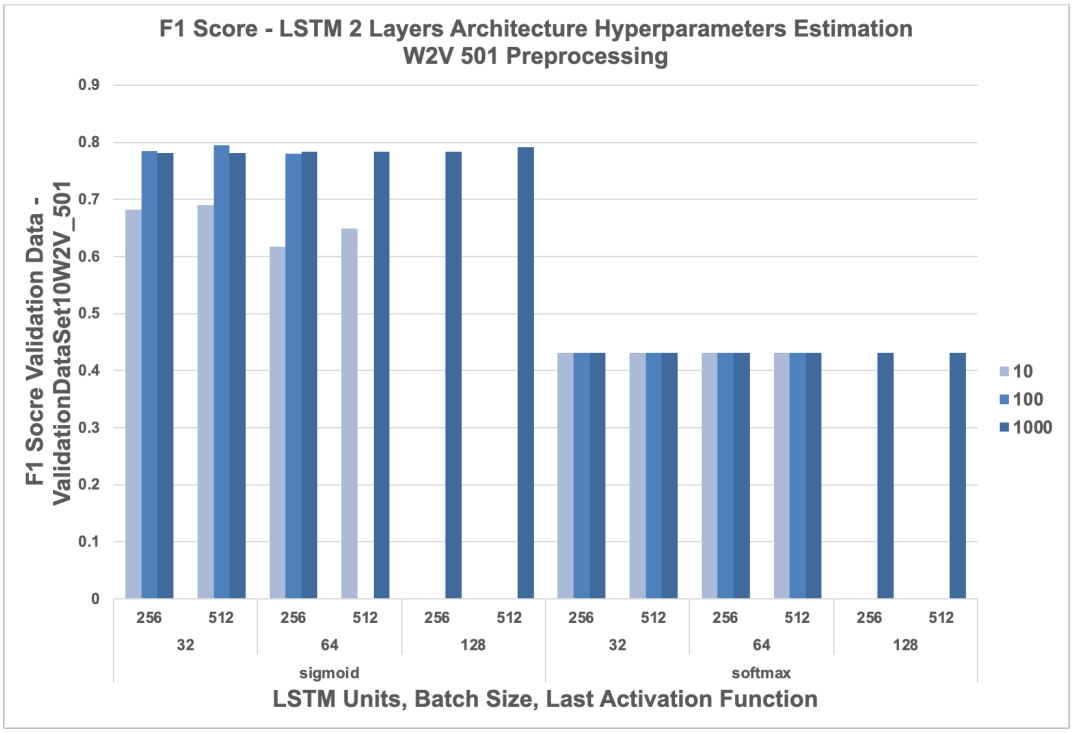


Chart 20- F1 Score per LSTM Units, Batch Size, Last Activation Function for LSTM 2 Layers. Each series represents training epochs – RNN LSTM Architecture. W2V 501 Matrix Preprocessing. Source: Author

	Experiment Iteration		7	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.817940772
	5438	541	PPV	0.77382946
	FN	TP	ACC	0.884372711
	412	1851	F1	**0.795273899**
LSTM Units	Batch Size	Epochs	LSTM Layers	
512	32	100	2	

Table 52- Confusion Matrix RNN LSTM Architecture Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 7. Source: Author

	Experiment Iteration		35	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.800265133
	5474	505	PPV	0.781951666
	FN	TP	ACC	0.88388741
	452	1811	F1	**0.791002402**
LSTM Units	Batch Size	Epochs	LSTM Layers	
512	128	1000	2	

	Experiment Iteration		5	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.796729982
	5449	530	PPV	0.772824705
	FN	TP	ACC	0.879883528
	460	1803	F1	**0.7845953**
LSTM Units	Batch Size	Epochs	LSTM Layers	
256	32	100	2	

Table 53- Confusion Matrix RNN LSTM Architecture Hyperparameters Estimation W2V 501 Matrix Preprocessing 35, 5. Source: Author

	Experiment Iteration		31	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.7976138
	5439	540	PPV	0.769722819
	FN	TP	ACC	0.878912866
	458	1805	F1	**0.783420139**
LSTM Units	Batch Size	Epochs	LSTM Layers	
512	64	1000	2	

	Experiment Iteration		33	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.807335377
	5404	575	PPV	0.760616124
	FN	TP	ACC	0.877335608
	436	1827	F1	**0.783279743**
LSTM Units	Batch Size	Epochs	LSTM Layers	
256	128	1000	2	

Table 54- Confusion Matrix RNN LSTM Architecture Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 31, 33. Source: Author

From the results presented above, the best performance model is experiment 7 with F1 Score 0.7952 and has the smallest false positives results. The figure below presents the model.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_12 (LSTM)               (None, 2, 512)            1562624
_________________________________________________________________
lstm_13 (LSTM)               (None, 512)               2099200
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,662,337
Trainable params: 3,662,337
Non-trainable params: 0
```

Figure 32- RNN LSTM Architecture Model Experiment 7. W2V 501 Matrix Preprocessing. Source: Author

Now after selecting the LSTM architecture size and shape hyperparameters, the optimization hyperparameters will be estimated. An experimental setup with 32 combinations has been constructed. The idea of this second round of hyperparameter estimations is to focus on the optimization hyperparameters. These are:

- LSTM Activation (LSTM_activation)
- LSTM Dropout (LSTM_dropout)
- LSTM Recurrent Activation (LSTM_recurrent_activation)
- LSTM Recurrent Dropout (LSTM_recurrent_dropout)
- Learning Rate (lr)
- Optimizer (optimizer)

The details of the combinations hyperparameters are presented in the table below.

Optimization Hyperparameters Selection: W2V 501 Preprocessing – RNN LSTM Network		
Parameter	**Parameter ID**	**Permutation Values**
LSTM Internal Activation Function	**LSTM_activation**	**['relu','tanh']**
LSTM Internal Recurrent Activation Function	**LSTM_recurrent_activation**	**['sigmoid']**
LSTM Internal Dropout	**LSTM_dropout**	**[0, 0.5, 0.8333]**
LSTM Internal Recurrent Dropout	**LSTM_recurrent_dropout**	**[0, 0.5]**
Learning Rate	**lr**	**[0.5, 0.8333, 1, 2.75]**
LSTM Units	**first_neuron**	**[512]**
LSTM Layers	**hidden_layers**	**[2]**
Batch Size	**batch_size**	**[32]**
Epochs	**epochs**	**[100]**
Optimizer	**optimizer**	**[Adam]**

Optimization Hyperparameters Selection: W2V 501 Preprocessing – RNN LSTM Network		
Parameter	**Parameter ID**	**Permutation Values**
Losses	**losses**	**['binary_crossentropy']**
Last Activation	**last_activation**	**[sigmoid]**

Table 55- Optimization Hyperparameters Selection W2V 501 Matrix Preprocessing – RNN LSTM Network. Source: Author

Due to the processing cost in terms of time, not all the combinations are recorded. For the best optimization hyperparameter combination, which returned a result of 0.8007 in F1 Score KPI, we raised again the training epochs hyperparameter to 1000. This model resulted in the best hyperparameter estimation for this preprocessing and RNN architecture proposal. The final value of the F1 Score KPI for the selected model is 0.803. The charts below present the behavior of each experiment iteration in terms of the F1 Score KPI.

The first chart presents the result of the combined hyperparameters leaving fixed the LSTM Recurrent Dropout = 0, and in the second chart the results are presented with the LSTM Recurrent Dropout hyperparameter fixed to 0.5.

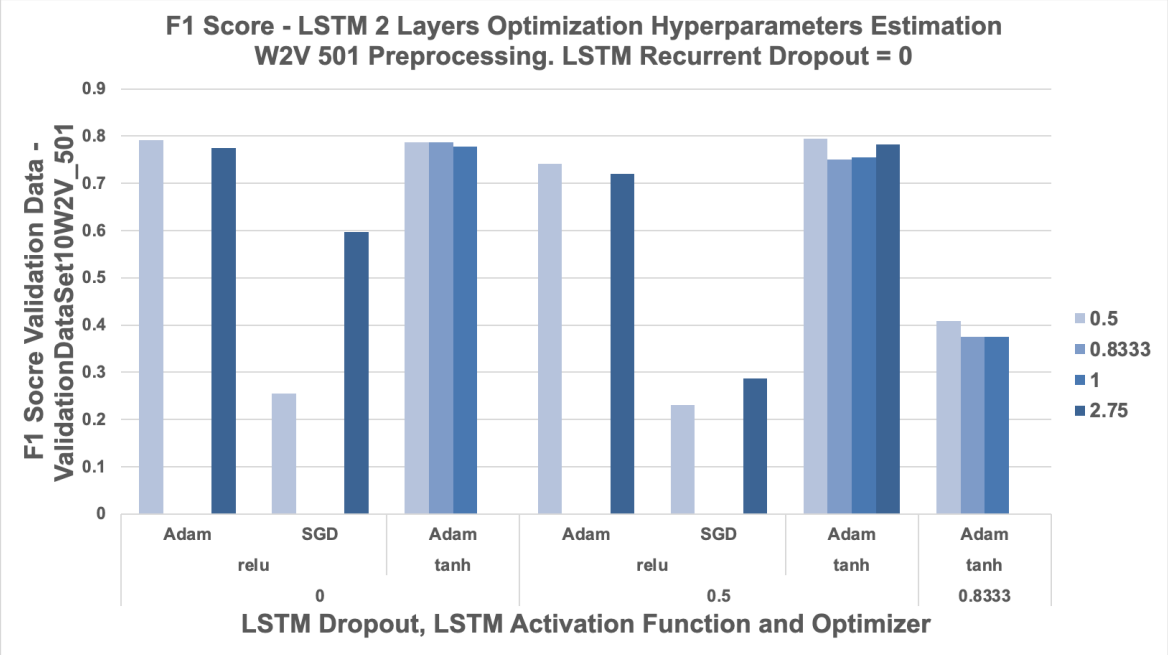


Chart 21- F1 Score per LSTM Dropout, LSTM Activation Function and Optimizer for LSTM 2 Layers. Each series represents learning rates. LSTM Recurrent Dropout = 0 – RNN LSTM Architecture. W2V 501 Matrix Preprocessing. Source: Author

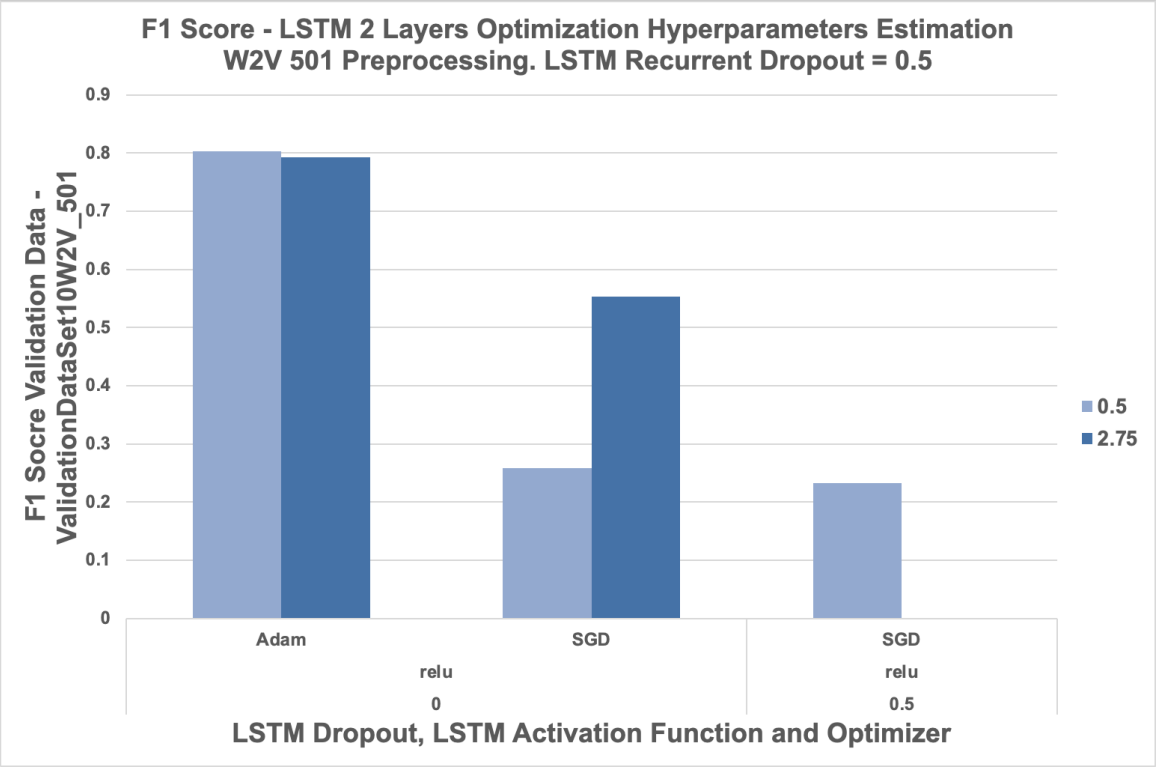


Chart 22- F1 Score per LSTM Dropout, LSTM Activation Function and Optimizer for LSTM 2 Layers. Each series represents learning rates. LSTM Recurrent Dropout = 0.5 – RNN LSTM Architecture. W2V 501 Matrix Preprocessing. Source: Author

From the results above we can conclude that the best value for the optimizer hyperparameter is 'Adam'. This is consistent with the literature and with all the other results presented in this investigation. Therefore, not all the combinations for the 'SGD' optimizer were evaluated.

Finally, the best combination of hyperparameters in experiment 14 that ran with 100 training epochs, was selected to be trained using 1000 training epochs. This resulted in the best model that will be selected as the best hyperparameter estimation model for the LSTM RNN architecture using W2V 501 Matrix Data Set preprocessing approach. This is experiment iteration 39. The details of the top five models are presented in the following tables.

	Experiment Iteration		39	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.819708347
	5477	502	PPV	0.787017405
	FN	TP	ACC	0.889589906
	408	1855	F1	**0.803030303**
LSTM Activation	LSTM Dropout	LSTM Recurrent Activation	LSTM Recurrent Dropout	LSTM Layers
relu	0	sigmoid	0.5	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	1000	Adam	sigmoid	Binary Cross Entropy

Table 56- Confusion Matrix RNN LSTM Optimization Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 39. Source: Author

	Experiment Iteration		14	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.855943441
	5341	638	PPV	0.752233028
	FN	TP	ACC	0.883038104
	326	1937	F1	**0.800744109**
LSTM Activation	LSTM Dropout	LSTM Recurrent Activation	LSTM Recurrent Dropout	LSTM Layers
relu	0	sigmoid	0.5	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	100	Adam	sigmoid	Binary Cross Entropy

	Experiment Iteration		6	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.815289438
	5441	538	PPV	0.774234176
	FN	TP	ACC	0.884008765
	418	1845	F1	**0.794231597**
LSTM Activation	LSTM Dropout	LSTM Recurrent Activation	LSTM Recurrent Dropout	LSTM Layers
tanh	0.5	sigmoid	0	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	100	Adam	sigmoid	Binary Cross Entropy

Table 57- Confusion Matrix RNN LSTM Optimization Hyperparameters Estimation W2V 501 Matrix Preprocessing 14, 6. Source: Author

	Experiment Iteration		16	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.830313742
	5377	602	PPV	0.757355928
	FN	TP	ACC	0.880368829
	384	1879	F1	**0.792158516**
LSTM Activation	LSTM Dropout	LSTM Recurrent Activation	LSTM Recurrent Dropout	LSTM Layers
relu	0	sigmoid	0.5	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
2.75	100	Adam	sigmoid	Binary Cross Entropy

	Experiment Iteration		10	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.803800285
	5461	518	PPV	0.778348327
	FN	TP	ACC	0.883280754
	444	1819	F1	**0.790869565**
LSTM Activation	LSTM Dropout	LSTM Recurrent Activation	LSTM Recurrent Dropout	LSTM Layers
relu	0	sigmoid	0	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	100	Adam	sigmoid	Binary Cross Entropy

Table 58- Confusion Matrix RNN LSTM Optimization Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 16, 10. Source: Author

From the results presented above, the best performance model is experiment 39 with F1 Score 0.8030 and has the smallest false positives results with 502 records. The figure below presents the model.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_12 (LSTM)               (None, 2, 512)            1562624
_________________________________________________________________
lstm_13 (LSTM)               (None, 512)               2099200
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,662,337
Trainable params: 3,662,337
Non-trainable params: 0
```

