MACKENZIE PRESBITERIAN UNIVERSITY
POSTGRADUATE PROGRAM IN
ELECTRICAL AND COMPUTING ENGINEERING

Bruno Cesar Dos Santos Lima

Application of a Swarm Intelligence Algorithm in the Selection of Bible Passages to Compose a Sermon

Doctoral Thesis presented to the Postgraduate Program in Electrical Engineering and Computing of Presbyterian University Mackenzie as part of the requirements for the obtaining the title of Doctor in Electrical and Computer Engineering

Advisor: Prof° Dr. Ismar Frango

São Paulo
2024

L752a Lima, Bruno Cesar dos Santos

Application of a swarm intelligence algorithm in the selection of Bible passages to compose a sermon / Bruno Cesar dos Santos Lima.

120 f.

Tese (Doutorado em Engenharia elétrica e computação) – Universidade Presbiteriana Mackenzie, São Paulo, 2024.
Orientador: Prof. Dr. Ismar Frango Silveira

Bibliografia: f. 111-118

1. Swarm intelligence 2. Natural Language Processing, 3. Ant Colony Optimization 4. Holy Bible. I. Silveira, Ismar Frango II. Título

CDD 006.3

Bibliotecária responsável: Maria Gabriela Brandi Teixeira – CRB 8 / 6339

Folha de Identificação da Agência de Financiamento

Autor: Bruno Cesar dos Santos Lima

Programa de Pós-Graduação *Stricto Sensu* em Engenharia Elétrica e Computação

Título do Trabalho: APPLICATION OF A SWARM INTELLIGENCE ALGORITHM IN THE SELECTION OF BIBLE PASSAGES TO COMPOSE A SERMON

O presente trabalho foi realizado com o apoio de [1]:

☒ CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

☐ CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico

☐ FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo

☒ Instituto Presbiteriano Mackenzie/Isenção integral de Mensalidades e Taxas

☒ MACKPESQUISA - Fundo Mackenzie de Pesquisa

☐ Empresa/Indústria:

☐ Outro:

[1] **Observação**: caso tenha usufruído mais de um apoio ou benefício, selecione-os.

UNIVERSIDADE PRESBITERIANA MACKENZIE

BRUNO CESAR DOS SANTOS LIMA

APPLICATION OF A SWARM INTELLIGENCE ALGORITHM IN THE SELECTION OF BIBLE PASSAGES TO COMPOSE A SERMON

Tese apresentada como requisito parcial ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie para obtenção do título de Doutor em Engenharia Elétrica e Computação.

ORIENTADOR: Prof. Dr. Ismar Frango Silveira

Aprovado em: 16/02/2024

BANCA EXAMINADORA

Prof. Dr. Ismar Frango Silveira

Universidade Presbiteriana Mackenzie (Orientador)

Prof. Dr. Fabio Silva Lopes

Universidade Presbiteriana Mackenzie

Prof. Dr. Gustavo Scalabrini Sampaio

Universidade Presbiteriana Mackenzie

Prof. Dr. Vagner da Silva

Universidade Cruzeiro do Sul

Prof. Dr Leandro Nunes de Castro

Florida Gulf Coast University

ACKNOWLEDGEMENTS

To God, first and foremost, I express gratitude for the gift of life and wisdom.

To my family, my parents Nasare and Genivaldo, and my aunts, uncles, and cousins, who provided the necessary support for the completion of this project.

To Paulo Cunha, my mentor and encourager throughout my academic career.

To my friends Reinaldo and Klicia, for the essential help and companionship in overcoming challenges.

To postgraduate friend Cristiano Benites.

To my friends Nivaldo and Israel Avansi, for their friendship and cooperation throughout the entire academic research journey.

To my former advisor, Prof. Dr. Nizam Omar, for his guidance, advice, inspiration, and great friendship.

To my advisor Prof. Dr. Ismar Frango, for his guidance, teachings, and the hours dedicated to this thesis.

To Prof. Dr. Leandro Nunes de Castro, for his mentoring and the hours dedicated to the theoretical development of this work, and for the great partnership throughout the project.

To researchers Prof. Dr. Lúcia Saito, Prof. Dr. Rafael Euzébio, Prof. Dr. Leandro Augusto, and Prof. Dr. Hugo Fragnito, for their patience and assistance during this research journey.

To Yopanan Rocha, secretary of the graduate program, for his assistance and camaraderie.

To CAPES and Mackpesquisa for the financial support provided.

"*Do not try to be successful, try to be a man of value.*"
"*Não tentes ser bem-sucedido, tenta antes ser um homem de valor*"

Albert Einstein

"*The fear of the LORD is the beginning of wisdom, and the knowledge of the holy is prudence. (Proverbs 9:10)*"
"*O temor do Senhor é o princípio da sabedoria, e o conhecimento do Santo a prudência. (Provérbios 9:10)*"

Holy Bible

Bíblia Sagrada

DEDICATION

To God be the Glory! *A Deus seja a Glória!*

ABSTRACT

Religious literature is undoubtedly one of the most widely read types of literature by humanity, regardless of the confessional spectrum (Christians, Jews, Muslims, etc.). It is through religious literature that religious leaders communicate their values and ideas, and this communication is generally referred to as sermons or homilies. In the Christian context, the Holy Bible constitutes this normative corpus. However, the Holy Bible is not a trivial literature from a hermeneutical perspective, due to its high degree of literary and linguistic variability. Therefore, sermon construction can become a laborious activity. Faced with this challenge, this thesis aimed to implement a combinatorial optimization methodology for the selection of Bible passages that will compose pastoral sermons. This methodology uses a hybrid approach, i.e., the combination of natural language processing and swarm intelligence. The thesis proposes the implementation of a SwarmaBle algorithm that simulates a colony of artificial ants traversing a biblical graph to find the best solution. This solution involves returning a specific number of Bible passages for the composition of a sermon given an input sentence or theme. The success of this method could facilitate its application to other complex textual corpora for optimized and efficient information retrieval. The methodology is innovative, and the obtained results are robust and promising, this means that an experimental battery was conducted, in which an average convergence above 70% of the fitness function was observed. In this optimization process of SwarmaBle, the retrieved verses were found to be compatible with the search sentence. This correspondence between the verses and the search sentence was validated by domain experts (theologians), who assigned an above-average performance to SwarmaBle.

Key words: *Swarm intelligence, Natural Language Processing, Ant Colony Optimization, Holy Bible.*

RESUMO

A literatura religiosa é, sem dúvida, um dos tipos de literatura mais amplamente lidos pela humanidade, independentemente do espectro confessional (cristãos, judeus, muçulmanos, etc.). É por meio da literatura religiosa que líderes religiosos comunicam seus valores e ideias, e essa comunicação é geralmente referida como sermões ou homilias. No contexto cristão, a Bíblia Sagrada constitui esse corpus normativo. No entanto, a Bíblia Sagrada não é uma literatura trivial do ponto de vista hermenêutico, devido ao seu alto grau de variabilidade literária e linguística. Portanto, a construção de sermões pode se tornar uma atividade laboriosa. Diante desse desafio, esta tese teve como objetivo implementar uma metodologia de otimização combinatória para a seleção de passagens bíblicas que comporão sermões pastorais. Essa metodologia utiliza uma abordagem híbrida, ou seja, a combinação de processamento de linguagem natural e inteligência de enxames. A tese propõe a implementação de um algoritmo SwarmaBle que simula uma colônia de formigas artificiais percorrendo um grafo bíblico para encontrar a melhor solução. Essa solução envolve retornar um número específico de passagens bíblicas para a composição de um sermão, dado uma sentença de entrada ou tema. O sucesso desse método poderia facilitar sua aplicação a outros corpora textuais complexos para recuperação de informação otimizada e eficiente. A metodologia é inovadora, e os resultados obtidos são robustos e promissores. Isso significa que foi conduzida uma bateria de testes experimental, na qual foi observada uma convergência média acima de 70% da função fitness. Nesse processo de otimização do SwarmaBle, os versículos resgatados mostraram-se compatíveis com a sentença de busca. Essa correspondência entre os versículos e a sentença de busca foi validada por especialistas de domínio (teólogos), os quais atribuíram um desempenho acima da média para o SwarmaBle.

Palavras-chave: *Inteligência de Enxames, Processamento de Linguagem Natural, Otimização por Colônia de Formigas, Bíblia Sagrada.*

List of Figures

TABLES

List of Tables

GLOSSARY

Gen Genesis

Ex Exodus

Lev Leviticus

Num Numbers

Deut Deuteronomy

Josh Joshua

Judg Judges

Ruth Ruth

1 Sam 1 Samuel

2 Sam 2 Samuel

1 Kings 1 Kings

2 Kings 2 Kings

1 Chron 1 Chronicles

2 Chron 2 Chronicles

Ezra Ezra

Neh Nehemiah

Esth Esther

Job Job

Psalm or Ps Psalms

Prov Proverbs

Eccl Ecclesiastes

Song of Sol Song of Solomon

Isa Isaiah

Jer Jeremiah

Lam Lamentations

Ezek Ezekiel

Dan Daniel

Hos Hosea

Joel Joel

Amos Amos

Obad Obadiah

Jonah Jonah

Mic Micah

Nah Nahum

Hab Habakkuk

Zeph Zephaniah

Hag Haggai

Zech Zechariah

Mal Malachi

Matt Matthew

Mark Mark

Luke Luke

John John

Acts Acts

Rom Romans

1 Cor 1 Corinthians

2 Cor - 2 Corinthians

Gal - Galatians

Eph Ephesians

Phil Philippians

Col Colossians

1 Thess 1 Thessalonians

2 Thess 2 Thessalonians

1 Tim 1 Timothy

2 Tim 2 Timothy

Titus Titus

Philem Philemon

Heb Hebrews

James James

1 Pet 1 Peter

2 Pet 2 Peter

1 John 1 John

2 John 2 John

3 John 3 John

Jude Jude

Rev Revelation

Contents

1 Introduction

Religion has been an important part of human culture and civilization for millennia, shaping the moral, ethics, and philosophy of societies worldwide (MAOZ; HENDERSON, 2013; AMORE, 2019; BEYERS, 2017). With more than 72% of the global population having access to a complete Holy Bible[1], it continues to be a source of influence and inspiration. Beyond its spiritual significance, the Bible represents a relevant study for both religious people and researchers in diverse fields.

A *sermon* is a spoken or written discourse delivered by a religious leader (e.g. a Priest or a Pastor), typically within the context of a religious service or gathering (LOWRY, 2001). Sermons are a common feature of many religious traditions, including Christianity, Islam, Judaism, and others, and they serve various purposes within these traditions. The primary purpose of a sermon is to provide religious instruction, guidance, and spiritual insight to the congregation or audience.

Pastors and religious leaders often create their sermons through a well-thought-out process that combines spiritual reflection, biblical interpretation, research, and effective communication. While the specific process may vary from one preacher to another and between different religious traditions, the creation of a sermon usually starts with prayer and spiritual reflection, the choice of a central message or theme to be delivered, the selection of the Biblical passages that will compose the sermon, and the study and interpretation of the selected passages, a process known as *exegesis* (KLEIN; BLOMBERG; HUBBARD, 2004; EIJNATTEN, 2008).

Automated sermon composition processes are not employed in this task, nor any type of optimization or automation. Works that use the biblical corpus in association with automation or Artificial Intelligence (AI) techniques are confined to building translation machines (ESAN et al., 2020), authorship identifiers (which use the biblical corpus to compare the performance of AI algorithms in the service of paleographers) (BRIA et al., 2018), and there are few works that deal with artificial intelligence algorithms with the goal of actually extracting information from the biblical text (MURAI, 2013).

Despite these works, no effort was found in the literature in the direction of auto-

[1] https://www.wycliffe.net/resources/statistics/

matically selecting Biblical passages to compose a sermon. This thesis hypothesis that by combining Natural Language Processing techniques with a search heuristics capable of selecting the most suitable passages, it is possible to assist in the construction of sermons, that is, retrieve optimized Bible passages for a specific sermon theme.

1.1 General Objectives

Swarm intelligence is widely used for solving complex combinatorial or NP-hard problems, and with the use of metaheuristics such as "Ant Colony Optimization" (ACO), optimal solutions have been found for a significant portion of problems.

This thesis represents a pioneering initiative in presenting an analytical study of implementing a swarm intelligence technique for the combinatorial optimization of biblical passages. Therefore, this thesis proposes the implementation of the SwarmaBle algorithm for the optimized selection of Bible verses to support the construction of pastoral sermons.

In summary, the general objectives in their respective fronts can be highlighted:

- Investigate the feasibility of swarm intelligence techniques for optimizing of Biblical passages.
- Support the construction of topical sermons based on the results obtained through optimization.

1.2 Specific Objectives

In terms of specific objectives, this thesis aims to achieve the following actions:

- Identify in the literature the state of the art of applications of artificial intelligence techniques in the Biblical corpus.
- Conduct a sensitivity analysis (statistical analysis of the method).
- Promote validation with domain experts (theologians) of the proposed algorithm.

1.3 Justification

The esteem that society holds for the biblical text is reflected in the statistics of Christianity itself, which leads in terms of the number of followers worldwide [2]. Thus, utilizing tools that assist in extracting knowledge from this textual corpus could be of global interest and benefit. Considering the social value that religion adds to humanity, as evidenced in the works of SEN, COLUCCI e BROWNE (2022) and ATEN et al. (2019), this endeavor holds potential for worldwide significance.

This assumption serves as the basis for justifying the relevance of the proposed work. In addition to this aggregating factor, another reason for undertaking this thesis is that if natural language processing techniques associated with swarm intelligence prove promising for the linguistic characteristics of the biblical corpus, which is complex, these same techniques could exhibit similar performance in texts with characteristics analogous to those of the biblical corpus. This could contribute to the optimized retrieval of knowledge from complex textual corpora and also contribute to the theological field in the creation of sermons automated.

1.4 Structure the Thesis

This thesis is organized as follows:

In Chapter 2, we address the technical and theoretical aspects of the study areas in this work, encompassing the fundamental theoretical principles of natural language processing, swarm intelligence, and the biblical corpus.

In Chapter 3, a systematic review is conducted to present the state of the art of artificial intelligence applications in biblical literature.

Chapter 4 is dedicated to discussing the proposal of this thesis as well as its implementation.

Chapter 5 covers discussions and obtained results, including validation with domain experts in theology.

Finally, Chapter 6 includes the conclusion and suggestions for future work.

[2] https://www.cia.gov/the-world-factbook/field/religions/

2 Theoretical Background

This chapter aims to briefly review the theoretical principles of the areas that underpin this work. It is important to emphasize that this thesis has a multidisciplinary nature, therefore, the theoretical background presented here will be limited to a brief and non-exhaustive approach to the relevant subjects covered: Natural Language Processing, the Ant Colony Optimization algorithm, and an analysis of Biblical Text and Sermon Construction.

2.1 Natural Language Processing

Computational linguistics, in one of its branches known as *"Natural Language Processing"* (NLP), aims to process the natural language of humans for machine or computer understanding. Human language or communication is not easily assimilated by computers, as humanity can communicate in various forms, such as sounds, writing, gestures, etc.

Linguistic science can be envisioned in a rudimentary manner through the following disciplines: phonology, morphology, lexicography, syntax, semantics, and pragmatics. In brief, each of these disciplines will be portrayed, as found in the work of MITKOV (2003):

- **Phonology**: This field of study is dedicated to the sounds used in language, derived from letters, syllables, words, and sentences. It emphasizes that each person produces the same phoneme with distinct sound, which characterizes their voice or timbre.
- **Morphology**: Deals with the structure, composition, or formation of words, not necessarily their relationship in a sentence but as individual units. An example of computational morphology is the spell checkers in text editors.
- **Lexicography**: A collection of words existing in a language or a dictionary of words. Another synonymous definition is to consider lexicography as the vocabulary of a language.
- **Syntax**: The study of the sequence of words in a sentence to present their structure

in accordance with a specific grammar.

- **Semantics**: The study of the meaning of language, whether diachronic or synchronic. One way to distinguish semantics from pragmatics is that semantics seeks a dyadic relationship, while pragmatics involves a triadic interaction. In other words, semantics is concerned with a more literal and restricted meaning of a sentence or phrase, while pragmatics focuses on the author's intention or the discourse/dialogue itself.

- **Pragmatics**: The study that seeks to find a contextual meaning in a dialogue or written text, aiming to disambiguate sentences and phrases.

2.1.1 Syntactic Analysis

Syntactic analysis aims to produce an investigation into the grammatical structure of a phrase or sentence, and such an approach results in conveying meaning to a phrase or set of words. Figure 1 illustrates an example of the grammar of the English phrase *"the old man a ship."*

A coherent mechanism for reproducing a syntactic investigation is through a syntactic tree, and the syntactic tree conveys all the derivation steps of the sentence from the root node, as shown by HERBRICH e GRAEPEL (2010). Figure 2 presents a syntactic tree of the English phrase *"the old man a ship"*. The Python programming language and the adoption of the spaCy library for Natural Language Processing (NLP) make it possible to portray a practical example of this approach with the English phrase *"Demonstrative phrase for the elaboration of the grammatical analysis"*, as shown in Figure 3.

S	→	NP VP	Det	→	*a* | *an* | *the*
NP	→	Det NBar	Adj	→	*old*
NBar	→	Adj Noun	Noun	→	*man* | *men* | *ship* | *ships*
NBar	→	Noun	Verb	→	*man* | *mans*
NBar	→	Adj			
VP	→	Verb			
VP	→	Verb NP			

Figure 1: Example of grammar. Adapted from (HERBRICH; GRAEPEL, 2010)

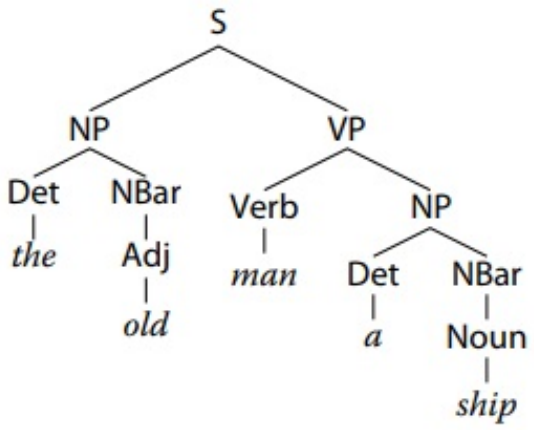

Figure 2: Syntactic tree. Adapted from (HERBRICH; GRAEPEL, 2010)

```
Demonstrative ADJ
phrase NOUN
for ADP
the DET
elaboration NOUN
of ADP
the DET
grammatical ADJ
analysis NOUN
. PUNCT
```

Figure 3: Example of grammar (Python).

2.1.2 Part-of-Speech Tagging (POS)

POS is a simplified activity of morphology that aims to present discourse analysis in a phrase or sentence. This is achieved by adding a label to the existing words in the sentence and marking the correct positions of the words in the discourse. Figures 4 and 5 illustrate this labeling for the English sentences *"This is an example sentence"* and *"Perform in the future"* (HERBRICH; GRAEPEL, 2010).

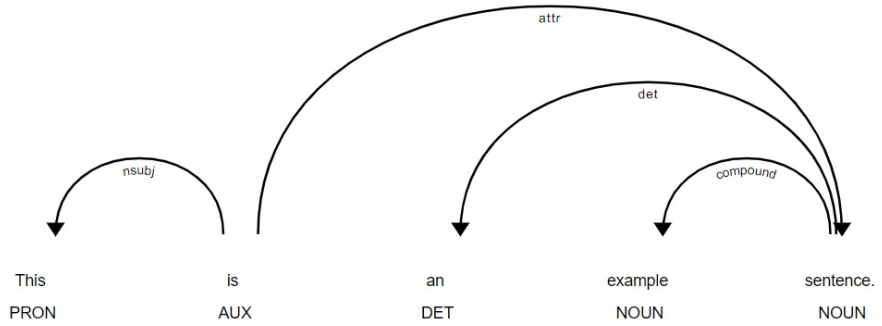

Figure 4: Example of POS 1 (Python).

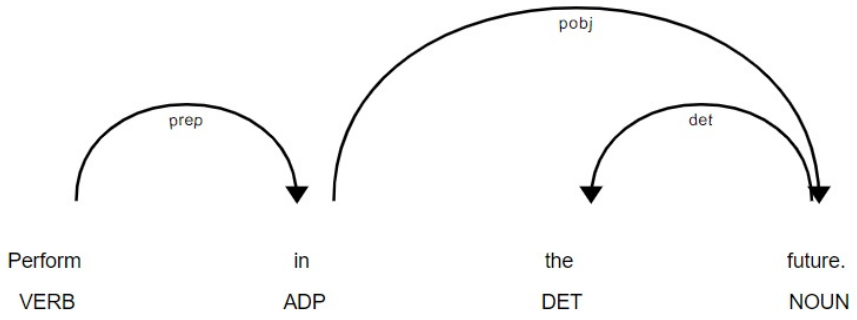

Figure 5: Example of POS 2 (Python).

2.1.3 Semantic Analysis

Due to the scope of this work, it is emphasized that the semantic analysis illustrated in this section is reduced and merely illustrative. With this caveat, this section will present the relational concept of lexical semantics. The Princeton WordNet database is a compilation of these syntactic semantic relations in the English language, and these relations are grouped by synsets. An example provided by Princeton University[3] illustrates these relations: a bed belongs to the furniture category and can be extended to classify a bunk bed as furniture in an equivalent manner. Both bunk bed and bed fall under the furniture category. A practical example using WordNet in Python illustrates the power of these

[3]https://wordnet.princeton.edu/

relations for the word “door”, as shown in Figure 6. (HERBRICH; GRAEPEL, 2010)

```
In [67]: import nltk
         from nltk.corpus import wordnet

         wordnet.synsets('porta', lang='por')

Out[67]: [Synset('car_door.n.01'),
          Synset('door.n.01'),
          Synset('door.n.05'),
          Synset('doorway.n.01'),
          Synset('exit.n.01'),
          Synset('gate.n.01'),
          Synset('interface.n.04')]
```

Figure 6: Example of WordNet (Python).

2.1.4 Information Extraction

Information extraction aims to track textual elements based on some semantic criteria, and is a robust task carried out by natural language processing. The extraction of information is accomplished through several techniques, including name extraction, entity extraction, relation extraction, and event extraction. In this subsection, each of these information extraction tasks will be briefly considered.

- **Name extraction**: Names can refer to various entities, including people, organizations, places, animals, and others. Therefore, for a proper understanding of names, i.e., understanding what a particular name refers to, some computational approaches and manipulations are necessary. While this distinction occurs naturally for us, it is much more complex for machines. Figure 7 illustrates this method of name extraction. Some computational manipulations that have been implemented for name extraction include regular expressions and the use of supervised learning methods with pre-trained models, as in the case of the Python NLP library (spaCy), as shown in Figure 8, where name recognition is performed on the English sentence *“In 1998 Miguel, who lives in New York in the United States and worked at Apple located in North America, decided to leave for Brazil”* (CLARK; FOX; LAPPIN, 2010).

Mr	O
Harry	B-PERSON
Hoople	I-PERSON
was	O
named	O
CEO	O
of	O
Harry's	B-ORG
Hogs	I-ORG
of	O
San	B-LOCATION
Francisco	I-LOCATION
.	O

Figure 7: Name extraction. Adapted from (CLARK; FOX; LAPPIN, 2010)

Figure 8: Example of name extraction (Python).

