NOVA IMS Information Management School

MDSAA

Master Degree Program in

Data Science and Advanced Analytics

AUTOMATION AND OPTIMISATION OF SEO CONTENT GAP ANALYSIS TASK WITH SEMANTIC CLUSTERING OF PORTUGUESE SEARCH KEYWORDS

Inês Rosado Vargem Gomes Roque

Internship Report

presented as partial requirement for obtaining the Master Degree Program in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

AUTOMATION AND OPTIMISATION OF SEO CONTENT GAP ANALYSIS TASK WITH SEMANTIC CLUSTERING OF PORTUGUESE SEARCH KEYWORDS

by

Inês Rosado Vargem Gomes Roque

Internship report presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a Specialisation in Data Science

Supervisor: Roberto Henriques

November 2023

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledge the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Inês Roque

Lisbon, November 12th 2023

ABSTRACT

In the field of digital marketing, efficient content gap analysis is crucial for developing effective SEO strategies. Traditional approaches to this task are time-consuming, representing a challenge for Organic Performance teams, who must process large volumes of data to stay competitive. This project addresses the problem within a media investment company by automating and optimising content gap analysis through semantic clustering of Portuguese search keywords.

The present work first uses the pre-trained Multilingual Universal Sentence Encoder to generate word embeddings for the Portuguese search terms, selected by its strong alignment with human semantic understanding. This word representation phase is followed by embeddings' dimensionality reduction via UMAP to facilitate more effective clustering, and the HDBSCAN algorithm is then applied to the reduced embeddings to group them into semantically coherent clusters. Finally, a systematic approach is employed to name the clusters, translating the technical output into accessible insights for SEO experts.

The results are significant, yielding a substantial reduction in time spent on content gap analysis and providing the SEO team with a more organised and accessible way to present data to clients.

The project contributes to the SEO and data science fields by demonstrating the practical application and benefits of employing and integrating advanced computational techniques into the digital marketing landscape. It sets the stage for future advancements, suggesting a scalable process that can be adapted to other SEO contexts, fostering innovation in digital marketing practices.

KEYWORDS

Natural Language Processing; Search Engine Optimisation; Semantic Clustering; Word Embeddings; Dimensionality Reduction; Portuguese Search Keywords;

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

SEO	Search Engine Optimisation
NLP	Natural Language Processing
SERP	Search Engine Results Page
BoW	Bag-of-Words
TF-IDF	Term Frequency-Inverse Document Frequency
CBOW	Continuous Bag of Words
GloVe	Global Vectors for Word Representation
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer
UMAP	Uniform Manifold Approximation and Projection
MSE	Mean Squared Error

1. INTRODUCTION

In today's digital world, search engines significantly influence how information is accessed and consumed. In 2019, around 68% of online experiences began with a search engine, and organic search alone accounted for approximately 53% of all website traffic (BrightEdge, 2019). These numbers highlight the relevance that visibility in search results represents for businesses' success, particularly the importance of organic search – which represents the natural search results determined by the search engine's algorithms based on the relevance to the user's query. In the context of organic search, the Search Engine Optimization (SEO) field plays a crucial role in defining the best strategies to improve online visibility by refining the website's quality and content to boost its ranking in organic search results.

The current document presents a project that occurred during a 9-month internship between September 2021 and May 2022 as a data scientist at GroupM, the world's leading media investment company, part of WPP's group, which has the mission to deliver the best advertising to people while generating value for clients of several business areas (GroupM, 2023).

Within the Organic Performance team, one of the developed projects, and the one described in the present document, was focused on the task of content gap analysis – which identifies the keywords, also called search terms, for which the competitors appear in the organic search results but for which our client does not, identifying areas of improvement to create or optimise the client's website's content. Typically, this task requires a substantial manual effort and time from the team to perform since it involves analysing a file with a large volume of keywords, looking into keywords one by one, and identifying, given the business context of the client in question and two keyword performance metrics (average monthly search volume and keyword difficulty[1]), the most promising keywords where the team should allocate the optimisation work. The long time consumed to perform this task and the type of steps it involves generated an ideal case for applying automation and optimisation.

This project aimed to automate and refine the content gap analysis task, leveraging Natural Language Processing (NLP) techniques and clustering algorithms to group Portuguese keywords into semantic topics.

This internship report details the journey of meeting this goal, illustrating the powerful combination of NLP and clustering techniques with SEO and exploring how these techniques can be used to understand the Portuguese language complexities and give businesses an advantage in the quickly evolving digital market.

The present document is structured to first guide the reader through a literature review (section 2), presenting some of the foundational aspects of SEO and of the content gap analysis task, the principles and practices of NLP with particular detail for the Portuguese language, and the dimensionality reduction and clustering algorithms used to develop this project. Then, the methodology section (section 3) describes how all the techniques were applied, followed by section 4, where the results obtained in the project are explained and discussed, giving a clear view of the process's main output.

[1] Keyword Difficulty is a value that represents how difficult it is to rank a search keyword in the search results page

At the end of the document, the conclusions and limitations of the project are presented to summarise the process and clarify the possible future work.

2. TECHNICAL BACKGROUND

This chapter presents a comprehensive overview of the foundational concepts and techniques that were employed to develop the process of automating the SEO content gap analysis with Portuguese search keywords semantic clustering.

Guiding the reader with the details of the team environment where the project was developed and to fully understand the task that the present project proposes to automate, the chapter starts by introducing SEO (section 2.1.), explaining some of its structural concepts, its importance in digital marketing strategies and its evolution into the present state in the field. Then, keyword research and content gap analysis tasks are explained to clearly describe how the process is traditionally performed and its relevance in identifying opportunities for SEO improvements.

Then, both the Natural Language Processing and Word Representations sections (2.2. and 2.3.) are presented. First, it introduces the broader topic of NLP, explaining the field of NLP and its importance to several tasks, such as the one explained in the present document, and explains some basic NLP concepts crucial to understanding the steps performed in the project and more advanced techniques that use those concepts as their basis. Relevant considerations and challenges for the Portuguese language are also presented, given that the project focuses on Portuguese search keywords. Delving into the word representations section (2.3.), the most famous traditional methods used to represent words are explained to give the reader an understanding of the methods used to create more complex models, followed by the word embeddings topic that explains the several word embeddings types and methods employed in this project's process.

Section 2.4. mentions the importance of dimensionality reduction when dealing with high-dimensional spaces and explains how the dimensionality reduction algorithm used in the project, UMAP, works.

To conclude the technical background, section 2.5. explains two clustering types that form the basis of the chosen algorithm for the solution, combined with the explanation of the referred chosen algorithm, HDBSCAN.

2.1. SEARCH ENGINE OPTIMISATION

2.1.1. Introduction to Search Engine Optimisation

Search Engine Optimisation is a digital marketing strategy that aims to increase a website's visibility for specific keywords or phrases in the organic (unpaid) search engine results on the Search Engine Results Page (SERP) (Yalçin & Köse, 2010). The SERP displays the web pages that a search engine has found in response to a keyword search, and it usually includes both paid advertisements and organic results.

Although both organic and paid results appear on the SERP, their position and presentation differ. While organic results are ranked based on factors like relevance to the search query and the site's authority, paid results, also called Pay-Per-Click advertising, are displayed based on monetary investments made by advertisers (Li et al., 2014).

The main objective of SEO is to make a website more visible in organic search results, and it includes several technical and creative components to achieve that, such as working on the website's structure, content, external links, and user experience (Jerri L. Ledford, 2009).

2.1.2. Evolution and Importance of SEO

Search Engine Optimisation has changed substantially since the beginning of the World Wide Web, reflecting the quick and profound changes in digital technology and internet usage. While initial SEO tactics typically consisted of just filling websites with relevant keywords to increase their rankings in search results, current strategies are more complex because they have evolved due to the sophisticated algorithms that search engines have adopted to give users more accurate and relevant results (Almukhtar et al., 2021).

Nowadays, SEO is a crucial component of digital marketing strategies, and modern businesses use SEO to make sure their websites are easily discoverable by customers in a cluttered digital landscape (Matta et al., 2020). Additionally, more than just attracting visitors to a website, it is essential to attract valuable consumers interested in a company's goods or services. Beyond just using keywords, SEO now includes a wide range of on- and off-site tactics, such as high-quality backlink building, content optimisation, user experience design, and mobile friendliness(Enge et al., 2015).

2.1.3. Keyword Research

Keyword research is an essential task of SEO teams that analyses search queries to determine which keywords or phrases users most enter when searching for information on a particular topic or industry. Its primary purpose is to understand user language and preferences when searching online so that content creators and marketers can modify their tactics and content in order to target the most appropriate audiences. For instance, knowing that the search query "sneakers for running" is more searched than "running trainers" can influence marketing decisions for a sports website, although the two search terms represent the same type of footwear.

Keyword research also helps determine how competitive the market is for specific terms by evaluating the number of other entities targeting the same keywords and the likelihood of having profit when doing it. Additionally, related terms or semantic variations are also identified to enrich the understanding of user search behaviour (Stockwell, 2011).

In summary, keyword research ensures that content aligns with consumer needs, setting the stage for more sophisticated SEO methods like content gap analysis.

2.1.4. Content Gap Analysis

Content gap analysis is a method used to compare a website to its competitors or industry benchmarks for identifying missing, underrepresented, or poorly optimised content areas. This approach has its roots in competitive intelligence, and the main objective is to find opportunities for improvement or differentiation (Chaffey & Smith, 2008).