Figure 33- RNN LSTM Architecture Model Experiment 7. W2V 501 Matrix Preprocessing. Source: Author

This experiment has provided the best KPI's and has minimized type I and type II error in the confusion matrix, increasing up to 0.8030 the F1 Score. As a result we have concluded the hyperparameter estimation for this preprocessing method and architecture. The table below summarize the selected hyperparameters to promote to the next phase of the investigation.

Hyperparameters Selection: W2V 501 Preprocessing – RNN LSTM Network		
Parameter	**Parameter ID**	**Combination Values**
LSTM Internal Activation Function	**LSTM_activation**	**['relu']**
LSTM Internal Recurrent Activation Function	**LSTM_recurrent_activation**	**['sigmoid']**
LSTM Internal Dropout	**LSTM_dropout**	**[0]**
LSTM Internal Recurrent Dropout	**LSTM_recurrent_dropout**	**[0.5]**
Learning Rate	**lr**	**[0.5]**
LSTM Units	**first_neuron**	**[512]**
LSTM Layers	**hidden_layers**	**[2]**
Batch Size	**batch_size**	**[32]**
Epochs	**Epochs**	**[1000]**
Optimizer	**Optimizer**	**[Adam]**
Losses	**Losses**	**['binary_crossentropy']**
Last Activation	**last_activation**	**[sigmoid]**

Table 59- Hyperparameters Selection W2V 501Matrix Preprocessing – RNN LSTM. Source: Author

10.2.6 RNN LSTM Architecture with Dense Output Layer – W2V 501 Matrix Data Set

As a final adjustment to this experimental array, we have included a dense layer between the LSTM hidden layers and the output layer. This in order to increase the effectiveness of the experiment and follow best practices. The experiment will only replicate the last hyperparameter training scenario presented in section 10.2.5, using parameters defined in *Table 59*. The figure below presents the model.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 2, 512)            1562624
_________________________________________________________________
lstm_5 (LSTM)                (None, 512)               2099200
_________________________________________________________________
dense_4 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 257
=================================================================
Total params: 3,793,409
Trainable params: 3,793,409
Non-trainable params: 0
```

Figure 34- RNN LSTM Architecture Model Experiment With Dense Layer. W2V 501 Matrix Preprocessing. Source: Author

As expected, we see improvements in all performance KPI's, specifically the F1 Score which reached a value of 0.8052. The following table presents the results.

	Experiment Iteration		40	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.825894833
	5469	510	PPV	0.785624206
	FN	TP	ACC	0.890317857
	394	1869	F1	**0.805256355**
LSTM Activation	LSTM Dropout	LSTM Recurrent Activation	LSTM Recurrent Dropout	LSTM Layers
relu	0	sigmoid	0.5	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	1000	Adam	sigmoid	Binary Cross Entropy

Table 60- Confusion Matrix RNN LSTM Optimization Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 40. Source: Author

This will be the model selected for this specific approach in order to promote to the test evaluation phase.

10.2.7 RNN Bi Directional LSTM Architecture – W2V 501 Matrix Data Set

Our fifth experimental setup includes the selection of the RNN – Bi Directional LSTM architecture and the Word2Vec 501 Matrix Data Sets of preprocess data. The objective to achieve with this setup is to test a different architecture approach with the same preprocessing methodologies, in order to compare the neural network architecture performance. With this RNN LSTM architecture, we will use by default two (2) bi directional layers using LSTM units. This experiment fixture uses the same hyperparameters setup defined in section 10.2.5

For each estimation experiment a set of parameters dictionary has been configured based in the available estimation parameters provided in Talos Scan[37] library. The dictionary includes the following selected parameters:

[37] Talos Scan Library: https://github.com/autonomio/talos/blob/master/docs/Scan.md

Hyperparameters Selection: W2V 501 Preprocessing – RNN BiLSTM Network		
Parameter	**Parameter ID**	**Permutation Values**
LSTM Internal Activation Function	**LSTM_activation**	**[‘tanh’]**
LSTM Internal Recurrent Activation Function	**LSTM_recurrent_activation**	**[‘sigmoid’]**
LSTM Internal Dropout	**LSTM_dropout**	**[0]**
LSTM Internal Recurrent Dropout	**LSTM_recurrent_dropout**	**[0]**
Learning Rate	**lr**	**[0.5]**
LSTM Units	**first_neuron**	**[256,512,1024]**
LSTM Layers	**hidden_layers**	**[2]**
Batch Size	**batch_size**	**[32,64,128]**
Epochs	**epochs**	**[10,100,1000]**
Dropout	**dropout**	**[0]**
Optimizer	**optimizer**	**[Adam]**
Losses	**losses**	**['binary_crossentropy']**
Last Activation	**last_activation**	**[sigmoid]**

Table 61- Hyperparameters Selection W2V 501 Matrix Preprocessing – RNN LSTM Network. Source: Author

With this experimental array, we present 41 experiments for architecture hyperparameter estimation. We will test the performance iterating with number of number of LSTM Units, different batch sizes, and training epochs.

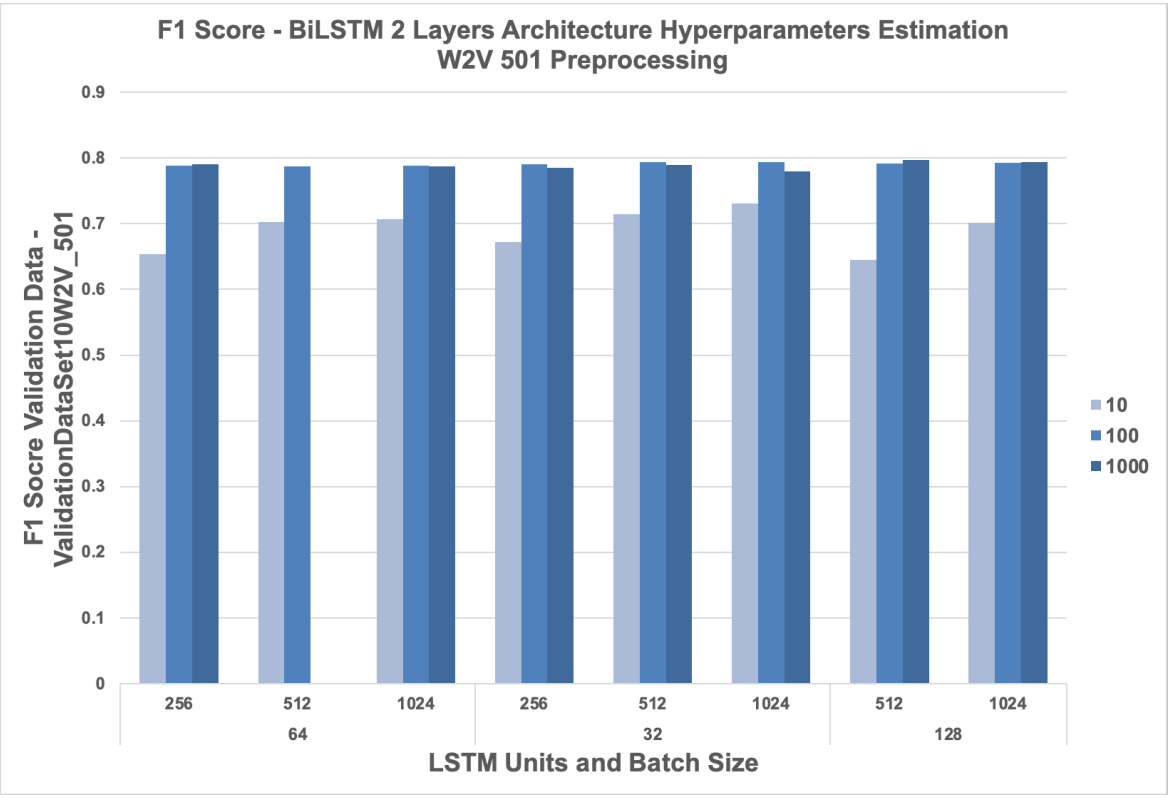


Chart 23- F1 Score per LSTM Units and Batch Size for Bi-LSTM 2 Layers. Each series represents training epochs. – RNN Bi-LSTM Architecture. W2V 501 Matrix Preprocessing. Source: Author

The best combination of architecture hyperparameters is found in experiment 14 that ran with 1000 training epochs, 512 LSTM Units per layer, 2 Bi-LSTM Layers and Batch Size 128, with a F1 Score KPI of 0.7975, was selected to be trained using 1000 training epochs. This resulted in the best model that will be selected as the best hyperparameter estimation model for the LSTM RNN architecture using W2V 501 Matrix Data Set preprocessing approach. This is experiment iteration 39. The details of the top five models are presented in the following tables.

	Experiment Iteration		14	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.817940772
	5451	528	PPV	0.778057992
	FN	TP	ACC	0.885950029
	412	1851	F1	**0.797501077**
LSTM Units	Batch Size	Epochs	BiLSTM Layers	
512	128	1000	2	

Table 62- Confusion Matrix RNN Bi-LSTM Architecture Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 14. Source: Author

	Experiment Iteration		16	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.819708347
	5424	555	PPV	0.769709527
	FN	TP	ACC	0.883159399
	408	1855	F1	**0.793922534**
LSTM Units	Batch Size	Epochs	BiLSTM Layers	
1024	32	100	2	

	Experiment Iteration		4	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.804684043
	5474	505	PPV	0.782889068
	FN	TP	ACC	0.885100722
	442	1821	F1	**0.793636958**
LSTM Units	Batch Size	Epochs	BiLSTM Layers	
512	32	100	2	

Table 63- Confusion Matrix RNN Bi-LSTM Architecture Hyperparameters Estimation W2V 501 Matrix Preprocessing 16, 4. Source: Author

	Experiment Iteration		23	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.797171891
	5498	481	PPV	0.78949672
	FN	TP	ACC	0.885950029
	459	1804	F1	**0.793315743**
LSTM Units	Batch Size	Epochs	BiLSTM Layers	
1024	128	1000	2	

	Experiment Iteration		22	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.7976138
	5490	489	PPV	0.786835194
	FN	TP	ACC	0.885100722
	458	1805	F1	**0.792187843**
LSTM Units	Batch Size	Epochs	BiLSTM Layers	
1024	128	100	2	

Table 64- Confusion Matrix RNN Bi-LSTM Architecture Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 23, 22. Source: Author

From the results presented in Table 62, the best performance model is experiment 14 with F1 Score 0.7975 and has the smallest false positives results. But taking into account that the processing time versus experiment 16 is 2.3 times larger, 6,263 seconds, and the gain in F1 Score KPI is just 0.45%, we will select model 16 to continue to optimization hyperparameter estimation. Figure 35 presents the model. Take into account that each bi-directional layer has 2,048 units, which means it has 1,024 (hyperparameter value) going “forward” and 1,024 going “backward”.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional (None, 2, 2048)           10444800
_________________________________________________________________
bidirectional_1 (Bidirection (None, 2048)              25174016
_________________________________________________________________
dense (Dense)                (None, 1)                 2049
=================================================================
Total params: 35,620,865
Trainable params: 35,620,865
Non-trainable params: 0
```