- **Entity extraction**: Entity extraction is an extension of name extraction, aiming to broaden its range of entities/names, including pronouns, noun phrases, and so on. Therefore, the analogy with name extraction is relevant. The work of CLARK, FOX e LAPPIN (2010) illustrates entity extraction.

- **Relation extraction**: Entities may have relationships with each other, and the machine needs to identify these relationships to enhance information extraction and the understanding of the examined text. For example, affiliation and organization, meaning recognizing that a particular person in a text may have an "affiliation" relationship with some organization mentioned in the text (CLARK; FOX; LAPPIN, 2010).

- **Event extraction**: The goal is to identify events or actions in the text based on the relationships between entities (CLARK; FOX; LAPPIN, 2010).

Natural language processing plays a unique role in the process of extracting textual information from a corpus. Without natural language processing techniques, the textual

representation and AI algorithms in text mining will not be sufficient to retrieve textual information with the linguistic nuances that need to be considered.

2.1.5 Textual Representation

Textual data needs to undergo conversion (from letters and words to numbers or numerical representation) because the algorithms used in text mining are limited to working with numbers. Some techniques for numerical representation of textual corpora will be briefly reviewed here.

Textual data needs initial processing, where the data is organized to optimize the execution of algorithms for proper performance. This step is called "preprocessing". The steps that constitute this preprocessing may vary depending on the implemented solution, but a typical approach would be: standardizing the textual format, handling stop words, and tokenizing. In a few words, we will describe what each of these phases does in preprocessing (WEISS et al., 2005):

- **Text Standardization**: Texts often come in various formats, from ASCII to scans and images. Converting the textual corpus to be worked on into a single format is convenient.

- **Stopwords**: Since many text mining applications use the word frequency mechanism for their analyses, terms such as articles, prepositions, conjunctions, and pronouns are common in any writing and do not convey significant information. Therefore, they introduce noise into the analyses and increase the textual base with elements that are not relevant. It is thus common to remove stopwords while preprocessing texts for analysis.

- **Tokenizing**: In the process of tokenization, characters are divided into words or "tokens". In languages delimited by spaces, each token is identified by spaces between words or specific punctuation marks. In other words, tokenization allows us to observe the number of words in a given text.

The *"Bag of Words"* (BOW) is a simplistic textual representation approach. In BOW, a vector representation of the textual corpus is created, where each document becomes

Table 1: Bag of Words (BOW) representation.

Documents	a	bright	idea	star	person
A bright idea	1	1	1	0	0
A bright star	1	1	0	1	0
A bright person	1	1	0	0	1

a vector, and the frequency of each word in the document is computed. In other words, each row represents a document, each column represents a word or term, and it indicates how many times each word in the columns appears in the documents. Table 1 illustrates the construction of this BOW matrix.

The Bag of Words (BOW) vector representation is not discriminatory in identifying important words; however, the TF-IDF (Term Frequency-Inverse Document Frequency) representation addresses this gap. In other words, the TF-IDF representation discriminates expressive words and reduces the weight for unimportant words, as stated by SRIVASTAVA e SAHAMI (2009). This is achieved as illustrated in the Equation 1 (AGGARWAL; ZHAI, 2012):

$$tfidf(w) = tf * \log \frac{N}{df(w)} \tag{1}$$

where tf is the term frequency (the number of occurrences of words in a document), $df(w)$ is the document frequency (the number of documents containing the word), N is the number of documents in the corpus, and $tfidf(w)$ is the relative weight of the feature in the vector.

As a result, the TF-IDF representation reduces the weight for words that appear in many documents and increases the weight for words that appear in few documents, producing an identification of truly promising words for analysis. However, the TF-IDF representation fails to capture the semantic characteristics of words in the text, as summarized by the authors WEISS et al. (2005); BERRY e KOGAN (2010).

These two representations do not capture the semantic relationships between words, in addition to this limitation, representations like Bag of Words (BoW) and Term Frequency-

Inverse Document Frequency (TF-IDF) place greater emphasis on word frequency and face the challenge of sparsity. In response to this, the scientific community has proposed methods of vector representation using dense vectors.

In this context, words can be represented by vectors in a multidimensional space, and the distances between these vectors reflect the similarities between the words. In other words, the closer the vectors, the greater the semantic similarity between the words. (PILEHVAR; COLLADOS, 2021) Figures 9 and 10 illustrate this concept.

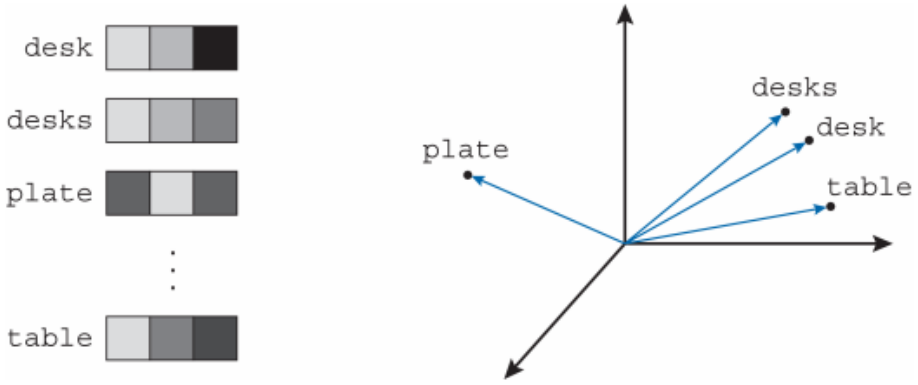

Figure 9: Vector textual representation 1. Adapted from (PILEHVAR; COLLADOS, 2021)

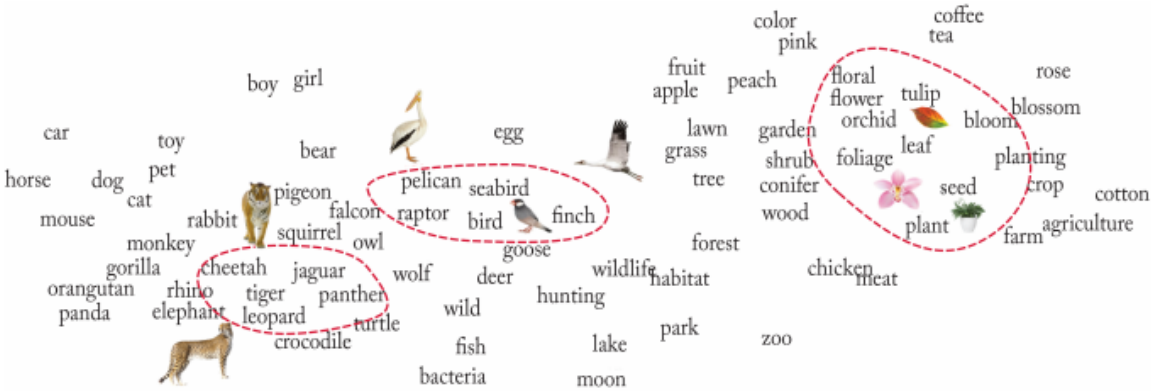

Figure 10: Vector textual representation 2. Adapted from (PILEHVAR; COLLADOS, 2021)

Neural networks have significantly improved the ability to vectorize words in Natural Language Processing (NLP), introducing robust performance in semantic assimilation

within *"embeddings"*. These neural networks have successfully trained robust models for these embeddings, with the most popular ones being Word2Vec, Glove, and fastText (PILEHVAR; COLLADOS, 2021). The architecture of Word2Vec is depicted in Figure 11.

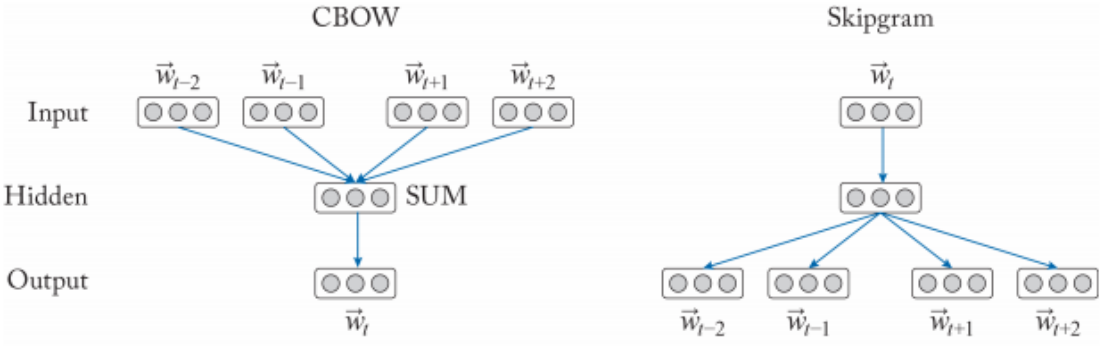

Figure 11: Word2Vec architecture. Adapted from (PILEHVAR; COLLADOS, 2021)

2.2 The Ant Colony Optimization Algorithm

2.2.1 Natural Computing

Nature has served humanity with its natural resources throughout history. Lately, nature has also been a source of inspiration for the development of *"bioinspired computing"* (DE CASTRO, 2007).

Natural computing is not the first discipline to draw inspiration from natural concepts. Current technologies have already emulated, to some extent, natural characteristics in their solutions, such as airplanes (inspired by birds), submarines (inspired by fish), sonar systems (inspired by bats), and so on (DE CASTRO, 2007).

According to the author DE CASTRO (2007), natural computing can be defined as follows: "A computational version of the process of extracting ideas from nature for the development of 'artificial' systems, i.e., the use of natural materials and mechanisms to perform computation".

Natural computing branches into three main segments (DE CASTRO, 2007), which are:

- **Nature-Inspired Computing**: Nature serves as inspiration for implementing techniques to solve problems.
- **Simulation and Emulation of Nature through Computing**: It synthesizes forms, patterns, and behaviors analogous to those already known in nature.
- **Computing with Natural Materials**: Involves the use of new raw materials for computing.

Natural computing is a field of science that, by its essence, is interdisciplinary. Physicists, chemists, mathematicians, engineers, biologists, among others, are necessary for effective development and understanding of natural phenomena and their emulation for computing.

According to DE CASTRO (2007), the implementation of natural computing methods is done in a simplified manner, that is, through simplified models, for the following reasons: simplifications are necessary to make computing possible, the use of models is sufficient to achieve the expected objectives, and there are details of the functioning of the phenomenon that are unknown.

The reasons for the applicability of natural computing techniques will depend on the characteristics of the problems to be solved. In other words, natural computing techniques are alternatives, and there are other methods to solve the same problem. However, when certain problems have characteristics such as:

- Complexity;
- Existence of more than one solution option, and the optimal solution is unknown;
- Difficulties in modeling problems;
- A single solution is not sufficient.

The adoption of some natural computing approach can be beneficial.

According to Ballard (1997), natural computation is grounded in 5 central ideas, namely: Fitness, Programs, Data, Dynamics, and Optimization.

- Fitness: Effective programs require a fitness function that evaluates their performance, i.e., assesses the cost of programs in terms of data. However, constructing fitness functions is not a simple task. For example, what fitness functions are used by the human brain to weigh its decisions?

- Programs: Programs operate by searching through a space of states using operators that govern transitions from one state to another. Therefore, problem-solving can be understood as choosing a good sequence of operators to specify a path through the space of states.

- Data: These are continuous state spaces that can be formally represented as a vector. This data is commonly produced as the output of sensors, among various other situations.

- Dynamics: Movement in the space of states is known as the system's dynamics. Such trajectories are differentially described in terms of the rate of change of the state vector with time. Any physical system, like neurons or muscles, does not respond immediately in time; however, it will have a time-varying response called dynamics.

- Optimization: Optimization theory is at the core of learning algorithms; in other words, all these algorithms work by searching through a space of states to make improvements in the objective function. In simpler terms, minimizing or maximizing the objective function defines optimization.

In general terms, natural computing encompasses other important concepts, which will be briefly described (DE CASTRO, 2007), namely:

- Individuals, Entities, and Agents: The characteristic of collectivity permeates natural computing, such as a population of individuals, a colony of insects, among others. Thus, an agent can be defined as a subroutine of a program or an intelligent organism capable of autonomy and identity.

- Parallelism and Distributivity: This principle of distributed parallelism emerges from the observation of various natural phenomena, such as social agents, the genome, and Darwinian evolution.

- Interactivity: Natural systems can interact with each other and the environment. This interaction can be diverse, including reproductive, competitive, cooperative aspects, among others.

- Connectivity: In these systems, information is encoded through nodes or connections in a network. A popular example of a connectionist system is artificial neural networks.

- Stigmergy: Indirect communication made by some insects and mediated by the environment.

- Adaptation: The ability to vary conditions depending on the environment's characteristics, always seeking a more appropriate condition for a given scenario.

- Learning: The ability to absorb knowledge through experience or interactions. Considering that it is a gradual and not instantaneous process.

- Evolution: The name given to Darwin's theory of species evolution, where an individual undergoes changes over time, providing a greater chance of survival. Unlike adaptability, evolution requires some conditions to exist, such as a population of individuals capable of reproducing, undergoing genetic variations, and natural selection.

- Feedback: Feedback occurs when a system's output is directed back to its input, i.e., a return from the output to the input again.

- Self-Organization: Self-organization spontaneously emerges through internal interactions within the system without any external intervention.

- Complexity: A complex system cannot be understood or studied by separating its composing parts. The global behavior emerges from the individual behavior of agents or phenomena.

The main subareas of natural computing can be identified as follows (DE CASTRO, 2007):

- Evolutionary Computing

- Genetic Algorithms
- Artificial Neural Networks
- Swarm Intelligence
- Fractal Geometry in Nature
- Artificial Life
- DNA Computing
- Quantum Computing

2.2.2 Swarm Intelligence

Natural Computing, as described earlier, takes inspiration from nature, and this inspiration branches into some subdivisions inspired by areas like biology, physics, and chemistry. The algorithms encapsulated in biological inspiration are termed "bioinspired". Therefore, we can observe a hierarchy of these research fields: nature-inspired, bioinspired, which is encompassed in nature inspiration, and swarm intelligence that is contained within bioinspiration (HASSANIEN; EMARY, 2016). Figure 12 presents this hierarchical overlap of these areas.

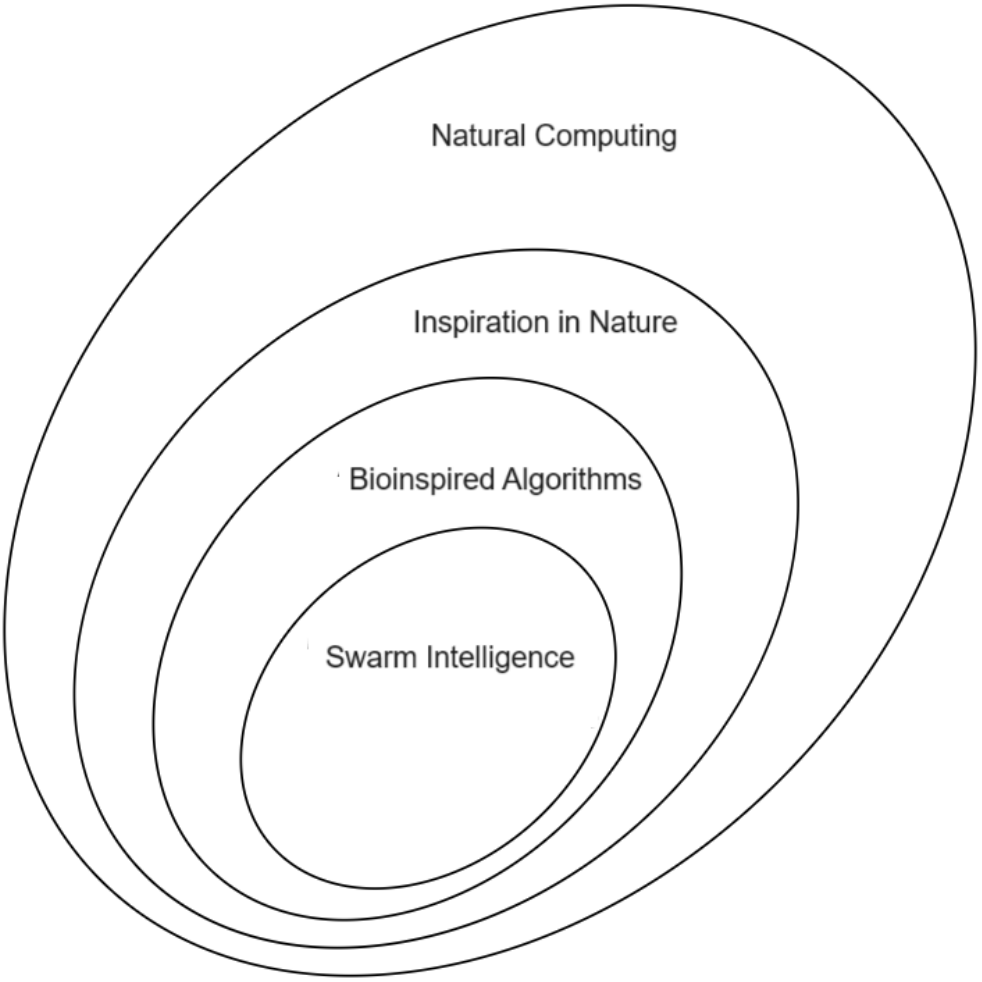

Figure 12: Subareas - Natural Computing

The foundation of swarm intelligence is based on several principles, including stochastic behavior, emergence, optimization, agents, and social life (KENNEDY; EBERHART, 2016):

- **Stochastic Behavior**: Stochastic or random behavior can be defined as "randomness exists when repeated occurrences of the same phenomenon can result in different outcomes". Often, stochastic situations are deterministic, that is, there is a cause, albeit unknown, and it is this unknown that induces randomness.
- **Emergence**: The concept of emergence is challenging to define in the scientific community; nevertheless, some of its characteristics are identifiable. Emergence becomes visible when we observe a particular organization and stability/equilibrium of a set of interactions, as if there were an invisible hand guiding the process. Thus, this emergence is the result of the quality of many interactions that compose it. Cellular automata, the economic system idealized by Adam Smith, among others, exemplify the concept of emergence.
- **Optimization**: Kennedy e Eberhart (2016) define optimization as follows: "Optimization generally refers to a process of adjusting a system to achieve the best possible outcome".
- **Agents**: According to Kennedy e Eberhart (2016), various examples can be considered agents, such as robots, chatbots, functions in computer programs, etc. However, the concept of an agent is more related to the ability to make decisions, possess autonomy, perceive the environment, among others.
- **Social Life**: A termite mound, an ant colony, or a beehive fascinates biologists and prompts the following questions: "who governs such beings?" and "how do they organize in society?" Upon a closer and more investigative observation of the behavior of these biological agents, it can be noted that there is individuality and that these agents do not act by subordination. Instead, the individual behavior of these agents, when combined, gives rise to the organization of the society of these agents (BONABEAU; DORIGO; THERAULAZ, 1999).

Hassanien e Emary (2016) also emphasize that most new optimization-based algorithms have some kind of inspiration from nature, and this inspiration is predominantly

bioinspired. Swarm intelligence is a representative of this bioinspiration.

A more formal and generic description of the optimization process is (HASSANIEN; EMARY, 2016):

$$\overset{min}{x \in \Re^n} f_i(x), (i = 1, 2, ...M) \tag{2}$$

$$Subject \quad to \quad h_j(x) = 0, (j = 1, 2, ...J) \tag{3}$$

$$g_k(x) \leq 0, (k = 1, 2, ..., K) \tag{4}$$

where M is the number of objective functions to be optimized, $f_i(x)$ is the objective function representing the target i, and x are the design variables to be discovered by the optimization algorithm, and their size is n. $\Re^n$ is the space spanned by the decision variables and is referred to as the search space. The equalities for h_j and the inequalities for g_k are called constraints.

The optimization problem with $M = 1$ is called single-objective optimization, while optimization with $M > 1$ is called multi-objective optimization. If no constraints are imposed on the problem ($J = 0$ and $K = 0$), the optimization is considered unconstrained. However, in cases where $J > 0$ or $K > 0$, the problem is considered a constrained problem.

The term "Swarm Intelligence", according to Kennedy e Eberhart (2016), was originally coined to describe a specific paradigm in robot research. However, over time, it has become closely associated with solving problems inspired by the social behavior of insects and animals. It is worth noting that the term "swarm" is not restrictively aligned with insects but rather with the structure of agents in social interaction. For example, a flock of birds, a swarm of cars (traffic), a swarm of economic agents (economy), etc.

Swarm intelligence algorithms are population-based, meaning a population of individuals cooperating with each other (PANIGRAHI; SHI; LIM, 2011). According to the same authors, swarm intelligence algorithms were originally conceived to solve unconstrained single-objective optimization problems. With advances in research, new types of algorithms have emerged for constrained optimization problems, multi-objective opti-

mization, combinatorial optimization, and many other types of problems.

Some real-world application problems (PANIGRAHI; SHI; LIM, 2011), previously intractable by other methods, have been solved by swarm intelligence algorithms, and the field has been invigorated by the significant contributions already achieved (PANIGRAHI; SHI; LIM, 2011).

Social insects typically live in communities or colonies, meaning a colony is a large family of insects. In a colony, each insect has its own agenda; however, the end result is a well-organized colony, which means there is no need for supervision, but the concept of self-organization is present.

A characteristic of the social life of insects is the presence of nests (termite mounds, ant nests, beehives, etc.), and these insects always venture out of these nests in search of a food source, such as ants foraging in the environment (DE CASTRO, 2007).

Interactions are often present in the social life of insects, both direct and indirect. Direct interactions are exemplified by obvious movements, such as antennation, food exchange (trophallaxis), mandibular contact, among others. Indirect interactions are subtle and occur when an agent or insect modifies the environment through substances, and another insect responds to this modification; this action is called "stigmergy" (BONABEAU; DORIGO; THERAULAZ, 1999).

Some examples of activities of social insects, besides the search for food, include nest construction or expansion, division of labor, inhibition and combat of predators, among others (DE CASTRO, 2007).

2.2.3 Ant Colonies

Ants have robust characteristics compared to other insects. They can support and carry up to 20 times their own weight, have two stomachs in their abdomen, can live up to 60 days, and there are about 10,000 known species of ants (DE CASTRO, 2007).

In an ant colony, there are various roles that an ant can play, such as collecting and distributing food, building the nest, taking care of the nest, eggs, and larvae, among others (DE CASTRO, 2007).

In the task of food collection by ants, they perform this activity through foraging, i.e., they exploit rich food sources without losing the ability to explore the environment.

This foraging is done randomly and is often mediated by a chemical substance deposited by the ants in the process of randomly exploring the environment, called pheromone. This mediation is called stigmergy. Figure 13 illustrates this foraging process.