Usually, this procedure consists of several steps:

1. Competitive Analysis: SEO teams start by determining their main competitors and evaluating their content, which creates the basis for comparison.

2. Keyword Analysis: They investigate the search keywords that are driving traffic to competitors' websites but are missing or underperforming on their websites.
3. Content Evaluation: The SEO teams assess the website's current content to see if it corresponds with the keywords and subjects identified as gaps.
4. Strategic Planning: A content plan is designed to either create new content or optimise existing content based on the gaps found.

The content gap analysis method has real strategic consequences for digital marketing. It can improve businesses' organic search visibility and drive more targeted traffic, leading to better user engagement and potentially increased revenue(Dave Chaffey & Fiona Ellis-Chadwick, 2019).

2.2. NATURAL LANGUAGE PROCESSING

2.2.1. Introduction to Natural Language Processing

Natural language processing can be defined as a group of computing methods to perform analysis and representation of human language to execute several tasks, such as machine translation, sentiment analysis, or text summarisation. Being a combination of Artificial Intelligence and Linguistics, NLP focuses on making it possible for machines to understand and generate language like humans (Cambria & White, 2014).

Many NLP methods and technology advancements have been seen in the last years, which is highly relevant for several industry applications, particularly for the work described in this document since it is the basis for clustering Portuguese search keywords into semantic topics. While NLP has been extensively researched for English and other major languages, the focus on Portuguese presents unique challenges and opportunities, which are discussed in the following sections (Quarteroni, 2018).

2.2.2. Important Concepts in NLP

The Natural Language Processing field includes several techniques and methodologies to analyse and understand human language. The following list outlines some of the field's fundamental concepts relevant to this thesis.

- Syntax and Semantics - syntax is the study of the rules that govern the structure of sentences in a language, while semantics refers to the meaning of those sentences. Syntax analysis often involves techniques such as parsing, whereas semantic analysis deals with understanding the context, which is usually required for applications like machine translation and summarisation (Jurafsky & Martin, 2014).
- Tokenisation - the process of breaking down a text corpus into smaller units, also known as tokens, that can be as small as characters or as large as words. This is a crucial step in NLP pipelines that is the foundation for more advanced processes (Manning et al., 2008).
- Stopword Removal - Stopwords are words filtered out before or after the text analysis process because they are considered noise in the text. Examples of stopwords are "and," "the," and "is." While stopword removal can significantly reduce the computational load, depending on the task, it may not always be beneficial (Manning et al., 2008).

- Stemming and Lemmatization - Both stemming and lemmatisation aim to reduce the inflectional forms of a word to a common base form. However, stemming is generally cruder than lemmatisation, cutting off prefixes or suffixes to find the stem. On the other hand, lemmatisation involves a vocabulary and morphological analysis to identify each word's lemma correctly (Manning et al., 2008).

These foundational concepts in NLP serve as the building blocks for more complex tasks, including those addressed in this.

2.2.3. Relevant Considerations for the Portuguese Language

While the most significant advancements seen in NLP technologies are focused on widely spoken languages like English or Mandarin, Portuguese presents unique challenges that need special consideration. The Portuguese language, spoken by more than 260 million people globally (Camões - Instituto da Cooperação e da Língua, 2023), has its own syntactic, semantic, and morphological particularities that impact the performance of NLP models.

One of the aspects to take into account is the grammar. For instance, Portuguese grammar presents an extensive system of verb conjugations and a more flexible word order than English, and these features can impact tasks like tokenisation and named entity recognition, requiring adaptations to standard NLP models (Gonalves et al., 2006).

Another consideration is the presence of regional dialects, particularly the differences between European Portuguese and Brazilian Portuguese that can significantly affect the performance of language models, especially in tasks such as sentiment analysis and machine translation, since the same word can have almost opposite meanings.

Moreover, Portuguese has a relatively smaller corpus of annotated text available for machine learning compared to languages like English, which often leads to the use of transfer learning or domain adaptation techniques when implementing NLP models.

2.3. WORD REPRESENTATIONS IN NLP

2.3.1. Introduction to Word Representations

Word representations form the basis of NLP application, serving as a bridge between human language and the computational models that aim to understand or generate text, its primary purpose is to capture the semantic essence of words in a format that Machine Learning algorithms can easily use.

In the early stages of NLP, words were often represented using simple methods such as one-hot encoding, where each word was associated with a unique identifier, essentially an index in a vocabulary. While these methods were straightforward, they lacked the depth to capture semantic relationships between words. Traditional approaches like Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) expanded on these foundational techniques to offer richer text representations. However, the actual transformation in word representation came more recently with the rise of word embeddings, leveraging advancements in machine learning, neural networks, and deep learning techniques to more effectively encapsulate nuanced semantic relations between words (Mikolov et al., 2013b).

As the NLP field grows, word representations continually adapt to incorporate more semantic and syntactic nuances, and this is especially important when applying NLP models to less commonly studied languages, like Portuguese, which has its linguistic challenges and fewer annotated resources.

The following sections will delve into traditional methods of word representations and several types of word embeddings.

2.3.2. Traditional Methods

2.3.2.1. N-Grams

N-grams are sequences of n words in a text in the same order, which serve as features, capturing context and aiding in capturing patterns in sequences of words.

N represents the number of combined words: a single word is a unigram, two forms a bigram, and three constitute a trigram. Illustrating with the sentence "I love to read books.":

- Unigram: (I), (love), (to), (read), (books)
- Bigram: (I, love), (love, to), (to, read), (read, books)
- Trigram: (I, love, to), (love, to, read), (to, read, books)

Besides tokenisation, n-grams are important in probabilistic language models, where each sequence, or n-gram, is associated with a probability $P(w|h)$ that represents the likelihood of a word w appearing after the sentence h.

Using n-grams in tasks such as speech recognition, spelling correction, and predictive text may be helpful since they harness the immediate lexical context of words (Jurafsky & Martin, 2014).

2.3.2.2. One-Hot Encoding

One-hot encoding is a method where a unique vector with a length equal to the size of the vocabulary represents each word in the vocabulary. The vector representing a word is composed of zeros in all elements except in the position corresponding to the word, where it is marked with a one. For instance, in a vocabulary ["books", "movies", "music"], the word "movies" would be represented as [0, 1, 0].

This method is simple and offers a distinct representation for each word. However, it is computationally inefficient for large vocabulary because it leads to high dimensional feature vectors and does not capture any semantic relationship between words (Naseem, 2020).

2.3.2.3. Bag-of-Words

The Bag-of-Words model transforms raw text into a fixed-size vector based on the vocabulary extracted from the text. While it maintains the multiplicity of words, it disregards the order or structure in which they appear.

The BoW model creates a vocabulary of all the distinct words in a given corpus. Each document or piece of text is subsequently represented as a vector, and each dimension of this vector corresponds to a word in the vocabulary, with the value in each dimension being the frequency of that word in the document (Jurafsky & Martin, 2014).

To exemplify, given the sentence "Books about books are unique books.", the vocabulary would be ["books", "about", "are", "unique"]. The BoW model would vectorise the sentence based on the frequency of each word in the vocabulary, and the respective vector would be [3, 1, 1, 1].

One of BoW's main advantages is its simplicity, being easy to understand and implement, and the fact that, regardless of the text's length, the output vector remains of a fixed size, equal to the size of the vocabulary. On the other hand, it loses the order of words, which means that nuances arising from word order or sentence structure are not captured. Additionally, for large vocabularies, the BoW representation can be very sparse, composed of many zeros, which leads to memory inefficiencies, and it does not capture the semantic meanings of similar words in different forms, treating them as distinct.

2.3.2.4. Term Frequency – Inverse Document Frequency

Term Frequency-Inverse Document Frequency is built on the principle that words frequently appearing in a specific document but being scarce across the corpus should be considered more valuable.

The concept is formulated by combining two fundamental metrics: Term Frequency (TF) and Inverse Document Frequency (IDF). TF(t,d) measures the occurrence of a term, t, within a document, d, and IDF(t, D) quantifies the significance of a term, t, across all documents, D, penalising those terms that are widespread, as they often carry less informational value.

Mathematically, for term t, in document d, inside the corpus D:

$$TFIDF(t,d,D) = TF(t,d) * IDF(t,D) = TF(t,d) * log\frac{|D|}{|\{d \in D : t \in d\}|}$$

This weight refines the distinction between texts, which is beneficial in several tasks, such as text classification and information retrieval. However, TF-IDF cannot capture semantic relationships or the syntactic roles of words in a document since it represents each word independently (Jones, 1972).

2.3.3. Word Embeddings

2.3.3.1. Introduction to Word Embeddings

One of the main challenges in the NLP field is the representation of text data in a manner that captures its underlying semantic and syntactic structures. While traditional methods like the ones mentioned in the previous section use sparse and high-dimensional representations, word embeddings shifted toward dense and continuous vector representations.

Word embeddings can be defined as dense vector representations of words in a continuous space where the semantic relationship between words corresponds to geometric relationships in the vector space. Specifically, word embeddings map words or phrases from the vocabulary into vectors, leveraging the context in which they appear in large corpora. In mathematical terms, each word is represented as a point in a high-dimensional space, and its position is determined by its contextual meaning, making words with similar meanings or that frequently appear in similar contexts closer within this space.