Figure 35- RNN Bi-LSTM Architecture Model Experiment 7. W2V 501 Matrix Preprocessing. Source: Author

Now after selecting the Bi-LSTM architecture size and shape hyperparameters, the optimization hyperparameters will be estimated. An experimental setup with 48 combinations has been constructed. The idea of this second round of hyperparameter estimations is to focus on the optimization hyperparameters. These are:

- LSTM Activation (LSTM_activation)
- LSTM Dropout (LSTM_dropout)
- LSTM Recurrent Activation (LSTM_recurrent_activation)
- LSTM Recurrent Dropout (LSTM_recurrent_dropout)
- Learning Rate (lr)
- Optimizer (optimizer)
- Losses

The details of the combinations hyperparameters are presented in the table below.

Optimization Hyperparameters Selection: W2V 501 Preprocessing – RNN Bi-LSTM Network		
Parameter	**Parameter ID**	**Permutation Values**
LSTM Internal Activation Function	**LSTM_activation**	**['relu','tanh']**
LSTM Internal Recurrent Activation Function	**LSTM_recurrent_activation**	**['sigmoid']**
LSTM Internal Dropout	**LSTM_dropout**	**[0, 0.3333, 0.5, 0.8333]**
LSTM Internal Recurrent Dropout	**LSTM_recurrent_dropout**	**[0, 0.5]**
Learning Rate	**lr**	**[0.5, 0.8333, 1, 2.75]**
LSTM Units	**first_neuron**	**[1024]**
LSTM Layers	**hidden_layers**	**[2]**
Batch Size	**batch_size**	**[32]**
Epochs	**epochs**	**[100]**
Optimizer	**optimizer**	**[Adam]**
Losses	**losses**	**['binary_crossentropy', 'logcosh']**
Last Activation	**last_activation**	**[sigmoid]**

Table 65- Optimization Hyperparameters Selection W2V 501 Matrix Preprocessing – RNN Bi-LSTM Network. Source: Author

With this experimental array, we present 48 experiments for optimization hyperparameter estimation. We will test the performance iterating with number of LSTM Internal Activation Function, LSTM Internal Dropout, LSTM Internal Recurrent Dropout, Losses, and Learning Rate hyperparameters. Due to the processing cost in terms of time, not all the combinations are recorded. For the best optimization hyperparameter combination, which returned a result of 0.7983 in F1 Score KPI, we raised again the training epochs hyperparameter to 1000, and change to 512 LSTM Units, as the best model result in the previous architecture hyperparameters selection. This additional combination model resulted in a lower F1 Score of 0.7891. Therefore the final selected model is 33 with 0.7983

F1 Score KPI. The charts below present the behavior of each experiment iteration in terms of the F1 Score KPI.

As the processing time is very high in this architecture fixture, we decided to iterate not all models at a time, but we started first reviewing the impact of LSTM Activation Function and the Optimizer for four (4) different values of LSTM Dropout. From these experiments we concluded that we will use Adam optimizer and tanh and relu LSTM Activation Functions for the next iterations. The chart below presents these results.

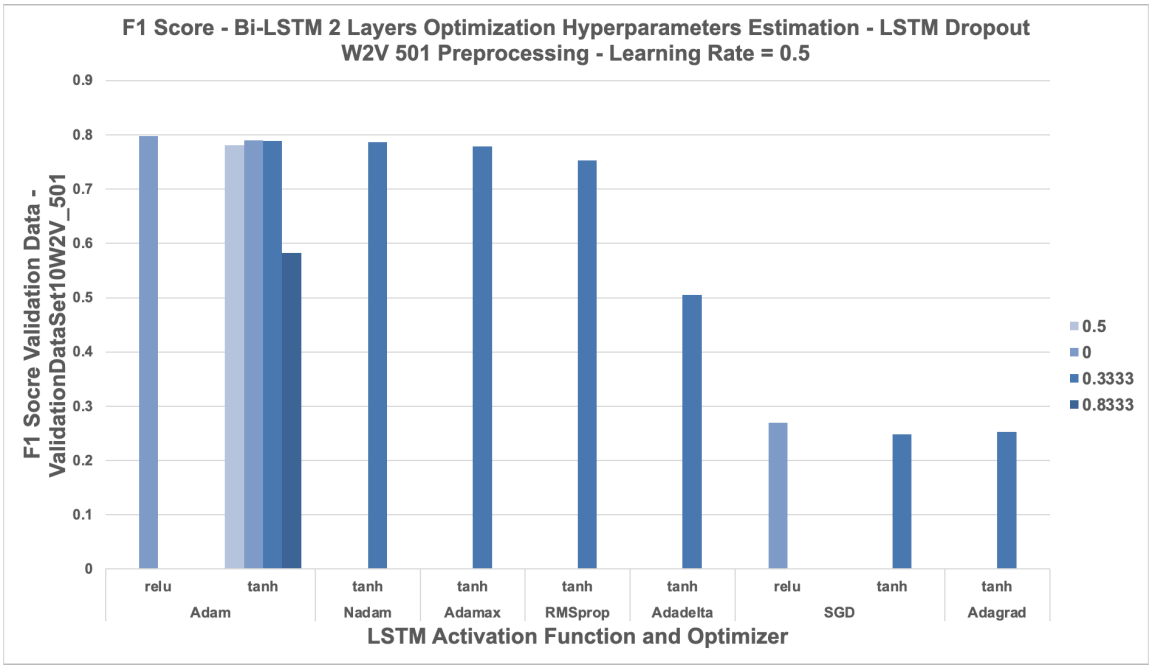


Chart 24- F1 Score per LSTM Activation Function and Optimizer for Bi-LSTM 2 Layers. Each series represents LSTM Dropout values. Learning Rate = 0.5 – RNN Bi-LSTM Architecture. W2V 501 Matrix Preprocessing. Source: Author

Now, we decided to iterate the LSTM Dropout with the Learning Rate and the LSTM Activation Function, fixing the optimizer value to Adam. The results pointed out that the selected LSTM Dropout best hyperparameter is 0, for relu and tanh. In the later one, iterating with four (4) different values for the learning rate. The chart below presents these results.

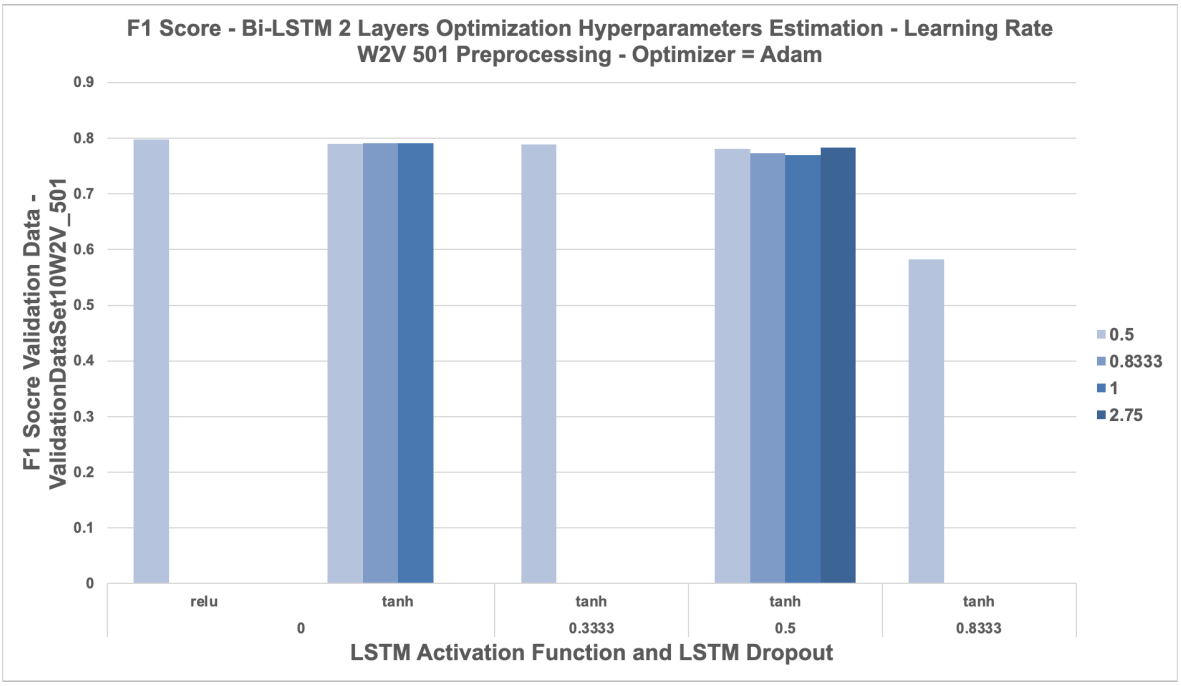


Chart 25- F1 Score per LSTM Activation Function and LSTM Dropout for Bi-LSTM 2 Layers. Each series represents Learning Rate values. Optimizer = Adam – RNN Bi-LSTM Architecture. W2V 501 Matrix Preprocessing. Source: Author

Finally, the LSTM Recurrent Dropout was reviewed, presented the following results.