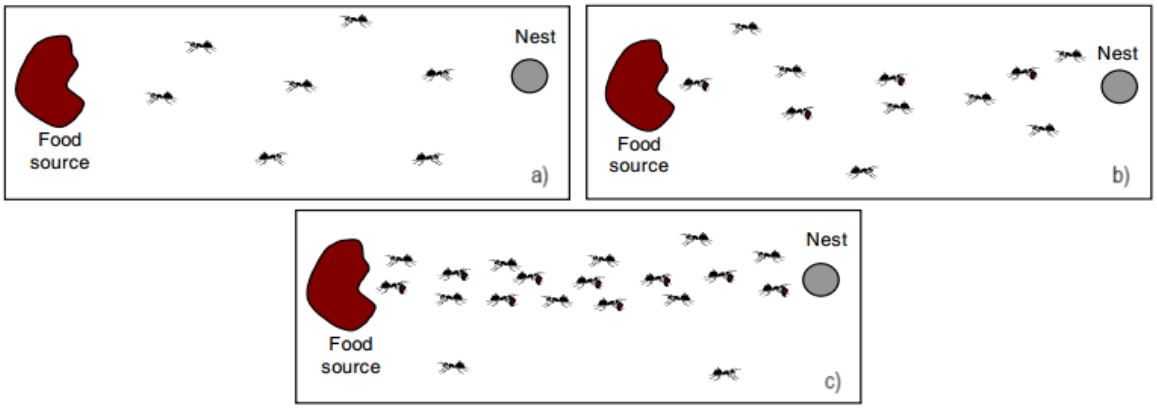

Figure 13: Ant foraging. a) Ants foraging randomly. b) Ants starting to converge. c) Ants converged. Adapted from (DE CASTRO, 2007)

Stigmergy is the indirect communication carried out by ants through the environment. The concept of stigmergy was introduced by Grassé in 1959 (SOLNON, 2010) to refer to the coordinated nest construction by termites of the genus Macrotermes. He observed how termites act independently in a structure without communicating directly with each other. Therefore, the environment mediates communication or interaction among individuals, and similarly, ants interact with each other. Figure 14 illustrates how stigmergy influences ant foraging decisions.

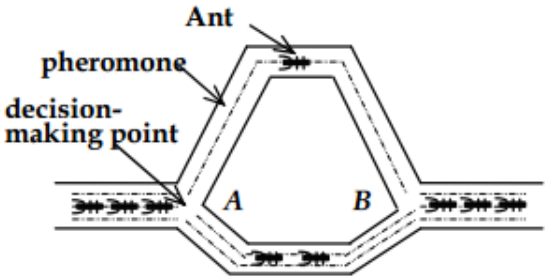

Figure 14: Ant decision based on pheromone trail. Adapted from (OSTFELD, 2011)

The probabilistic and optimization-driven phenomenon of ant foraging behavior was observed in the double-bridge experiment by researchers Deneubourg and colleagues (DORIGO; STüTZLE, 2004). In other words, ants walk between the nest and a food source, depositing a certain amount of pheromone, thus forming a trail. Once an ant finds a food source, and if the quality of that source is high, it returns to the nest depositing a larger amount of pheromone.

Therefore, the probabilistic characteristic identified by the experiment is that, in the random foraging of ants, they make probabilistic decisions based on the amount of pheromone deposited on the trail (the higher the concentration of pheromone, the higher the probability that ants will choose that path). The optimization capability lies in the concentration of pheromone on the path, as pheromones undergo evaporation. In other words, shorter paths will have a higher concentration of pheromone because ants spend less time between the nest and the food source, resulting in a greater number of ants that have passed through, evaporation was not sufficient to decrease the concentration, and due to the quality of the source, maintaining a higher pheromone deposit. The figures 15 and 16 illustrate that, over time, ants converged to the shortest path in the double-bridge experiment.

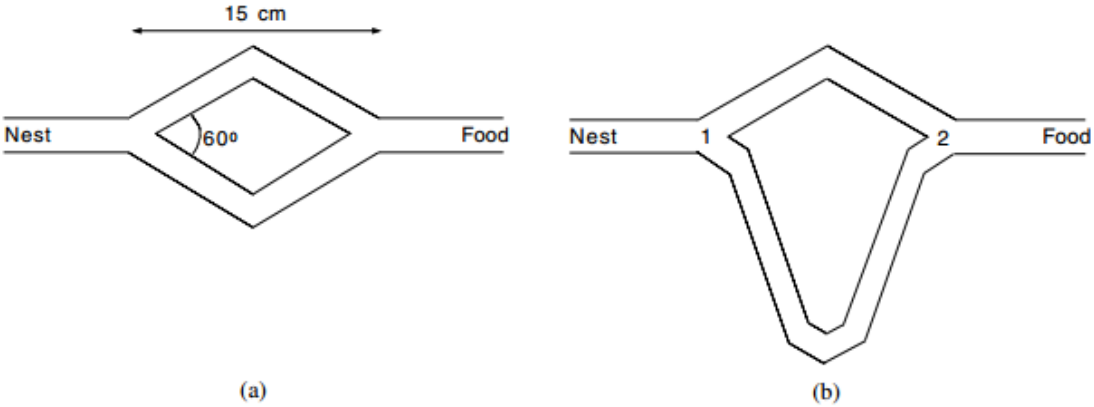

Figure 15: Experiment of the double bridge. a) Bridge with equal paths. b) Bridge with different path lengths. Adapted from (DORIGO; STüTZLE, 2004)

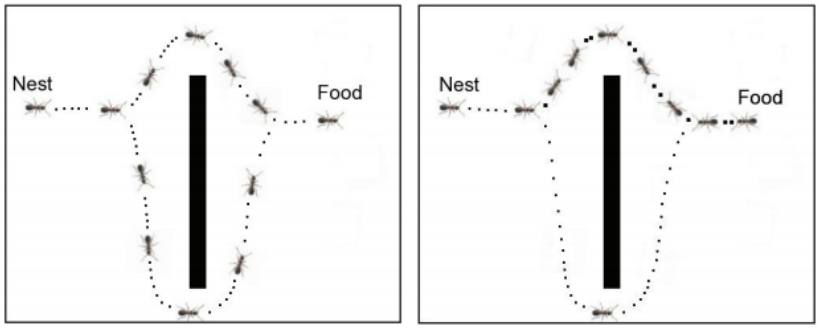

Figure 16: Convergence to the shortest path. Adapted from (OSTFELD, 2011)

The authors Dorigo e Stützle (2004) state that in this experiment, the autocatalytic or positive feedback effect was observed, ensuring the self-organized behavior of ants, where microscopic actions of the ants emerge into a macroscopic pattern of organization and optimization.

The double-bridge experiments showed that ant colonies have an inherent optimization capability; through probabilistic rules based on local information, they can find the shortest path between two points in their environment (DORIGO; STüTZLE, 2004). Therefore, researchers delved into how to artificially emulate this behavior.

The researchers Deneubourg et al. (SOLNON, 2010) devised a continuous mathe-

matical model for the double-bridge experiment using differential equations. Dorigo and Stützle adapted this continuous model to a discrete one for solving combinatorial problems.

The first step in developing an algorithmic implementation for minimum path optimization in graph-based problems was the Simple - Ant Colony Optimization (S-ACO); however, it serves more as a didactic mechanism for understanding the essential characteristics of an Ant Colony Optimization (ACO) algorithm (DORIGO; STüTZLE, 2004).

The S-ACO maintains the fundamental characteristics of the behavior of biological ants and adds an additional component, which is the use of memory for previously traveled paths to execute the optimization of shorter paths in graphs and avoid loops (DORIGO; STüTZLE, 2004).

In an experiment with a more extended double-bridge, it was shown that the S-ACO struggles to find the shortest path in more elaborate graphs. Therefore, the ants would have to make a series of correct choices to generate optimal paths. The figure 17 illustrates the graph of the extended bridge experiment.

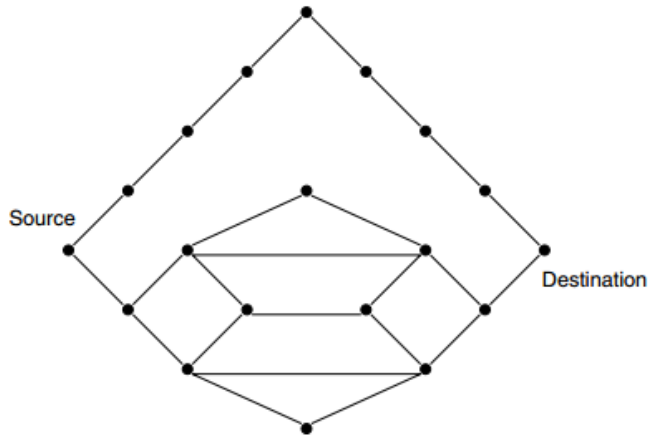

Figure 17: Expanded graph. Adapted from (DORIGO; STüTZLE, 2004)

2.2.4 Ant Colony Optimization (ACO) as a Metaheuristic

Combinatorial optimization problems are challenging, as they can sometimes be trivial and, on other occasions, exhibit NP-hard nature, meaning they are problems that cannot achieve an optimal solution within polynomial computing time. Therefore, in some cases, it is necessary to employ techniques that result in near-optimal solutions in a reasonably short time. Methods of this kind are called heuristics, which are algorithms that utilize specific problem knowledge to enhance the returned solutions. (DORIGO; STüTZLE, 2004).

Research in this area of heuristics has advanced, leading to the emergence of a subfield known as metaheuristics. According to the author Dorigo e Stützle (2004) a metaheuristic is a set of concepts and algorithms that can be used to define heuristic methods applicable to a broad range of distinct problems.

The ACO is a metaheuristic in which inspiration from the optimization of ant colonies is applied to find good solutions for difficult discrete optimization problems.

The ACO was initially proposed by Dorigo e Stützle (2004) to be applied to the TSP (Traveling Salesman Problem). The TSP has been extensively studied in the literature due to being a significant NP-hard optimization problem and easily adaptable to the ACO analogy. Additionally, it can be tested to evaluate the performance of ACO without obscuring its technical details. (DORIGO; STüTZLE, 2004). Therefore, an explanation of the TSP's functioning is necessary for its application in ACO.

The traveling salesman problem is defined by a traveler seeking to find the shortest path on a journey, starting from a hometown to a destination, visiting each city only once and returned to his hometown. In other words, the TSP is the problem of finding a minimum path in a Hamiltonian circuit. The representation of the TSP is given by a weighted graph $G = (N, A)$ with N being the set of $n = |N|$ nodes (cities) and A the set of arcs that fully connect the nodes. Each edge $(i, j) \in A$ is assigned a weight d_{dj} representing the distance between city i and j (DORIGO; STüTZLE, 2004).

Therefore, an optimal solution for the TSP is a permutation π of the node indices $1, 2, ..., n$ in such a way that the length $f(\pi)$ is minimized, where $f(\pi)$ is given by:

$$f(\pi) = \sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)} \tag{5}$$

ACO - Ant Colony Optimization

The ACO proposed by Dorigo et al. for the TSP problem can be mathematically represented through its main metaphors, i.e., the probabilistic choice made by the ants, pheromone update, and pheromone evaporation rate (DORIGO; STüTZLE, 2004).

The probabilistic equation for choosing the next node is given by:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{1 \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \quad \text{if} \quad j \in N_i^k \tag{6}$$

where $\eta_{ij} = 1/d_{ij}$ is a pre-defined heuristic value, α and β are two parameters that determine the relative influence of the pheromone trail and heuristic information, and N_i^k is the neighborhood of ant k when it is in city i, i.e., the set of cities that ant k has not visited yet. The probability of choosing a city outside of N_i^k is 0. According to this probability rule, the choice of a specific arc (i, j) increases with the value of the associated pheromone trail τ_{ij} and the value of heuristic information η_{ij}.

The pheromone deposit update equation is given by:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^k, \quad \forall (i, j) \in L \tag{7}$$

where $\Delta\tau_{ij}^k$ is the amount of pheromone deposit that ant k places on the arcs it has visited, and this is represented as follows.

$$\Delta\tau_{ij}^k = \begin{cases} 1 \quad /C^k \\ 0 \end{cases}, \text{if arc (i,j) belongs to} \quad T^k; \quad \text{otherwise;} \tag{8}$$

where C^k, the length of the tour T^k constructed by the k-th ant, is calculated as the sum of the lengths of the arcs belonging to T^k.

The pheromone evaporation equation is given by:

$$\tau \leftarrow (1 - \rho)\tau_{ij}, \quad \forall (i, j) \in L \tag{9}$$

where $0 < \rho \leq 1$ is the pheromone evaporation rate. The parameter ρ is used to prevent the unlimited accumulation of pheromone and enables the algorithm to forget poor paths chosen earlier.

A pseudocode that synthesizes the application of ACO in the TSP is illustrated in the algorithm 1 (SOLNON, 2010).

Algorithm 1 Ant System for the Traveling Salesman Problem

Input:

a complete non-directed graph $G = (V, E)$

a distance function $d : E \to R$

a set of numerical parameters $\{\alpha, \beta, \rho, \tau_0, Q, \text{nbAnts}\}$

Output:

Returns a Hamiltonian cycle of G

Begin:

1. for each edge $(i, j) \in E$ do $\tau_{ij} \leftarrow \tau_0$
2. **while** stopping criteria not reached **do**
   (a) for each ant $k \in \{1, \ldots, \text{nbAnts}\}$ **do**
      i. put ant k on a randomly chosen vertex of V
      ii. **while** ant k has not visited all vertices of V **do**
         A. let i be the vertex on which ant k is currently located
         B. let $Cand$ be the set of unvisited vertices
         C. randomly choose $j \in Cand$ with respect to probability
$$p_{ij} = \frac{\tau_{ij}^{\alpha} \cdot (1/d_{ij})^{\beta}}{\sum_{l \in Cand} \tau_{il}^{\alpha} \cdot (1/d_{il})^{\beta}}$$
         D. move ant k to vertex j
   (b) for each edge $(i, j) \in E$ **do** $\tau_{ij} \leftarrow \tau_{ij} \cdot (1 - \rho)$
   (c) for each ant $k \in \{1, \ldots, \text{nbAnts}\}$ **do**
      i. let lk be the length of the cycle built by ant k
      ii. for each edge (i, j) of the cycle built by ant k **do** $\tau_{ij} \leftarrow \tau_{ij} + \frac{Q}{lk}$
3. **return** the best Hamiltonian cycle built during the search process

2.3 Analysis of Biblical Text and Sermon Construction

The word "Bible" is derived from Latin, which in turn comes from the Greek term "βιβλος" "biblos", means "written". The Bible organizes the sacred literature of two monotheistic religions in the world (Christianity and Judaism), as summarized by COMFORT (1998).

The Holy Bible, however, differs for both religions. In Judaism, the Bible is known as the "Tanakh" or Old Testament and consists of 24 books (Table 2). In Christianity, the Bible is composed of 66 books, 39 in the Old Testament and 27 in the New Testament (Tables 3 and 4) (CHAPMAN; SWEENEY, 2016).

The Holy Bible is the normative text for these two religions, that is, the manual of faith and conduct. Therefore, it is a literature that holds the highest position in the importance ranking for these two religious worldviews (FORSTER, 2019).

Currently, the Bible is produced in paper format as a book (except in Judaism, where the use of scrolls is preserved) and arranged in chapters and verses. The Protestant Christian Holy Bible has 1,189 chapters, 31,102 verses, and 66 books[4].

2.3.1 Structure and Division of the Hebrew / Protestant Bible

[4]https://www.biblebelievers.com/believers-org/kjv-stats.html

Table 2: Division of the Hebrew Bible (Tanakh)

Law (Torah)	The Prophets (Nevi'im)	The Writings (Ketuvim)
Genesis	**A. Former Prophets**	**A. Poetic Books**
Exodus	Joshua	Psalms
Leviticus	Judges	Proverbs
Numbers	Samuel	Job
Deuteronomy	Kings	**B. Five Scrolls (Megilloth)**
	B. Latter Prophets	Song of Solomon
	Isaiah	Ruth
	Jeremiah	Lamentations
	Ezekiel	Esther
	The Twelve	Ecclesiastes
		C. Historical Books
		Daniel
		Ezra-Nehemiah
		Chronicles

Table 3: Division of the Old Testament (Christian/Protestant Bible)

Pentateuch	Poetry	History	Prophets
Genesis	Job	Joshua	**A. Major**
Exodus	Psalms	Judges	Isaiah
Leviticus	Proverbs	Ruth	Jeremiah
Numbers	Ecclesiastes	1 Samuel	Lamentations
Deuteronomy	Song of Solomon	2 Samuel	Ezekiel
		1 Kings	Daniel
		2 Kings	**B. Minor**
		1 Chronicles	Hosea
		2 Chronicles	Joel
		Ezra	Amos
		Nehemiah	Obadiah
		Esther	Jonah
			Micah
			Nahum
			Habakkuk
			Zephaniah
			Haggai
			Zechariah
			Malachi

Table 4: Division of the New Testament (Christian/Protestant Bible)

Gospels	History	Epistles (Letters)	Prophecy
Matthew	Acts of the Apostles	Romans	Revelation
Mark		1 Corinthians	
Luke		2 Corinthians	
John		Galatians	
		Ephesians	
		Philippians	
		Colossians	
		1 Thessalonians	
		2 Thessalonians	
		1 Timothy	
		2 Timothy	
		Titus	
		Philemon	
		Hebrews	
		James	
		1 Peter	
		2 Peter	
		1 John	
		2 John	
		3 John	
		Jude	

2.3.2 The original languages of the Holy Bible

The Holy Bible originally began to be crafted by the Hebrews (Jews) in the Hebrew language. Hebrew consists of a consonantal alphabet of 22 consonants (KERR, 1948). However, in the Middle Ages, the Masoretes (a group of Jewish scholars) added a system of vocalization so that the Hebrew would not lose its pronunciation over time. With this addition, the Hebrew text used for academic studies of the Hebrew Bible is the Masoretic Text. Nevertheless, there are portions of Aramaic in the Old Testament.

Around the 3rd century BCE, the Old Testament or Hebrew Bible was translated into Greek, and this version of the Old Testament scriptures is known as the "Septuagint" (LXX). It is worth noting that a translation already undergoes some degree of text interpretation, as the translation from one language to another is often challenging due to the heterogeneous grammar of languages. Therefore, for the translation to convey meaning, an interpreter's exercise is necessary, as pointed out by CHAPMAN e SWEENEY (2016).

The Christian New Testament is composed solely in the Greek language, owing to the Hellenization process of the time when Greek was the predominant language. However, the New Testament has an ancient translation that influenced subsequent translations, the Latin Vulgate (4th century AD). Similar to the Septuagint (LXX), it requires some degree of scriptural interpretation in the translation process (ROGERSON; LIEU, 2006).

In summary, the Holy Bible was produced in three distinct languages: Hebrew and Aramaic in the Old Testament and Greek in the New Testament. These are ancient languages with challenging lexicography, carrying cultural and grammatical elements that need to be understood for an adequate translation process to convey the message comprehensibly (FORSTER, 2019).

Throughout the history of human civilization, the Holy Bible has gained the status of a classic of world literature, equivalent to the ancient Greek writings. Therefore, the Holy Bible has literary characteristics that need to be listed to demonstrate its complexity of interpretation, due to the combination or concatenation of these various literary styles combined in a single volume.

- **Literary Genres:**

1. **Narrative:** The narrative genre discusses facts, history, speeches, and actions of characters or the writer itself (DELL; JOYCE, 2013).

2. **Poetic:** The poetic genre is exemplified by the largest book in the Holy Bible, the book of "Psalms," and the poetry of the psalter, as the book of Psalms is known among scholars. It is rich, blending figurative language, emotion, rhyme/meter, and the use of images. The poetic language of Hebrew emphasizes the parallelism of ideas. Consequently, this concatenation of linguistic phenomena in biblical poetry reveals its figurative essence and interpretive complexity (CHAPMAN; SWEENEY, 2016), (BENTHO, 2003).

3. **Wisdom:** A literary genre found in the biblical text, especially in the books of Proverbs, Job, and Ecclesiastes. This type of text sought answers related to God, life, and the world—a kind of Jewish philosophy (ROGERSON; LIEU, 2006), (CHAPMAN; SWEENEY, 2016).

4. **Biographical:** The Gospels (Matthew, Mark, Luke, and John) are an example of the biographical text. They narrate the story of the ministry of the Lord Jesus Christ, his teachings, actions, and conduct. Despite their biographical nature, the content of the Gospels retains its normative essence for Christianity (ROGERSON; LIEU, 2006).

5. **Prophetic:** Written messages of oracles distinguish biblical literature from other writings produced by humanity. In other words, biblical prophecy is the distinctive feature of biblical text, and this literary genre is rich in linguistic phenomena (ROGERSON; LIEU, 2006), (CHAPMAN; SWEENEY, 2016).

6. **Letters:** The use of this literary genre occupies a significant space in the content of the New Testament—specifically, the apostles' letters. These letters follow a structural pattern, including greetings, addresses, requests and recommendations, instructions, and others (ROGERSON; LIEU, 2006).

7. **Apocalyptic:** The apocalyptic literature of the Bible is linked to the idea of "revealing" the unknown, often implying an eschatological vision. Apocalyptic language is marked by symbolism and metaphors, making it challenging to comprehend casually, as discussed by ROGERSON e LIEU (2006), CHAPMAN e SWEENEY (2016).

- **Linguistic Phenomena:**

  Linguistic phenomena follow the concept of the nature of language in its communication task. In spreading the message, the speaker or writer needs to use various linguistic resources to continue propagating their message to the recipient. Thus, biblical authors used diverse linguistic devices to proclaim their message. For illustrative purposes, some examples are cited, including parables, metaphors, allegories, similes, typology, and others (VIRKLER, 1980). It is emphasized that for a coherent understanding of the message's meaning, the interpreter needs to be attentive to this linguistic phenomenology.

2.3.3 Interpretative Methods (Hermeneutics)

The linguistic information outlined above demonstrates the complexity of interpretation that the biblical text presents to its readers. It is a textual corpus that requires significant effort for satisfactory textual comprehension. In light of this, scholars of biblical literature have felt the need to employ methods that aid in the accurate interpretation of the text, and one of these techniques is the use of hermeneutics.

Hermeneutics is conceptualized by author BARTON (1998) as "a critical reflection on the nature and objectives of reading and interpretation, or the understanding of the acts and processes of communication". Another author defines biblical hermeneutics in this way: Forster (2019) states, "Hermeneutics is the theory of interpretation and understanding." In a broader sense, biblical hermeneutics analyzes and investigates not only the text but also the author's intention, the cultural context, and other relevant factors (FORSTER, 2019).

The word hermeneutics derives from the Greek word "ερμηνευειν" "hermeneuein," which means "to interpret," as stated by the author BENTHO (2003). Therefore, interpretative techniques are applied in the process of extracting the meaning of the textual message. Some techniques can be exemplified as follows: contextual analysis (immediate and remote context), grammatical analysis (lexicography and syntax), historical-cultural analysis, and others (VIRKLER, 1980).

According to the author BARTON (1998), after 2,000 years of New Testament scrip-

ture history, is there room for new research? The author's response is affirmative, asserting that there is indeed an opportunity for new research due to the emergence of new information and new methods. According to the same author, research in the field of interpretation will always exist. Authors CHAPMAN e SWEENEY (2016) ensure that in recent centuries, there has been a growing interest in academic research in biblical literature. According to these authors, studies focused on religious literature have achieved academic status, transcending solely religious and theological realms.

Biblical literature has influenced society in various ways. Linguistically, it played a role in the literacy of European nations, as demonstrated in the work of Gawthrop e Strauss (1984). Its influence extends beyond language to the political sphere. According to the study by HOFMANN (1999), the Holy Bible significantly contributed to the shaping of American politics in the 13th century. This biblical intervention in Western culture is now referred to as "Judeo-Christian culture" [5].