2.3.3.2. Foundational Embedding Models

2.3.3.2.1. Word2Vec

Word2Vec is a word representation model introduced by (Mikolov et al., 2013b) that aims to create dense vector representations of words using shallow neural networks, differing significantly from prior count-based methods.

The Word2Vec model comprises two primary architectures: Continuous Bag of Words (CBOW) and Skip-Gram. While CBOW predicts a word based on its context, Skip-Gram does the opposite by predicting the surrounding context given a word. Comparing the two approaches, Skip-Gram often showcases superior performance, especially with smaller datasets and capturing infrequent terms.

One of Word2Vec's most celebrated features is its ability to distinguish semantic and syntactic relationships. This model captures relationships in the vector space, such as the renowned analogy where "Berlin" minus "Germany" plus "France" results in a vector near "Paris" (Mikolov et al., 2013a).

2.3.3.2.2. GloVe

GloVe (Global Vectors for Word Representation), introduced by (Pennington et al., 2014), was developed to combine the benefits of count-based methods and predictive methods.

The Glove method uses a co-occurrence matrix from the corpus that registers how frequently pairs of words appear together within a specified context window. So, for a given word pair (i, j), the corresponding matrix entry represents the likelihood that word j will appear in the context of Word i. Instead of directly using this co-occurrence matrix, GloVe's innovation lies in its factorisation, learning word vectors such that their dot product mirrors the logarithm of the words' co-occurrence probabilities, which enables the encoding of meaningful semantic and syntactic relationships into these vectors. The intuition is that the structure in the co-occurrence statistics holds valuable semantic information, and the embeddings seek to represent this structure in a dense form.

2.3.3.2.3. FastText

The FastText model presents a different approach from the previously mentioned models since it treats words as compositions of character n-grams, enabling it to recognise some semantic meanings from words by analysing morphological structures inside it (Bojanowski et al., 2017).

Unlike models that generate embeddings solely for whole words, FastText considers groups of n-grams to represent words. For instance, considering the word "model" with n=3, the word representation would be: "<mo", "mod", "ode", "del", and "el>". This approach allows FastText to capture the meaning of different word parts.

Representing words as a bag of these character n-grams allows FastText to encapsulate subword semantics, which is beneficial, especially for languages with rich morphological rules where a single word root can lead to multiple word variants based on prefixes, suffixes, or infixes.

Another advantage of FastText is its capability to generate embeddings for out-of-vocabulary words. Leveraging subword information provides a solution for representing words not seen during training, addressing a notable challenge in the NLP community. Furthermore, handling subword nuances

enables FastText to improve performance on various NLP tasks, particularly in languages where morphology plays a pivotal role in word construction and meaning.

2.3.3.3. Contextualised Word Embeddings

Contextualised word embeddings are, as the name suggests, word representations capable of capturing the semantic meaning of a word based on its context within a sentence and, unlike previously discussed word representations, contextualised embeddings can represent multiple meanings of a word, which provides a richer understanding of the text.

With the developments in NLP, several pre-trained language models have been created, setting new benchmarks for several tasks and leveraging deep learning and transformer architectures to capture deep semantic and syntactic structures of languages. Transformer models use self-attention mechanisms that weigh the importance of different text parts without directly relating them to their position, leading to a better understanding of context.

2.3.3.3.1. BERT

BERT (Bidirectional Encoder Representations from Transformers), unlike other models that analyse text in a unidirectional manner, is designed to consider the context from both directions (left to right and right to left). This bidirectional context is achieved using the Masked Language Model, where random words in a sentence are masked, and the model is trained to predict them based on context (Devlin et al., 2019).

BERT has also been used as the base of some variants and improvements of this model, such as RoBERTa (Liu et al., 2019), which modifies the training procedure for better performance, and ALBERT(Lan et al., 2019), which uses parameter reduction techniques to scale the model without increasing the computational requirements significantly. Other pre-trained models based on BERT were adapted to consider other languages, like Portuguese. An example is BERTimbau, introduced by (Souza et al., 2020), which provides resources for several NLP tasks.

2.3.3.3.2. GPT

With a different approach, the Generative Pre-trained Transformer (GPT) models are generative language models that employ a unidirectional approach and focus on generating coherent and contextually relevant text, by predicting the next word in a sequence based on the preceding words. These models demonstrated excellent capabilities in several generative tasks and showed the potential of transformer models in complex language generation (Radford et al., 2018) (Radford et al., 2019).

2.3.3.4. Sentence-Level Embeddings

To conclude the word representation section of this literature review, there are also models designed to generate embeddings at the sentence level that are worth mentioning, capable of capturing contextual relationships and semantic meaning of entire sentences.

2.3.3.4.1. Universal Sentence Encoder

Universal Sentence Encoder (USE) is a model capable of generating sentence embeddings by converting textual data into high-dimensional vectors that accurately represent the meaning of the

sentences. It uses deep learning techniques to produce those embeddings, with an architecture with two encoders, a Transformer for longer texts, and a DAN (Deep Averaging Network) for shorter texts, which ensures the model's flexibility to deal with different types of textual data (Cer et al., 2018).

Essentially, USE distinguishes itself by its ability to generate embeddings that consider the entire context of a sentence, being especially relevant for short texts or phrases where meaning is highly dependent on the context.

2.4. DIMENSIONALITY REDUCTION

2.4.1. Introduction to Dimensionality Reduction

Dimensionality reduction is an essential step in data preprocessing, especially when dealing with high-dimensional data, and it is usually helpful to reduce the computational load for following analysis and to mitigate the curse of dimensionality that can generate off-sides in terms of ML algorithms' performance. In the context of word or sentence embeddings, dimensionality reduction represents an essential part of the workflow since it highly influences the subsequent tasks that can be performed. Techniques for dimensionality reduction can be classified into feature selection and feature extraction methods, and while the first involves selecting a subset of the most informative features, the latter transforms data into a lower-dimensional space, keeping as much information as possible.

2.4.2. UMAP

Uniform Manifold Approximation and Projection (UMAP), introduced by (McInnes, Healy, & Melville, 2018), is a dimensionality reduction technique that excels in preserving both local and global structures of high-dimensional data, making it a good tool for sentence embeddings generated by language models. The algorithm works on the principle of manifold learning by assuming that within complex and high-dimensional data, a much simpler underlying structure (the manifold) can be captured and represented in lower dimensions. The main goal is to discover this lower-dimensional structure and use it to project the high-dimensional data into a space where similar points are closer together while still reflecting the original data's intrinsic geometric and topological properties.

The UMAP process starts by approximating the manifold on which the data lies using a k-nearest neighbor graph, where each point in the high-dimensional space is connected to its k-nearest neighbors, forming a weighted graph that captures the local structure of the manifold. This graph's weights are calculated by transforming the distance between points into probabilities representing the likelihood of the points being connected on the manifold.

When the weighted graph is created, it is transformed into a fuzzy simplicial set – a topological structure that retains the essential features of the manifold by representing the relationships between points as simplices (abstract representations of triangles, tetrahedra, and their n-dimensional analogs).

After that, to create the lower-dimensional projection that preserves the topological structure, UMAP initialises the layout using spectral embedding, which serves as a starting point for the optimisation with the Stochastic Gradient Descent where the objective is to minimise cross entropy between the high-dimensional fuzzy simplicial set and its low-dimensional representation.

During several iterations, UMAP optimises the points' positions in the low-dimensional space, reducing the disparities between the connectedness of points in both high and low-dimensional spaces.

By the end of the process, the UMAP algorithm returns an embedding that closely mimics the original data's essential geometric and topological properties.

2.5. CLUSTERING ALGORITHMS

2.5.1. Introduction to Clustering

Clustering can be defined as a type of unsupervised learning method used for categorising and grouping unlabeled data into clusters based on similarities in their features. There are several algorithms to perform clustering, which can be categorised into several types, including partition-based clustering, density-based clustering, hierarchical Clustering, distribution-based clustering, and several more (Xu & Tian, 2015).

This section describes the clustering methods related to this project's process and details the used algorithm, HDBSCAN.

2.5.2. Hierarchical Clustering

Hierarchical clustering is a method of cluster analysis that seeks to build a hierarchy of clusters, and that has been used in several scientific domains for exploratory data analysis and pattern detection. Usually, the approaches used in hierarchical clustering can be classified into two main categories:

- **Agglomerative Hierarchical Clustering**: in this bottom-up approach, each data point starts by belonging to a separate cluster and, as the process evolves and goes up on the hierarchy, pairs of clusters are merged. The process continues by continuously merging clusters until all points are combined into one single cluster.
- **Divisive Hierarchical Clustering**: opposed to the agglomerative approach, this top-down approach starts with all data points inside the same clusters and then the cluster is divided into smaller clusters in a step-by-step process.

Figure 1 from (Widia Sembiring et al., 2010) shows a dendrogram, usually the visualisation used for these algorithms' applications, to illustrate the agglomerative and divisive hierarchical clustering approach. The height at which two clusters are merged represents the distance between those clusters.