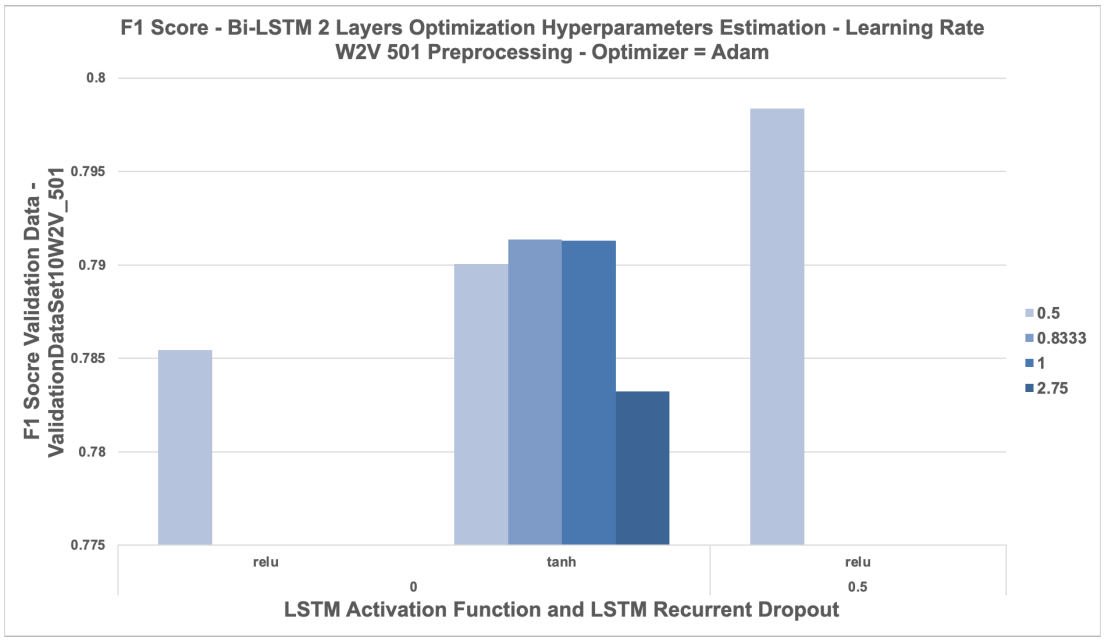


Chart 26- F1 Score per LSTM Activation Function and LSTM Recurrent Dropout for Bi-LSTM 2 Layers. Each series represents Learning Rate values. Optimizer = Adam – RNN Bi-LSTM Architecture. W2V 501 Matrix Preprocessing. Source: Author

Now, after the execution of the 48 experimental iterations, which in average took 4,945 seconds to execute per iteration, with a max processing time for experiment 59 which took 41,061 seconds to

execute, we found that the best model is 33 with a F1 Score KPI of 0.7983. The tables below present the top five results.

	Experiment Iteration		33	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.825011075
	5432	547	PPV	0.773405135
	FN	TP	ACC	0.885586023
	396	1867	F1	**0.798375027**
Bi-LSTM Activation	Bi-LSTM Dropout	Bi-LSTM Recurrent Activation	Bi-LSTM Recurrent Dropout	Bi-LSTM Layers
relu	0	sigmoid	0.5	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	100	Adam	sigmoid	Binary Cross Entropy

Table 66- Confusion Matrix RNN Bi-LSTM Optimization Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 33. Source: Author

	Experiment Iteration		50	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.810428619
	5441	538	PPV	0.77318716
	FN	TP	ACC	0.882674098
	429	1834	F1	**0.791370011**
Bi-LSTM Activation	Bi-LSTM Dropout	Bi-LSTM Recurrent Activation	Bi-LSTM Recurrent Dropout	Bi-LSTM Layers
tanh	0	sigmoid	0	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.8333	100	Adam	sigmoid	logcosh

	Experiment Iteration		48	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.803358376
	5465	514	PPV	0.779588342
	FN	TP	ACC	0.88364476
	445	1818	F1	**0.791294886**
Bi-LSTM Activation	Bi-LSTM Dropout	Bi-LSTM Recurrent Activation	Bi-LSTM Recurrent Dropout	Bi-LSTM Layers
tanh	0	sigmoid	0	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
1	100	Adam	sigmoid	Binary Cross Entropy

Table 67- Confusion Matrix RNN Bi-LSTM Optimization Hyperparameters Estimation W2V 501 Matrix Preprocessing 50, 48. Source: Author

	Experiment Iteration		46	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.794078648
	5490	489	PPV	0.786089242
	FN	TP	ACC	0.884130061
	466	1797	F1	**0.79006375**

Bi-LSTM Activation	Bi-LSTM Dropout	Bi-LSTM Recurrent Activation	Bi-LSTM Recurrent Dropout	Bi-LSTM Layers
tanh	0	sigmoid	0	2
Learning Rate	**Epochs**	**Optimizer**	**Last Activation**	**Losses**
0.5	100	Adam	sigmoid	Binary Cross Entropy

	Experiment Iteration		58	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.797171891
	5474	505	PPV	0.781290591
	FN	TP	ACC	0.883038104
	459	1804	F1	**0.789151356**

Bi-LSTM Activation	Bi-LSTM Dropout	Bi-LSTM Recurrent Activation	Bi-LSTM Recurrent Dropout	Bi-LSTM Layers
relu	0	sigmoid	0.5	2
Learning Rate	**Epochs**	**Optimizer**	**Last Activation**	**Losses**
0.5	1000	Adam	sigmoid	Binary Cross Entropy

Table 68- Confusion Matrix RNN Bi-LSTM Optimization Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 46, 58. Source: Author

From the results presented in Table 67 and Table 68, the best performance model is experiment 33 with F1 Score 0.7983. The figure below presents the model.

Figure 36 presents the model.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional (None, 2, 2048)           10444800
_________________________________________________________________
bidirectional_1 (Bidirection (None, 2048)              25174016
_________________________________________________________________
dense (Dense)                (None, 1)                 2049
=================================================================
Total params: 35,620,865
Trainable params: 35,620,865
Non-trainable params: 0
```

Figure 36- RNN Bi-LSTM Optimization Model Experiment 33. W2V 501 Matrix Preprocessing. Source: Author

This experiment has provided the best KPI's and has minimized type I and type II error in the confusion matrix, increasing up to 0.7983 the F1 Score. As a result we have concluded the hyperparameter estimation for this preprocessing method and architecture. The table below summarize the selected hyperparameters to promote to the next phase of the investigation.

Hyperparameters Selection: W2V 501 Preprocessing – RNN Bi-LSTM Network		
Parameter	**Parameter ID**	**Combination Values**
LSTM Internal Activation Function	**LSTM_activation**	**['relu']**

Hyperparameters Selection: W2V 501 Preprocessing – RNN Bi-LSTM Network		
Parameter	**Parameter ID**	**Combination Values**
LSTM Internal Recurrent Activation Function	**LSTM_recurrent_activation**	**['sigmoid']**
LSTM Internal Dropout	**LSTM_dropout**	**[0]**
LSTM Internal Recurrent Dropout	**LSTM_recurrent_dropout**	**[0.5]**
Learning Rate	**lr**	**[0.5]**
LSTM Units	**first_neuron**	**[1024]**
LSTM Layers	**hidden_layers**	**[2]**
Batch Size	**batch_size**	**[32]**
Epochs	**Epochs**	**[100]**
Optimizer	**Optimizer**	**[Adam]**
Losses	**Losses**	**['binary_crossentropy']**
Last Activation	**last_activation**	**[sigmoid]**

Table 69- Hyperparameters Selection W2V 501Matrix Preprocessing – RNN Bi-LSTM. Source: Author

10.2.8 RNN Bi Directional LSTM Architecture with Dense Output Layer – W2V 501 Matrix Data Set

As a final adjustment to this experimental array, we have included a dense layer between the Bi-Directional LSTM hidden layers and the output layer. This in order to increase the effectiveness of the experiment and follow best practices. The experiment will only replicate the last hyperparameter training scenario presented in section 10.2.7, using parameters defined in *Table 71*. The figure below presents the model.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional (Bidirectional (None, 2, 2048)           10444800
_________________________________________________________________
bidirectional_1 (Bidirection (None, 2048)              25174016
_________________________________________________________________
dense (Dense)                (None, 512)               1049088
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 36,668,417
Trainable params: 36,668,417
Non-trainable params: 0
```

Figure 37- RNN Bi-LSTM Architecture Model Experiment With Dense Layer. W2V 501 Matrix Preprocessing. Source: Author

As expected, we see improvements in all performance KPI's, specifically the F1 Score which reached a value of 0.8032. The following table presents the results.

	Experiment Iteration		60	
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.838709652
	5414	565	PPV	0.770604968
	FN	TP	ACC	0.887163281
	365	1898	F1	**0.803216251**
Bi-LSTM Activation	Bi-LSTM Dropout	Bi-LSTM Recurrent Activation	Bi-LSTM Recurrent Dropout	Bi-LSTM Layers
relu	0	sigmoid	0.5	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	100	Adam	sigmoid	Binary Cross Entropy

Table 70- Confusion Matrix RNN Bi-LSTM Optimization Hyperparameters Estimation Result W2V 501 Matrix Preprocessing 60. Source: Author

This will be the model selected for this specific approach in order to promote to the test evaluation phase.

10.3 Hyperparameter Estimation Summary

This section summarizes the hyperparameter estimation process and results. The results of the hyperparameter estimation correspond to a total of 1,453 experiments where hyperparameters were estimated accordingly. In addition, the estimation was carried out using a systematic approach for each of the preprocessing proposals and the neural network architecture. At the end we have five (5) models that will be trained with the 100% of the Train Data Set. Then each of these will be tested with the corresponding Test Data Sets, and furthermore they will be evaluated using the Golden Standard Data Set.

Hyperparameter Experiment Count Combinations				
Pre-Processing Approach	Selected NN Architecture	Architecture Hyperparameters Estimation	Optimization Hyperparameters Estimation	Total
DL	MLP	90	924	1014
W2V	MLP	36	63	99
DL	LSTM	81	42	123
W2V	LSTM	94	33	127
W2V	BiLSTM	41	49	90

Table 71- Hyperparameter Experiment Count Iterations. Source: Author

F1 Score On Validation 10% Data Set				
Pre-Processing Approach	**Selected NN Architecture**	**Architecture Hyperparameters Estimation**	**Optimization Hyperparameters Estimation**	**Best Model Hyperparameters Estimation**
DL	**MLP**	0.7353	0.8088	770
W2V	**MLP**	0.682	0.8087	96
DL	**LSTM**	0.8227	0.8258	38
W2V	**LSTM**	0.7952	0.8052	40
W2V	**BiLSTM**	0.7975	0.8032	60

Table 72- F1 Score KPI Results for Hyperparameter Best Model Selection – Validation 10% Data Set. Source: Author

Also, we point out in the results the False Positive class result for each of the above F1 Score KPI for each model. From this analysis we can early conclude that the DL Similarity Matrix preprocessing approach helps minimize the merge of different products, that have been labeled as duplicates (false positives). But the W2V 501 preprocessing approach has a much better overall result.

False Positives On Validation 10% Data Set				
Pre-Processing Approach	**Selected NN Architecture**	**Architecture Hyperparameters Estimation**	**Optimization Hyperparameters Estimation**	**Best Model Hyperparameters Estimation**
DL	**MLP**	33	36	770
W2V	**MLP**	425	534	96
DL	**LSTM**	91	100	38
W2V	**LSTM**	541	510	40
W2V	**BiLSTM**	528	565	60

Table 73- False Positives Results for Hyperparameter Best Mode Selection – Validation 10% Data Set. Source: Author

11 Train, Test, and Golden Standard Model Evaluation

The following sections will evaluate the performance of each of the five proposed models using the Test Data Sets, which contain fresh data according to the data corpus preprocessing and preparation performed in section 8.4. Prior to the test evaluation, each model will be trained with the corresponding data sets using the 100% of the training records. Results for training and testing will be presented for each model proposal independently. In addition, all five models will be evaluated using the Golden Record Standard data sets as an extra evaluation. Finally a summary comparing all the models will be presented.

11.1 Multi-layer perceptron / DL Similarity Matrix Preprocessing

In this section we will present the results of testing this model with the corresponding test and golden standard data sets. First the model will be trained and the tested.

11.1.1 MLP - DL Similarity Matrix Preprocessing Train Evaluation

As presented in section 10.2.2 the best the select hyperparameters for this model produced a F1 Score KPI result of 0.8088. We will train this model using the TrainDataSet described in section 9.2.2.