2.3.4 Sermon Construction

Creating a sermon is not a trivial task, as several variables must be considered for its development. One initial point to be observed is the audience to whom the sermon will be delivered. Another important aspect is the choice of sermon type, because there are at least four types of sermons (BROADUS, 2003): the textual sermon, the topical sermon, the textual-topical sermon, and the expository sermon. Each of these types of sermons has its advantages and disadvantages.

To ensure that the reading and exposition of a text are not influenced by subjectivity, hermeneutics is adopted as a tool for text interpretation, as this field of science provides guidelines that readers must follow so that they do not subjectively and partially absorb the message of a text. One of the mandatory rules of hermeneutics is the reader's observation of the context of the text, whether it is immediate or remote (FORSTER, 2019). This is essential for the reader and the expositor to understand the author's intention of the text and avoid extracting their own or inappropriate interpretations.

[5] https://jus.com.br/artigos/24834/influencia-da-etica-judaico-crista-nos-ordenamentos-juridicos-da-atualidade

3 Systematic Review: AI Applied to the Analysis of Biblical Text

Over the past years, AI has been extensively used in the analysis of documents, mainly combining Text Mining, Natural Language Processing, Neural Networks, Machine Learning and other fields of investigation. Altogether, they provide a powerful suite of tools for machine translation, authorship identification, tagging, semantic annotation, and even interpretation. In this direction, to the best of the author's knowledge, this thesis provides the first systematic survey on the use of AI in the analysis of Biblical Scriptures.

The thesis combines Kitchenham's and PRISMA-P protocols (KITCHENHAM, 2004); (MOHER et al., 2015), considering nine different, but complementary, search expressions over the Scopus and Web of Science repositories. From among 115 papers retrieved, 34 were included in the survey based on the inclusion and exclusion criteria. Three research questions were proposed: i) What are the main tasks solved by AI methods in the analysis of the Bible? ii) What are the main AI algorithms used in the analysis of the Bible? and iii) What are the main limitations of AI approaches in the analysis of the Bible?

Was applied some standard text mining methods in the 34 papers selected and found that machine learning, neural networks and deep learning are the most common AI techniques used in the literature. Furthermore, it was possible to observe the following main tasks being solved: machine translation; authorship identification; tagging; semantic annotation; clustering; categorization; and interpretation. Based on this knowledge, a systematic review was carried out by answering the three research questions, and then discussing the contents of most papers in each of these sections. Greater emphasis is given to the advancements and the limitations of the field, as well as the most widely used algorithms and their achieved results.

3.1 Overview of the Bible

The Holy Bible is a singular document in history. According with the Book of Records (GUINNESS. . . , 2022), the Bible is the world's most sold book, reaching 5 billion printed copies. The influence of the Bible achieved overall relevance in the literature, not only by

its extraordinary number of copies, but also for its crucial role in the alphabetization of medieval European populations, particularly in the 16th Century Germany (GAWTHROP; STRAUSS, 1984)), and for its participation in the formation of the early American political rationale of the 18th Century (LUTZ, 1984). Such prestige made the Bible a book to be studied for hundreds of years by theologists, critics, and enthusiasts.

The document widely identified as "The Holy Bible" is a collection of 66 books written by 40 authors from the Levant Region, whose existence comprehend a timespan of 1,600 years. It is divided in 1,189 chapters and 31,102 verses[6]. As the Holy Bible is studied by its numbers, its interpretative complexity is revealed. For example, when the Old Testament is tokenized, it generates 1.5 million tokens in vocabulary (BLEIWEISS, 2017). It also possesses a diversity of literary genres, such as poetic, narrative, metaphorical, among others.

The traditional method of Biblical knowledge extraction is human-based *hermeneutics*, that is, the method and theory of interpretation, mostly used for biblical texts (ZIMMERMANN, 2015). However, this is a process limited by the examiner's subjectivity and by the large extension of the Biblical text.

3.2 Research Protocol

This research combines Kitchenham's and PRISMA-P protocols to perform systematic reviews Kitchenham (KITCHENHAM, 2004; MOHER et al., 2015). The methodology was organized based on three main steps: *planning*, *conducting*, and *reporting*. The first step, planning, consists of defining the review protocol, that is, specifying the objective, the research questions, keywords, databases, and the inclusion and exclusion criteria. After defining the protocol, the survey of related works is made, selecting primary studies, performing quality assessment, extracting and synthesizing data. The final step is the papers' reporting itself, specifying the dissemination mechanisms and formatting the main report.

[6]https://www.biblebelievers.com/believers-org/kjv-stats.html

3.2.1 Research Sources and Terms

Scopus and Web of Sciences were chosen as search engines because they index a large number of scientific papers, including high impact journals. The main search expressions were based on a combination of the following terms: Bible, Artificial Intelligence, Text Mining, Neural Network, NLP, Machine Learning, Deep Learning, Computational Intelligence, Data Science and Data Mining. Table 5 sumarizes the search expressions, and the number of papers retrieved from each source. English was chosen as the language of the papers eligible to review. Figure 18 summarizes the methodology block diagram, which was implemented following the PRISMA-P protocol with the initial planning three phases of identification (data capture), screening, and inclusion.

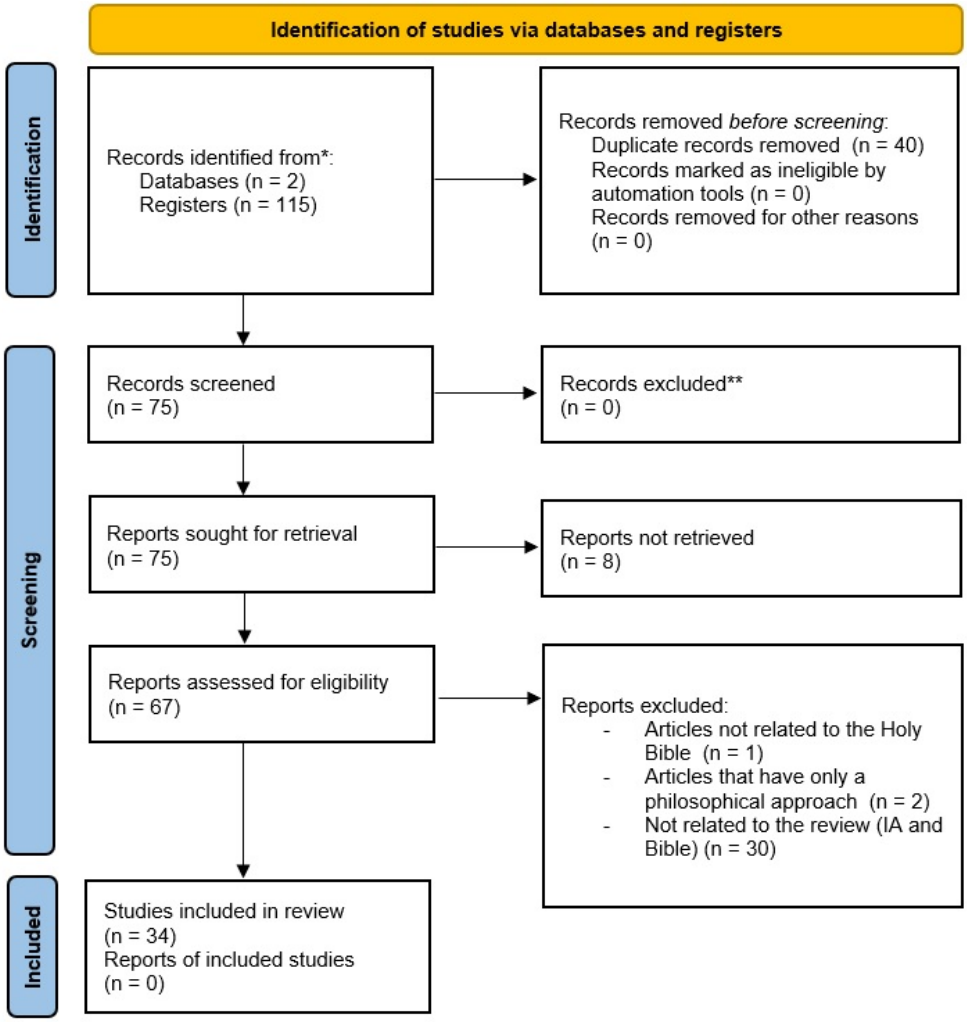

Figure 18: PRISMA Protocol.

Table 5: Search terms and papers retrieved from each search engine.

Keywords	Scopus	Web of Science	Selected papers
Bible AND Artificial Intelligence	17	5	22
Bible AND Text Mining	5	2	7
Bible AND Neural Network	10	4	14
Bible AND NLP	13	5	18
Bible AND Machine Learning	15	7	22
Bible AND Computation Intelligence	0	0	0
Bible AND Data Science	1	0	1
Bible AND Data Mining	9	2	11
Bible AND Deep Learning	8	12	20
Number of selected papers			115

3.2.2 Inclusion and Exclusion Criteria

The eligibility criteria for inclusion in the systematic review consist of being an original paper written in English, having a full text available, and displaying the application of AI to the Holy Bible texts. A key aspect for inclusion is the presentation and use of AI, and the chosen subareas, as a tool to extract knowledge from the Biblical text. Duplicates, abstract only, and works focused solely on the philosophical study of the Bible were excluded from the review. Table 6 summarizes the inclusion and exclusion criteria.

Table 6: Inclusion and exclusion criteria.

Inclusion Criteria	Exclusion criteria
Original papers	Duplicates
Papers written in English	Non-English languages
Complete Text	Abstract only (partial content)
Application of AI techniques in the Holy Bible	Lack of utilization of AI techniques in the Holy Bible
Use of AI to interpret the biblical text	Purely philosophical papers

3.2.3 Research Questions

The research questions aim at summarizing the field of knowledge related with the intelligent data analysis of the Bible, providing a broad view of state-of-the-art works in this area. The research questions that will be addressed are the following:

- Question 1: What are the main tasks solved by AI methods in the analysis of the Bible?
- Question 2: What are the main AI algorithms used in the analysis of the Bible?
- Question 3: What are the main limitations of AI approaches in the analysis of the Bible?

3.3 Systematic Review

This section starts by applying some simple text mining methods to the surveyed papers in order to present their context. It then follows with a presentation of the main AI techniques and applications in Biblical text analysis. In this direction, the works were divided into four main groups: machine translation and authorship identification; grammatical and semantic analysis; clustering and categorization; and Biblical interpretation.

3.3.1 Selected Papers and Their Context

To have a better understanding of the context, the frequency histograms for bigrams and trigrams extracted from the selected papers are presented in Figures 19 and 20, respectively. The bigrams show mainly the methods used and tasks solved in the literature. It can be observed that the main AI approaches used are machine learning, neural networks and deep learning, whilst the main tasks solved are machine translation, digital image processing, language processing, information retrieval, PoS-tagging (Part of Speech tagging) and topic modelling. The trigrams presented in Figure 20 reinforce these findings.

Figure 21 shows the bigrams from the abstracts of the excluded papers. As can be observed, the majority of the most frequent expressions in these papers are not related

with neither the Bible, nor artificial intelligence, supporting our exclusion criteria.

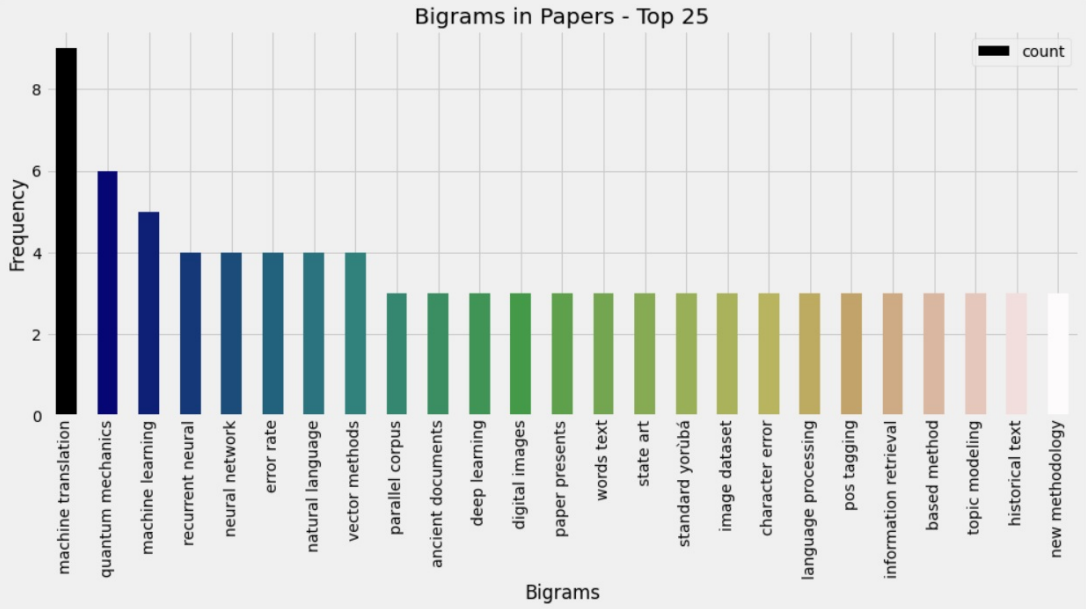

Figure 19: .

Bigrams generated from the selected papers.

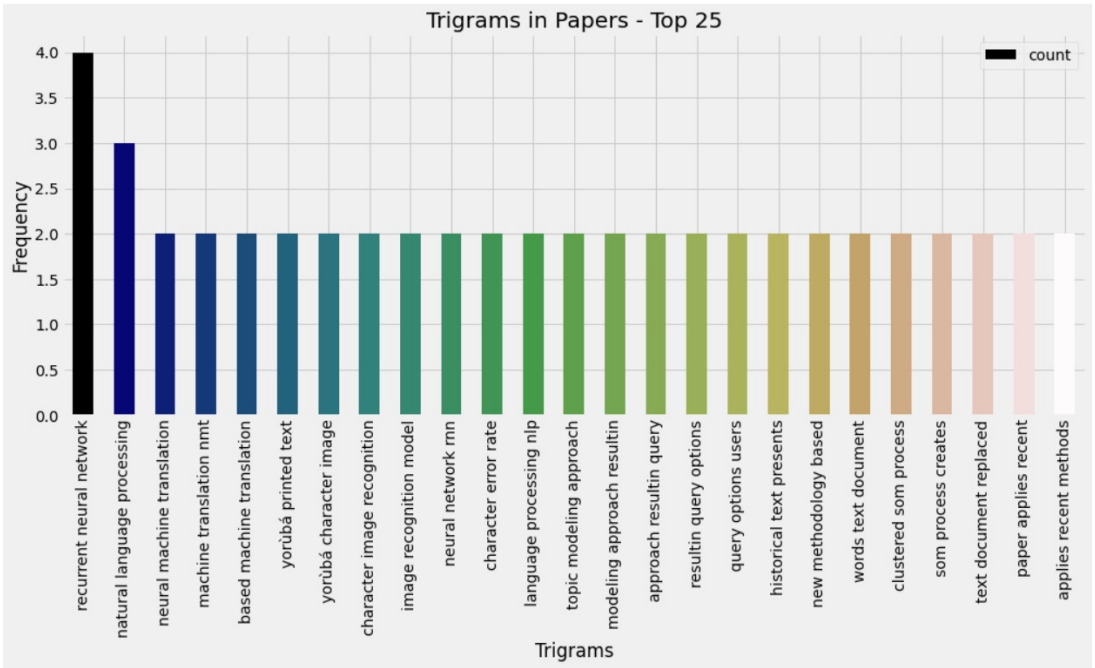

Figure 20: Trigrams generated from the selected papers.

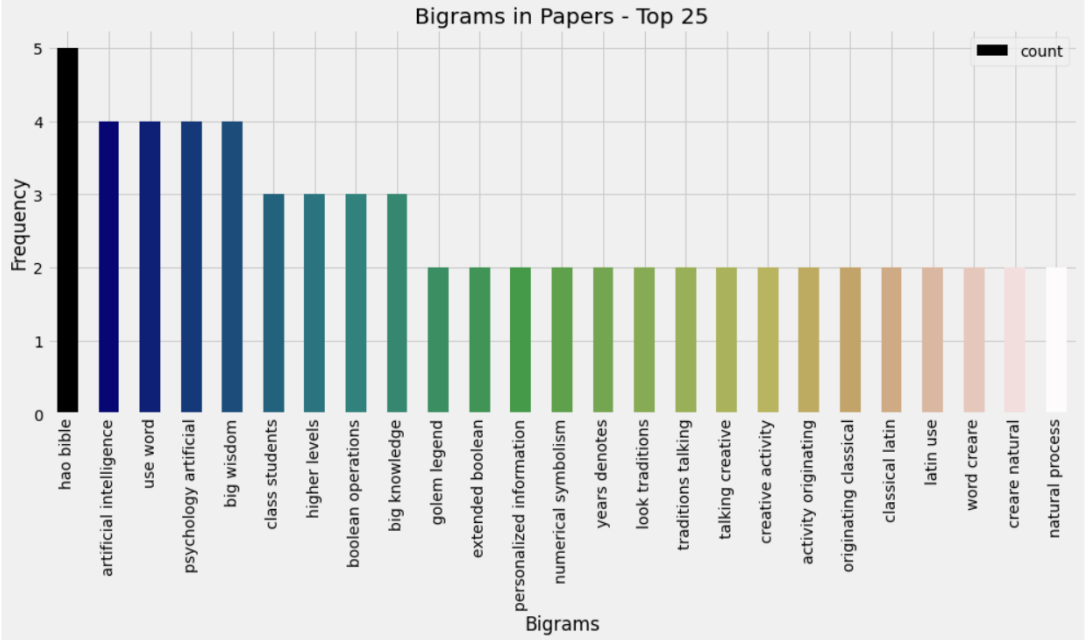

Figure 21: Bigrams generated from the excluded papers.

3.3.2 AI Methods and Applications in Biblical Text Analysis

The interest in scientific works where AI is applied to the Biblical literature is as recent as the beginning of the present millennium (2000), and has grown more intense over the past five years. After reading the selected papers, it was possible to organize in more detail the main application areas and AI techniques used, as will be discussed in the following sections. The discussions in each section will bring the answers to the three research questions proposed in this thesis.

Machine Translation and Authorship Identification

The selected papers show that the Biblical corpus is being used in association with artificial intelligence mostly for the development of automatic translators or authorship identification methods (ASHENGO; AGA; ABEBE, 2021; BRIA et al., 2018; EDER, 2016; CILIA et al., 2020a; CILIA et al., 2020b; ESAN et al., 2020; GOLOVIN et al., 2016; STEFANO et al., 2018; TSCHUGGNALL; SPECHT, 2016; ZIRAN et al., 2020; ÓNí; ASAHIAH, 2020; COVINGTON; POTTER; SNODGRASS, 2015; RISTA; KADRIU, 2021; THOMAS; VALENZUELA, 2020). Thus, textual recognition (text mining and natural language processing) is the main methodology in use. The preference for such method may be explained by the growing demand of the pattern recognition field as a

whole (VINOTHENI; LAKSHMANA, 2019).

Algorithms frequently used in papers that perform authorship recognition works are the ones related to deep learning and support vector machines (SVM) (ASHENGO; AGA; ABEBE, 2021; BRIA et al., 2018; CILIA et al., 2020b). The results shown in these works are robust and indicate the capability of these algorithms to recognize textual patterns and identify the textual styles of authors, even in ancient and grammatically complex languages from the medieval literature (CILIA et al., 2020a; BRIA et al., 2018). This goal of building translating machines for Biblical corpora is attractive, because many dialects and exotic languages suffer with the absence of automatic translators. This is because AI techniques are applied most commonly to broadly spoken languages, such as English. Limitations in this field include high numbers of false positives in identifying the authorship of medieval manuscripts via SVM (EDER, 2016), the high volume of hyperparameters in deep neural networks (CILIA et al., 2020a; BRIA et al., 2018), and a dependance of a symmetrical source comparable to the training source coupled to a post-processing stage to reduce the characters recognition error rate by using Long Short-Term Memory (LSTM) networks.

Machine Translation

In the paper ESAN et al. (2020), the authors used a Recurrent Neural Network (RNN) to train a machine-translation solution using a standard text mining process: data loading, tokenization, vocabulary building, model training and evaluation. A Biblical Corpus was used for both training and evaluation purposes. The works of Tschuggnall e Specht (2016), Thomas e Valenzuela (2020), Rista e Kadriu (2021) adopted a similar methodology.

In the work of Ashengo, Aga e Abebe (2021), a bilingual dictionary was used along with a Context-Based Machine-Translation network (CBMT). Also, a Recurrent Neural Network Machine-Translation (RNNMT) was used as the output of the CBMT. In this case, a biblical corpus was used to evaluate the model. In the research of Ziran et al. (2020), the authors used two Faster R-CNN models, the first one focused on recognizing generic words and the second one used to recognize reference words. The Gutenberg Bible was used to evaluate the proposed machine-translation model.

Authorship Identification

The work presented in Bria et al. (2018) was conducted with five trained Deep Neural Network (DNN) models: VGG19, ResNet50, InceptionResNetV2, InceptionV3 and NASNetLarge. With a workflow involving transfer learning and fine tuning, the authors evaluated the performance of the authorship identification solution by making use of a 12th Century Biblical corpus. In the work of Eder (2016), by contrast, the author proposed a stylometric method combining supervised learning and sequential analysis to assess mixed authorship. Three different versions of the method were proposed: Rolling SVM; Rolling NSC, based on the Nearest Shrunken Centroids method; and Rolling Delta, based on the Burrowsian measure of similarity. Among different datasets, he used a 15th century translation of the Bible into Polish.

In the research of Cilia et al. (2020b), the adopted methodology was the creation of two Convolutional Neural Networks (CNN), the first one focused on detecting each line of the manuscript, and the second one responsible for identifying an author to each respective line. The processing occured in two training stages: transfer learning and fine tuning. A copy of a medieval Bible (Avila Bible), was utilized to evaluate the model. The authors justified the use of the medieval Bible because it is a robust source of data that allows for authorship identification of its scholars. In the work of Stefano et al. (2018), by contrast, the implementation of a set of algorithms was performed, including decision trees, K-Nearest Neighbor, neural networks and SVMs.

In the works of Cilia et al. (2020a), Cilia et al. (2020b), the authors proposed a system to identify writer in medieval documents, and used digitized manuscripts of the Avila Bible to assess their proposal. The authorship identification process was divided in three stages: a detector of objects to identify each line on a page; the construction of a deep neural network with transfer learning; and a weighted-majority vote to assign a writer to each page. The goal was to investigate the use of DL with relatively small data sets. The following works applied a similar approach: (ÓNí; ASAHIAH, 2020; COVINGTON; POTTER; SNODGRASS, 2015).

3.3.3 Part of Speech Tagging and Semantic Annotation

Other common tasks solved by AI in the analysis of the Bible include Part of Speech tagging and Semantic Annotations (DIONE; KUHN; ZARRIESS, 2010; FRANCIS; NAIR,

2014; COECKELBERGS; HOOLAND, 2016; VARGHESE; PUNITHAVALLI, 2019; YU et al., 2016; ÖSTLING; TIEDEMANN, 2017; AZAWI; AFZAL; BREUEL, 2013; CERNANSKY et al., 2007; BILOVICH; BRYSON, 2008; JAENISCH et al., 2002). These works usually aim to collect information related to word use in an attempt to build dictionaries. The most commonly used algorithms for this kind of research are the classic ones from machine learning, such as Decision Trees, Support Vector Machines, Conditional Random Fields (CRF), K-Nearest-Neighbors (KNN), Bagging, Random Forests, Gradient Tree Boosting and Topic Modelling. The AI techniques that presented the most promising results were SVM, where their performance exceeded those of related papers (YU et al., 2016) and Topic Modelling, which generated insights regarding Hebraic literature diachrony and added the creation of a historical Hebraic ontology (COECKELBERGS; HOOLAND, 2016). The main limitations of these works are how the performance relies on the size of the dataset.