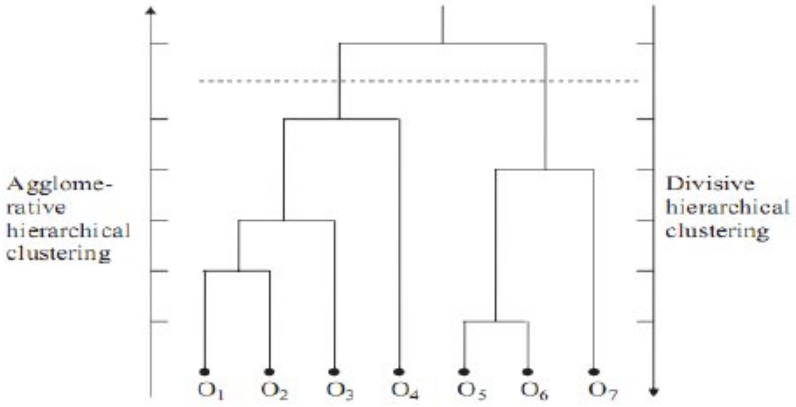

Figure 1 - Illustration of Agglomerative vs Divisive Hierarchical Clustering

To decide which clusters should be combined (agglomerative) or split (divisive), dissimilarity measures are used, like Euclidean or Manhattan distances, to calculate the distances between all pairs of observations in the two clusters in analysis and to use the linkage criteria further to determine how to compute the already calculated distance between those clusters. The linkage criteria can have different definitions, and some common examples are the single/minimum linkage clustering, where the distance between two clusters is defined as the minimum distance between any two points from the two distinct clusters, or the average linkage clustering, where the considered distance is the average distance between each point in one cluster and each point in the other cluster. Both linkage criteria and distance metric parameters are crucial in the performance of hierarchical clustering, and it is important to have enough knowledge of the data and the clustering objective to obtain good results.

In summary, hierarchical clustering is particularly useful for exploratory data analysis but can also be computationally expensive, especially for large datasets.

2.5.3. Density-Based Clustering

Density-based clustering is a group of clustering algorithms that identifies clusters as dense regions of data points separated by areas of lower point density and capable of discovering clusters of arbitrary shape, providing good performance when dealing with noise and outliers. Moreover, unlike partition-based methods such as K-Means, these algorithms do not require prior knowledge of the number of clusters, which provides flexibility to different data structures.

The main idea behind these algorithms is to grow and expand clusters based on the density of data points until the density in a region exceeds a certain threshold (this means that for each point within a specific distance, there is a minimum number of points within this distance to qualify a region as a dense region) and there are two fundamental principals:

- Density Reachability: A point is part of a cluster if it is within a certain distance from a core point of the cluster and there is a minimum number of points within this distance.
- Density Connectivity: A point belongs to a cluster if it is connected to a core point through a chain of points that satisfy the density reachability condition.

These two conditions guarantee that clusters are formed based on the local density of points, which enables the detection of clusters of several shapes and sizes.

2.5.3.1. DBSCAN: Density-Based Spatial Clustering of Applications with Noise

One of the most well-known and used density-based algorithms is DBSCAN, introduced by (Ester et al., 1996), operates with two parameters, ε being the neighborhood distance threshold and *MinPts* being the minimum number of data points required to form a dense region, and has the following workflow:

1. Start: Choose an arbitrary unvisited point;
2. Neighbor Search: Retrieve all points within distance ε. If the number of points in it is equal or greater than *MinPts*, initiate a new cluster;
3. Expand Cluster: For each new point in the cluster, check its ε-neighborhood, adding new points to the cluster as necessary;
4. Repeat: Continue until all points are either assigned to a cluster or marked as noise.

A point that does not belong to a cluster is treated as noise or outlier, but if a noise point is found within a distance ε from a core point of a cluster, it is reclassified as a border point in that cluster.

Although DBSCAN presents the advantages of not requiring previous knowledge of the number of clusters, being capable of identifying clusters of various shapes, and effectively distinguishing noise and outliers, it also has some challenges and limitations associated especially with the fact that the parameters definition can be challenging and with its struggles with data that has varying density regions, since a single set of parameters will not be suitable for all regions.

2.5.4. HDBSCAN: Hierarchical DBSCAN

HDBSCAN is a clustering algorithm that extends DBSCAN by adapting it into a hierarchical clustering algorithm (Ricardo .J.G.B. et al., 2013). By removing the need to define an optimal ε value, thus automating what is often a challenging step, and introducing the concept of minimum cluster size as a single intuitive parameter, it overcomes some of the primary limitations of DBSCAN.

The main idea behind this algorithm is to create a hierarchy of clusters following several steps, which are simplified in the explanation below:

1. Start: Similar to DBSCAN, HDBSCAN starts by transforming the space according to density, but instead of picking a particular limit (ε), it looks at how these clusters connect in a hierarchy (since the limit is varied);
2. Build a Hierarchy: It uses a hierarchy of connected components, building a cluster tree, by employing the concept of mutual reachability – which adds an additional factor to the distance metric – ensuring that clusters are formed based on varying densities;
3. Condense the Tree: The tree is then simplified and transformed to emphasise the most prominent clusters by pruning branches that do not contain enough points;
4. Extract Clusters: Clusters are extracted from the transformed tree based on stability, which measures how persistently points stay together through the hierarchy.

Besides the already mentioned, one of the main advantages of this algorithm is its ability to perform soft clustering. Unlike hard clustering algorithms that force every point to belong to a cluster, even if it is an outlier, HDBSCAN allows points to be classified as noise and clusters to be ranked by the

likelihood of points belonging to them, which provides a significant advantage, especially for data with border points that may not be clear to which cluster they belong.

2.6. Related Studies

The paper proposed by Hartmann et al. presents an extensive evaluation of word embedding models for the Portuguese language, where 31 word embedding models were trained and evaluated, utilising algorithms such as FastText, GloVe and Word2Vec. This study's primary outcome lies in its extrinsic evaluation, which assessed the performance of the models on Part-of-Speech tagging and sentence similarity tasks, which is particularly relevant to the present project since one of its most critical steps is to correctly represent keywords as word embeddings in a way that can capture the semantic similarity between them. The authors found a lack of correlation between the models' performances on syntactic and semantic analogy tests (intrinsic evaluations) and their effectiveness in practical NLP tasks like semantic similarity (extrinsic evacuation). This insight reinforced the importance of extrinsic evaluation metrics, namely Mean Squared Error and Pearson correlation, to assess the effectiveness of word embeddings in capturing semantic relationships within Portuguese text (Hartmann et al., 2017).

Another relevant study for this project was proposed by Muhammad Asyaky and Rila Mandala, where the authors address the challenges in clustering short texts, such as tweets. They recognise that short texts often suffer from sparsity and lack of word co-occurrences, making extracting semantic information difficult. This paper proposed a novel method integrating word embeddings, FastText and Bert, with the HDBSCAN clustering algorithm. The high-dimensional data resulting from the embeddings was reduced using UMAP before being fed into HDBSCAN, given that HDSCAN alone struggles with high-dimensional data. This approach outperformed the baseline in terms of purity and Normalised Mutual Information, which means that the method employed can effectively cluster short texts by enhancing the semantic understanding of words (Asyaky & Mandala, 2021). This study is particularly relevant to the described project in the present report as it validates the effectiveness of using UMAP for dimensionality reduction in combination with HDBSCAN for clustering. The focus on short texts and the combination of these specific techniques align closely with the present project development, which also leverages UMAP and HDBSCAN but in the context of Portuguese search keywords.

3. METHODOLOGY

This section describes the project's workflow by giving all the details about each process step. It is relevant to mention that a requirement from the team was to have a tool that worked for several clients from several industries. The tests and results in the present document were done on data from a client in the retail industry, specifically in the beauty area, but the idea was always to develop a process that could be applied to every industry.

3.1. DATA GATHERING AND UNDERSTANDING

The dataset used for this project was gathered from ahrefs[2], an SEO tool that makes it possible to get detailed keyword information about specific web pages and their competition. The ahrefs tool was chosen because it is capable of retrieving relevant keyword information and also due to its usual usage within the SEO team's workflow for content gap analysis, which would help align with the team's established practices.

More specifically, the data is obtained through the ahrefs' Site Explorer feature, and given one or more competitor's web pages, it highlights keywords for which competitors were successfully ranking and which were missing from the client's visibility. This information is highly relevant to identifying content gaps on the client's pages, and its optimisation can offer significant advantages regarding search engine ranking and overall digital presence.

The initial dataset is composed of 286 keywords, and for each keyword, it presents two metrics essential to understanding the context and relevance of each keyword in the digital landscape:

- Search Volume: Serving as a measure of a keyword's popularity, the search volume metric captures the number of times, on average, that users entered a search query, or keyword, into a search engine over a month. This information is usually important when analysing the potential user interest and determining the keyword's overall significance.

- Keyword Difficulty: Besides popularity, the keyword difficulty provides a vision into the competitive landscape of that term. In other words, this integer value gives an idea of how difficult it is to rank a search query in the SERP results. A keyword with high difficulty suggests that many entities are competing for its top-ranking place, making it crucial for strategising content decisions.

It is also important to note the length of the keywords, in terms of number of words, given that a significant limitation that was found is the lack of context due to the short number of words in search keywords. Figure 2 shows that the search terms subject to analysis are composed of approximately three words on average, and around 40% of the terms have three words (113 search terms).