11.1.2 MLP - DL Similarity Matrix Preprocessing Test Evaluation

After training and saving this model, it has been evaluated with the corresponding TestDataSet. The table below presents the evaluation result.

MLP-DL	Testing Iteration with Data Set: TestDataSetDL 35,564 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.725510415
	25401	465	PPV	0.938008266
	FN	TP	ACC	0.912074007
	2662	7036	F1	**0.818187104**
Dropout	**Optimizer**	**Learning Rate**	**Neurons per Layer**	**Activation Function**
0	Nadam	0.0055	10	selu
Losses	**Last Activation**	**Batch Size**	**Epochs**	**Hidden Layers**
Binary Cross Entropy	sigmoid	1000	100	4

Table 74- Confusion Matrix MLP DL Similarity Matrix Test Evaluation. Source: Author

11.1.3 MLP - DL Similarity Matrix Preprocessing Golden Standard Evaluation

Finally, the saved model has also been evaluated using the Golden Standard data set. The following table presents this evaluation result.

MLP-DL	All GS Testing with Data Set: all_gsDataSetDL 4,400 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.685
	3075	125	PPV	0.868004224
	FN	TP	ACC	0.885681818
	378	822	F1	**0.765719609**
Dropout	**Optimizer**	**Learning Rate**	**Neurons per Layer**	**Activation Function**
0	Nadam	0.0055	10	selu
Losses	**Last Activation**	**Batch Size**	**Epochs**	**Hidden Layers**
Binary Cross Entropy	sigmoid	1000	100	4

Table 75- Confusion Matrix MLP DL Similarity Matrix Test Evaluation. Source: Author

From the tables above we can conclude that this model, using a DL Similarity Matrix preprocessing technique, with a MLP Neural Network has very consistent performance when evaluated with fresh data sets. The F1 Score KPI from training results to test evaluation variated in 0.05%, and compared to the Golden Standard evaluation result in -6.79%.

11.2 Multi-layer perceptron / W2V_501 Matrix Preprocessing

In this section we will present the results of the training and testing this model with the corresponding data sets. First the model will be trained and the tested.

11.2.1 MLP – W2V_501 Preprocessing Training

As presented in section 10.2.3 the best the select hyperparameters for this model produced a F1 Score KPI result of 0.8087. We will train this model using the TrainDataSet described in section 9.2.2.

11.2.2 MLP – W2V_501 Preprocessing Test Evaluation

After training and saving this model, it has been evaluated with the corresponding TestDataSet. The table below presents the evaluation result.

MLP-W2V_501	Testing Iteration with Data Set: TestDataSetW2V_501 35,564 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.835120643
	23625	2241	PPV	0.783268859
	FN	TP	ACC	0.892025644
	1599	8099	F1	**0.808364108**
Dropout	Optimizer	Learning Rate	Neurons per Layer	Activation Function
0.3333	Adadelta	0.5	500	relu
Losses	Last Activation	Batch Size	Epochs	Hidden Layers
Binary Cross Entropy	sigmoid	1000	1000	10

Table 76- Confusion Matrix MLP W2V_501 Matrix Test Evaluation. Source: Author

11.2.3 MLP – W2V_501 Preprocessing Golden Standard Evaluation

Finally, the saved model has also been evaluated using the Golden Standard data set. The following table presents this evaluation result.

MLP-W2V_501	All GS Testing with Data Set: all_gsDataSetW2V_501 4,400 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.6575
	1956	1244	PPV	0.388096409
	FN	TP	ACC	0.623863636
	411	789	F1	**0.488091556**
Dropout	Optimizer	Learning Rate	Neurons per Layer	Activation Function
0.3333	Adadelta	0.5	500	relu
Losses	Last Activation	Batch Size	Epochs	Hidden Layers
Binary Cross Entropy	sigmoid	1000	1000	10

Table 77- Confusion Matrix MLP W2V_501 Matrix Golden Standard Evaluation. Source: Author

From the tables above we can conclude that this model, using a W2V_501 Matrix preprocessing technique, with a MLP Neural Network does not have a consistent performance when evaluated with fresh data sets. First we can see that the F1 Score KPI from training results to test evaluation variated in -14.51%, and compared to the Golden Standard evaluation result in -48.38%. This means that this model has overfitted training results and, due to its hyperparameters, specifically the batch size, when presented to a small evaluation data set as the Golden Standard, the performance reduces drastically.

11.3 Long Short Term Memory – DL Similarity Matrix Preprocessing

In this section we will present the results of the training and testing this model with the corresponding data sets. First the model will be trained and the tested.

11.3.1 LSTM – DL Similarity Preprocessing Training

As presented in section 10.2.4 the best the select hyperparameters for this model produced a F1 Score KPI result of 0.8258. We will train this model using the TrainDataSet described in section 9.2.4.

11.3.2 LSTM – DL Similarity Preprocessing Test Evaluation

After training and saving this model, it has been evaluated with the corresponding TestDataSet. The table below presents the evaluation result.

LSTM-DL	Testing Iterarion with Data Set: TestDataSetDL 35,564 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.648690452
	25827	39	PPV	0.993838863
	FN	TP	ACC	0.903104263
	3407	6291	F1	**0.785001248**
LSTM Activation	**LSTM Dropout**	**LSTM Recurrent Activation**	**LSTM Recurrent Dropout**	**LSTM Layers**
tanh	0	sigmoid	0	3
Learning Rate	**Epochs**	**Optimizer**	**Last Activation**	**Losses**
0.5	1000	Adam	sigmoid	Binary Cross Entropy

Table 78- Confusion Matrix LSTM DL Similarity Matrix Test Evaluation. Source: Author

11.3.3 LSTM – DL Similarity Preprocessing Golden Standard Evaluation

Finally, the saved model has also been evaluated using the Golden Standard data set. The following table presents this evaluation result.

LSTM-DL	All GS Testing with Data Set: all_gsDataSetDL 4,400 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.565
	3176	24	PPV	0.965811966
	FN	TP	ACC	0.875909091
	522	678	F1	**0.712933754**
LSTM Activation	LSTM Dropout	LSTM Recurrent Activation	LSTM Recurrent Dropout	LSTM Layers
tanh	0	sigmoid	0	3
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	1000	Adam	sigmoid	Binary Cross Entropy

Table 79- Confusion Matrix LSTM DL Similarity Matrix Golden Standard Evaluation. Source: Author

From the tables above we can conclude that this model, using a DL Similarity Matrix preprocessing technique, with a LSTM Recurrent Neural Network has consistent performance when evaluated with fresh data sets. First we can see that the F1 Score KPI from training results to test evaluation variated in 0.21%, and compared to the Golden Standard evaluation result in -8.99%. This means that this model has a good generalization to identify duplicates on different data sets. In addition, we can observed that it constantly has the lowest value of false positives, in our experiment the class we will like to minimize.

11.4 Long Short Term Memory – W2V_501 Matrix Preprocessing

In this section we will present the results of the training and testing this model with the corresponding data sets. First the model will be trained and the tested.

11.4.1 LSTM – W2V_501 Matrix Preprocessing Training

As presented in section 10.2.6 the best the select hyperparameters for this model produced a F1 Score KPI result of 0.8052. We will train this model using the TrainDataSet described in section 9.2.4.

11.4.2 LSTM – W2V_501 Preprocessing Test Evaluation

After training and saving this model, it has been evaluated with the corresponding TestDataSet. The table below presents the evaluation result.

LSTM-W2V_501	Testing Iterarion with Data Set: TestDataSetW2V_501 35,564 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.827077748
	23936	1930	PPV	0.806049643
	FN	TP	ACC	0.898577213
	1677	8021	F1	**0.816428317**
LSTM Activation	LSTM Dropout	LSTM Recurrent Activation	LSTM Recurrent Dropout	LSTM Layers
relu	0	sigmoid	0.5	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	1000	Adam	sigmoid	Binary Cross Entropy

Table 80- Confusion Matrix LSTM W2V_501 Matrix Test Evaluation. Source: Author

11.4.3 LSTM – W2V_501 Preprocessing Golden Standard Evaluation

Finally, the saved model has also been evaluated using the Golden Standard data set. The following table presents this evaluation result.

LSTM-W2V_501	All GS Testing with Data Set: all_gsDataSetW2V_501 4,400 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.64
	1933	1267	PPV	0.377395577
	FN	TP	ACC	0.613863636
	432	768	F1	**0.474806801**
LSTM Activation	LSTM Dropout	LSTM Recurrent Activation	LSTM Recurrent Dropout	LSTM Layers
relu	0	sigmoid	0.5	2
Learning Rate	Epochs	Optimizer	Last Activation	Losses
0.5	1000	Adam	sigmoid	Binary Cross Entropy

Table 81- Confusion Matrix LSTM W2V_501 Matrix Golden Standard Evaluation. Source: Author

From the tables above we can conclude that this model, using a W2V_501 Matrix preprocessing technique, with a LSTM Recurrent Neural Network does not have a consistent performance when

evaluated with fresh data sets. First we can see that the F1 Score KPI from training results to test evaluation variated in -14.17%, and compared to the Golden Standard evaluation result in -50.09%. This means that this model has overfitted training results and, due to its hyperparameters, specifically the training epochs, when presented to a small evaluation data set as the Golden Standard, the performance reduces drastically. Also it is important to annotate that the this model has a better performance on reducing false negatives than on false positives. In terms of business objectives, it is desired to not mark as duplicates two different products, than mark to duplicates as non-duplicate products and presenting as two different master records throughout the value chain.

11.5 Bidirectional Long Short Term Memory – W2V_501 Matrix Preprocessing

In this section we will present the results of the training and testing this model with the corresponding data sets. First the model will be trained and the tested.

11.5.1 Bi-LSTM – W2V_501 Matrix Preprocessing Training

As presented in section 10.2.8 the best the select hyperparameters for this model produced a F1 Score KPI result of 0.8052. We will train this model using the TrainDataSet described in section 9.2.4.

11.5.2 Bi-LSTM – W2V_501 Preprocessing Test Evaluation

After training and saving this model, it has been evaluated with the corresponding TestDataSet. The table below presents the evaluation result.

Bi-LSTM-W2V_501	Testing Iterarion with Data Set: TestDataSetW2V_501 35,564 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.819962879
	23813	2053	PPV	0.794802599
	FN	TP	ACC	0.893178495
	1746	7952	F1	**0.807186723**
LSTM Activation	**LSTM Dropout**	**LSTM Recurrent Activation**	**LSTM Recurrent Dropout**	**LSTM Layers**
relu	0	sigmoid	0.5	2
Learning Rate	**Epochs**	**Optimizer**	**Last Activation**	**Losses**
0.0005	100	Adam	sigmoid	Binary Cross Entropy

Table 82- Confusion Matrix Bi-LSTM W2V_501 Matrix Test Evaluation. Source: Author

11.5.3 Bi-LSTM – W2V_501 Preprocessing Golden Standard Evaluation

Finally, the saved model has also been evaluated using the Golden Standard data set. The following table presents this evaluation result.

Bi-LSTM-W2V_501	All GS Testing with Data Set: all_gsDataSetW2V_501 4,400 Records			
	Actual Class		KPI's	
Predicted Class	TN	FP	TPR	0.625
	1951	1285	PPV	0.368550369
	FN	TP	ACC	0.608881876
	450	750	F1	**0.463678516**
LSTM Activation	**LSTM Dropout**	**LSTM Recurrent Activation**	**LSTM Recurrent Dropout**	**LSTM Layers**
relu	0	sigmoid	0.5	2
Learning Rate	**Epochs**	**Optimizer**	**Last Activation**	**Losses**
0.0005	100	Adam	sigmoid	Binary Cross Entropy

Table 83- Confusion Matrix Bi-LSTM W2V_501 Matrix Golden Standard Evaluation. Source: Author

From the tables above we can conclude that this model, using a W2V_501 Matrix preprocessing technique, with a Bidirectional LSTM Recurrent Neural Network does not have a consistent performance when evaluated with fresh data sets. First we can see that the F1 Score KPI from training results to test evaluation variated in -14%, and compared to the Golden Standard evaluation result in -50.6%. This means that this model has overfitted training results and, due to its hyperparameters, specifically the training epochs, when presented to a small evaluation data set as the Golden Standard, the performance reduces drastically.