Part of Speech Tagging (PoS Tagging)

In the work of Dione, Kuhn e Zarrieß (2010), the authors created tags in conformity with the EAGLES guidelines to elaborate PoS Tagging. These tags were conceived to reflect verbal inflections within the sentences or in their context. The authors designed and developed annotated corpus resources to support PoS Tagging for a language from the Niger-Congo called Wolof. As the first effort to build a publicly available NLP resource for Wolof, they used part of the Bible as the gold standard corpus. For the training of the PoS tags, 26,846 tokens were generated from the Gospel of Matthew of Wolof's Bible. Two well-known machine-learning taggers were used, namely TnT tagger and TreeTagger, and the results were compared with a baseline that assigns the most frequent tag from the training set to each known word.

In the work of Francis e Nair (2014), a hybrid PoS Tagging model was proposed combining elements from a Rule-Based method and an n-gram model. In both approaches the authors executed morphological, lexical and syntax analyses. When displaying their results, the authors showed that the proposed machine-learning-based solution performed better than SVM.

Yu et al. (2016) proposed a delexicalized tagging method for cross-language transfer of PoS Tagging models that requires only a raw corpus of the target language. Their

proposal was composed of four main steps. The first focuses on the identification of the original languages (of which they already possessed the labels) and the identification of the destination languages. In the second stage a feature vector was produced, and also the destination language labels were obtained. In the third stage, the origin language vectors are used as training set for machine learning classifiers, such as KNN and SVM. Finally, in the fourth stage, its efficiency is evaluated for the destination language.

Azawi, Afzal e Breuel (2013) proposed a new approach for the normalization of historical texts for applications such as PoS Tagging. They made use of Martin Luther's Bible (1545 version) as the textual corpus. They also implemented a deep neural network (LSTM – Long Short-Term Memory), using words from modern and ancient German, to make comparisons during training.

Semantic Annotation

Coeckelbergs e Hooland (2016) adopted a dataset with annotations from the Hebrew Bible "SHEBANQ" to run a topic modeling technique, more specifically, the Linear Discriminant Analysis (LDA) algorithm. They created a tool that readers can use to study the biblical text in Hebrew. In the work of Varghese e Punithavalli (2019), the authors started with the hypothesis that there is intersection between the Bible, Tanakh and Quran. The goal was to perform text analytics on sacred texts to find similarities among them using NLP, ontology and ML methods. In the research of Östling e Tiedemann (2017), a project of semantic relations of languages was conceived, aiming to infer connections among them using continuous vector representation of languages. The experiments used a volumous biblical corpus with 1,303 translations in 990 languages. Their model was based on a deep neural network (LSTM) and various word emmbeddings. The papers written by Cernansky et al. (2007), Bilovich e Bryson (2008), Jaenisch et al. (2002), adopt similar methodologies.

3.3.4 Clustering and Categorization

The works related to the grouping or categorization of Biblical texts are modest when compared to the other aforementioned fields (VALDIVIA; VEGA; LÓPEZ, 2003; BLEI-WEISS, 2017; POPA; GOGA; GOGA, 2015; GESSNER; KÖTTERITZSCH; LAUER, 2013; VISA et al., 2001; ARI et al., 2014). The main idea in segmenting the Biblical text

is to investigate semantic similarities in order to evaluate the capacity of some algorithms to infer contextual synonymity.

The are few works in this area and the main obstacles are the fact that the algorithms have to categorize a multilingual textual corpus (VALDIVIA; VEGA; LÓPEZ, 2003), to identify similarities among authors from different geographical regions (WIDDOWS; COHEN, 2009), and the vectors used in the representation of words or sentences (BLEIWEISS, 2017).

The algorithms employed in this field are mostly unsupervised, such as Rocchio algorithm, Widrow-Hoff algorithm, Kivinen-Warmuth algorithm, Learning Vector Quantization (LVQ), Vector Space Model (VSM), Latent Semantic Analysis, Self Organizing Maps (SOM), K-means and some deep neural networks. This makes evident that the techniques used to find textual patterns in the Biblical literature are under the same paradigm as the ones used in data mining, where the unsupervised approach is not related with content search. This reinforces the necessity of neutrality and of removing biases in the grouping of Biblical texts, so one can in fact identify textual patterns.

The main limitation of this approach is related with the fact that in representation vectors building, the number of dimensions implies in excessive need of memory space to process the texts (BLEIWEISS, 2017; CERNANSKY et al., 2007).

Clustering

(WIDDOWS; COHEN, 2009) conducted a comprehensive research about the clustering of a Biblical corpus by using machine-learning and linear algebra techniques. Their research aimed to investigate if the algorithms are capable of identifying similarities among the three synoptic gospels (Matthew, Mark and Luke) in contrast with the gospel of John. In another stage of the research, it was attempted to relate Biblical characters with geographical regions. The machine-learning algorithm used to perform clustering was the K-means. The papers from Bleiweiss (2017), Popa, Goga e Goga (2015), Geßner, Kötteritzsch e Lauer (2013), adopted similar methodologies.

Categorization

In the work of Valdivia, Vega e López (2003), the authors proposed a text categorization method based on unsupervised and competitive learning paradigms. The authors

reported comparisons between Learning Vector Quantization (LVQ) and other machine-learning algorithms, concluding that LVQ presented better results for text categorization. The other algorithms compared were the Rocchio, Widrow–Hoff and Kivinen–Warmuth algorithm. Two biblical translations were used for training the categorization algorithms: the Reina Valera edition for Spanish; and the American Standard Version for English.

Visa et al. (2001) presented a prototype solution whose goal was to extract knowledge from textual content. Such solution used Self Organizing Maps (SOM) and a vector representation of the words. Categorization was performed in three levels: word, sentence and paragraph. The authors declared to have reached their goals with the proposed method. The paper by Ari et al. (2014) used a similar approach.

3.3.5 Biblical Interpretation

When it comes to the use of AI to interpret the Bible, scientific contributions are shy in numbers (MURAI, 2013; HU, 2012; ZHAO; LIU, 2018). The AI techniques employed are more related with representation than extracting implicit knowledge. Thus, an exploration of the use of AI techniques, such as DNN and segmentation algorithms, that aim to examine semantic aspects as a whole could be of interest. One of the limitations identified in this area is the difficulty of extracting contextual knowledge, such as historical and cultural facts. However, the utilized methods are those of representation, and no extrapolation techniques were applied (MURAI, 2013). In the work of Zhao e Liu (2018), it was noticed that the simplest model (RNN) displayed a better result when compared with a more complex one (BiDAF). It concluded that Bible translations that follow a more literal style eventually display lower performance, since the coding was limited to consider only the semantic of the sentences and not their syntax.

In the work of Hu (2012), an analysis of the Book of Psalms and the Book of Proverbs was conducted in order to achieve similar conclusions to the ones found by hermeneutics scholars. The authors reported to have been able to extract novel information from the texts. The applied method was relatively simple, making use of the Latent Dirichlet Allocation (LDA) algorithm. In the paper of Zhao e Liu (2018), the development of a system based on questions and answers from the Biblical text was conducted. To reach this goal, the authors used two textual datasets (SQuAD e BibleQA) as well as a word

enbedding model (Word2Vec).

3.4 Discussion and Open Issues

The Biblical literature is relevant for society, given its publication statistics and reach. Thus, the knowledge obtained via text analysis aids the interpretation of the Bible. However, the findings of this systematic review showed that the use of AI in the interpretation cycle of the Biblical text is among the most scarce ones when compared with other, more constrained, works. It was possible to observe that the algorithms employed in the papers reviewed are, in their majority, the same as the ones applied in typical text mining and natural language processing applications.

The search involving the use of AI in Biblical text analysis was performed using nine search terms in Scopus and Web of Science, and retrieved 115 papers, from which 34 were selected for review. This is mainly due to the popularization of techniques like machine and deep learning. As reviewed, most works deal with machine translation, authorship identification, PoS-tagging, semantic annotation, clustering, categorization, and Biblical interpretation.

Recurrent neural networks were the most frequently used approach in Biblical text analysis, mainly because one of the main characteristics of the Biblical literature is the abundance of symbology, typology, and textual genres, which make evident its non-triviality. The Bible also possesses two literary genres that are more prevalent than others: narrative and poetry.

In terms of data, the Avila Bible was the most frequently used, maybe because it is a text written by different scholars in the Medieval period and which enabled the design and implementation of authorship identification solutions. The King James version is, however, the most widely adopted by the general public.

Was observed that not all analysis techniques used were open-source or freely accessible, some proprietary frameworks were chosen in specific papers. Thus, the scientific discussion of these methods is difficult, as there is no detail of the AI mechanisms behind such frameworks. This is the case for the Text2Onto and Gertrude frameworks.

Last, to better understand the performance of the AI methods used, it is necessary

to have a clearer description of the metrics used to assess their performances. However, many papers surveyed did not contain such description and a better formalism in this direction is certainly the subject for future study.

The present systematic review investigated scientific works that used AI applications for the discovery of implicit knowledge in the Bible. Three research questions guided the review: the main tasks solved by AI; the main algorithms used; and the limitations of AI approaches when applied to the Bible text analysis.

In terms of the three research questions, the following were the main findings:

- *Question 1: What are the main tasks solved by AI methods in the analysis of the Bible?* The main tasks involving the use of AI techniques in the Biblical literature are machine translation, authorship identification, semantic annotation, PoS tagging and, more scarcely, categorization, clustering and textual interpretation.

- *Question 2: What are the main AI algorithms used in the analysis of the Bible?* The techniques most commonly used are KNN, K-means, Deep Learning (LSTM, RNN, DNN, CNN), SVM, Decision Trees and Self-Organizing Maps. It is worth noting that Deep Neural Networks were the preferred method, achieving consistent results in most works reviewed.

- *Question 3: What are the main limitations of AI approaches in the analysis of the Bible?* The limitations found in the papers correspond to classical problems generally found in data mining, such as memory storage (BLEIWEISS, 2017), dataset size (FRANCIS; NAIR, 2014)) and asymmetry between the training data and the real or test data (ÓNí; ASAHIAH, 2020). However, some limitations are specific to AI applications in the biblical literature, such as the identification of contextual elements in biblical texts (MURAI, 2013), which is a highly complex task. This complexity can be explained by the diversity of genres that compose the biblical text and its semantics heavily charged with symbology and typology. This is a bottleneck that must be overcome for the application of AI in biblical interpretation. Other limitations include a high number of false positives in classification tasks (EDER, 2016), the difficulty in finding suitable deep network architectures for dealing with the Bible text (ZHAO; LIU, 2018), and the lack of standardized

performance metrics in the field.

The findings showed that this field is still recent and with a scarce literature. The main goal of using AI in the Bible text analysis is the development of translation machines and not knowledge extraction, as could be expected. Recurrent neural networks have shown better performance for translation machines and authorship identification. It is worth noting the fact that there is a gap when it comes to applying AI for the Bible text interpretation. It is reasonable to conceive that the scientific development in this direction will contribute to the works of theologists and natural language processing research in general, because the biblical literature encompasses an emblematic and diverse textual format. Algorithms that achieve a satisfactory performance level when applied to the biblical literature may also do so in texts with similar characteristics.

As future works in the field it is possible to detach the used of AI in the interpretation of biblical corpora, a deeper investigation into performance measures in the analysis of the Bible with AI, the experimentation with other AI algorithms (e.g., text mining and NLP approaches); and the adoption of visualization tools for the knowledge extracted from the Bible texts. And, finally, apply optimization techniques and approaches from natural computing since there is no scientific work with these techniques applied to the biblical corpus. Therefore, this thesis seeks to contribute to filling this gap.

4 SWARMABLE: An Ant Colony Optimization Algorithm for the Selection of Bible Passages

Within the overall objectives of this work, the author aims to develop a hybrid solution that employs Natural Language Processing (NLP) and optimization techniques for the selection of biblical passages, considering a specific search query for sermon composition.

This solution is called SwarmaBle and adopts a multidisciplinary approach, combining techniques from various areas, such as:

- NLP: involving the application of machine learning mechanisms, such as the construction of embeddings or other forms of textual representation.

- Swarm Intelligence: using meta-heuristic algorithms for optimizing combinatorial problems, such as ACO. Additionally, it incorporates the bioinspiration of these meta-heuristics derived from natural computing.
- Theology: by adopting the biblical corpus, the approach indirectly involves theological science and the interpretation of texts.

The diagram presented in Figure 22 illustrates the relationship between these areas and the proposed pipeline in the thesis.

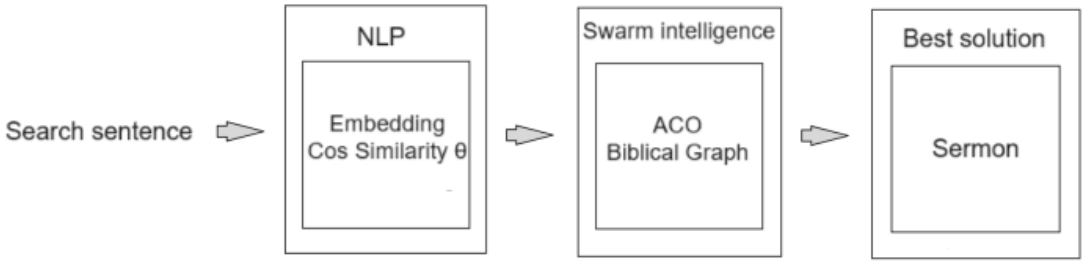

Figure 22: Proposal overview

The proposal of this thesis aims to contribute significantly to a noble task of developing contextual sermons, through the selection of biblical passages for pastoral preaching. The target audience that can benefit from the recommendations of biblical texts includes religious individuals (pastors/seminarians), students of sacred texts, hermeneutics scholars, exegetes, among others. The contribution to this field is innovative due to the combinatorial optimization of biblical verses. Furthermore, it contributes to the advancement of research in NLP and natural computing

4.1 SwarmaBle Flow

This thesis proposes a hybrid approach to constructing sermons from biblical passages returned by the optimized process. Initially, natural language processing is used to analyze the biblical text, followed by ant colony optimization in a subsequent phase.

The methodology employed to develop these two execution phases (NLP and ACO) will be described in two sections. The first section covers the natural language processing methodology, while the second section addresses the ant colony optimization methodology.

4.1.1 Natural Language Processing

As described in the 2.1 section, human language is a complex phenomenon, representing a significant computational challenge. In response to this, researchers developed natural language processing to enable machines to approach the understanding of human language.

The natural language processing adopted in this work revolves around three distinct stages: textual preprocessing, textual representation, and text matching or similarity textual.

For the feasibility of natural language processing, the Python programming language was used, employing standard NLP libraries, namely, SpaCy and NLTK. The algorithm 2 for natural language processing is illustrated below:

- Preprocessing: Text preprocessing involved several tasks, including tokenization and stopwords removal.

    1. Tokenization: The tokenization performed in this work is done considering the white spaces between words. In other words, at each white space, it is understood that there is a token.
    2. Stopwords: The process of removing stop words utilized the standard approach. In other words, the removal involved discarding frequently occurring words without analytical meaning, such as “and,” “a,” “the,” etc., using the default stopword list from the Python NLTK libraries, without employing an additional list or specific words..

- Textual Representation: In order to broaden the semantic space of the textual representation of the Bible corpus, vector representation by word embeddings was adopted using the FastText architecture. FastText is an approach aimed at capturing semantic information, taking subwords into account. Unlike traditional word

Algorithm 2 Search Similar Verses - NLP

```
1: procedure SearchSimilarVerses(referenceSentence, corpusURL, similarityThreshold)
2:     Download necessary libraries and resources (NLTK, spaCy, etc.)
3:     Preprocess data and load the Bible corpus
4:     for each book in the corpus do
5:         for each verse in the book do
6:             Preprocess the verse
7:             Calculate similarity with the reference sentence
8:             if similarity above the threshold then
9:                 Store result in the format book/chapter/verse/similarity
10:            end if
11:        end for
12:    end for
13:    Sort results by similarity in descending order
14:    Return the results
15: end procedure
```

embeddings such as Word2Vec or GloVe, FastText incorporates character n-grams, allowing the representation of out-of-vocabulary words and capturing morphological information. The result of using FastText embeddings is a vector representation of words in the text, where each word is associated with a high-dimensional vector in a continuous space.

- Text Matching or Similarity: The semantic similarity of the Bible corpus will be measured using cosine similarity. This choice is based on the robustness of this similarity measurement method in the literature, as evidenced by the work AGGARWAL e ZHAI (2012). It calculates the cosine of the angle between two vectors, ranging from -1 (completely dissimilar) to 1 (identical). The formula for the cosine similarity between vectors u and v is:

$$Cosine(u, v) = \frac{u \cdot v}{\|u\|\|v\|} \tag{10}$$

4.1.2 Database

The textual corpus that will be used is the King James Holy Bible, an English version composed of 66 books. This textual corpus is available on the GitHub repository[7] in txt format.

No modifications are made to this textual corpus, meaning the original format provided by the repository is used.

According to the authors HERBRICH e GRAEPEL (2010), creating a textual corpus is not trivial and involves extensive effort and cost. The corpus needs to be representative and distributed coherently. Furthermore, as stated by the same authors HERBRICH e GRAEPEL (2010), a large corpus is more significant from both lexicographical and statistical perspectives.

[7] https://raw.githubusercontent.com/pstephens/kingjames.bible/main/kjv-src/kjv-1769.txt

4.1.3 Swarm Intelligence - Ant Colony Optimization

The present thesis proposes an innovative algorithm titled "*SwarmaBle*," which constitutes a swarm intelligence approach based on the Ant Colony Optimization (ACO) algorithm, implemented to select biblical passages for composing pastoral sermons. The scheme below (Figure 23) synthesizes the expected output at the end of the optimization process, In other words, biblical passages that will comprise a sermon, produced by SwarmaBle.

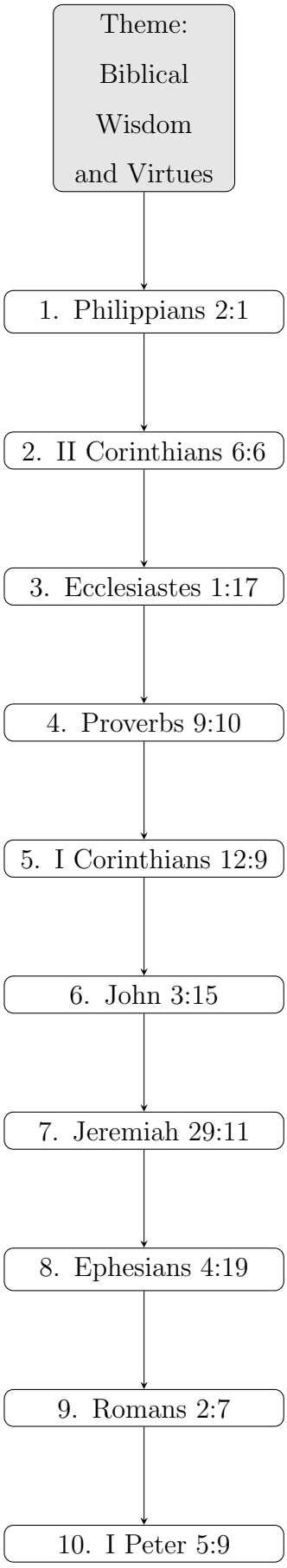

Figure 23: Biblical passages for sermon construction, generated by SwarmaBle.

SwarmaBle consists of a workflow summarized in Figure 24, with the following steps involved:

- Loading the Bible Corpus: SwarmaBle begins its pipeline with the loading of the Bible corpus. The user has the option to use the complete Holy Bible, composed of 66 books, or to select specific books during the search and sermon construction process. It is worth noting that any English version of the Bible can be loaded into SwarmaBle.

- Input Theme or Biblical Passage: In the second step, the main theme of the sermon is chosen. This can be performed either by simply choosing a theme (e.g., spiritual growth, happiness at work, harmonious family, etc.), or by selecting a specific Biblical passage that contains the message to be delivered in the sermon.

- Definition of Output Parameters: In a third step, the number (N) of passages that will compose the sermon is established, and simultaneously, the parameters for the scope of the search are defined. In other words, it is decided whether the search will be restricted to just one book, to a set of books, or to the entire Bible.

- Construction of the Bible Graph: In the fourth step, a graph is built that encompasses all the verses of the Holy Bible. In this graph, verses are represented as nodes, and connections between them are represented as edges. The weights assigned to the edges reflect the measure of similarity between the verses. Named the "Bible Graph," this structure is fundamental for the optimization process by the Ant Colony Optimization (ACO) Algorithm.

- Finally, a search is conducted in the Bible Graph for a path that minimizes the value, thus selecting the desired N verses to compose the sermon.

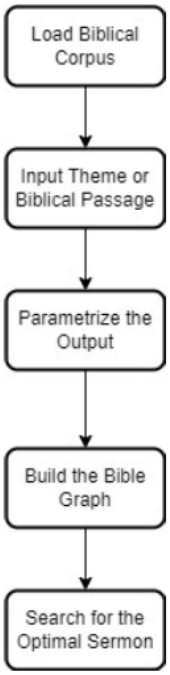

Figure 24: SwarmaBle workflow

4.2 Construction of the Bible Graph

The Bible graph is constructed based on the results obtained in the NLP phase, where similarities between biblical verses are calculated for the entire Bible or the set of book(s) chosen by the user.

In the bible graph, nodes or vertices represent the verses, while edges symbolize the connections between these verses. The graph is fully connected, and the edge weights reflect the calculated similarities based on the cosine similarity measure from NLP.

Formally, the Bible graph, denoted as B_g, is a complete graph, represented by $B_g = (V, E)$, where V is the set of vertices or nodes (biblical verses), and E is the set of edges connecting all the verses. The set of vertices V is defined as $V = v_1, v_2, v_3, ..., v_n$, representing each individual biblical verse. The set of edges E is established so that each pair of verses is connected by an edge, resulting in a complete graph $E = (v_i, v_j)|v_i, v_j \in V, i \neq j$. This implies that there is an edge connecting each verse v_i to all other verses v_j, where $i \neq j$. Thus, the Bible graph used by SwarmaBle is fully connected and weighted. Figure 25 illustrates this Bible graph.

The length of the path between nodes is given by the inverse of the sum of similarities.

This means that the larger the sum of similarities, the shorter the length of the path. The equation 11 illustrates this concept.

$$L_k = \frac{1}{\sum_{E=1}^{n} \eta_{kE}} \tag{11}$$

where, L_k is the length of the path, η is the similarity, and E is the verse.