[2] https://ahrefs.com/

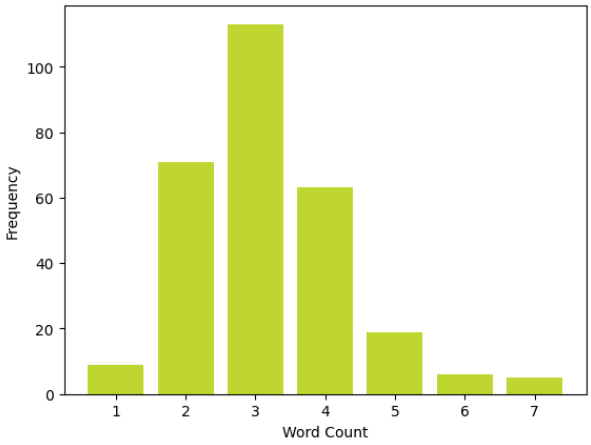

Figure 2 - Distribution of keywords word count

3.2. Data Cleaning and Preprocessing

This section describes the methodologies used to clean and preprocess the search keywords in the dataset to remove any noise or irrelevant information and guarantee that the search terms were all optimised for the Word embedding models and semantic clustering applications.

The first step in cleaning data is removing branded keywords from the dataset. As previously mentioned, extracted data reflects the keywords the competitors rank for, but the client in the analysis does not. So, it is expected to find specific keywords with its brand in the search terms. Although this information can be relevant to other types of studies, it is not relevant for the content gap analysis because the main purpose is to find the keywords that can be useful to the client's business for content optimisation or generation. The execution of this step requires the SEO team to define the client's competitors, and after that, the information is included in the process script to be considered.

After branded keywords removal, all the keywords containing punctuation marks (.,_:!?) were also discarded from the dataset due to their irrelevance to the study since the SEO optimisation is done for the exact expression and not for similar ones, which is the reason for not excluding just the punctuation marks and keep the rest of the search terms. In the same line of thought, all keywords comprising more than 75% numeric characters would also be removed from the list of search keywords. This decision is mainly based on the existence of phone numbers in the results, which would not make sense to include in the present study.

Lowercasing all keywords was also considered in the preprocessing phase to guarantee consistency and prevent the development of duplicated embeddings due to case differences.

Lemmatisation was also applied to the list of search keywords to reduce each word to its base or root form, ensuring that different forms of words are treated as the same entity. For instance, the keywords "produto cosmética" and "produtos cosméticos" will be transformed into "produto cosmético" and thus will be treated as the exact same search terms. This step was applied using spaCy's Python library[3]

[3] https://spacy.io/models/pt#pt_core_news_md

, and it is particularly important for tasks for languages with many variations, such as Portuguese, because it enhances data unification and reduces the dimensionality, which is essential for the clustering phase.

Before generating word embeddings for each keyword, stop words were also removed, using the NLTK[4] stop words corpus for Portuguese, where words like "o", "a", "e", "de", "como", and "com" were removed from the search terms because its presence would not contribute to the creation of word embeddings and would increase the dimensionality.

To conclude, it is important to note that no accentuation was removed before the Word embeddings generation because there are several cases where an accent in a word completely changes its meaning (e.g., "pôr" (to put/to place) and "por" (by/through)). Moreover, this technique will be used in the cluster's naming section in the methodologies chapter.

3.3. WORD EMBEDDINGS GENERATION

The selection of an appropriate word embedding model to represent Portuguese search keywords was the next logical step in the process workflow. Given the lack of internal pre-classified data to train models and analyse their performance, the choice was to use three public datasets created for evaluation in lexical similarity tasks, tailored for the Portuguese language, comprising pairs of words and human annotations with similarity scores(Querido et al., 2017):

- **LX-WordSim-353**: containing 352 pairs of words composed of nouns, adjectives, verbs, and named entities;
- **LX-SimLex-999**: containing 999 pairs of words (666 pairs of noun-noun, 222 pairs of verb-verb, and 111 pairs of adjective-adjective);
- **LX-Rare Word Similarity:** containing 2034 words (1017 pairs of words), extracted from Wikipedia and WordNet.

These three datasets served as a benchmark to evaluate the performance of pre-trained Word embedding models. The rationale behind the process was to find the model that would better mimic human judgment in terms of capturing semantic similarity between words, which is crucial for the clustering phase of the process since the primary purpose is to group similar search keywords into semantic topics.

To find the best word embedding model, after merging the three previously mentioned public datasets and obtaining a list with 4033 unique words distributed in 3385 pairs of words, all the words in the list were transformed into word embeddings using the following pre-trained models:

- **FastText** (Grave et al., 2018): a pre-trained model developed by Facebook that stands out for its ability to generate word vectors while considering the internal structure of words. The model used in this study, implemented using the fastText Python library[5], was trained on a large corpus (with Common Crawl and Wikipedia data) and uses CBOW architecture

[4] https://www.nltk.org/
[5] https://fasttext.cc/docs/en/python-module.html

with position weights. The generated word vectors have 300 dimensions and are generated with character n-grams of length 5, a context window of size 5, and 10 negative samples.

- **BertTimbau**(Devlin et al., 2019)**:** pre-trained BERT model tailored for Brazilian Portuguese (although there are some differences between Brazilian and European Portuguese, these models were still tested). This model has demonstrated promising results on several NLP tasks, such as sentence textual similarity. It has two available variants: Base (with 12 transformer layers and 110 million parameters) and Large (with 24 transformer layers and 335 million parameters). Both variants were tested using the Hugging Face Transformers library (Wolf et al., 2020).
- **RoBERTa_PT**(Santos et al., 2021): variant of the RoBERTa model (Liu et al., 2019), pre-trained on a corpus with 10 million Portuguese and 10 million English sentences from Oscar corpus and is composed of six transformer layers and 68 million parameters.
- **GPT-2**(Radford et al., 2019): pre-trained transformer-based language model designed to generate text that performs well in text similarity tasks. It is based on an architecture focused on predicting the next word in a sentence, performs well when capturing the context of words, and has 1.5 billion parameters. It was implemented using the Hugging Face Transformers library.
- **Universal Sentence Encoder**(Cer et al., 2018): two variants of Universal Sentence Encoder were tested in the present study, the Large version 5 (USE Large v5) and the Multilingual version 3 (USE Multilingual v3). USE Large's architecture enables it to understand the contextual relationship between words in a sentence, providing a rich text representation. USE Multilingual(Yang et al., 2019) uses a convolution neural network and is capable of understanding 16 different languages. Both variants produce 512-dimensional vectors as outputs and were implemented using TensorFlow Hub repository and library[6].
- **PTT5**(Carmo et al., 2020): a model based on the T5 model's architecture – a transformer-based encoder-decoder structure that leads with NLP tasks as text generation problems, providing a unified approach to several tasks (Raffel et al., 2019). PTT5 is pre-trained on a large corpus extracted from Portuguese webpages, and it leverages the T5 structure to provide embeddings that are contextually rich and semantically accurate for Portuguese text. This model was also implemented using the Hugging Face Transformers library.

The choice of testing the pre-trained models listed above was based on their specialisation tasks related to text similarity and their ability to work with the Portuguese language.

After generating the Word embeddings, it is necessary to compare and select the best-performing pre-trained model to apply to the search keywords' data and proceed to the clustering phase. To do this, two metrics were calculated for each pair of words and used as a comparison with the human-assigned scores, Cosine Similarity and Euclidean distance, whose formulas are the following, for two vectors A=$[a_1, a_2, \dots, a_n]$ and B=$[b_1, b_2, \dots, b_n]$:

$$Cosine = \frac{a^T b}{|a| \cdot |b|} \qquad d(A,B) = \sqrt{\sum_{n=1}^{N} (a_n - b_n)^2}$$

[6] https://www.tensorflow.org/hub

One necessary step to compare calculated similarity scores and the human-assigned scores was the normalisation of the calculated metrics, so all the values would be in the range between 0 and 10, the same value interval observed in the human scoring.

After similarity metrics calculation and respective normalisation, Mean Squared Error (MSE) and Pearson Correlation were calculated to determine which model best aligns with the human judgment, considering both cosine similarity and Euclidean distance.

The calculations behind this step are the following:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2 \qquad Pearson = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2(Y_i-\bar{Y})^2}}$$

Where X is the array of human-assigned scores, Y the calculated similarity scores (one at a time), and n is the number of records. $\bar{X}$ and $\bar{Y}$ represent the mean of the values of the respective variable.

Finally, the selected model should be the one presenting the highest correlation between human annotations and the calculated cosine and Euclidean metrics combined with the lowest values for MSE.

When the results' analysis is finished, and the pre-trained Word embedding model is selected, the mentioned model should be employed to generate embeddings for the search keywords in the original dataset, setting the stage for the following clustering phase.

3.4. CLUSTERING SEARCH KEYWORDS

When the transformation of search keywords is completed, the next step is to group those search keyword embeddings into semantically consistent clusters that can be delivered to the SEO team.

Before the clustering algorithm application, a dimensionality reduction algorithm was employed to reduce the number of dimensions present in Word embeddings to avoid issues related to the Curse of Dimensionality, where distance measures crucial for clustering can become unreliable. The chosen dimensionality reduction algorithm was UMAP due to its advantages over other algorithms - projecting data into a lower-dimensional space while preserving its inherent structure. UMAP was implemented using the UMAP Python library(McInnes, Healy, Saul, et al., 2018).