12 RESULTS ANALYSIS

12.1 ANALYSIS ON MODEL'S EVALUATIONS

Comparing the results on all five model approaches, using the F1 Score as main performance KPI, we can see that the most consistent is the MLP – DL Similarity Matrix Preprocessing model, getting the best results in F1 Score KPI with the Test and Golden Standard evaluation data sets. It also presented that the variation of performance among the three data sets is the lowest one. This can be observed in the following chart.

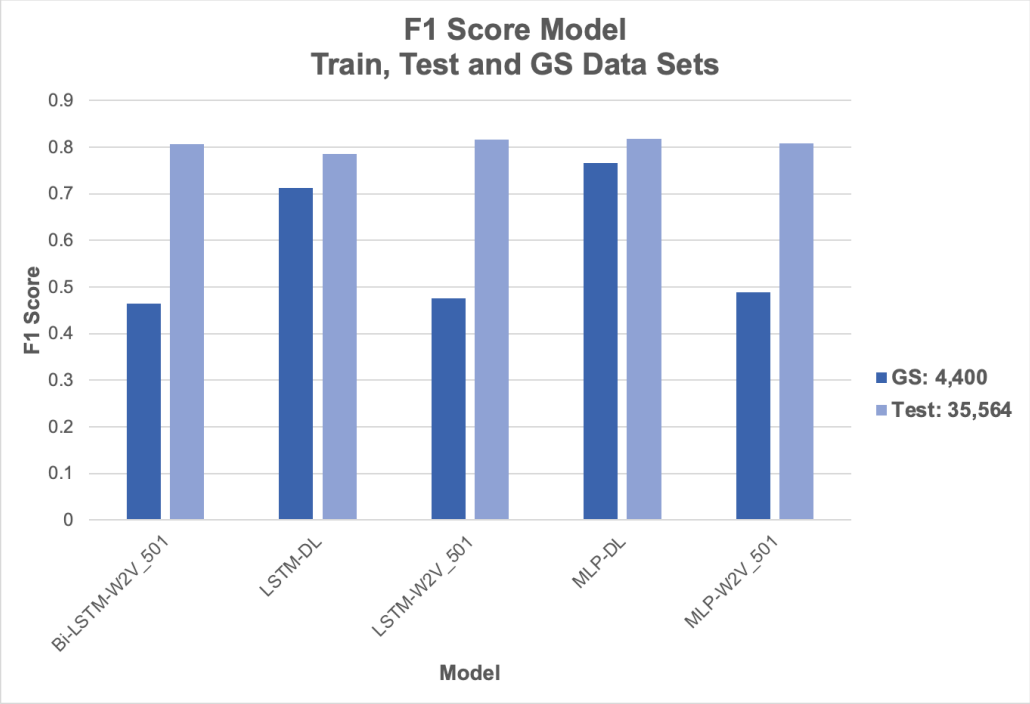


Chart 27- F1 Score KPI Results per Model and Data Set. Source: Author

Also from the above chart we can observe that the models using the W2V_501 preprocessing approach, where just the "Title" attribute pair was fed to the corresponding networks they have a F1 Score KPI results score just above 0.8, performing just about the same as the best performant model. In addition it must be noted that these three models, MLP-W2V_501, LSTM-W2V_501, and BiLSTM-W2V_501, reduce drastically they performance when evaluated with the Golden Standard Data Set. The F1 Score decreases to 0.47.

In the following chart we can observe the behavior of the other three performance KPIs analyzed to select the best performing model. These are:

- ACC-Binary Accuracy of the classifier
- PPV-Precision
- TPR-Recall

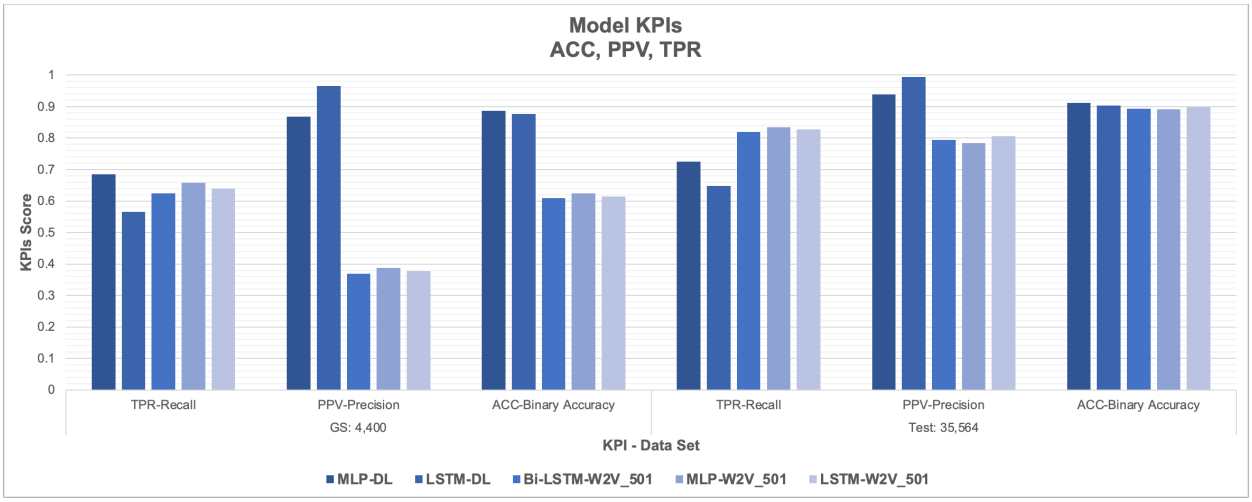


Chart 28- ACC, PPV, TPR Score KPI Results per Model and Data Set. Source: Author

From the chart above, we can see that the PPV and ACC KPIs on Golden Standard and Test data sets for models using the DL-Similarity Preprocessing Matrix approach in both proposed architectures (MLP and LSTM), outperform the other three. We can see that this is due to the fact that the models using W2V_501 preprocessing method, with just a the “Title” attribute as feature to be provided to the network does not classify correctly the “true” class, duplicate products, increasing the false positives. It also presents the fact that these models minimize the false negative class instead.

This result can be detailed in the following chart where we can see that the false negative results are practically the same across all models in evaluated using the Golden Standard data set, but the false positives grow systematically across the models using the W2V_501 preprocessing method, with just a the “Title” attribute as feature. Additionally, we can observe in the Test data sets that the false negatives are significantly larger in the models using the DL Similarity Matrix preprocessing approach, with results almost two times as big as the model using W2V_501 preprocessing method. But when we depict the behavior for false positives, the models using W2V_501 preprocessing method score values at least five times larger than the other two.

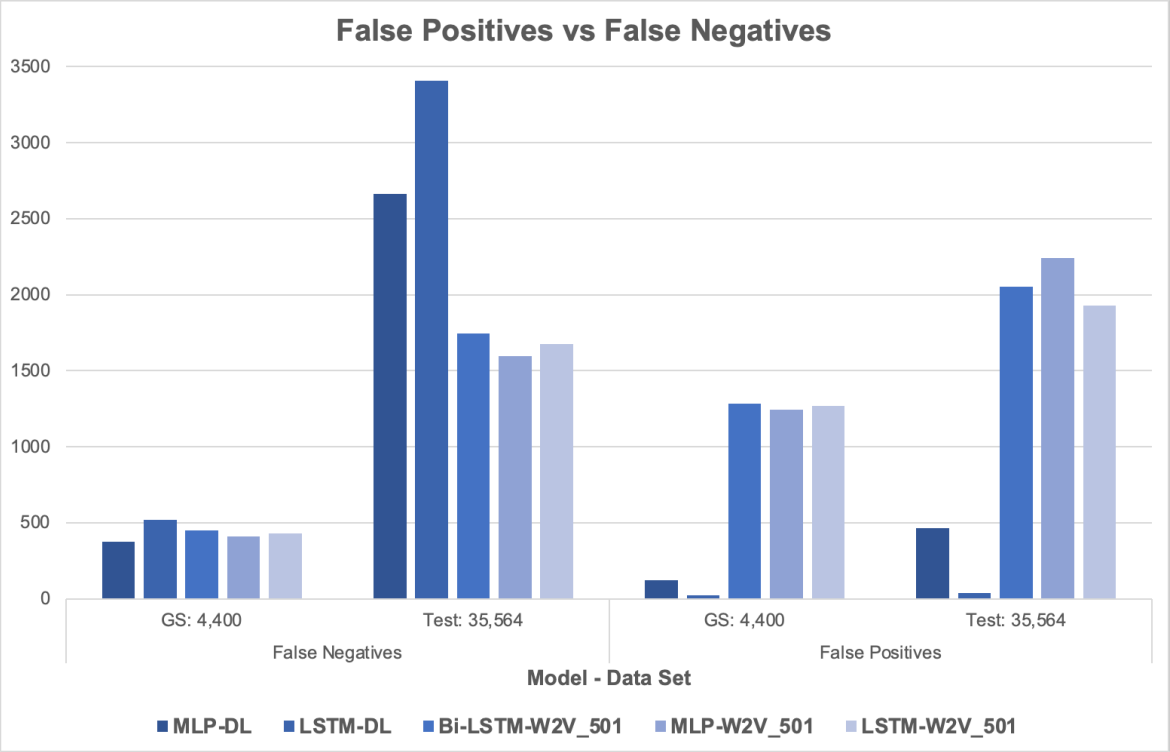


Chart 29- False Negatives vs. False Positives per Model and Data Set. Source: Author

Furthermore, we need to address the time consumption and complexity of each model in terms of number of parameters to use in the neural networks. During the training of each model, we recorded the time effort in order to complete the training task. The following table presents the number of parameters to estimate in each model and the time to complete the training task in seconds. In terms of training efficiency we can conclude that the MLP-DL model is the most efficient one. It achieved its KPI's in less time and with a small number of trainable parameters compared to the other models. It also stands out that even if the number of trainable parameters for the LSTM-DL model is fourteen times smaller, it takes up to four times more time to train. One of the reasons is that as presented in section 10.2.4, this model uses 1,000 epochs to train, whereas the Bi-LSTM-W2V_501 uses 100 epochs only.

Model	Training Time (s)	Trainable Parameters
MLP-DL	100	561
MLP-W2V_501	1,000	2,756,001
LSTM-DL	40,000	2,531
LSTM-W2V_501	60,515	3,793,409
Bi-LSTM-W2V_501	10,500	36,668,417

Table 84- Training Time and Trainable Parameters for each Model. Source: Author

12.2 Model Results Business Applications

The five models provided by this investigation are perfectible suitable to use in real world business applications. The first reality we need to address is that data sets, especially in Product Master Data and Website / E-Commerce Product and Offer's Data, will vary significantly. Sometimes you will have all data in different attributes or features, where you can explode the usage of the DL Similarity Matrix Preprocessing and use the MLP or LSTM model. As seen on section 12.1, these two models provide the all-around best KPI results, including the highest F1 Score among the Test and Golden Standard Data Sets, and minimizing the false positive's class. But in various business applications, specifically in the retail / marketplace business use case, where retailers open their product catalogs to sellers all around the globe, not every data entry will have such a rich data. Moreover, as presented in section 8.3.3, only the product Title feature and the product Category feature is present in all the records. This is a clear representation of real world behavior. Also, from the literature review, it has been proof that in e-commerce ecosystems, specifically in the retail industry, up to 80% of the product features and or attributes are included in the product Title. So when this real life scenario is present, the W2V_501 Preprocessing Method with the LSTM or Bi-LSTM network architectures produces excellent results. In the Test data set these models provide 0.8 or above F1 Score results, which is only decreased when tested with the Golden Standard. These models also produce a much better balance between false negatives and false positives. This behavior provides also business value, specially where the retailer wants to reduce the *"multi-product"* situation on their e-commerce platforms. As a final word on the five models produced in this investigation, is that each one can suite different business use cases and scenarios, providing a huge competitive advantage, as these models are ready to use and deploy within their ecosystems, and have proven satisfactory performance under various business scenarios.

13 Conclusions

13.1 Main Research Conclusion

The main objective of this investigation was to address, study, and propose a model that will de-duplicate product master data records using machine learning techniques. This model will aim to reduce the manual de-duplication review tasks, using a labeled data set for training and testing. This model should be provide performance KPI results in order to measure its effectiveness and possibilities to be applied as part of a product master "golden record" for e-Commerce, marketplace, and omnichannel application.