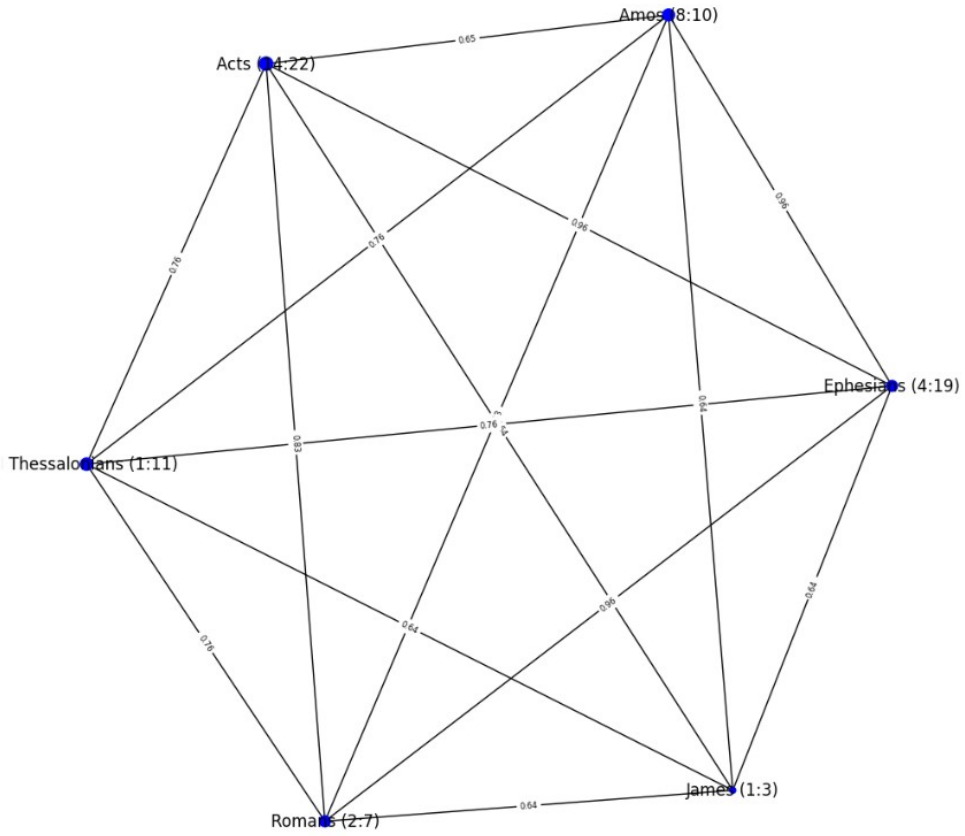

Figure 25: Bible Graph

4.3 ACO for Search in the Sacred Text

The ACO used in this thesis is classical, meaning it has the essential structure of the ACO proposed by Dorigo (1991). However, there is a specificity in which the ants do not need to visit all nodes but only a subset of nodes. Algorithm 3 outlines the operation of the ACO used.

Algorithm 3 ACO

```
1: procedure ACO
2:     Initialization
3:     Hyperparameters
4:     Ants
5:     while t < maxit do
6:         Build solutions (Search Strategy)
7:         for each ant do
8:             Evaluate its solution
9:         end for
10:        Update pheromones (Pheromone Update)
11:        t = t + 1
12:    end while
13:    Output solutions
14: end procedure
15: procedure SearchStrategy
16:    for each ant do
17:        while ant has not visited N nodes do
18:            Select next node (Eq. 12)
19:            Move ant to the selected node
20:            Update pheromones (Eq. 13)
21:        end while
22:    end for
23: end procedure
24: procedure PheromoneUpdate
25:    for each edge in the graph do
26:        Evaporate pheromone on the edge
27:        if edge used by the best ant(s) then
28:            Deposit additional pheromone
29:        end if
30:    end for
31: end procedure
```

ACO involves the following two main procedures (DORIGO; STüTZLE, 2004; DORIGO; STÜTZLE, 2019):

- **Solution Construction Search Strategy**: Artificial ants construct solutions incrementally by making probabilistic decisions in relation to the nodes to be traversed based on the pheromone levels and a heuristic function. The heuristic function provides information about the quality of a solution component. This solution construction mechanism blends global exploration and local exploitation. Ants explore the solution space, and local search procedures may be applied to improve the solutions.
- **Pheromone Update**: After constructing solutions, the pheromone levels are updated, simulating the natural decay of pheromone and reinforcing paths taken by ants that lead to better solutions.

The algorithm iterates until a predefined *termination criterion* is met, such as a maximum number of iterations (*maxit*), a time limit, or a desired solution quality. In the current context of this thesis, the stopping criterion is reaching the maximum number of iterations.

The problem to be solved is modeled as a graph, where the nodes represent different items (e.g., a state, a configuration, a city, etc.), and the edges represent transitions between these items. Each edge has an associated weight, which could represent physical distance, time, monetary cost, similarity degree, or any other metric that is appropriate for the problem at hand. In the context of the current thesis, the nodes represent (verses), and the edges represent (similarity).

The goal is to find the optimal path through this graph that minimizes (or maximizes) the total cost. The ants build solutions by moving from node to node along the edges of this graph, guided by pheromone trails and heuristic information, until a certain predefined stopping criterion is met. The probability p_{ij} that an ant moves from node i to node j is given by:

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{k \in N_i}(\tau_{ik})^{\alpha}(\eta_{ik})^{\beta}} \tag{12}$$

where τ_{ij} is the amount of pheromone on edge (i, j), η_{ij} is the desirability of edge (i, j),

N_i is the feasible neighborhood of node i, α and β are parameters that control the relative importance of the pheromone versus the desirability, respectively.

While traversing an edge, an ant lays an amount of pheromone on it, $\Delta\tau_{ij}$ proportionally to the quality, L_k, of the route chosen by ant k and a predefined constant amount Q:

$$\Delta\tau_{ij} = \frac{Q}{L_k} \tag{13}$$

where $\Delta\tau_{ij}$ is the amount of pheromone laid on edge (i, j) by the k-th ant, Q is a constant pheromone amount (input parameter), and L_k is the cost of the k-th ant's tour.

The pheromone update rule is given by:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \tag{14}$$

where ρ is the evaporation rate, and $\Delta\tau_{ij}$ is the amount of pheromone ants deposit on edge (i, j): $\Delta\tau_{ij} = \sum_k \Delta\tau_{ij}^k$, and k is the index of ants.

Therefore, as presented in the pseudocode, the ACO algorithm implemented in this work has three essential parts:

4.3.1 Initialization

In SwarmaBle, the number N of ants is equal to the number of verses that will compose the sermon. Hyperparameters α and β will be initialized with different values to assess the sensitivity of the algorithm to them. Parameters ρ and Q will be set using standard values from the literature.

After initializing the hyperparameters, the N ants are initially placed in N randomly chosen nodes (verses) of the graph and the iterative process starts and continues until the maximal number of iterations, $maxit$, is achieved. This values $maxit$ was chosen empirically.

4.3.2 Search Strategy

The DeployAnts procedure is responsible for controlling the ants' movements within the Bible graph. Each ant has to select a next node to move to until it has passed through N nodes in the graph. It is important to realize that, differently from the standard use of ACO in routing problems in which all nodes in the graph have to be traversed by the ants, in SwarmaBle only a fraction of the graph's nodes are selected by the ants. The selection of the next node to move to follows the probabilistic rule of Eq. 12. The last step in DeployAnts procedure is the deposition of pheromone in the traversed edges following Eq. 13.

4.3.3 Pheromone Update

The last procedure in the ACO algorithm is responsible for the pheromone evaporation (Eq. 14) and the application of elitism, that is, the reinforcement of one or more best routes (ants) in the swarm.

4.3.4 Fitness Function and Objective

The measure of solution quality, or "fitness," in this thesis is the sum of all edge weights (similarity) in each route (sermon) to be constructed by the ants. The equation 15 summarizes this concept.

Fitness of ant k

$$f(k) = \sum_{i=1, i \in S}^{N} w_i(k) \tag{15}$$

where N is the number of verses in the sermon, and S is the set of verses selected by ant k.

The goal of the SwarmaBle search is to maximize the "fitness" function. In other words, the problem consists of finding the maximum path, where the length of this path represents the quality of the "fitness" function. The higher the "fitness" value, the more similar the sermon will be to the search sentence. Equation 16 represents this reasoning.

Objective function:

$$max_k = (f(k)) \tag{16}$$

5 Performance Evaluation

The methodology and experimental resources used will be described to ensure the reproducibility of the results achieved.

5.1 Materials and methods

The materials used for the experimental phase of this thesis involve the use of a local coding environment, with the Python programming language being used in all stages of the research. The Python libraries used in the implementation were the classic ones for mathematics, data manipulation, graph visualization, NLP, and others.

The experimental method adopted in this work consists of performing a sequence of executions of the SwarmaBle algorithm, basically varying two parameters: α and β. The parameters ($maxit$, ρ, and Q) have fixed values in the algorithm executions, namely: $maxit = 3,000$, $\rho = 0.001$, and $Q = 0.01$. The stopping criterion adopted in this thesis was $maxit$ iterations reached.

The values of α will be alternated among the following values, with $\beta \in 0.5, 1.0, 2.0, 5.0$, and the values of β will be alternated in an equivalent manner for the following values of $\alpha \in 0.5, 1.0, 2.0, 5.0$. Table 7 below summarizes the variation of the parameters α and β.

For each parameter α and β, 5 executions are performed to quantify the mean and variance values, as the algorithm has stochastic nature. Table 8 summarizes the values of this experimental battery.

It is worth noting that, for each set of SwarmaBle algorithm executions, a distinct search sentence was used. That is, for the search sentence (X_1), executions were performed varying the α and β parameters with a sequence of 5 repetitions to assess the static performance values of the algorithm.

The author of this work chose to use two distinct approaches for search sentences or

sermon themes. In other words, one approach involves constructing a sentence containing the theme for sermon development, for example: 'How to have a blessed life?' Another approach to the algorithm's search sentence would be to use the biblical text itself as a sentence, i.e., choose a verse that synthesizes the theme for composing a sermon, for example: Matthew 19:1.

Table 7: Experimental Scheme - SwarmaBle

		Beta				ρ	Q	$maxit$	
Search Sentence	**Alpha**	0.5	1	2	5				
X_1	0.5	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	0.001	0.01	3,000	
X_1	1	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	0.001	0.01	3,000	
X_1	2	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	0.001	0.01	3,000	
X_1	5	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	5 runs of the SwarmaBle algorithm	0.001	0.01	3,000	
Total runs		20	20	20	20				Total absolute 80 runs

End of table

Table 8: Values from the experimental phase

Quantity of variations of parameters (α or β)	Quantity of times SwarmaBle is executed for each parameter value (α or β)	Total executions for each sentence	Total of sentences	Total passages Biblical per sermon	Total sermons generated	Total passages biblical passages generated in the experimentation
4	5	80	6	10	480	4,800

5.2 Experimental Results and Discussion

5.2.1 Results of the experimental phase of SwarmaBle (sensitivity)

The experimental results are presented in Table 9.

Table 9: Experimentation SwarmaBle - Sensitivity

		Beta			
Search Sentence	**Alpha**	0.5	1	2	5
Gn 1.1	0.5	Avg = 0.49 Std = ± 0.29	Avg = 0.50 Std = ± 0.08	Avg = 0.51 Std = ± 0.05	Avg = 0.52 Std = ± 0.05
Gn 1.1	1	Avg = 0.51 Std = ± 0.01	Avg = 0.52 Std = ± 0.07	Avg = 0.52 Std = ± 0.08	Avg = 0.52 Std = ± 0.05
Gn 1.1	2	Avg = 0.50 Std = ± 0.08	Avg = 0.50 Std = ± 0.01	Avg = 0.50 Std = ± 0.02	Avg = 0.51 Std = ± 0.05
Gn 1.1	5	Avg = 0.47 Std = ± 0.08	Avg = 0.47 Std = ± 0.08	Avg = 0.48 Std = ± 0.03	Avg = 0.51 Std = ± 0.07
Jn 3.16-17	0.5	Avg = 0.58 Std = ± 0.01	Avg = 0.60 Std = ± 0.04	Avg = 0.62 Std = ± 0.05	Avg = 0.63 Std = ± 0.04
Jn 3.16-17	1	Avg = 0.61 Std = ± 0.05	Avg = 0.63 Std = ± 0.07	Avg = 0.63 Std = ± 0.05	Avg = 0.63 Std = ± 0.04
Jn 3.16-17	2	Avg = 0.61 Std = ± 0.01	Avg = 0.61 Std = ± 0.01	Avg = 0.63 Std = ± 0.01	Avg = 0.63 Std = ± 0.04
Jn 3.16-17	5	Avg = 0.52 Std = ± 0.01	Avg = 0.54 Std = ± 0.57	Avg = 0.58 Std = ± 0.01	Avg = 0.63 Std = ± 0.05

Continues on the next page

Table 9 – *Continuation*

		Beta			
Mk 9.1	0.5	Avg = 0.67 Std = ± 0.01	Avg = 0.67 Std = ± 0.01	Avg = 0.69 Std = ± 0.08	Avg = 0.71 Std = ± 0.0
Mk 9.1	1	Avg = 0.69 Std = ± 0.01	Avg = 0.70 Std = ± 0.05	Avg = 0.70 Std = ± 0.05	Avg = 0.70 Std = ± 0.04
Mk 9.1	2	Avg = 0.66 Std = ± 0.01	Avg = 0.67 Std = ± 0.02	Avg = 0.68 Std = ± 0.01	Avg = 0.71 Std = ± 0.0
Mk 9.1	5	Avg = 0.61 Std = ± 0.02	Avg = 0.65 Std = ± 0.01	Avg = 0.65 Std = ± 0.01	Avg = 0.69 Std = ± 0.08
Phrase 1	0.5	Avg = 0.88 Std = ± 0.01	Avg = 0.89 Std = ± 0.01	Avg = 0.90 Std = ± 0.05	Avg = 0.92 Std = ± 0.0
Phrase 1	1	Avg = 0.90 Std = ± 0.01	Avg = 0.92 Std = ± 0.01	Avg = 0.92 Std = ± 0.01	Avg = 0.92 Std = ± 0.04
Phrase 1	2	Avg = 0.85 Std = ± 0.08	Avg = 0.86 Std = ± 0.01	Avg = 0.90 Std = ± 0.01	Avg = 0.92 Std = ± 0.05
Phrase 1	5	Avg = 0.83 Std = ± 0.02	Avg = 0.82 Std = ± 0.02	Avg = 0.85 Std = ± 0.01	Avg = 0.90 Std = ± 0.01
Phrase 2	0.5	Avg = 0.76 Std = ± 0.01	Avg = 0.78 Std = ± 0.01	Avg = 0.80 Std = ± 0.01	Avg = 0.81 Std = ± 0.0

Continues on the next page

Table 9 – *Continuation*

		Beta			
Phrase 2	1	Avg = 0.79 Std = ± 0.08	Avg = 0.80 Std = ± 0.01	Avg = 0.80 Std = ± 0.01	Avg = 0.81 Std = ± 0.0
Phrase 2	2	Avg = 0.75 Std = ± 0.01	Avg = 0.77 Std = ± 0.02	Avg = 0.80 Std = ± 0.05	Avg = 0.80 Std = ± 0.08
Phrase 2	5	Avg = 0.73 Std = ± 0.01	Avg = 0.73 Std = ± 0.03	Avg = 0.76 Std = ± 0.01	Avg = 0.79 Std = ± 0.01
Phrase 3	0.5	Avg = 0.80 Std = ± 0.08	Avg = 0.83 Std = ± 0.01	Avg = 0.86 Std = ± 0.07	Avg = 0.86 Std = ± 0.04
Phrase 3	1	Avg = 0.84 Std = ± 0.08	Avg = 0.85 Std = ± 0.08	Avg = 0.86 Std = ± 0.08	Avg = 0.87 Std = ± 0.0
Phrase 3	2	Avg = 0.81 Std = ± 0.03	Avg = 0.80 Std = ± 0.02	Avg = 0.84 Std = ± 0.01	Avg = 0.86 Std = ± 0.04
Phrase 3	5	Avg = 0.74 Std = ± 0.03	Avg = 0.75 Std = ± 0.03	Avg = 0.80 Std = ± 0.02	Avg = 0.85 Std = ± 0.08

End of table

This thesis aimed to statistically analyze the performance of the SwarmaBle algorithm, using graphs to illustrate the algorithm's evolution over iterations, as well as to highlight the relationship between the parameters used and the variables.

Plot 26 shows the evolution of SwarmaBle over iterations. The curves plotted in graph 26 indicate that SwarmaBle progresses evolutionarily towards convergence of the objective function, indicating that the best solution is being improved. It is worth noting that the curve of the average of the solutions in each iteration deviates by about 15% from the curve of the best solution or fitness. The SwarmaBle algorithm seems to reach convergence around the 1,000th iteration.

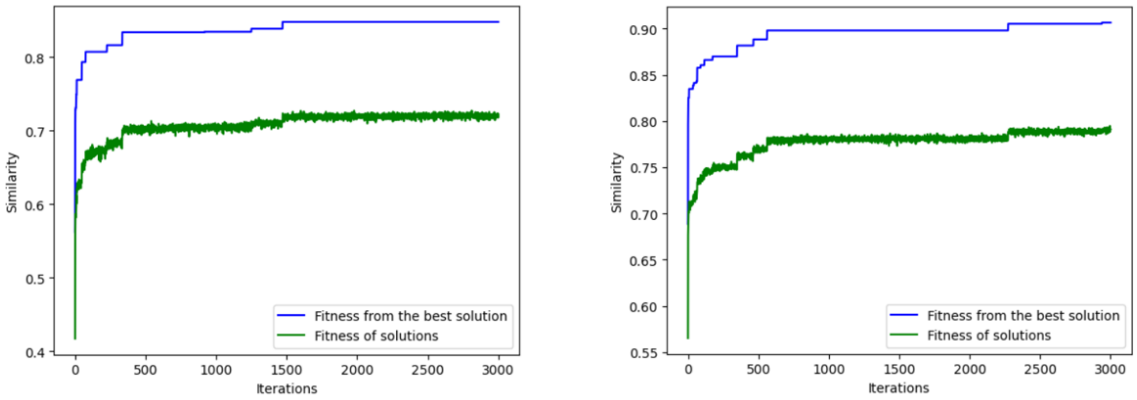

Figure 26: SwarmaBle Iteration Curves

The temporal plot 27 presents the performance behavior of the SwarmaBle algorithm concerning variations in the α and β parameters. It can be observed that, over time, as α increases, the overall performance of SwarmaBle decreases, especially for α rates above 2, as illustrated by the temporal plot 27. Another noteworthy observation is that, for the search sentence Jn 3.16-17, an outlier is observed in the standard deviation behavior.

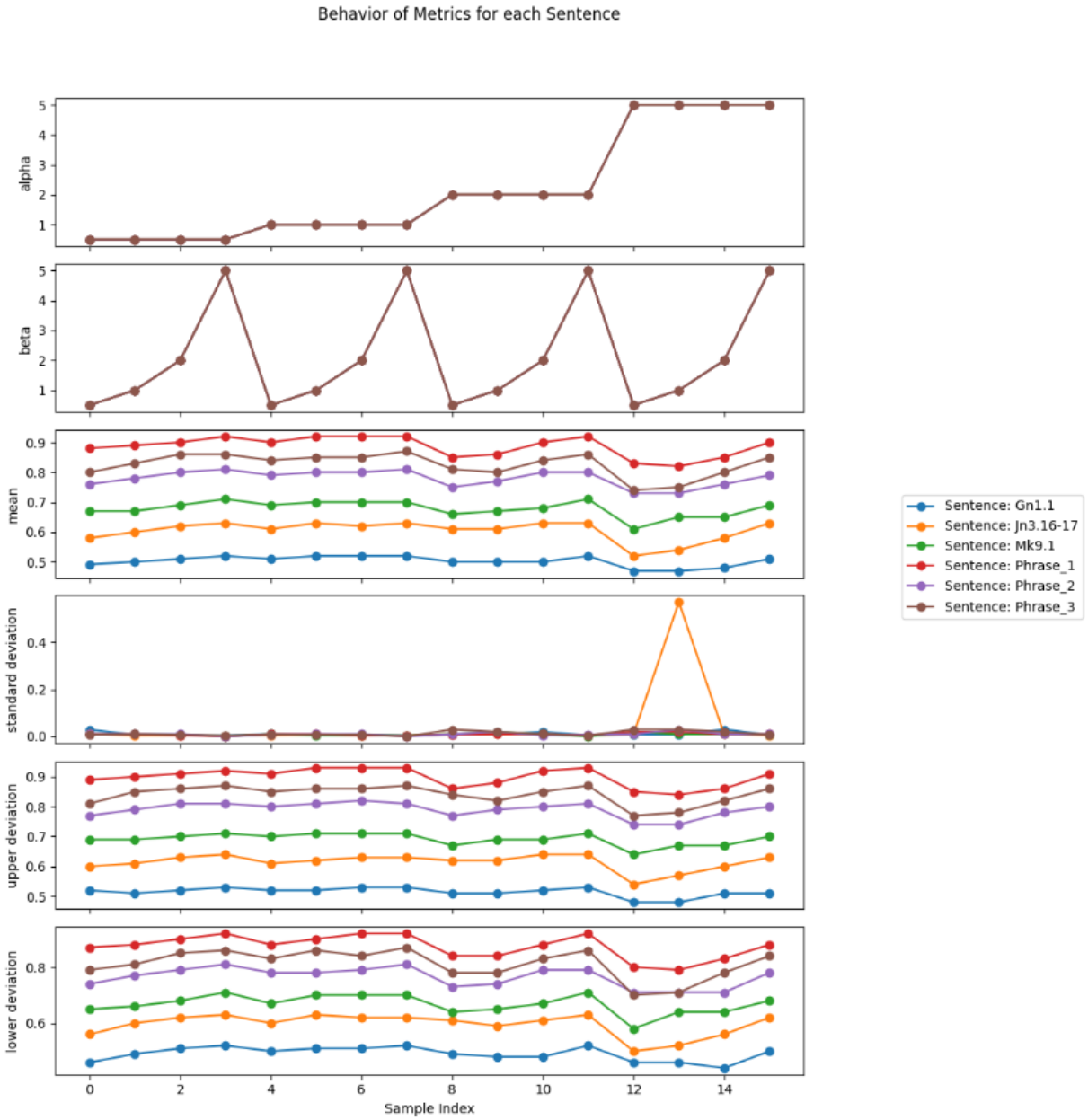

Figure 27: Iteration Curves SwarmaBle

The plots 28 display the individual means of each variable (mean, standard deviation, plus deviation, and minus deviation). The box plots highlight that the mean and deviations ± remain stable, around 70%. Therefore, it is possible to conclude that, for the entered search sentences and the volume of executions performed, the SwarmaBle algorithm has an overall performance above 70%, emphasizing that this performance can be higher depending on the search sentence.