After dimensionality reduction, the chosen clustering algorithm to perform the search keywords grouping is HDBSCAN, which was implemented using the HDBSCAN Python library (McInnes et al., 2017). This choice was due to the model's capabilities of generating clusters without previously specifying the total number of clusters – which is a requirement of the project since one of its primary goals is to be as automated as possible – and its ability to deal with noise and ambiguity in data by providing soft clustering capabilities. The aptitude of being able to assign membership probabilities to each point for all clusters, allowing the analysis of uncertainty and potential overlaps between clusters, is particularly important in the context of the present study because even for humans some search keywords can be difficult to categorise in only one category since its nature can be ambiguous (e.g. from the data in analysis we have the keyword "removedor de verniz" (nail polish remover) which can

be easily included in a cluster where keywords are about nails or in another cluster focused on cleaning products).

To optimise the performance of both UMAP and HDBSCAN, hyperparameter tuning was performed using Random Search to find the best combination of parameters for UMAP and HDBSCAN. For UMAP, the parameters that must be defined are the number of neighbors, which controls how the model balances local and global structure in data, and the number of components, which defines the dimensionality of the resulting vectors after dimensionality reduction. In the case in analysis, the options given for the number of neighbors are [3, 4, 5, 10, 15, 20] - considerably low values to guarantee the UMAP model concentrates on the local structure and not be too broad – and for the number of components are [2, 3, 4] – although reducing a lot the number of dimensions, tests showed that the results were better when using lower-dimensional vectors. On the other hand, for the HDBSCAN model, the only parameter that must be defined is the minimum cluster size, for which the values included in the random search are [5, 10, 15, 20].

All combinations resultant of random search were evaluated and compared using a calculated cost metric that uses the previously mentioned assigned probabilities to each point inside the clusters, defined as:

$$Cost = \frac{Count(cluster\ probabilities < 0.05)}{Total\ Number\ of\ Keywords}$$

This metric essentially represents the proportion of keywords with a low probability (under 5%) of belonging to their assigned cluster, indicating potential misclassification in the clustering results. Additionally, to choose the combination of parameters to proceed, there is also the definition by the SEO team, given the volume of keywords and their business knowledge on the client website, of a maximum and minimum value for the number of created clusters to limit it and avoid very low or high number of created groups.

Finally, the best-performing combination of parameters in terms of cost, aligned with the limits established, is applied to the dataset, generating the final clusters. In this phase, there was also a qualitative assessment after generating clusters, involving manual inspection to understand if the process results were satisfactory.

3.5. CLUSTERS NAMING

Next to obtaining clusters for the search keywords, it is necessary to transform the numerical cluster labels into descriptive and meaningful names so they can be easily understood by the SEO team and facilitate the further application of the results into the generation of actionable insights.

The naming process was automated using a Python function, which looks into the original search keywords instead of their word embeddings.

The process starts by removing all accents from the keywords. This is done to avoid terms duplication, which would cause issues in the terms count inside the clusters (as already mentioned at the end of the Data Cleaning and Preprocessing section of this chapter, this was not done in the early stages of

the project because it was considered that it could make a relevant difference in the meaning of words; in this phase, it already makes sense to apply this method because keywords are already grouped by their meaning).

After keyword normalisation, a frequency analysis is performed to register the number of times every single word and bigram (pair of adjacent words) appear in each cluster.

In the end, a test is made for each cluster: if the most frequent word in the cluster appears more than twice the times the most frequent bigram appears, then the cluster gets its name with the most frequent word; otherwise, it gets the most frequent bigram as its name.

4. RESULTS AND DISCUSSION

After the word cleaning step, the final dataset comprised 265 distinct keywords. The volume of keywords distributed by word count is presented in Figure 3:

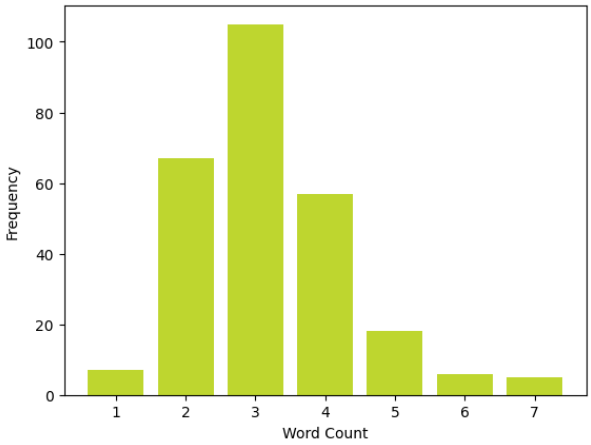

Figure 3 - Distribution of keywords word count after data cleaning

A study on Word embeddings was conducted after word cleaning and preprocessing, as mentioned in the methodology chapter.

A dataset with 3385 pairs of words was used to compare word similarity scores assigned by humans with similarity metrics calculated on top of Word embeddings created by the pre-trained word vectors listed in the methodology. The similarity metrics used were the Cosine Similarity and the Euclidean distance, and after their calculation, both Mean Squared Error and Person correlation were calculated between the two computed similarity metrics and the human-annotated labels. Table 1 details the average value for those calculations.

Model Name	MSE(Human, CosineSim)	MSE(Human, EuclideanDist)	PearsonCorr (Human, CosineSim)	PearsonCorr (Human, EuclideanDist)
Fast_PT	9.98	5.96	0.44	0.27
BERT_base	15.98	16.52	0.20	0.2
BERT_large	21.86	20.28	0.23	0.19
RoBERTa_PT	12.57	22.28	0.18	0.17
GPT2	34.19	22.03	0.06	0.16
USE_large	9.92	5.26	0.13	0.17
USE_multilingual	11.3	4.92	0.44	0.39
PTT5	33.8	7.78	0.04	0.14

Table 1 - Summary of Word embedding model's results

Analysing the obtained results, the main goal was to find the best combination of values of MSE and Pearson Correlation. The best combination of values would be the one with lower MSE, indicating that the used model's estimates are closer to the human judgements and the highest Pearson Correlation value, meaning a higher positive correlation between human scores and the calculated similarity metrics.

Although the FastText also showed good results within the results obtained, the chosen pre-trained word embedding was the multilingual variant of Universal Sentence Encoder since it showed the best results for MSE between human annotations and the calculated Euclidean Distance between word pairs and the highest Pearson Correlation values.

After choosing the pre-trained Word embeddings to use, a Random Search was performed to select the best set of parameters to run both UMAP and HDBSCAN. As already mentioned in the methodology, the evaluation metric used to decide the parameters to use is the calculated cost based on cluster probabilities, and this metric is then combined with the pre-defined limits for the minimum and maximum number of clusters that can be created. For the case in the study, the minimum number of clusters is 10, and the maximum is 30. Table 2 shows the top 10 results of the random search.

n_neighbors	n_components	min_cluster_size	label_count	Cost
5	3	5	20	0.000000
10	3	5	19	0.011321
20	3	20	3	0.033962
15	3	15	3	0.033962
15	4	20	3	0.033962
20	4	20	3	0.033962
15	3	20	3	0.033962
5	4	5	21	0.037736
4	2	5	22	0.041509
5	2	5	22	0.041509

Table 2 - Top 10 random search results

As we can observe, the best combination of parameters presented a cost of 0, meaning no point has less than 5% probability of belonging to the assigned clusters. Consequently, there will be no data points classified as outliers. Although this happened with this client and respective data being analysed, further work on other clients showed that this does not happen regularly, and for that reason, the automated process was created considering this; thus, the final output, delivered to the SEO team, has the information on the total number of keywords classified as outliers, as well as the detailed list of which keywords are those.

After running Random Search, using the results to generate clusters and naming them, as explained in the methodology, the results were pretty satisfactory, having created the clusters and corresponding labels for all the groups. Figure 4 shows the resulting distribution of keywords along the cluster names.

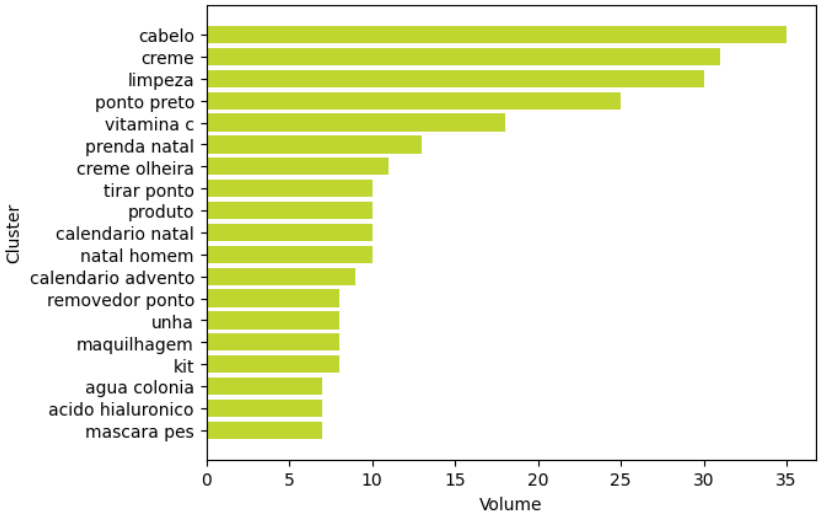

Figure 4 - Number of keywords by cluster name

Table 3 shows some examples of keywords and the corresponding assigned clusters.