As it has been presented in the above sections of this document, this investigation has proposed five (5) different models. Each model has been developed using a systematic approach for selecting the solution proposal, building the proposed model, testing and evaluating the each suggested model. In addition, the main input for model's training and testing was the data corpus of labeled data. This data corpus fits accordingly the needs of this research as it addresses the exact same goal, describing if two product master data pair records are duplicates or not. Also, two different preprocessing methods have been studied, developed and applied in order to compare each one. Furthermore, two different solution architectures have been addressed, to increase the applicability and generalization of the proposed solution. The combination of the above resulted in the five proposed models, achieving the investigation main objective.

In summary according to the evaluation KPI's defined as metrics, the all-around best model is the DL Similarity Matrix Preprocessing / MLP Neural Network Architecture presented in section 11.1. Now as depicted in section 12.2, according to the business needs, and data structure available to compare product pair records and identify duplicates, this investigation results propose the following model decision table in order to find the best fitted model.

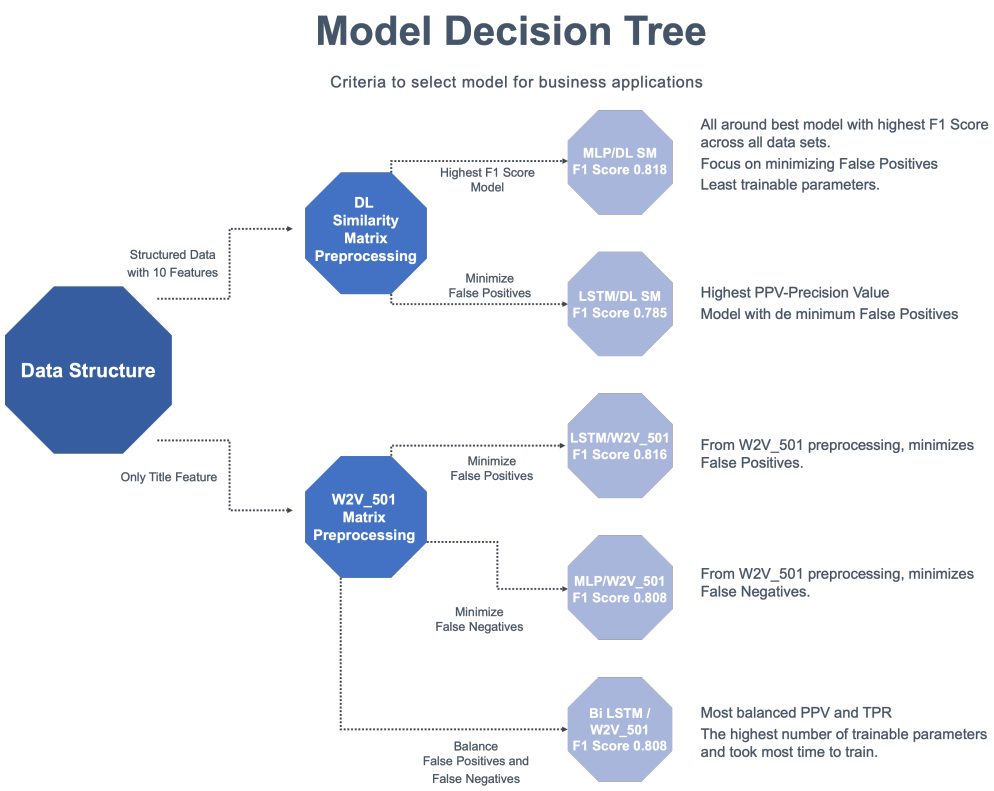


Figure 38- Model Selection Decision Tree. Based on available data structure and busines goals. Source: Author

From the proposed decision tree presented in Figure 38, we can easily navigate to the best solution model depending on the current data status and the desired outcome. First the decision tree allows to select a path according to the data structured available. Basically the first level of decision making takes into account if the available data is structured in a way that the 10 features can easily be identified, or if there is only the Title feature available. That points out to the selection of the preprocessing method, which can be DL Similarity Matrix, developed in this investigation, or the Word 2 Vec skipgram 501 method adapted in this investigation. Then the second level of decision involves the KPI that the business wants to pursue. At the end of the tree each model presents its benefits for the specific business application.

13.2 Specific Conclusions

13.2.1 Data Corpus

As detailed in section 8, a data corpus suited for the investigation objectives had to be selected. The WDC Product Data Corpus and Gold Standard for Large-Scale Product Matching - Version 2.0[3]

suited perfectly the investigation needs. This data set was studied, analyzed and described in order to provide the corresponding data sets for all the training, validating, and testing subsequent tasks. We were able to analyze the data corpus schema in section 8.2.1, in order to decided which features/attributes of the data corpus to use, and how did the attribute density could affect the investigation results. Also, in section 8.3.4 the data corpus balance between the two classes, duplicate or non-duplicate, was examined. This helped us construct correctly our data sets, honoring the exact same imbalanced behavior presented in the data corpus, which is of 27.27% of duplicates versus non-duplicate pairs.

In addition, as described in section 9 two different preprocessing methods were proposed in order to pre-process the data and compare results on the proposed solution models. The first method called *"Damerau-Levenshtein Similarity Matrix"* described in section 9.2.1 takes the each pair feature/attribute and compares its single-edit similarity ratio. With this approach a matrix for each pair record was built, with ten features that represents *"how similar"* does the specific feature of each pair record is, under the single-edit distance approach proposed in [45]. Also, in order to address the industry problem where just the product "Title" feature/attribute is available, and taking into account that according to researches 80% of a product attribution can be found in its title/description, we proposed to use the *"Word 2 Vec"* skipgram preprocessing technique, which is detailed in section 9.2.3. Implementing this preprocessing method we were able to transform each title feature of the pair record into a vector of 250 elements, building our W2V_501 matrixes for each data set.

Therefore, besides selecting a fitted data corpus, data sets for each phase of the investigation were built. These were systematically analyzed, described and constructed, using two different preprocessing methods, one that has been proposed in this investigation from scratch. As a result, ten (10) well suited data sets have been produce to use in the subsequent phases of the investigation.

13.2.2 Machine Learning Techniques for Solution Model

This investigation used two different machine learning approaches to solve this classification problem using self-trained techniques. In accordance to what the state of the art scientific literature proposes, neural networks techniques have been applied to propose solution models in this investigation. In detail two different architectures or neural networks were selected. The first one, the multi-layer perceptron feed forward with back propagation neural network architecture has been studied in order to build the proposed solutions. From the implementation of this model we found that even though, it is a simple architecture, it resulted to be very robust in order to solve the binary classification problem proposed in this investigation. As described in sections 10.1.2, 10.2.2, and 10.2.3, this model is simple to build, easy for hyperparameter estimation task accomplishment, and very light in terms of amount of trainable parameters and training time.

The second selected architecture to develop proposed solutions in this investigation is Recurrent Neural Network, specifically we selected Long Short Term Memory as it has been proof great application in natural language processing. This fact suits perfectly our goal of de-duplicating product master records just by analyzing its "Title/Description". This model approach has proof to be a very good fit to when processing large amount of data, even though they are expensive in terms of trainable parameters and training effort. Once the model has overcome the hyperparameter estimation phase, and the training, they generalize pretty well. In contrast of the models using MLP, these models provided better insight in decreasing false negative results, returning better performance KPI's, specifically PPV, TPR and F1 Score. These can be reviewed in sections 10.1.3, 10.2.4, 10.2.6, and 10.2.8. Furthermore, we were able to expand and test a more specific architecture proposal within the LSTM RNN, the bi-directional RNN. This architecture proof that even if it has a

huge amount of trainable parameters, during the hyperparameter estimation process, it did not required heavy amount of training epochs, in order to achieve similar KPI performance results compared to the regular LSTM architecture models.

Finally, we can summarize that two very well suited machine learning techniques, which were studied, selected, and implemented with a systematic engineering method, were addressed in the development and completion of this investigation, achieving the specific objective proposed.

13.2.3 Build a Solution Model using ML for Product De-Duplication

As it has been presented in sections 10.2.2, 10.2.3, 10.2.4, 10.2.6, and 10.2.8 five solution models have been proposed in this investigation. They address the product master record de-duplication problem using two different preprocessing methods and three different type of neural network architectures. All of these models have been constructed using a systematic engineering method, where first the general hyperparameters applicable for each model were defined. Then a set of values on which each hyperparameter for each proposed model iterated have been proposed. This resulted in 1,453 experimental combinations which were evaluated during the experimentation phase. The main benefit of achieving this specific objective in this investigation was the possibility to propose different solution models depending on the available resources and information had at hand once we want to implement the solution. We have models that allow different attribution/features to be used as input parameters to solve the de-duplication task. These are the models that use the DL-Similarity Matrix Preprocessing approach. As a part of the concluding results provided in sections 8.3.3 and 8.3.4, we found that not all the attributes/features are statistically suited as input parameters. Basically this conclusion was drawn due to attribute density, as from a real life data corpus we found an attribute density pattern in the data corpus point to ten (10) basic attributes/features in the pair record which will help us de-duplicate the record. With this approach when these attributes are at hand, the DL-Similarity Matrix Preprocessing method can be used to feed any of the two proposed ML architectures, MLP and LSTM. From the results these two architectures have very similar performance. On the other hand, we have proposed three models that will address the de-duplication task when just the product master record “Title” attribution/feature is available as input parameter of the pair record evaluated. This expands the possibility to achieve similar performance results which a less data input parameters at hand. These models use the W2V_501 Matrix preprocessing method, leveraging from TensorFlow Hub[48] Word 2 Vec 250 preprocessing algorithm. As presented in sections 10.2.5, 10.2.6, 10.2.7, and 10.2.8, two models using LSTM and Bi-LSTM architectures were built, trained and evaluated to achieve this specific experimental array. Furthermore a fifth model combining MLP and W2V_501 preprocessing method, combining the two approaches depicted above, was proposed.

Concluding, five solution models have been propose as result of this investigation. Each one has addresses the product master record de-duplication problem, allowing different data preprocessing methods and achieving results with different input parameters available to execute the task. These can be used independently from each other or even in combination according to the required needs.

13.2.4 Evaluate the Proposed Solution Models

As presented with great amount of detail in section 11, all the models were trained, tested and evaluated with a golden standard data set. Performance KPI’s were used to rank each model according to desired result stated in the problem of this investigation. We found that the best performance KPI’s were achieved by the Multi-Layer Perceptron Neural Network, which used DL-Similarity Matrix preprocessing method. This conclusion was made upon the fact that across the training, test and golden standard data sets, this model is the one the on average performed the best

with a 0.80 F1 Score performance KPI. The runner up model is the Long Short Term Memory Recurrent Neural Network model, using the DL-Similarity Matrix preprocessing method, which scored on average among the three data sets an F1 Score of 0.76. In addition, as presented in section 12.1, these models present the lower false positive values. In term of business, these means that these models will provide less errors of marking a pair record as the same product master data record, a duplicate, which is the desired behavior in this specific industry application. From these we can also extrapolate that having independent features to be inputted to the neural networks, regardless of the selected architecture, performs better, as the result KPI's present stable results among all the data sets.

But even if the models using the W2V_501 preprocessing method, in conjunction with the LSTM, Bi-LSTM and MLP architectures, do not performed well with the golden standard data set, they performed extremely good with the Test Data Set, achieving a 0.81 F1 Score. The challenge with this models is that the false positive class is pretty high compared to the other two models. This was addressed in section 12.1, specifically in Chart 29. As part of the proposed follow ups for this investigation, a detail study should address this particular result when using just the title attribute as input for the de-duplication task. Many hypothesis can be stated. Is the pattern sequence of the words in the title the same in duplicates and non-duplicates records? Is there really available product characteristics in the product title and these are suited in a pattern that will help the models de-duplicate correctly?

As a conclusion, not just a single model has been proposed and evaluated. Five (5) proposed solution model have been evaluated and rank using the exact same data sets, methods and KPI's. Furthermore, we have pointed different approaches where each model can suit better the product master record de-duplication task. In addition, under specific circumstances, and available data, we have ranked the MLP-DL model as the best performer of all, taking into account that it has the bets F1 Score among all data sets and it minimizes the false positive class.