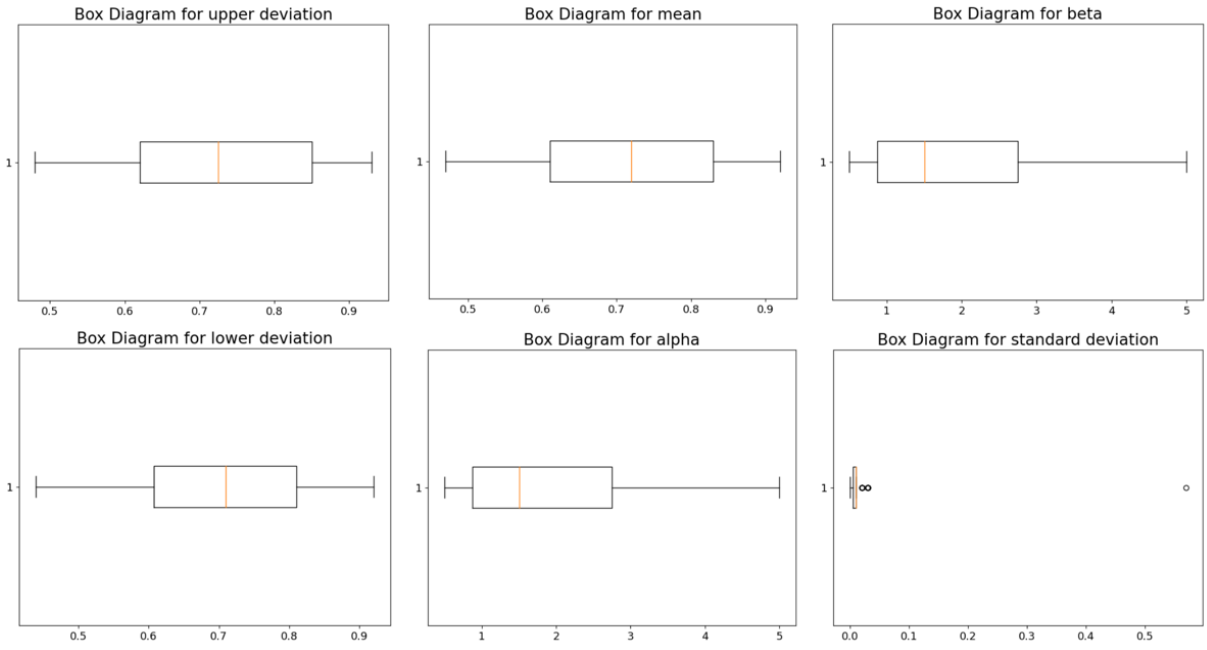

Figure 28: Box plot of variables

An analysis of the behavior of means concerning some parameters was conducted. In the plot 29, it is presented how the parameter α influences the mean of the executions. In the plot 30, the average behavior of the algorithm is demonstrated as a function of β. The behavior of the box plots on the influence of α on the overall mean indicates a significant increase with α rates around 0.5 and 1 in the algorithm's performance; however, when the α rate reaches values above 2, the performance decreases. An important point to consider is that even with this reduction in the algorithm's performance at higher α rates, the algorithm's performance remains above 70%. In contrast to the influence of β on the overall mean, a considerable improvement in the algorithm's performance can be observed when increasing the values of β. β values around 5 bring the algorithm closer to achieving a performance close to 80%.

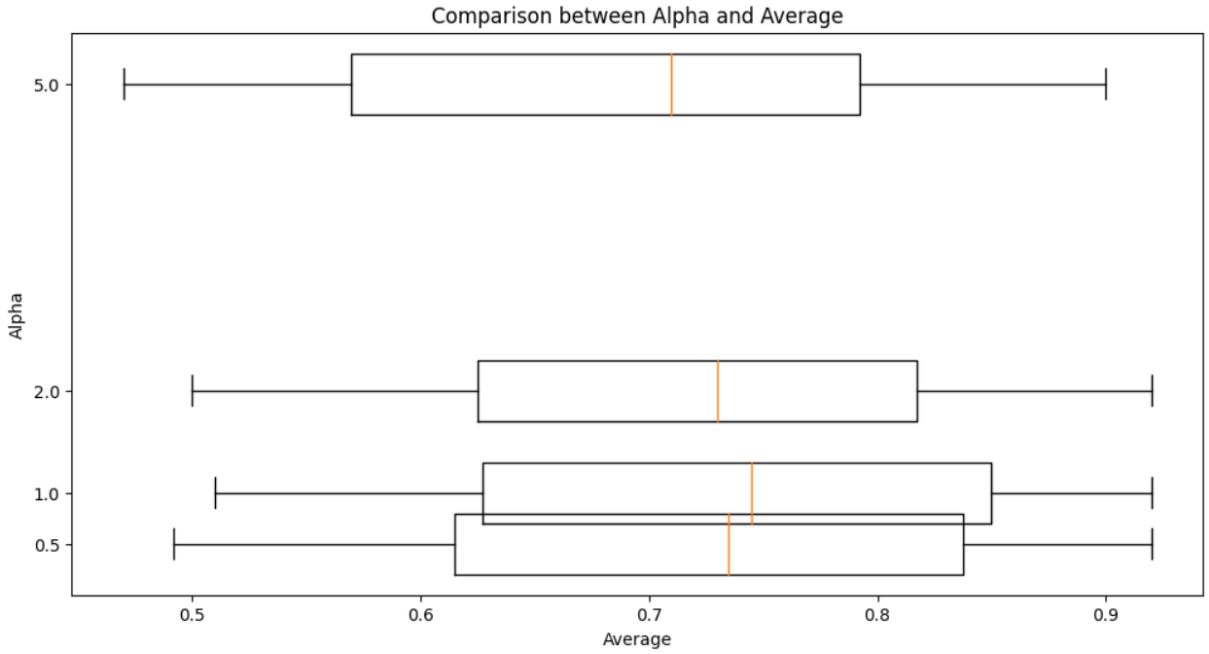

Figure 29: Box plot of mean as a function of α

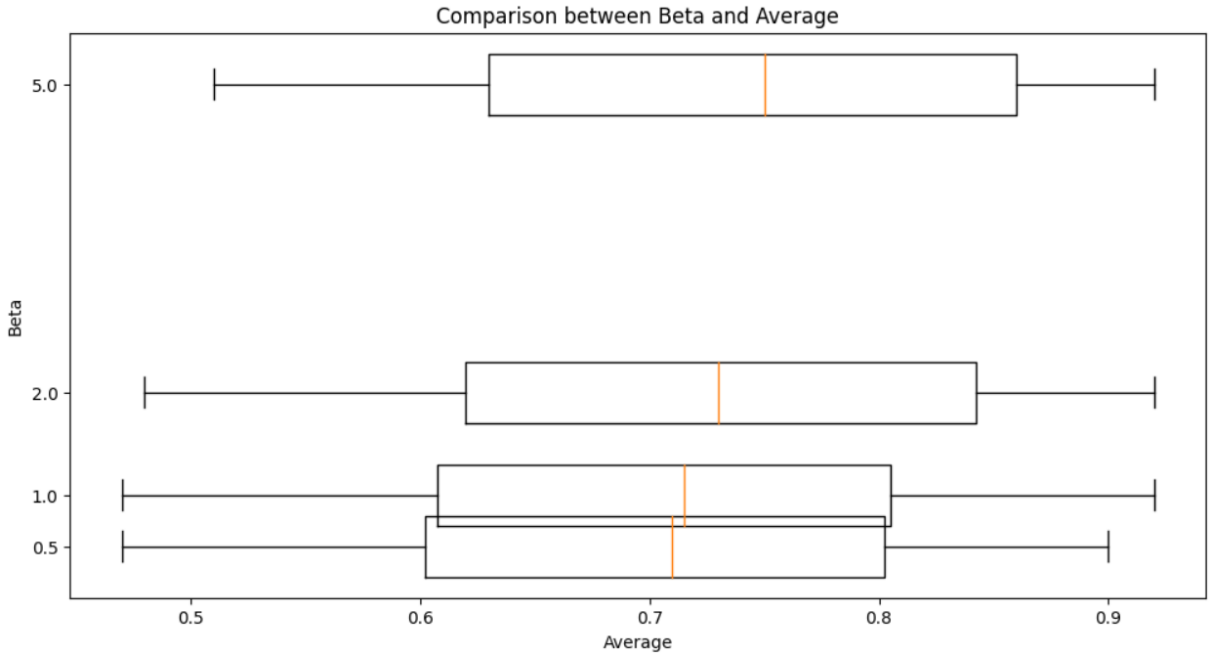

Figure 30: Box plot of mean as a function of β

The scatter plots in figure 31 display the dispersion of results values in relation to α and the mean, while the plots in figure 32 illustrate the dispersion of results in relation to

β and the mean. The scatter plots in 31 and 32 support the findings from the box plots, showing this performance reduction in α compared to β, especially when α reaches values of 5.

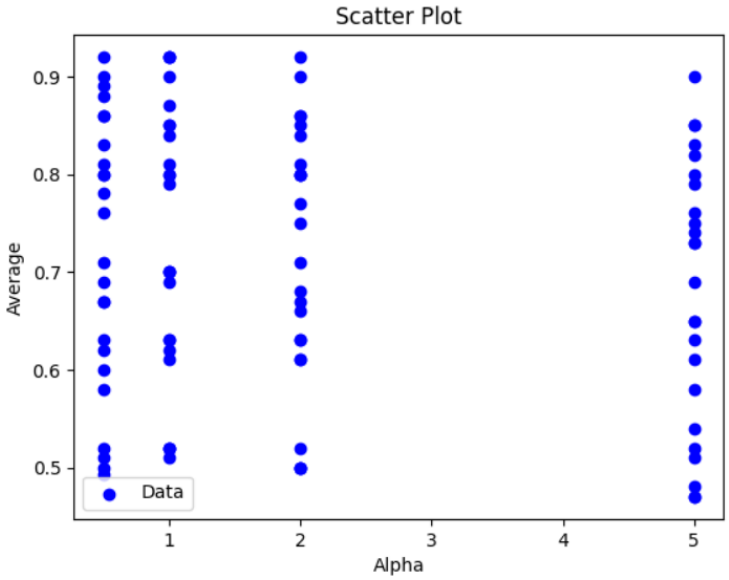

Figure 31: Scatter plot of mean as a function of α.

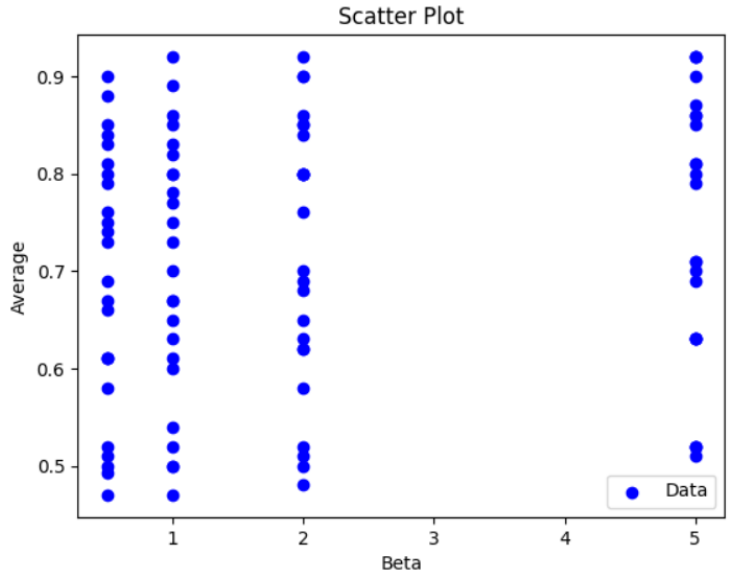

Figure 32: Scatter plot of mean as a function of β.

The scatter plots in 33 show the dispersion of values in relation to the association of themes with the mean achieved for each theme. In this specific graph, it is evident that the constructed search sentences have higher efficiency than the search sentences

with biblical text, i.e., the verses. Sentence 1 stands out among all search sentences, reaching an average above 90% fitness. The reason for this behavior remains unclear, as the verses should perform better than the sentences. However, it's possible to consider as a hypothesis that sentences can more effectively construct the sermon's theme than verses, as often only one introductory verse may not be sufficient to encompass the central idea of a theme.

The bubble charts show the influence of the parameters α and β in comparison to the search sentences or themes in 34, 35. The bubble charts align with the evidence from the other plotted graphs, indicating that alpha has a significant influence on performance at lower values for all sentences, while beta has a more significant influence at higher values.

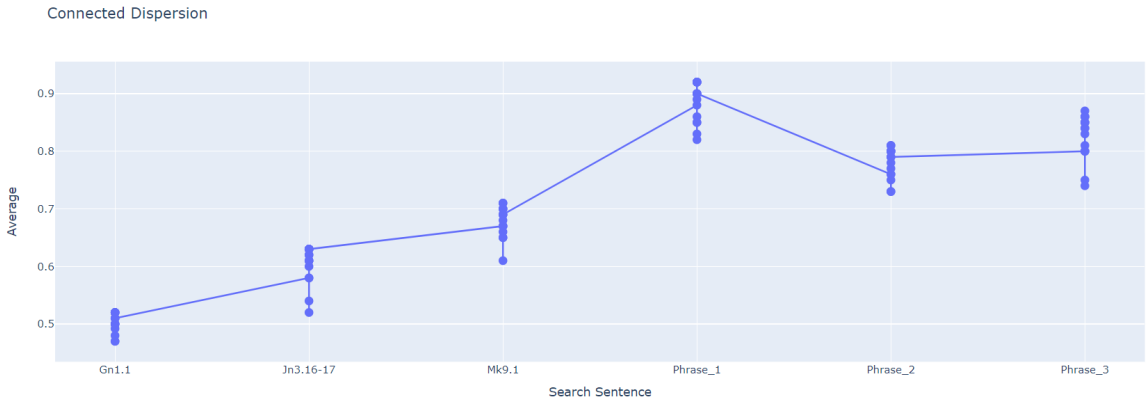

Figure 33: Scattering based on search sentences

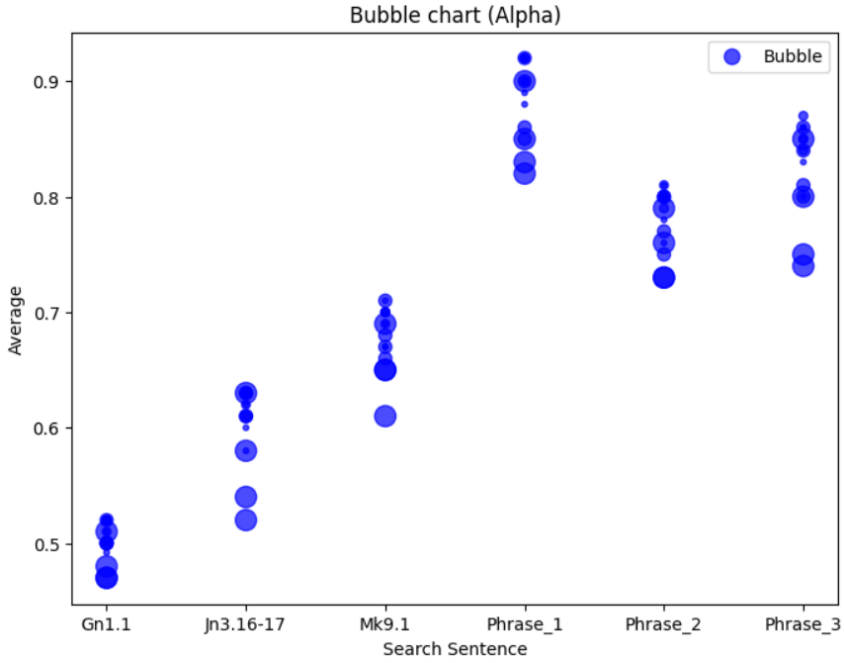

Figure 34: Bubble chart α

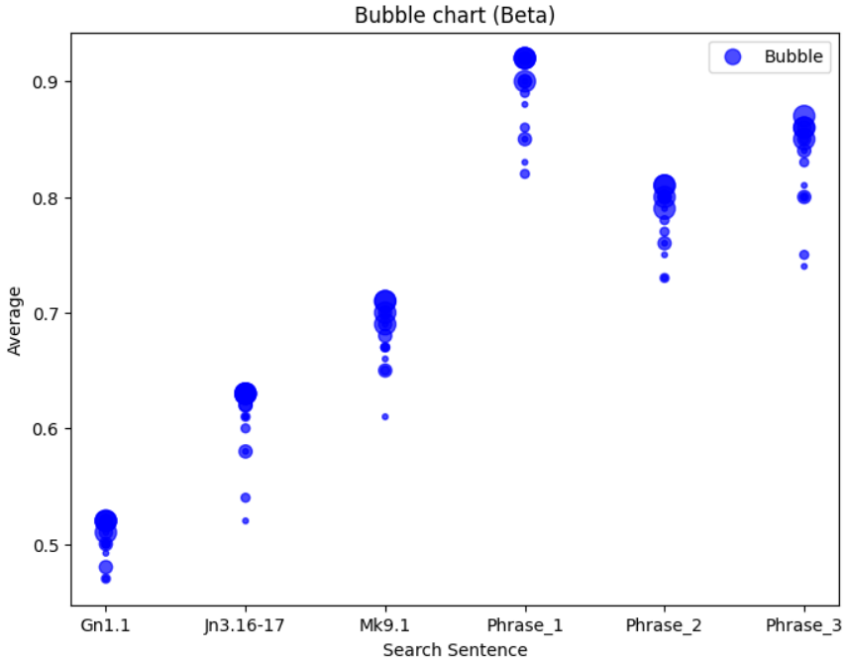

Figure 35: Bubble chart β

5.2.2 Results of the experimental phase of SwarmaBle (contextual)

This thesis was concerned with subjecting the results to an evaluation and validation process conducted by domain experts. After all, it is these professionals who will use the application, making it crucial to obtain a comprehensive view of the solution's performance from their perspective.

It is worth noting that the evaluators did not have conflicts of interest with the thesis researcher. The criteria for composing the panel of evaluators included the requirement of pastoral experience and theological training, ensuring that all evaluators had an academic foundation to some degree.

The panel of evaluators consists of three professionals. The author of this thesis expressed the desire to obtain a broader sample to ensure greater consistency; however, due to the significant time commitment required from the evaluators for the analysis of biblical texts, only this group of three evaluators was gathered. The data generated by SwarmaBle and submitted to the evaluation panel are presented in Table 10.

The adopted method for the evaluation process involved conducting a battery of six SwarmaBle executions. In other words, each execution provided a set of ten biblical passages for each input sentence or sermon theme. Eight questions were formulated for the evaluators to assign scores, using the Likert scoring scale ranging from 0 to 5, where 0 represents the worst score and 5 the best score. See table 11

The subsequent table displays the formulated questions. The responses were obtained through the completion of an online form.

The statistical results of the evaluation are presented in the graphs 38, 37 and 36.

Table 10: SwarmaBle Experimentation - Contextual

Search Sentence	Sermon Verses	α e β	Fitness of the best solution
Gn 1.1	1 Ch 16.14; Jn 1.2; Gn 1.1 Ps 68.8; Hb 11.16; 1 Kgs 8.27; Is 37.16; Act 17.24; 2 Kgs 19.15; Rev 14.19	$\alpha = 05$ e $\beta = 5$	53%
How to achieve happiness and faith in the midst of suffering? (Phrase 1)	Pv 9.10; Gal 5.22; 1 Tim 2.15; Phil 2.1; Eph 4.19; 1 Thess 1.3; 1 Pe 5.9; Col 1.11; Jn 3.15; Rom 2.7;	$\alpha = 5$ e $\beta = 5$	86%
Mar 9.1	1 Kgs 22.8; Matt 16.28; 2 Kgs 3.14; Num 9.8; 1 Sam 16.5; Ezek 13.6; Exod 3.7; Mar 9.1; Lk 9.27; Is 35.4;	$\alpha = 05$ e $\beta = 2$	69%
How should we pray to God? (Phrase 2)	Jon 2.1; 1 Cor 3.17; 2 Cor 6.16; Ps 43.4; 2 Kgs 19.15; Jer 23.23; Act 7.32; Mal 1.9 1 Jn 4.15; Eph 6.23;	$\alpha = 5$ e $\beta = 1$	84%
Jn 3.16-17	Mar 10.45; Ezr 10.2; Jn 3.16; Gal 1.4; Matt 27.54; 1 Jn 4.9; Hb 11.7; Rom 5.10; Jas 1.27; Tit 1.2;	$\alpha = 5$ e $\beta = 5$	63%
What do you need to do to live a life of sanctification and repentance? (Phrase 3)	Matt 10.39; Dt 30.19; 1 Tim 4.8; Rom 2.7; Lk 18.30; Hb 7.25; Jn 12.25; 2 Thess 1.11; 2 Tim 3.7 1 Cor 3.22	$\alpha = 5$ e $\beta = 5$	80%

Table 11: Validation Questions

Numbers of the questions	Questions
Question 1	What was your opinion about the selected passages for the theme? Did they meet your expectations?
Question 2	Do you believe that the selected Bible passages chosen for composing the sermon exhibit any diversity?
Question 3	Do you think the Bible passages chosen by the algorithm have any immediate context with the proposed theme?
Question 4	Do you believe the Bible passages chosen by the algorithm have any remote context with the proposed theme?
Question 5	Do the chosen Bible passages have any theological context related to the theme or the entry sentence, in your opinion?
Question 6	Is it possible to create a sermon (topic) using the selected passages so that the preacher can develop a theme based on them?
Question 7	Would you use the SwarmaBle algorithm to find passages for your sermons?
Question 8	Evaluate the overall performance of the SwarmaBle algorithm for the combination of Bible passages given a search sentence (theme)