Cluster Name	Keywrod
cabelo	protetor calor cabelo
	anti frizz
	serum para cabelo
	melhores shampoos
	alisamento vegan
unha	manicure francesa
	manicure pedicure
	unhas à francesa
ponto preto	creme pontos negros
	como tirar pontos pretos do nariz
	mascara pontos pretos

Table 3 - Examples of clusters' results

Analysing the output results, given that there was no pre-classified dataset to ease the performance analysis, manual classification of success was performed to better understand the process's quality. Of the 265 keywords, 28 were wrongly assigned to the cluster, representing approximately 10% of incorrect assignations over all the observations. Moreover, some cases were considered misclassifications due to bad cluster naming and not necessarily due to insufficient performance in clustering. An example of this is present in the cluster named with "vitamina c" (vitamin C), where the

keyword "citrato de mágnesio" (magnesium citrate) is included. For instance, if the cluster name was "supplement", it would be considered a reasonable classification.

Figure 5 shows the representation of each search keyword in a 2-dimensional space, where their colours correspond to the assigned clusters. This representation was achieved using the UMAP algorithm and Plotly's Python library[7].

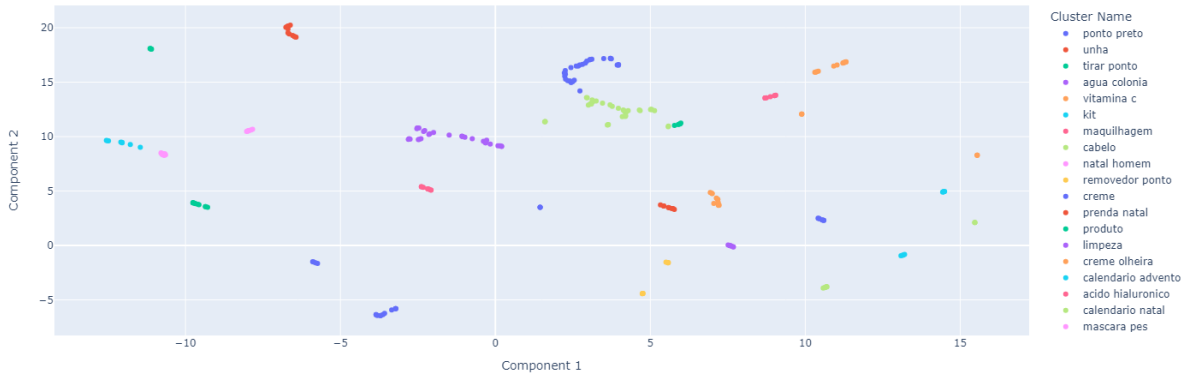

Figure 5 - 2-Dimensional representation of keyword clusters

It is possible to observe formed groups in space with keywords close to each other. In order to provide a clearer view of the clusters, Figure 6 is an adapted figure from Figure 5, in which there are circles rounding the clusters to facilitate their representation, visualisation, and interpretation. In order to make this readable, there was a need to omit three clusters from the representation ("vitamina c", "ponto preto" and "produto") since their volume and intra-cluster distances produced some noise and made the visualisation hard to interpret.

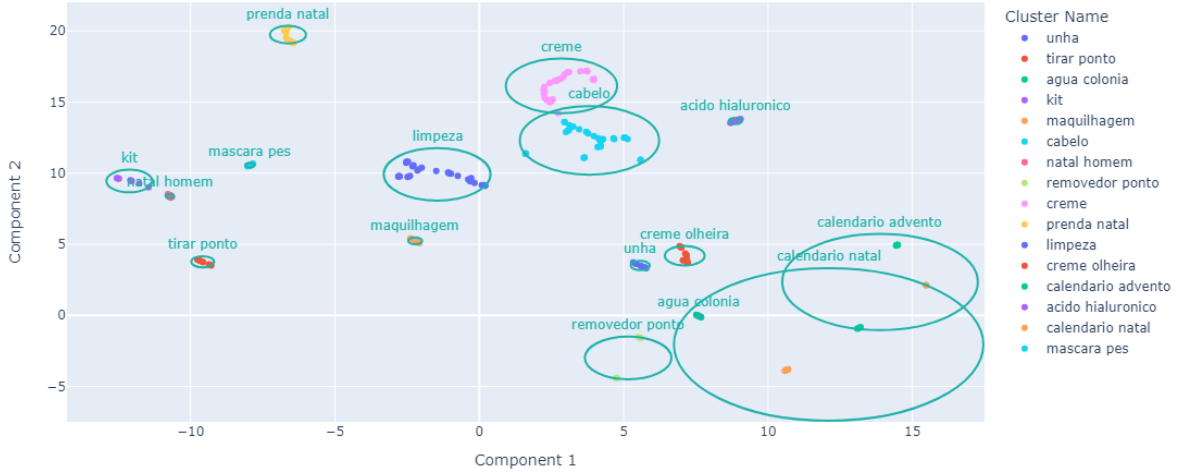

Figure 6 - 2-Dimensional visualisation of a part of the clusters

[7] https://plotly.com/python/

Finally, Figure 7 and Figure 8 show two visualisations that are part of the output retrieved to the SEO team through an Excel file. These visualisations are essential for the team because, when analysed together, they give valuable insights to guide the work that should come next, making clear which topic is more worth optimising in the client's pages.

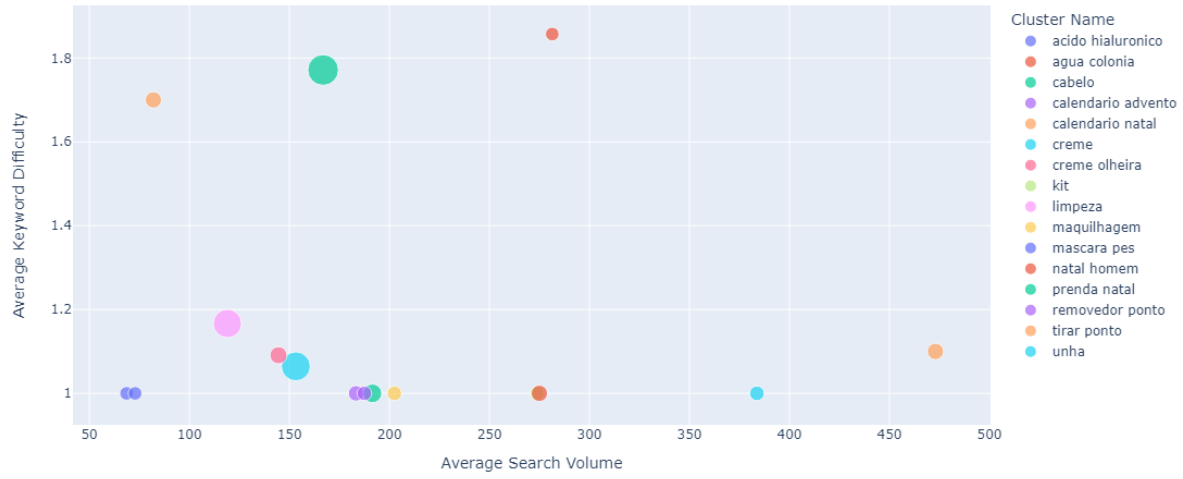

Figure 7 - Bubble chart with Average Keyword Difficulty over Average Search Volume, with size proportional to bubble size

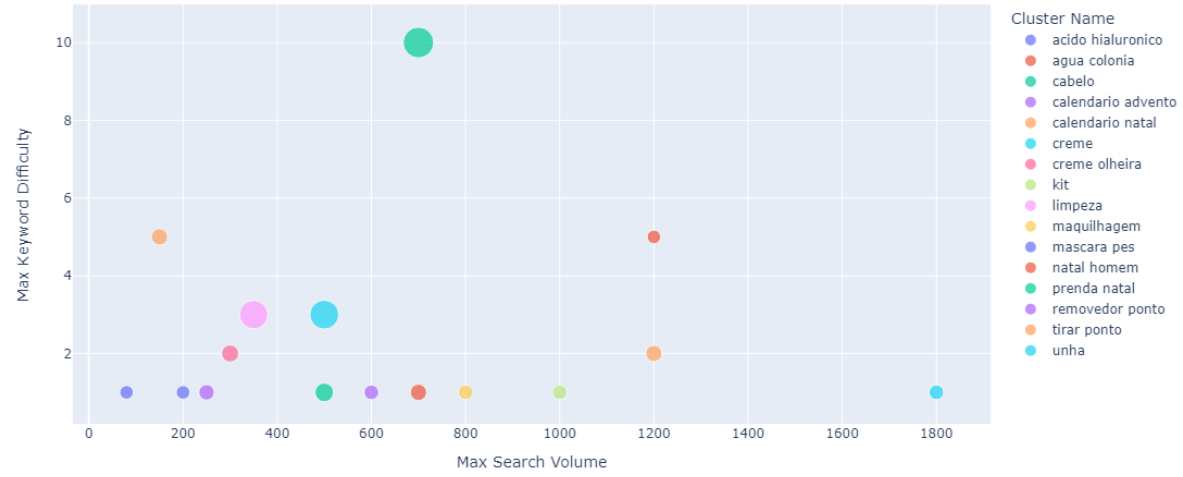

Figure 8 - Bubble chart with Maximum Keyword Difficulty over Maximum Search Volume, with size proportional to bubble size

From these visualisations, one insight that can be taken is that the clusters "tirar ponto" and "unha" look like the most promising ones, given their high average search volume value and low average keyword difficulty, being the priority terms to be taken into account for the proceeding tasks of pages' optimisation and content generation.