13.3 Achievement of Expected Results

Stated in this investigation pre-project document, there was four expected results. The first one was to acquire, study and preprocess a data corpus suited to achieve the main objective of this investigation. As you can observe in sections 8, 9, and 13.2.1, this result has been achieved. The second expected result that has been achieved is the proposal of a solution model using ML. As it has been presented in sections **Error! Reference source not found.**, 11, and 13.2.3, five solution m odels have been proposed as part of this investigation. They use two different types of neural network architectures, and two data preprocessing methods. The third expected result of this investigation that has also been achieved is the systematic evaluation of the proposed models. Depicted in sections 12.1, and 13.2.4 the models have been evaluated using performance KPI's under an engineering method, making a quantitative evaluation and also a qualitative evaluation.

13.4 Further Investigation Recommendations

After finalizing and concluding on this research project, several questions rose that can lead to further investigation on this subject, based on the results proposed in this investigation. Here a list of the most relevant questions and topics that will make a good fit to continue exploring in the future.

- Can these models be applied one after the other in order to have a second round and better results?

- Is it the precondition that pair records belong to the same product category something worth investigating? And can that precondition be applied in real business scenarios?
- In terms of the selected neural network architectures, can a hybrid model stacking a LSTM layer with a MLP perform better?

These three questions are good fits to extend the present results and findings, in order to continue with a more deep understanding and solutioning of product master data de-duplication.

14 BIBLIOGRAPHY

[1] K. Singh, G. Gupta, G. Shroff, and P. Agarwal, “Automated product-attribute mapping,” 21st Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2017 held in conjuction with the Workshop on Machine Learning for Sensory Data Analysis, MLSDA 2017, Workshop on Biologically Inspired Data-Mining Techniques, BDM 2017, Pacific Asi, vol. 10526 LNAI. Springer Verlag, TCS Research, Gurgaon, 122003, India, pp. 163–175, 2017.

[2] P. Petrovski, V. Bryl, and C. Bizer, “Learning regular expressions for the extraction of product attributes from e-commerce Microdata,” in 2nd International Workshop on Linked Data for Information Extraction, LD4IE 2014, Co-located with the 13th International Semantic Web Conference, ISWC 2014, 2014, vol. 1267, pp. 45–54.

[3] R. Peeters, A. Primpeli, and C. Bizer, “WDC Product Data Corpus and Gold Standard for Large-Scale Product Matching - Version 2.0,” University of Mannheim, 2019. [Online]. Available: http://webdatacommons.org/largescaleproductcorpus/v2/index.html.

[4] E. I. Sanchez Kerguelen, M. F. Quiñonez, and J. Carmona, “Prospectiva para el comercio electrónico en Colombia,” Bogota, 2019.

[5] I. Van Dam, N. Nijenhuis, G. Van Ginkel, D. Vandic, W. Kuipers, and F. Frasincar, “Duplicate detection in web shops using LSH to reduce the number of computations,” in 31st Annual ACM Symposium on Applied Computing, SAC 2016, 2016, vol. 04-08-Apri, pp. 772–779.

[6] A. Hartveld et al., “An LSH-based model-words-driven product duplicate detection method,” 30th International Conference on Advanced Information Systems Engineering, CAiSE 2018, vol. 10816 LNCS. Springer Verlag, Erasmus University Rotterdam, PO Box 1738, Rotterdam, 3000 DR, Netherlands, pp. 409–423, 2018.

[7] F. Haneem, R. Ali, N. Kama, and S. Basri, “Resolving data duplication, inaccuracy and inconsistency issues using Master Data Management,” in International Conference on Research and Innovation in Information Systems, ICRIIS, 2017.

[8] V. Dsilva et al., “Forecast: Infrastructure Software Markets, Worldwide, 2017-2023, 1Q19 Update,” 2019.

[9] C. Sapp, “Laying the Foundation for Artificial Intelligence and Machine Learning: A Gartner Trend Insight Report (ID:G00373110),” 2018.

[10] N. Shen, A. Linden, C. J. Idoine, and J. Hare, “Six Pitfalls to Avoid When Planning Data Science and Machine Learning Projects (ID:G00325649),” 2017.

[11] Centro Nacional de Consultoría, Observatorio e-Commerce, and Camara Colombiana de Comercio Electrónico, “Medición de Indicadores de consumo del Observatorio eCommerce,” Bogota, 2019.

[12] BlackSip, “BlackIndex: REPORTE DEL ECOMMERCE EN COLOMBIA,” Bogota, 2019.

[13] B. Otto, “Quality and Value of the Data Resource in Large Enterprises,” Inf. Syst. Manag., vol. 32, no. 3, pp. 234–251, 2015.

[14] A. V. Levitin and T. C. Redman, “Data as a Resource : Properties . Implications . and Prescriptions,” Sloan Manage. Rev., 1998.

[15] B. Heinrich and M. Klier, “Assessing data currency - A probabilistic approach,” J. Inf. Sci., 2011.

[16] A. Even and G. Shankaranarayanan, “Utility-driven assessment of data quality,” ACM SIGMIS Database, 2007.

[17] J. van den Hoven, “Information resource management: Stewards of data,” Inf. Syst. Manag., 1999.

[18] Dama, “No Title,” DAMA Dict. data Manag., 2008.

[19] V. Y. Yoon, P. Aiken, and T. Guimaraes, “Managing organizational data resources: quality dimensions,” Inf. Resour. Manag. J., 2000.

[20] B. Otto, “How to design the master data architecture: Findings from a case study at Bosch,” Int. J. Inf. Manage., vol. 32, no. 4, pp. 337–346, 2012.

[21] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, “Methodologies for data quality assessment and improvement,” ACM Comput. Surv., 2009.

[22] B. Otto, “Data governance,” Bus. Inf. Syst. Eng., vol. 3, no. 4, pp. 241–244, 2011.

[23] V. Khatri and C. V Brown, “Designing data governance,” Commun. ACM, vol. 53, no. 1, pp. 148–152, 2010.

[24] C. Oppenheim, J. Stenson, and R. M. S. Wilson, “Studies on information as an asset III: Views of information professionals,” J. Inf. Sci., 2004.

[25] B. Otto, K. M. Hüner, and H. Österle, “Toward a functional reference model for master data quality management,” Inf. Syst. E-bus. Manag., vol. 10, no. 3, pp. 395–425, 2012.

[26] H. A. Smith and J. D. McKeen, “Master Data Management: Salvation Or Snake Oil?,” Commun. AIS, 2008.

[27] K. Weber, B. Otto, and H. Österle, “Ones size does not fit all -A contingency approach to data governance,” J. Data Inf. Qual., vol. 1, no. 1, 2009.

[28] P. Weill and J. W. Ross, “IT governance; How top performers manage IT decisions rights for superior results,” in IT Governance, 2004.

[29] M. Mosley, M. Brackett, S. Earley, and D. Henderson, “No Title,” DAMA Guid. to Data Manag. Body Knowl., 2009.

[30] B. Otto, Y. W. Lee, and I. Caballero, “Information and data quality in business networking: A key concept for enterprises in its early stages of development,” Electron. Mark., vol. 21, no. 2, pp. 83–97, 2011.

[31] R. Vilminko-Heikkinen and S. Pekkola, “Establishing an Organization’s Master Data Management Function: A Stepwise Approach,” 2013 46th Hawaii International Conference on System Sciences. pp. 4719–4728, 2013.

[32] B. Otto and A. Reichert, “Organizing master data management: Findings from an expert survey,” in Proceedings of the ACM Symposium on Applied Computing, 2010, pp. 106–110.

[33] W. Yeoh and A. Koronios, “Critical success factors for business intelligence systems,” J. Comput. Inf. Syst., vol. 50, no. 3, pp. 23–32, 2010.

[34] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” Neurocomputing, 2017.

[35] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. The MIT Press, 2016.

[36] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural Comput., vol. 18,

no. 7, pp. 1527–1554, 2006.
[37] B. Liao, J. Xu, J. Lv, and S. Zhou, “An image retrieval method for binary images based on DBN and softmax classifier,” IETE Tech. Rev. (Institution Electron. Telecommun. Eng. India), vol. 32, no. 4, pp. 294–303, 2015.
[38] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., 1997.
[39] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, 1997.
[40] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” in Neural Networks, 2005.
[41] A. Graves, N. Jaitly, and A.-R. Mohamed, “Hybrid speech recognition with Deep Bidirectional LSTM,” in 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2013 - Proceedings, 2013, pp. 273–278.
[42] A. Kannan, I. E. Givoni, R. Agrawal, and A. Fuxman, “Matching unstructured product offers to structured product specifications,” in 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’11, 2011, pp. 404–412.
[43] I. 2020 Bar Code Graphics, “GTIN DEFINITION : INFORMATION,” GTIN INFO, 2020. [Online]. Available: https://www.gtin.info. [Accessed: 31-May-2020].
[44] S. Karpischek, F. Michahelles, and E. Fleisch, “The not so unique global trade identification number: Product master data quality in publicly available sources,” in ACM International Conference Proceeding Series, 2012, pp. 39–40.
[45] F. J. Damerau, “A technique for computer detection and correction of spelling errors,” Commun. ACM, vol. 7, no. 3, pp. 171–176, 1964.
[46] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, 2013.
[47] L. Boytsov, “Indexing methods for approximate dictionary searching: Comparative analysis,” ACM J. Exp. Algorithmics, vol. 16, 2011.
[48] Google, “wiki-words-250-with-normalization,” TensorFlow Hub Text Embedding, 2020. [Online]. Available: https://tfhub.dev/google/Wiki-words-250-with-normalization/2. [Accessed: 07-Aug-2020].
[49] Google, “tf.nn.embedding_lookup_sparse,” TensorFlow Hub Text Embedding, 2020. .
[50] W. Zong, F. Wu, L. K. Chu, and D. Sculli, “Identification of approximately duplicate material records in ERP systems,” Enterp. Inf. Syst., vol. 11, no. 3, 2017.
[51] H. Palangi et al., “Deep Sentence embedding using long short-term memory networks: Analysis and application to information retrieval,” IEEE/ACM Trans. Audio Speech Lang. Process., vol. 24, no. 4, pp. 694–707, 2016.
[52] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang, “Distributed representations of tuples for entity resolution,” in Proceedings of the VLDB Endowment, 2018, vol. 11, no. 11, pp. 1454–1467.
[53] N. Kooli, R. Allesiardo, and E. Pigneul, Deep Learning Based Approach for Entity Resolution in Databases, vol. 10752 LNAI. 2018.
[54] R. Vinayakumar, H. B. Barathi Ganesh, M. Anand Kumar, K. P. Soman, and P. Poornachandran, “DeepAnti-PhishNet: Applying deep neural networks for phishing email detection CEN-AISecurity@IWSPA-2018,” in CEUR Workshop Proceedings, 2018, vol. 2124, pp. 39–49.
[55] B. Agarwal, H. Ramampiaro, H. Langseth, and M. Ruocco, “A deep network model for paraphrase detection in short text messages,” Inf. Process. Manag., vol. 54, no. 6, pp. 922–937, 2018.
[56] B. Ye, G. Feng, A. Cui, and M. Li, “Learning Question Similarity with Recurrent Neural Networks,” in Proceedings - 2017 IEEE International Conference on Big Knowledge, ICBK 2017, 2017, pp. 111–118.
[57] E. Hunt et al., “Machine learning models for paraphrase identification and its applications on plagiarism detection,” in Proceedings - 10th IEEE International Conference on Big Knowledge, ICBK 2019, 2019, pp. 97–104.
[58] S. P. Singh, A. Kumar, H. Darbari, A. Rastogi, S. Jain, and N. Joshi, Building machine learning system with deep neural network for text processing, vol. 84. 2018.
[59] M. Kotila, “Talos: Hyperparameter Optimization for Keras.” github, 2020.
[60] D. M. W. Powers, “Evaluation: from precision, recall and f-factor,” Tech. Rep. SEI-07-001, 2007.
[61] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” BMC Genomics, 2020.
[62] T. Fawcett, “An introduction to ROC analysis,” Pattern Recognit. Lett., 2006.

15 APPENDIX

15.1 ATTACHMENTS

Bellow the list of attachments to this document is presented.

- GitHub URL:
    - https://github.com/JulioXa69/Thesis_Public_Repository/blob/Thesis/Tesis_Maestria_Ing_Julio_Hallo_201020022065_v3.ipynb