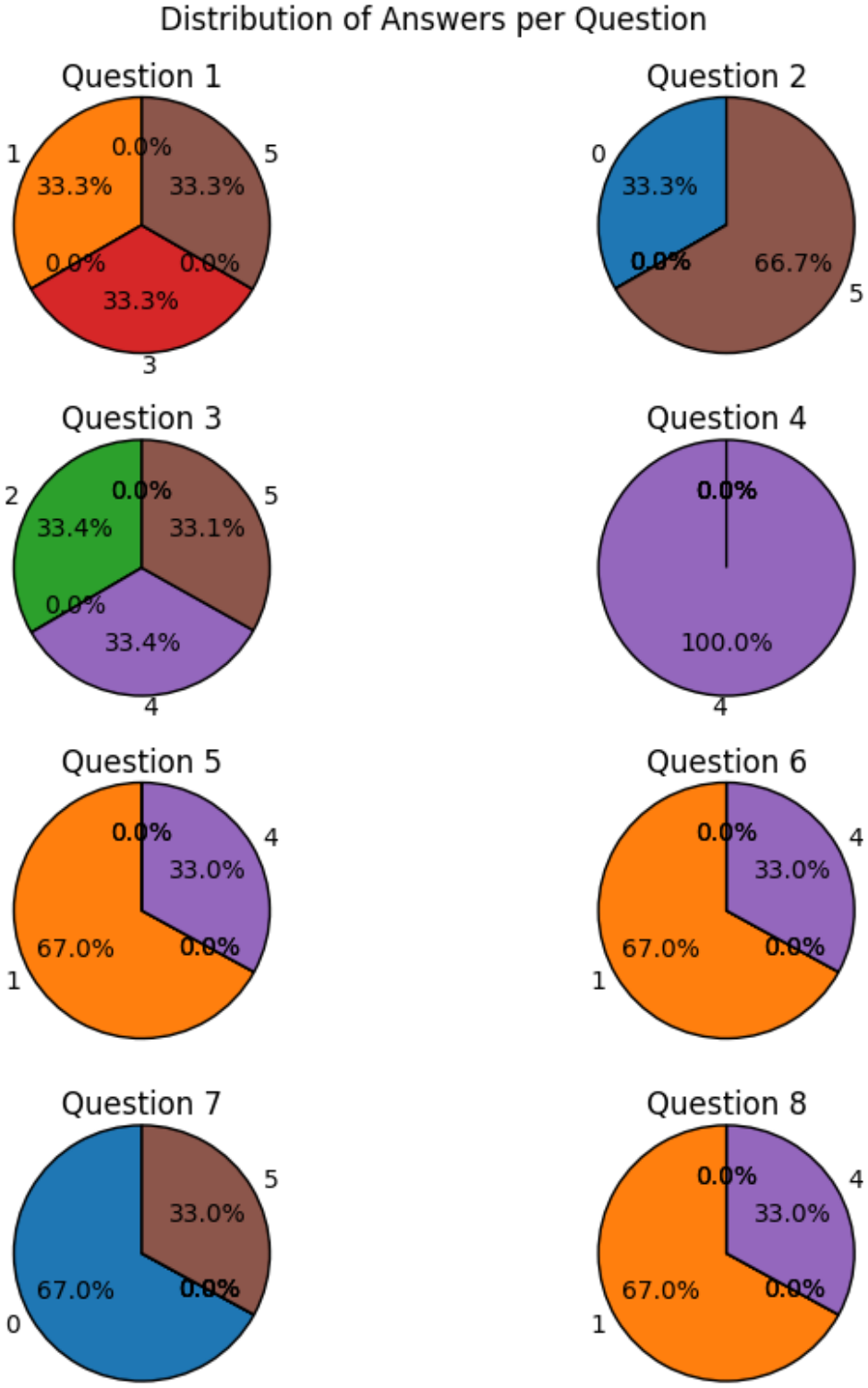

Figure 36: Pie chart - Validation

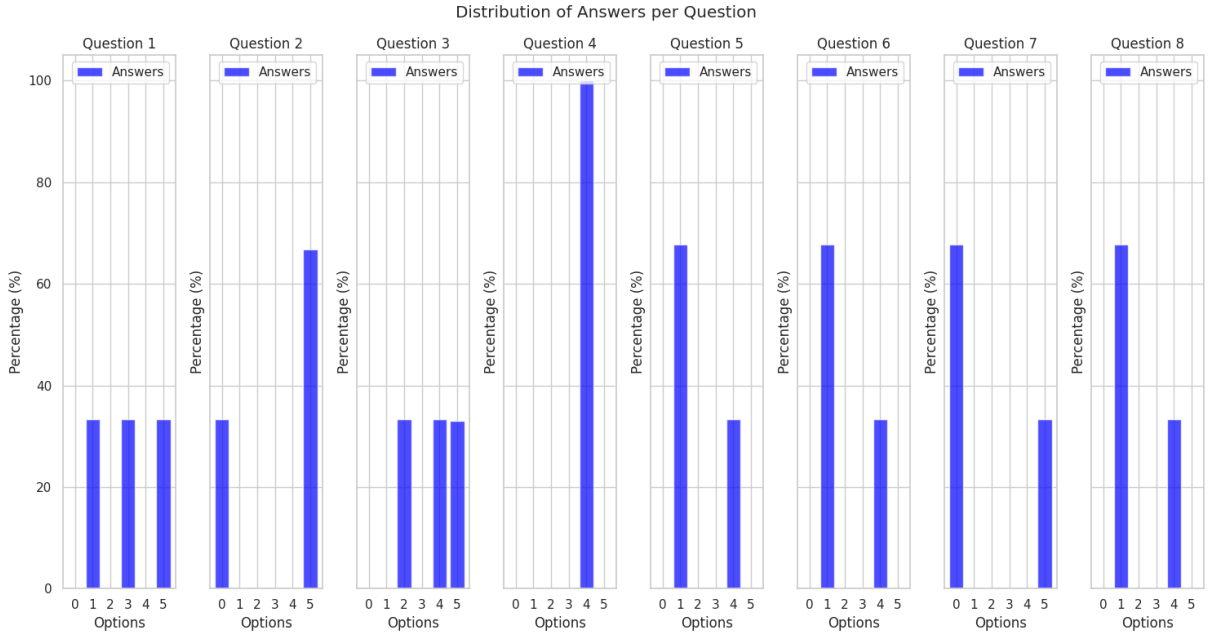

Figure 37: Bar chart - Validation

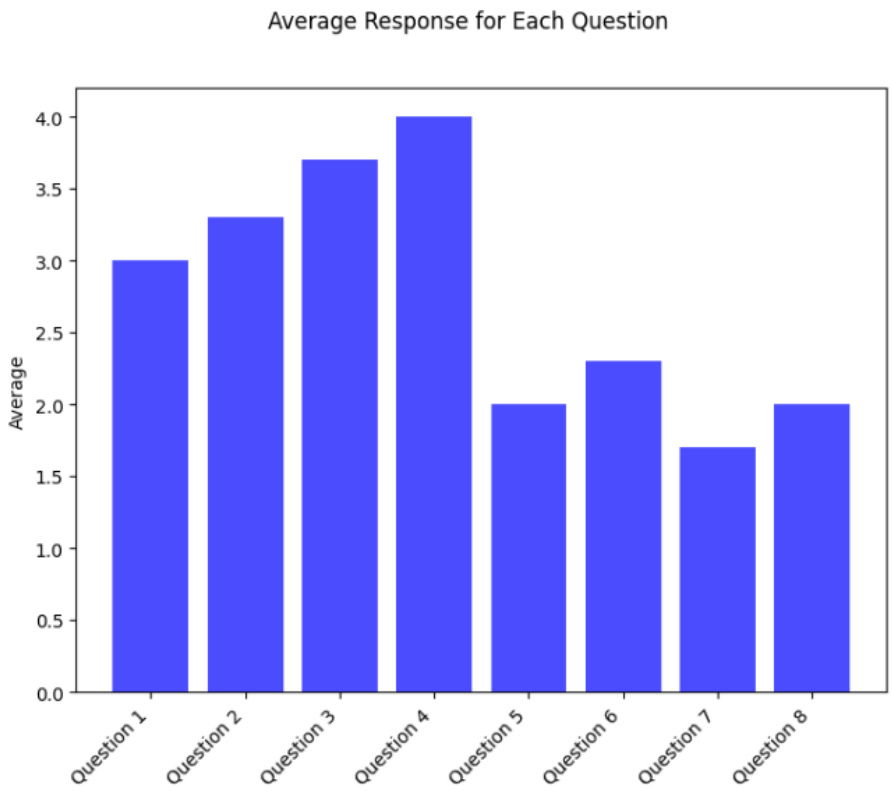

Figure 38: Bar chart - Validation

In general terms, it is possible to observe that the performance of SwarmaBle was within or above average, according to the evaluators. In other words, in the first four questions, SwarmaBle shows performance above average, while in the last four questions, it remains in the average performance range, falling just below average in question 7. Refer to Figure 38.

It is worth noting that the first four questions are quite specific inquiries about SwarmaBle's performance, i.e., about the contextual capability of SwarmaBle to generate passages for sermons. According to the evaluators, SwarmaBle demonstrated satisfactory performance in these tasks.

A significant indicator is question 1, which queries whether the passages returned by the SwarmaBle algorithm met the evaluators' expectations. In other words, it is expected that passages consistent with such sermon topics be chosen. The result of the domain experts' evaluation is that SwarmaBle achieves above-average performance.

Questions 6, 7, and 8 are not directly related to the algorithm's performance but rather to the evaluators' willingness to use technology to aid in sermon preparation. Therefore, these questions will reveal the enthusiasm of the theological community in adopting such technologies. It is important to consider that innovative technologies may encounter immediate resistance to their adoption and assimilation in the community. Additionally, there is a taboo when it comes to the use of techniques involving artificial intelligence. Thus, the results indicate resistance to the use of automation by theologians.

Based on these validation data, it can be stated that SwarmaBle's best performance lies in retrieving passages that convey some degree of contextualization (immediate and remote) with the theme. This highlights the richness of diversity in optimal solutions and the semantic robustness of natural language processing.

6 Conclusions and Future Work

Combinatorial problems have always posed a challenge for the scientific community, especially when seeking optimal solutions within short processing times. Bioinspired metaheuristics have achieved remarkable performance in solving problems of this nature.

In this thesis, the author dedicated themselves to studying the feasibility of applying a bioinspired metaheuristic, ACO (Ant Colony Optimization), for the optimization of Bible passages intended for sermon construction, through the construction of a weighted biblical graph.

Despite the unique characteristics of the biblical corpus, which make it a text set that is challenging to manipulate and interpret, the performance of ACO has proven to be solid and robust. Some highlights of the results are presented below:

- The consistency in the behavior of ACO throughout the experimental battery is noteworthy, providing greater confidence in the statements about the results. This is not an isolated behavior of the algorithm but rather a global trend.
- The algorithm exhibits relatively fast convergence; that is, in approximately 1000 iterations, signs of convergence to a local optimal solution can already be observed.
- In the statistical analyses, the average fitness function consistently remains above 70%, corroborating the fact that the algorithm consistently produces combinations with a high semantic value of textual similarity.
- The hyperparameter that had the most positive influence on the improvement of the fitness function was the heuristic information (beta).
- This statistical performance was confirmed during validation, as in questions more related to the performance of SwarmaBle, the scores remained above average.
- In the unanimous opinion of domain experts, SwarmaBle demonstrates above-average performance in the remote context, indicating that the Bible passages generated by SwarmaBle exhibit contextual diversity with respect to the theme or input sentence.
- It is important to note that the validation process highlighted significant resistance from domain experts in the theological field regarding the adoption of technologies. Thus, it becomes imperative to promote information and awareness campaigns about the benefits that technologies can provide, aiming to reduce this taboo.

Thus, the application of bioinspired optimization methods in the optimized textual information retrieval appears plausible in the context of the biblical corpus. However, the

results of SwarmaBle reveal a new research opportunity, as similar textual corpora may exhibit similar performances.

6.1 Future work

For future work, the author plans to explore the feasibility of other natural computing techniques in optimizing the biblical corpus, such as evolutionary algorithms, clonal selection algorithms, among others.

Additionally, there is an intention to investigate other linguistic phenomena beyond semantics, such as syntax, lexocographics, among others.

Performing individualized NLP treatment for each literary genre to assess whether there will be any impact on SwarmaBle performance.

Implementing multiple stopping criteria to investigate the convergence nature of the algorithm.

Promoting the investigation into why sentences perform better in SwarmaBle than the biblical texts themselves.

References

AGGARWAL, C.; ZHAI, C. *Mining Text Data*. New York: 1st Edition. Publishing company: Springer, 2012.

AMORE, R. C. *Religion and Politics: New Developments Worldwide*. Basel, Switzerland: MDPI, 2019.

ARI, V. et al. Contents matching defined by prototypes: Methodology verification with books of the bible. *Journal of Management Information Systems*, v. 18, p. 87–100, 2014.

ASHENGO, Y.; AGA, R.; ABEBE, S. Context based machine translation with recurrent neural network for english–amharic translation. *Machine Translation*, v. 35, p. 19–36, 2021.

ATEN, J. et al. The psychological study of religion and spirituality in a disaster context: A systematic review. *Psychological Trauma*, v. 11, p. 597–613, 2019.

AZAWI, M.; AFZAL, M.; BREUEL, T. Normalizing historical orthography for ocr historical documents using lstm. In: *ACM International Conference Proceeding Series*. New York: ACM, 2013. p. 80–85.

BALLARD, D. H. *An Introduction to Natural Computation*. Massachusetts: MIT Press, 1997.

BARTON, J. *Biblical Intepretation*. England: 1st Edition. Publishing company: Cambridge University Press, 1998.

BENTHO, E. *Hermenêutica Fácil e Descomplicada* . São Paulo: 1st Edition. Publishing company: Casa Publicadora das Assembleias de Deus, 2003.

BERRY, M.; KOGAN, J. *Text Mining Applications and Theory*. New Jersey: 1st Edition. Publishing company: John Wiley Sons, 2010.

BEYERS, J. Religion and culture: Revisiting a close relative. *HTS Theological Studies*, scieloza, v. 73, p. 1 – 9, 00 2017. ISSN 0259-9422.

BILOVICH, A.; BRYSON, J. Detecting the evolution of semantics and individual beliefs through statistical analysis of language use. In: *AAAI Fall Symposium - Technical Report*. Virginia: AAAI, 2008. p. 21–26.

BLEIWEISS, A. A hierarchical book representation of word embeddings for effective semantic clustering and search. In: *ICAART 2017 - Proceedings of the 9th International Conference on Agents and Artificial Intelligence*. Portugal: ICAART, 2017. v. 2, p. 154–163.

BONABEAU, E.; DORIGO, M.; THERAULAZ, G. *SWARM INTELLIGENCE From Natural to Artificial Systems*. England: Oxford University Press, 1999.

BRIA, A. et al. Deep transfer learning for writer identification in medieval books. In: *2018 Metrology for Archaeology and Cultural Heritage (MetroArchaeo)*. Italy: MetroArchaeo, 2018. p. 455–460.

BROADUS, J. A. *On the preparation and delivery of sermons*. Washington: Western Reformed Seminary, 2003.

CERNANSKY, M. et al. Text correction using approaches based on markovian architectural bias. In: *CEUR Workshop Proceedings*. Italy: CEUR, 2007. v. 284.

CHAPMAN, S.; SWEENEY, M. *The Hebrew Bible/Old testament*. England: 1st Edition. Publishing company: Cambridge University Press, 2016.

CILIA, N. et al. An end-to-end deep learning system for medieval writer identification. *Pattern Recognition Letters*, v. 129, p. 137–143, 2020.

CILIA, N. et al. An experimental comparison between deep learning and classical machine learning approaches for writer identification in medieval documents. *Journal of Imaging*, v. 6, p. 89, 2020.

CLARK, A.; FOX, C.; LAPPIN, S. *The Handbook of Computational Linguistics and Natural Language Processing*. England: 1st Edition. Publishing company: John Wiley Sons, 2010.

COECKELBERGS, M.; HOOLAND, S. Modeling the hebrew bible: Potential of topic modeling techniques for semantic annotation and historical analysis. *CEUR Workshop Proceedings*, v. 1595, p. 47–52, 2016.

COMFORT, P. *A Origem e Autenticidade da Bíblia*. São Paulo: 1st Edition. Publishing company: Casa Publicadora das Assembleias de Deus, 1998.

COVINGTON, M.; POTTER, I.; SNODGRASS, T. Stylometric classification of different translations of the same text into the same language. *Digital Scholarship in the Humanities*, v. 30, p. 322–325, 2015.

DE CASTRO, L. N. *Fundamentals of Natural Computing Basic Concepts, Algorithms, and Applications*. Florida: CRC Press Taylor & Francis Group, 2007.

DELL, K.; JOYCE, P. *Biblical Interpretation and Method*. England: 1st Edition. Publishing company: Oxford University Press, 2013.

DIONE, C.; KUHN, J.; ZARRIESS, S. Design and development of part-of-speech-tagging resources for wolof (niger-congo, spoken in senegal). In: *Proceedings of the 7th*

International Conference on Language Resources and Evaluation. LREC 2010. Malta: LREC, 2010. p. 2806–2813.

DORIGO, M. Positive feedback as a search strategy. *Tech. Rep. 91-016*, 1991.

DORIGO, M.; STÜTZLE, T. *Ant colony optimization: overview and recent advances*. New York: Springer, 2019.

DORIGO, M.; STüTZLE, T. *Ant Colony Optimization*. Massachusetts: The MIT Press, 2004.

EDER, M. Rolling stylometry. *Digital Scholarship in the Humanities*, v. 31, p. 457–469, 2016.

EIJNATTEN, J. van. *Preaching, Sermon and Cultural Change in the Long Eighteenth Century*. Leiden, The Netherlands: Brill, 2008. ISBN 978-90-47-42487-1.

ESAN, A. et al. Development of a recurrent neural network model for english to yoruba machine translation. *International Journal of Advanced Computer Science and Applications*, v. 11, p. 602–609, 2020.

FORSTER, M. N. *The Cambridge Companion to HERMENEUTICS*. England: Cambridge University Press, 2019.

FRANCIS, M.; NAIR, K. Hybrid part of speech tagger for malayalam. In: *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics. ICACCI 2014*. Delhi: ICACCI, 2014. p. 1744–1750.

GAWTHROP, R.; STRAUSS, G. Protestantism and literacy in early modern germany. *Oxford University Press on behalf of The Past and Present Society*, p. 31–55, 1984.

GESSNER, A.; KÖTTERITZSCH, C.; LAUER, G. Biblical intertextuality in a digital world: The tool gertrude. In: *ACM International Conference Proceeding Series*. UK: ACM, 2013. p. 6:1–5.

GOLOVIN, S. et al. Algorithmic handwriting analysis of judah's military correspondence sheds light on composition of biblical texts. *Proceedings of the National Academy of Sciences of the United States of America*, v. 113, p. 4664–4669, 2016.

GUINNESS world records. 2022. Accessed 02 May 2022. Https://www.guinnessworldrecords.com/world-records/best-selling-book-of-non-fiction.

HASSANIEN, A. E.; EMARY, E. *SWARM INTELLIGENCE Principles, Advances, and Applications*. Florida: CRC Press Taylor Francis Group, 2016.

HERBRICH, R.; GRAEPEL, T. *Handbook of Natural Language Processing* . England: 2st Edition. Publishing company: Taylor Francis Group, 2010.

HOFMANN, T. Probabilistic Latent Semantic Indexing. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

HU, W. Unsupervised learning of two bible books: Proverbs and psalms. *Sociology Mind*, v. 2, p. 325–334, 2012.

JAENISCH, H. et al. Graphics-based intelligent search and abstracting using data modeling. In: *Society of Photo-Optical Instrumentation Engineers (SPIE)*. Washington: SPIE, 2002. v. 4788, p. 135–146.

KENNEDY, J.; EBERHART, R. C. *Swarm Intelligence*. Massachusetts: Morgan Kaufmann Publishers, 2016.

KERR, G. *Gramática Elementar da Língua Hebraica*. Philadelphia: 1st Edition. Publishing company: United states of America Press of the Jewish Publication Society, 1948.

KITCHENHAM, B. Procedures for performing systematic reviews. *Keele. UK. Keele University*, v. 33, p. 1–26, 2004.

KLEIN, W. W.; BLOMBERG, C. L.; HUBBARD, R. L. *Introduction to Biblical Interpretation*. Tennessee: Thomas Nelson, 2004.

LOWRY, E. L. *The Homiletical Plot: The Sermon as Narrative Art Form*. Kentucky: Westminster John Knox Press, 2001.

LUTZ, D. The relative influence of european writers on late eighteenth-century american political thought. *The American Political Science Review*, v. 78, p. 189–197, 1984.

MAOZ, Z.; HENDERSON, E. A. The world religion dataset, 1945–2010: Logic, estimates, and trends. *International Interactions*, Routledge, v. 39, n. 3, p. 265–291, 2013.

MITKOV, R. *The Oxford Handbook of Computational Linguistics*. England: 1st Edition. Publishing company: Oxford Press, 2003.

MOHER, D. et al. Preferred reporting items for systematic review and meta-analysis protocols (prisma-p) 2015 statement. *Systematic reviews*, v. 4, p. 1–9, 2015.

MURAI, H. Exegetical science for the interpretation of the bible: Algorithms and software for quantitative analysis of christian documents. *Studies in Computational Intelligence*, v. 492, p. 67–86, 2013.

OSTFELD, A. *Ant colony Optimization - Methods and Applications*. London: intechopen Publisher, 2011.

ÖSTLING, R.; TIEDEMANN, J. Continuous multilinguality with language vectors. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valence: ACL, 2017. v. 2, p. 644–649.

PANIGRAHI, B. K.; SHI, Y.; LIM, M. *Handbook of Swarm Intelligence Concepts, Principles and Applications*. New York: Springer-Verlag Berlin Heidelberg, 2011.

PILEHVAR, M. T.; COLLADOS, J. C. *Embeddings in Natural Language Processing Theory and Advances in Vector Representations of Meaning*. Massachusetts: Morgan publishers, 2021.

POPA, R.; GOGA, N.; GOGA, M. Extracting knowledge from the bible: A comparison between the old and the new testament. p. 505–510, 2015.

RISTA, A.; KADRIU, A. Casr: A corpus for albanian speech recognition. In: *International Convention on Information, Communication and Electronic Technology (MIPRO)*. Opatija: MIPRO, 2021. p. 438–441.

ROGERSON, S.; LIEU, J. *BIBLICAL STUDIES*. England: 1st Edition. Publishing company: Oxford University Press, 2006.

SEN, H.; COLUCCI, L.; BROWNE, D. Keeping the Faith: Religion, Positive Coping, and Mental Health of Caregivers During COVID-19. *Frontiers in Psychology*, v. 12, p. 67–86, 2022.

SOLNON, C. *Ant Colony Optimization and Constraint Programming*. Jersey: John Wiley Sons, Inc., 2010.

SRIVASTAVA, A.; SAHAMI, M. *Text mining : classification, clustering, and applications*. England: 1st Edition. Publishing company: Taylor and Francis Group, 2009.

STEFANO, C. et al. Reliable writer identification in medieval manuscripts through page layout features: The "avila" bible case. *Engineering Applications of Artificial Intelligence*, v. 72, p. 99–110, 2018.

THOMAS, D.; VALENZUELA, R. Text mining analysis of the king james version new international version: Concerns and implications for esl readers. *Journal of Research on Christian Education*, v. 29, p. 259–271, 2020. Https://doi.org/10.1080/10656219.2020.1837696.

TSCHUGGNALL, M.; SPECHT, G. From plagiarism detection to bible analysis: The potential of machine learning for grammar-based text analysis. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. New York: Springer, 2016. v. 9853, p. 245–248.

VALDIVIA, M.; VEGA, M.; LÓPEZ, L. Lvq for text categorization using a multilingual linguistic resource. *Neurocomputing*, v. 55, p. 665–679, 2003.

VARGHESE, N.; PUNITHAVALLI, M. Lexical and semantic analysis of sacred texts using machine learning and natural language processing. *International Journal of Scientific and Technology Research*, v. 8, p. 3133–3140, 2019.

VINOTHENI, C.; LAKSHMANA, P. A state of art approaches on handwriting recognition models. *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, v. 1, p. 98–103, 2019.

VIRKLER, H. *Hermenêutica Avançada*. São Paulo: 1st Edition. Publishing company: Vida, 1980.

VISA, A. et al. Prototype matching - finding meaning in the books of the bible. In: *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*. Maui: Annual Hawaii International Conference on System Sciences, 2001.

WEISS, S. et al. *Text Mining Predictive Methods for Analyzing Unstructured Information*. New York: 1st Edition. Publishing company: Springer, 2005.

WIDDOWS, D.; COHEN, T. Semantic vector combinations and the synoptic gospels. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. New York: Springer, 2009. v. 5494, p. 251–265.

YU, Z. et al. If you even don't have a bit of bible: Learning delexicalized pos taggers. In: *Proceedings of the 10th International Conference on Language Resources and Evaluation. LREC*. Portoroz: LREC, 2016. p. 96–103.

ZHAO, H.; LIU, J. Finding answers from the word of god: Domain adaptation for neural networks in biblical question answering. In: *2018 Proceedings of the International Joint Conference on Neural Networks*. Rio de Janeiro: International Joint Conference on Neural Networks, 2018. p. 475:1–8.

ZIMMERMANN, J. *Hermeneutics: A very short introduction*. England: Oxford University Press, 2015.

ZIRAN, Z. et al. Text alignment in early printed books combining deep learning and dynamic programming. *Pattern Recognition Letters*, v. 133, p. 109–115, 2020.

ÓNí, Q.; ASAHIAH, F. Computational modelling of an optical character recognition system for yoruba printed text images. *Scientific African*, v. 9, p. e00415, 2020.