5. CONCLUSION

The primary objective of this project was to explore the automation and optimisation of content gap analysis through semantic clustering of Portuguese search keywords. Within the GroupM internship as a Data Scientist, the goal was to harness advanced data science techniques to enhance the SEO team's operational efficiency and insight generation.

The project was driven by the need to modernise a process that traditionally consumed significant time and resources. It was achieved by combining four crucial phases: the word representation of Portuguese search terms into word embeddings, the word embeddings' dimensionality reduction, the usage of clustering algorithms to group the keywords and finally, the assignment of names to those groups.

The journey began with testing several pre-trained word embeddings to select the one that would achieve the best alignment with human semantic understanding. A comparative analysis of the word similarities captured by the several models revealed a superior performance when using the Multilingual Universal Sentence Encoder to represent Portuguese words for semantic similarity tasks.

After generating the word embeddings for search keywords using the Multilingual USE, dimensionality reduction was performed to obtain lower-dimensional vectors with which clustering algorithms could deal. To complete this task, UMAP was used to effectively convert the complex high-dimensional space of word embeddings into a manageable form without sacrificing data integrity.

The subsequent phase of clustering the word embeddings employed the HDBSCAN algorithm, identifying dense regions of data points to organise the data into distinct semantically meaningful groups and form the clusters.

Translating the technical output of the clustering process into actionable marketing insights and providing an easily interpretable output to the SEO team, the final step of the process was the clusters' naming by systematically identifying the most representative terms within each group.

The feedback from the SEO experts was extremely positive, highlighting the reduction in the time spent on the content gap analysis, making their work more accessible and organised to present to the client.

In conclusion, the objectives of this project have been achieved, and the methodologies employed not only fulfilled the goal of automating content gap analysis but also opened the door to new possibilities in SEO strategy formulation. This work proves the power of leveraging advanced computational techniques to solve real-world marketing challenges, and it is a step forward in the ongoing journey of refining digital marketing strategies and workflows through innovative data science solutions.

6. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

During the process of automating and optimising the content gap analysis with semantic clustering of Portuguese search keywords, some challenges and limitations were faced.

Although a cost metric was used to evaluate the clustering performance, the absence of an already established dataset with a previously assigned labelling of keywords into semantic clusters led to the need for a human inspection to validate the actual effectiveness of the clustering process. The potential of the developed system could be further realised with access to a robust pre-classified dataset that would not only facilitate a more precise evaluation of the algorithm's output but also enhance the training process, thereby improving the model's accuracy and reducing the need for manual verification.

Moreover, future work could benefit from a broader comparative analysis, experimenting with a range of word embedding generation models and clustering techniques that could uncover more optimised approaches and provide insights into the scalability and adaptability of the system across different SEO contexts.

In conclusion, the process discussed in this report is suitable for further exploration. The development of comprehensive classified datasets for the Portuguese language and the continuous testing approach represent significant opportunities for advancing the automation of content gap analysis for Portuguese keywords, contributing to the efficiency and efficacy of SEO strategies in the dynamic landscape of digital marketing.

7. REFERENCES

Almukhtar, F., Mahmoodd, N., & Kareem, S. (2021). Search engine optimisation: A review. *Applied Computer Science*, *17*(1), 69–79. https://doi.org/10.23743/acs-2021-07

Asyaky, M. S., & Mandala, R. (2021). Improving the Performance of HDBSCAN on Short Text Clustering by Using Word Embedding and UMAP. *Proceedings - 2021 8th International Conference on Advanced Informatics: Concepts, Theory, and Application, ICAICTA 2021*. https://doi.org/10.1109/ICAICTA53211.2021.9640285

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). *Enriching Word Vectors with Subword Information*. https://doi.org/10.1162/tacl_a_00051/1567442/tacl_a_00051.pdf

BrightEdge. (2019). *2019 Channel Report*. https://www.brightedge.com/

Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. In *IEEE Computational Intelligence Magazine* (Vol. 9, Issue 2, pp. 48–57). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/MCI.2014.2307227

Camões - Instituto da Cooperação e da Língua. (2023). *Dia Mundial da Língua Portuguesa 5 de maio de 2023*.

Carmo, D., Piau, M., Campiotti, I., Nogueira, R., & Lotufo, R. (2020). *PTT5: Pretraining and validating the T5 model on Brazilian Portuguese data*. http://arxiv.org/abs/2008.09144

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. St., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., & Kurzweil, R. (2018). *Universal Sentence Encoder*. http://arxiv.org/abs/1803.11175

Chaffey, D., & Smith, P. (2008). *eMarketing eXcellence Planning and optimising your digital marketing* (3rd ed.).

Dave Chaffey, & Fiona Ellis-Chadwick. (2019). *Digital marketing: strategy, implementation and practice* (7th ed.).

Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. https://github.com/tensorflow/tensor2tensor

Enge, E., Spencer, S. M., & Stricchiola, J. (2015). *The Art of SEO : Mastering Search Engine Optimization* (3rd ed.).

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. www.aaai.org

Gonalves, T., Silva, C., Quaresma, P., & Vieira, R. (2006). *Analysing part-of-speech for Portuguese Text Classification*. http://www.visl.sdu.dk/

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). *Learning Word Vectors for 157 Languages*. http://arxiv.org/abs/1802.06893

GroupM. (2023). *About WPP and GroupM*. Https://Www.Groupm.Com/Wpp-Groupm/. https://www.groupm.com/wpp-groupm/

Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Rodrigues, J., & Aluisio, S. (2017). *Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks*. http://arxiv.org/abs/1708.06025

Jerri L. Ledford. (2009). *Search Engine Optimisation Bible* (2nd ed.). Wiley.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *60*, 11–21.

Jurafsky, D., & Martin, J. H. (2014). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (2nd ed.).

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. http://arxiv.org/abs/1909.11942

Li, K., Lin, M., Lin, Z., & Xing, B. (2014). Running and chasing - The competition between paid search marketing and search engine optimisation. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 3110–3119. https://doi.org/10.1109/HICSS.2014.640

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimised BERT Pretraining Approach*. http://arxiv.org/abs/1907.11692

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *An Introduction to Information Retrieval*.

Matta, H., Gupta, R., & Agarwal, S. (2020). Search Engine Optimisation in Digital Marketing: Present Scenario and Future Scope. *International Conference on Intelligent Engineering and Management (ICIEM)*.

McInnes, L., Healy, J., & Astels, S. (2017). hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, *2*(11), 205. https://doi.org/10.21105/joss.00205

McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. http://arxiv.org/abs/1802.03426

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

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). *Distributed Representations of Words and Phrases and their Compositionality*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). *Efficient Estimation of Word Representations in Vector Space*. http://arxiv.org/abs/1301.3781

Naseem, U. (2020). *Hybrid Words Representation for the classification of low quality text*.

Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe: Global Vectors for Word Representation*. http://nlp.

Quarteroni, S. (2018). Natural Language Processing for Industry: ELCA's experience. *Informatik-Spektrum*, *41*(2), 105–112. https://doi.org/10.1007/s00287-018-1094-1

Querido, A., Carvalho, R., Rodrigues, J., Garcia, M., Silva, J., Correia, C., Rendeiro, N., Valadas Pereira, R., Campos, M., & Branco, A. (2017). LX-LR4DistSemEval: a collection of language resources for the evaluation of distributional semantic models of Portuguese. *Revista Da Associação Portuguesa de Linguística*, *3*, 265–283. https://doi.org/10.26334/2183-9077/rapln3ano2017a15

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving Language Understanding by Generative Pre-Training*. https://gluebenchmark.com/leaderboard

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language Models are Unsupervised Multitask Learners*. https://github.com/codelucas/newspaper

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. http://arxiv.org/abs/1910.10683

Ricardo .J.G.B., C., Moulavi, D., & Sander, J. (2013). *Density-Based Clustering Based on Hierarchical Density Estimates*.

Santos, R., Rodrigues, J., Branco, A., & Vaz, R. (2021). *Neural Text Categorisation with Transformers for learning Portuguese as a Second Language*.

Souza, F., Nogueira, R., & Lotufo, R. (2020). BERTimbau: Pretrained BERT Models for Brazilian Portuguese. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12319 LNAI*, 403–417. https://doi.org/10.1007/978-3-030-61377-8_28

Stockwell, J. (2011). *Keyword Research Tools*. www.KeywordWorkshop.com

Widia Sembiring, R., Mohamad Zain, J., & Embong, A. (2010). *A Comparative Agglomerative Hierarchical Clustering Method to Cluster Implemented Course* (Vol. 2).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., … Rush, A. M. (2020). *Transformers: State-of-the-Art Natural Language Processing*. https://github.com/huggingface/

Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, *2*(2), 165–193. https://doi.org/10.1007/s40745-015-0040-1

Yalçin, N., & Köse, U. (2010). What is search engine optimisation: SEO? *Procedia - Social and Behavioral Sciences*, *9*, 487–493. https://doi.org/10.1016/j.sbspro.2010.12.185

Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Abrego, G. H., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., & Kurzweil, R. (2019). *Multilingual Universal Sentence Encoder for Semantic Retrieval*. http://arxiv.org/abs/1907.04307

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa