

The Democratic and Popular Republic of Algeria
Ministry of Higher Education and of Scientific Research
20 August 1955 University of Skikda
Faculty of Sciences - Computer Science Department

Master's Thesis
in
Computer Systems

Automatic Extraction of Collocations

Presented by

Djakhdjakha Omar Yasser
Mezghache Abdelkarim

Supervised by

Dr.Laroum Toufik

Juin 2022

Abstract

Collocational knowledge is necessary for language generation. The challenge is that collocations can take on a wide range of shapes. They could involve two, three, or more words, words from various syntactic categories, and more flexible or more strict word arrangements.. This thesis describes the procedures involving in creating a dedicated platform for the linguists and every person who is in a need for collocations, which address the problem of finding the most advantageous word that help him make better sentences for his contexts by filtering large corpora in 13 different language and extract best results.

Keywords

collocation, thesaurus, concordancer, search-engines, web development, devOps.

Table of Contents

Abstract.....	II
Introduction.....	1
Chapter 1: General Definitions.....	3
1.1. What is Collocation ?.....	3
1.2. What is Thesaurus.....	4
1.3. Search Engine.....	5
1.4. Natural Language Processing.....	7
Chapter 2: Functional Specifications.....	12
2.1. Database Design.....	12
2.1.1 Database Schema.....	12
2.1.2 Database denormalization.....	16
2.1.3 Database Partitioning.....	17
2.1.4 Database Index.....	17
2.2. Feeding Process.....	18
2.2.1 Downloading and Cleaning Corpus.....	18
2.2.2 Writing to Database.....	18
2.3. Indexing Process.....	19
2.4. Information Retrieval.....	19
2.5. User interface and User experience.....	20
2.6. Data Analysis.....	20
Chapter 3: Search Engine Optimization.....	22
3.1. The Beginning of SEO.....	22
3.2. What is Search engine Optimization.....	24
3.3. How SEO works.....	26
3.4. Techniques Used in Optimizing the Platform.....	27
3.4.1 Keyword Research.....	27
3.4.2 Link Building.....	30
3.4.3 On-page Optimization.....	32
3.4.4 Semantic Markup.....	37
Chapter 4: Tools and Implementation.....	38
4.1. Programming Languages.....	38
4.1.1 HTML 5.....	38
4.1.2 CSS 3.....	38
4.1.3 Python.....	38
4.1.4 JavaScript.....	38
4.2. Frameworks.....	39
4.2.1 Django.....	39
4.2.2 Spacy.....	40
4.2.3 Vue.js.....	41
4.2.4 Webpack.....	42
4.3. Deployment.....	42
4.3.1 Amazon Web services (AWS).....	42
4.3.2 Git.....	44
4.3.3 GitLab.....	44

4.3.4 Docker.....	45
4.3.5 Docker Compose.....	45
4.3.6 Nginx.....	45
4.3.7 Gunicorn.....	46
4.3.8 Cloudflare.....	46
4.3.9 Sentry.io.....	47
4.4. Implementation.....	48
Conclusion.....	55

Table of Figures

Database Schema of the platform.....	13
Token Table.....	14
Document Table.....	14
N-grams Table.....	15
Summary Table.....	15
Outlier Table.....	16
Search Table.....	16
SpaCy nlp pipeline.....	18
Google Analytics - Funnel.....	21
Google Analytics - Cohort.....	21
Illustration 11: Example of a user search.....	28
Monthly searched volume for adjective pattern using google trends.....	29
All backlinks for INSPIRASSION.....	31
twitter backlinks for INSPIRASSION.....	31
Desktop LCP Score.....	34
Mobile LCP Score.....	34
Sitemap links of all patterns of all languages.....	36
sitemap links of Adjective pattern for English Language.....	36
Generated image in website.....	37
Sentry Performance Monitoring.....	47
Sentry Errors.....	47
Inspirassion - French Home Page.....	48
Inspirassion - Arabic Home Page.....	48
Inspirassion - English Home Page.....	49
Inspirassion - Patterns Used.....	49
Inspirassion - Languages Supported.....	49
Inspirassion - Displays Implemented.....	49
Inspirassion - Words Mode.....	50
Inspirassion - Sentences Mode.....	50
Inspirassion - KWIC Mode.....	51
Inspirassion - Frequency Mode.....	51
Inspirassion - Polarity Mode.....	52
Inspirassion - Advanced Static Mode.....	53
Inspirassion - Advanced Dynamic Mode.....	53
Inspirassion - Versus Advanced Mode.....	54

Introduction

The syntactic and semantic limitations on word choice and word ordering have been the focus of language generation research on lexical choice. However, collocational limitations influence how words can appear in the same phrase. When one word is used in a certain context, it is frequently necessary to use one or more additional words in the same statement. While phrasal lexicons, in which lexical connections are pre-encoded, allow for the handling of some forms of collocations, they also have issues. The compilation of phrasal entries must be done by hand, which is both time consuming and inefficient. Furthermore, phrasal entries have a tendency to catch idiomatic formulations that are fairly strict.

There are several online platforms available that can extract collocations from corpora. However, very few platforms support multi language extraction. In this paper we will present the design of INSPIRASSION, a collocation retrieval platform besides some other features such as thesaurus and concordancer. INSPIRASSION retrieves the collocations from the corpora by taking a word and verify its part-of-speech then display the results with the different contexts from many trusted sources. The user interface of our platform tend to be more concise, simple and attractive for the user.

This work is an attempt of making a software solution that is easily usable and reusable and reliable as much as possible. This software has the mission to help writers, students and linguistics in general to choose words from different contexts that have different meanings for their sentences and aide them to create professional contents.

There's a whole world of collocations to be discovered. Because we begin to acquire words in larger groups or 'chunks'. Learning collocations is essential. Putting these bits of language together results in more fluent linguistic.

The document is composed of five chapters, including this introduction and 4 other chapters that contains the following:

1. in the first chapter We will go into greater information regarding collocation and the platforme

2. In the second chapter we explain how does this platform work and how it is structured.
3. The third chapter talks about optimizing the search engine research such as google, yahoo, bing ... etc
4. The fourth chapter talks about the tools used in building this platform besides the deployment process.

Chapter 1: General Definitions

We shall cover more information on collocation and Thesaurus also a brief about seo and nlp, the main actors of our platform in the first chapter.

1.1. What is Collocation ?

Collocation is defined in a variety of ways. A collocation, according to Firth (1968:182), is "real words in frequent company." "The co-occurrence of two or more lexical items as realizations of structural components within a particular syntactic pattern," according to Cowie . "A noteworthy collocation is one in which the two elements co-occur more often than might be expected on the basis of their individual frequencies and the length of the text under examination," according to Martin et al.. "A collocation is a sequence of words that appears more than once in same form and is grammatically well-structured," according to Kjellmer .

Collocations are "the occurrences of two or more words within a short distance of each other in a text," according to Sinclair (1991:170). "A collocation is an arbitrary and repeated word combination," says Benson (1990:23). A collocation, on the other hand, is defined by Svensén as "a set of words having a common meaning that tend to appear together."

Two methods to the analysis of collocations arise from the definitions above: a frequency-based approach and a semantically-based or phraseological approach .

Collocations are a sort of word combination, according to Cowie and Benson (1989), who use the phraseological method. Collocations are divided into two categories by Benson et al. grammatical collocations and lexical collocations. A grammatical collocation is a phrase that includes a dominating word (noun, adjective, verb) plus a preposition or grammatical structure like an infinitive or clause, as in rely on. Lexical collocations, on the other hand, do not contain prepositions, infinitives, or clauses, as in greatest remorse.

Type of collocation	examples
noun adjective	“religious leader” , “ single person”, “dear friend”
verb adverb	“walk slowly”, “work hard”, “sing sweetly”
verb object	“try your best”, “achieve that goal”, “type a letter”
noun noun	“night time”, “life insurance”, “book magazine”
noun verb	“developed wonderful skill”, “saw the sun”
adjective preposition	“sick of”, “familiar with” , “tired from”
verb preposition	“write to”, “see with”, “responsible for”
noun metaphor	“world is a pair”, “happiness is the sport”, “life is a rose”

1.2. What is Thesaurus

A thesaurus is a collection of terms related to a certain topic (s). It's also known as a vocabulary control tool that helps indexers and end users understand how to utilize phrases and improves the quality of search results. A thesaurus is a database that is used to index and retrieve documents in a certain subject domain. ERIC for education resources, macrothesaurus for economic resources, GESIS thesaurus for social science, R.L. Burton's legal thesaurus, and others are examples of topic areas covered by thesauri. A thesaurus is a crucial tool for organizing information and retrieving it later. The basic function of the thesaurus, according to Aitchison et al. , is information retrieval. Information retrieval is one of the most important aspects of librarianship as it deals with needs of end users of the library.

A thesaurus supplies indexers and end users with several forms of information. A thesaurus is made up of terminologies that are used to describe concepts and their semantic relationships. Preferred terms/descriptors, non-preferred terms/non-descriptors, related terms, narrow (15) terms(NT), broad terms(BT), USE and Used For(UF), and so on are some of these words. To utilize a thesaurus effectively while

indexing texts and designating relevant keywords, one must have a thorough grasp of its relationships. All the relationships employed in a thesaurus have been attempted to be defined.

In simple words thesaurus are synonyms, antonyms, rhymes, related words.

1.3. Search Engine

A search engine is a piece of software that assists individuals in finding information online by utilizing keywords or phrases. When a user types in a search phrase, the search engine examines the contents and keywords it has indexed, then employs algorithms to generate a list of websites, with the most relevant results at the top.

To create search results, each search engine employs a separate set of complicated mathematical calculations. The SERP then displays the results for a specified query. The important characteristics of a web page, such as the page title, content, and keyword density, are used by search engine algorithms to create a ranking for where to display the results on the pages. The algorithms of each search engine are different, therefore a high position on Yahoo! does not ensure a high ranking on Google, and vice versa. To make matters even more confusing, search engine algorithms are not only highly guarded secrets, but they are also regularly modified and revised. This means that the criteria to best optimize a site with must be surmised through observation, as well as trial and error — and not just once, but continuously.

The first internet search engines date back to 1982, and the Knowbot Information Service multi-network user search was first implemented in 1989, far before the launch of the Web in December 1990. [1] Archie, which debuted on September 10, 1990, was the first well-documented search engine that scanned content files, namely FTP files. [2].

The World Wide Web was totally indexed by hand before to September 1993. Tim Berners-Lee compiled a list of web servers, which was housed on the CERN webserver. Although a snapshot of the list from 1992 exists,[3] the central list could no longer keep up as more and more web servers came up. New servers were introduced on the NCSA website under the heading "What's New!".

Archie was the first tool on the Internet for searching content (rather than persons). The word "archive" is spelled without the "v." Alan Emtage, a computer science student at McGill University in Montreal, Quebec, Canada, developed it. The program downloaded directory listings for all files on public anonymous FTP (File Transfer Protocol) sites, creating a searchable database of file names; however, Archie Search Engine did not index the contents of these sites because the amount of data was so small that it could be easily searched manually.

Veronica and Jughead were born as a result of the development of Gopher (invented in 1991 by Mark McCahill at the University of Minnesota). They searched the file names and titles contained in Gopher index systems in the same way Archie did. Veronica (Very Easy Rodent-Oriented Net-wide Index to Computerized Archives) allowing to search the complete Gopher listings for most Gopher menu names. Jughead (Jonzy's Universal Gopher Hierarchy Excavation And Display) was a Gopher server menu information retrieval tool. While the search engine's name "Archie Search Engine" was not a connection to the Archie comic book series, the characters "Veronica" and "Jughead" are from the series, therefore it was a nod to their progenitor.

In the summer of 1993, there was no web search engine, albeit there were various specialized catalogues kept by hand. Oscar Nierstrasz of the University of Geneva created a set of Perl programs that mirrored and changed these sites on a regular basis. W3Catalog, the web's first crude search engine, was introduced on September 2, 1993, based on this.

Matthew Gray, while at MIT, created what is thought to be the first web robot, the Perl-based World Wide Web Wanderer, in June 1993, and used it to create the "Wandex" index. The Wanderer was created with the intention of measuring the extent of the World Wide Web, which it did until late 1995. Aliweb, the second search engine on the web, debuted in November 1993. Aliweb did not utilize a web robot and instead relied on website administrators informing them of the availability of an index file in a certain format for each site.

JumpStation (founded by Jonathon Fletcher in December 1993) employed a web robot to search and index web pages, and a web form as the interface to its query software. As a result, it was the first WWW resource-discovery tool to incorporate the three key elements of a web search engine (crawling, indexing, and searching), as detailed below. Because of the platform's low resources, the crawler's indexing and hence searching were confined to the titles and headers contained in the web pages it encountered.

WebCrawler, which debuted in 1994, was one of the earliest "all text" crawler-based search engines. Unlike its predecessors, it enabled users to search for any term on any webpage, a feature that has subsequently been adopted by all major search engines. It was also the search engine that the general public was familiar with. Lycos (which began at Carnegie Mellon University) was also founded in 1994 and quickly grew into a big commercial venture.

Yahoo! Search was the first widely used search engine on the Internet. Yahoo! Directory, which was created by Jerry Yang and David Filo in January 1994, was the company's initial product. A search tool was added to Yahoo! Directory in 1995, allowing users to search it. [4] It quickly became one of the most popular methods for

users to locate interesting online sites, however its search feature only worked on its web directory, not full-text versions of web pages.

Soon after, a slew of other search engines sprung up, all vying for attention. Magellan, Excite, Infoseek, Inktomi, Northern Light, and AltaVista were among them. Instead of doing a keyword-based search, information searchers might browse the directory.

Robin Li got a US patent for the RankDex site-scoring system in 1996, which he created for search engine results page ranking[5]. It was the first search engine to utilize hyperlinks to assess the quality of the pages it was indexing, two years before Google submitted a patent for a very similar algorithm. [6] Larry Page cited Li's work in some of his PageRank patents in the United States. Li's Rankdex technology was eventually adopted by Robin Li's Baidu search engine, which was introduced in China in 2000.

In 1996, Netscape wanted to grant an exclusive arrangement to a single search engine to be the prominent search engine on Netscape's web browser. Because of the high level of demand, Netscape instead signed arrangements with five large search engines, each of which would be included on the Netscape search engine website for \$5 million per year. Yahoo!, Magellan, Lycos, Infoseek, and Excite were the five engines. [7] .

In 1998, Google acquired the notion of selling search phrases from goto.com, a tiny search engine business. The search engine firm, which had been floundering, became one of the most profitable on the Internet as a result of this decision. [8]

In the late 1990s, search engines were also recognized as some of the brightest lights in the Internet investing frenzy.[9]

Several firms made outstanding debuts on the market, achieving record highs during their initial public offerings. Some companies, such as Northern Light, have shut down their public search engines and are instead offering enterprise-only editions. The dot-com bubble, a speculative market boom that peaked in March 2000, engulfed several search engine firms.

1.4. Natural Language Processing

Natural language processing (NLP) is the capacity of a computer software to interpret spoken and written human language, often known as natural language. NLP employs

artificial intelligence to accept real-world data, interpret it, and make sense of it in a way that a computer can comprehend. Computers have programs to read and microphones to gather sounds, much as people have diverse sensors such as ears to hear and eyes to see. And, just as people have a brain to process their inputs, computers have a software to do the same. The input is transformed to computer-readable code at some point throughout the processing.

Natural language processing is divided into two stages: data preprocessing and algorithm development.

Data preprocessing involves preparing and "cleaning" text data for machines to be able to analyze it. Preprocessing puts data in workable form and highlights features in the text that an algorithm can work with.

Many early language-processing systems were created using symbolic approaches, such as composing grammars or inventing heuristic stemming rules, which required hand-coding of a set of rules and a dictionary lookup:[13][14].

Recent machine-learning algorithms-based systems provide a number of benefits over hand-written rules:

- Machine learning learning techniques automatically focus on the most common scenarios, however when defining rules by hand, it's sometimes unclear where the effort should be focused.
- Automatic learning processes can employ statistical inference techniques to create models that are resistant to both novel and erroneous input (for example, input including words or structures that have never been seen before) (e.g. with misspelled words or words accidentally omitted). Handling such input gracefully using handwritten rules, or constructing systems of handwritten rules that make soft judgments in general, is highly difficult, error-prone, and time-consuming.
- The accuracy of systems based on automatically learning the rules can be improved simply by providing more input data. Handwritten rule systems, on the other hand, can only be improved by increasing the complexity of the rules, which is a far more challenging process. In particular, the complexity of systems based on handwritten rules has a limit beyond which they become more unmanageable. Creating additional data to feed into machine-learning systems, on the other hand, just necessitates an increase in the number of man-hours spent, with little to no increase in the complexity of the annotation process.

Statistical Methods

Much natural language processing research has depended largely on machine learning since the so-called "statistical revolution"[15][16] in the late 1980s and mid-1990s. Instead, the machine-learning paradigm proposes that large corpora (plural form of corpus) of typical real-world examples be analyzed using statistical inference to automatically learn such rules through the analysis of large corpora (plural form of corpus, is a set of documents, possibly with human or computer annotations).

Natural-language processing problems have been subjected to a variety of machine-learning methods. These algorithms take a high number of "features" created from the raw data as input. However, statistical models, which make soft, probabilistic judgements by assigning real-valued weights to each input attribute, have become increasingly popular (complex-valued embeddings,[17] and neural networks in general have also been proposed, for e.g. speech[18]). When used as part of a broader system, such models offer the benefit of expressing the relative certainty of many different potential responses rather than just one, resulting in more trustworthy conclusions.

Some of the first machine learning algorithms, such as decision trees, generated systems of hard if-then rules that were comparable to handwritten rules. Part-of-speech tagging, on the other hand, brought hidden Markov models to natural language processing, and research has increasingly concentrated on statistical models, which make soft, probabilistic judgments by assigning real-valued weights to the characteristics that make up the input data. Cache language models, which are now used by many speech recognition systems, are an example of statistical models. When given new input, especially information that contains mistakes (as is quite typical for real-world data), such models are more resilient, and when incorporated into a bigger system with several subtasks, they offer more dependable results.

Statistical approaches in NLP research have been mostly superseded by neural networks since the neural revolution. They are, nonetheless, still significant in situations where statistical interpretability and openness are necessary.

Neural networks

Statistical approaches have the disadvantage of necessitating extensive feature engineering. As a result, the discipline has mainly abandoned statistical approaches in favor of neural networks for machine learning since 2015[19]. Using word embeddings to capture semantic features of words and increasing end-to-end learning of a higher-level task (e.g., question responding) rather than depending on a pipeline of discrete intermediary tasks are two popular strategies (e.g., part-of-speech tagging

and dependency parsing). This transition has resulted in significant changes in how NLP systems are constructed in some domains, to the point that deep neural network-based techniques may be considered a new paradigm distinct from statistical natural language processing.

For example, the term neural machine translation (NMT) emphasizes that deep learning-based approaches to machine translation directly learn sequence-to-sequence transformations, eliminating the need for intermediate steps like word alignment and language modeling that were previously required in statistical machine translation (SMT).

The following is a list of some of the most applied natural language processing tasks. In our platform

Morphological analysis

- **Lemmatization:** the task of eliminating just inflectional ends and returning a word's base dictionary form, also known as a lemma. Another method for converting words to their normalized form is lemmatization. In this example, though, the transformation makes use of a dictionary to map words to their proper forms. [21]
- **Morphological segmentation:** distinguish words into separate morphemes and classify the morphemes. The difficulty of this task is mostly determined by the complexity of the language's morphology (i.e., the structure of words). Because English morphology is very easy, particularly inflectional morphology, it is often possible to neglect this work altogether and just represent all potential forms of a word (for example, "open, opens, opened, opening") as independent words. Such a method is not practical in languages like Turkish or Meitei,[22] a heavily agglutinated Indian language, because each dictionary entry contains hundreds of alternative word forms.
- **Part-of-speech tagging:** determine the part of speech (POS) for each word in a sentence. Many words, particularly common ones, may be used in several ways. "Book" can be a noun ("the book on the table") or a verb ("to book a flight"); "set" can be a noun, verb, or adjective; and "out" can be one of at least five parts of speech.
- **Stemming:** The process of reducing inflected (or sometimes derived) words to a base form (e.g., "close" will be the root for "closed", "closing", "close", "closer" etc.). Stemming yields similar results as lemmatization, but does so on grounds of rules, not a dictionary.

Conclusion : In this chapter, we cover details concerning collocation and thesaurus, as well as a quick overview of SEO and NLP.

Chapter 2: Functional Specifications

In this chapter introduces some core concepts and principles that we will describe the structure and operation of this platform.

2.1. Database Design

A poorly constructed database is more likely to produce mistakes, which can lead to poor judgments. A faulty database design can eventually self-correct: businesses with badly built databases frequently fail because their management lack timely (or even correct) information, allowing the bad database architecture to take control.

If the database is constructed to take use of the available power, the availability of a DBMS allows for significantly more complex usage of the data resources.

The types of data structures produced inside the database, as well as the breadth of their interactions, have a significant impact in defining the DBMS's effectiveness. As a result, database design has become a critical activity in the database world.

When we employ models, we can make database design lot easier. A database model is a set of logical constructs that are used to describe the database's data structure and data connections, or simplified abstractions of real-world events or situations. If the models aren't logically solid, the database designs that result from them won't be able to deliver on the database system's promise of effective information taken from a fast database. *"Good models yield good database design that are the basis for good applications"*.

2.1.1 Database Schema

The relational model's purpose is to provide a declarative method for specifying data and queries: users state directly what information the database contains and what information they want from it, and the database management system software handles describing data structures for storing the data and retrieval procedures for answering queries.

The SQL data definition and query language is used by most relational databases; these systems implement what may be considered an engineering approximation to

the relational paradigm. A predicate variable relates to a table in a SQL database schema; a table's contents to a relation; key constraints, other constraints, and SQL queries to predicates. SQL databases, on the other hand, stray from the relational model in a number of ways, and Codd was adamant about not compromising the basic ideas.

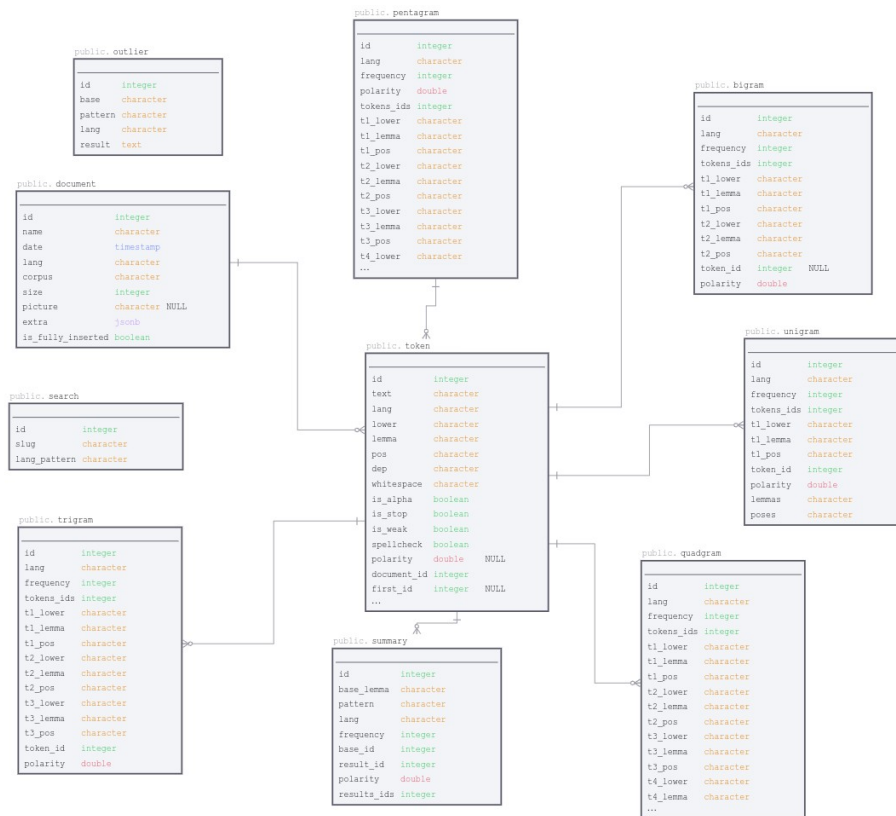


Illustration 1: Database Schema of the platform

The illustration above covers all the tables used construct the platform which is separated into main model that contains all tokens , than it will be denormalized to help us avoid costly joins in our database and tune the performance of the system to support time-critical operations.

Token Model

Illustration 2: Token Table

```
public. token
```

id	integer	
text	character	
lang	character	
lower	character	
lemma	character	
pos	character	
dep	character	
whitespace	character	
is_alpha	boolean	
is_stop	boolean	
is_weak	boolean	
spellcheck	boolean	
polarity	double	NULL
document_id	integer	
first_id	integer	NULL
...		

On this table we will insert all the words, punctuation, white space , symbols, in other words we will transfer text into a set of meaningful segments.

Document Model

This model contains all files to read the text from, which have the language of every file, name of the file, timestamp, language of written text, number of characters and sentences, which corpus belong to.

```
public. document
```

id	integer	
name	character	
date	timestamp	
lang	character	
corpus	character	
size	integer	
picture	character	NULL
extra	jsonb	
is_fully_inserted	boolean	

Illustration 3: Document Table

N-Gram Model

In this models we will write to the table a contiguous sequences of 1,2,3,4,5 items from the given sample of text, this table help us to fetch the most probable sentences that contains the N words (grams) on its contexts.

public. trigram	
id	integer
lang	character
frequency	integer
tokens_ids	integer
t1_lower	character
t1_lemma	character
t1_pos	character
t2_lower	character
t2_lemma	character
t2_pos	character
t3_lower	character
t3_lemma	character
t3_pos	character
token_id	integer
polarity	double

public. bigram	
id	integer
lang	character
frequency	integer
tokens_ids	integer
t1_lower	character
t1_lemma	character
t1_pos	character
t2_lower	character
t2_lemma	character
t2_pos	character
token_id	integer NULL
polarity	double

public. unigram	
id	integer
lang	character
frequency	integer
tokens_ids	integer
t1_lower	character
t1_lemma	character
t1_pos	character
token_id	integer
polarity	double
lemmas	character
poses	character

public. pentagram	
id	integer
lang	character
frequency	integer
polarity	double
tokens_ids	integer
t1_lower	character
t1_lemma	character
t1_pos	character
t2_lower	character
t2_lemma	character
t2_pos	character
t3_lower	character
t3_lemma	character
t3_pos	character
t4_lower	character
...	

public. quadgram	
id	integer
lang	character
frequency	integer
tokens_ids	integer
t1_lower	character
t1_lemma	character
t1_pos	character
t2_lower	character
t2_lemma	character
t2_pos	character
t3_lower	character
t3_lemma	character
t3_pos	character
t4_lower	character
t4_lemma	character
...	

Illustration 4: N-grams Table

Summary Model

In this table we tried to summarize all occurrences of the results into a just one result that contains base word ,its result and all other tokens that have the same result as the one searching for.

public. summary	
id	integer
base_lemma	character
pattern	character
lang	character
frequency	integer
base_id	integer
result_id	integer
polarity	double
results_ids	integer

Illustration 5: Summary Table

Outlier Model

Is the model where we store all the thesaurus (synonyms, antonyms, rhymes, related) scraped from other websites besides every other element fetched from an external source.

```
public.outlier
```

id	integer
base	character
pattern	character
lang	character
result	text

Illustration 6: Outlier Table

Search Model

The search table is responsible of collecting every query made by the user , partitioned by patterns of every language.

```
public.search
```

id	integer
slug	character
lang_pattern	character

2.1.2 Database denormalization

Denormalization is a performance-enhancing method that is applied to a previously normalized database. Denormalization is the practice of adding duplicate copies of data or grouping data in order to increase the read performance of a database at the loss of some write speed. It's usually motivated by performance or scalability in relational database software that has to do a lot of read operations. Denormalization differs from unnormalization in that the benefits of denormalization can only be fully realized on a normalized data model.

Token → Summary And Token → N-grams

The main idea of denormalizing the Token table into summary or n-grams was to store data in separate tables and attempt to minimize redundant data. We will have only one copy of each record. This helped us in retrieving data faster and we do only few joins, also we can make simple queries to get the data from the table.

2.1.3 Database Partitioning

With huge data being stored in our database, performance and scaling are two main factors that are affected. As table size increases with data load, more data scanning, swapping pages to memory, and other table operation costs also increase. So we decided that partitioning may be a good solution, as It can help divide the large tables we have into smaller ones and thus reduce table scans and memory swap problems, which ultimately increases performance.

We saw that the horizontal partitioning by language value, did help us provide good control to users besides the high flexibility by increasing the query performance compared to a single large table.

Most benefits of partitioning can be enjoyed when a single table is not able to provide them. So we can say that the large data that we have is going to be written on a single table at some point, our users need partitioning. With good planning and taking all factors into consideration, table partitioning did give us a great performance boost and scaled our PostgreSQL to larger datasets.

2.1.4 Database Index

A database index is a data structure that speeds up data retrieval operations on a database table in exchange for more writes and storage space. When the wanted table is consulted, indexes are utilized to rapidly identify data without having to scan every row in the table. Indexes can be built utilizing one or more columns from a database table, allowing for quick random lookups as well as efficient access to sorted items.

Our n-grams and summary tables are indexed on the token lemma ,token lower, token part-of-speech fields because these fields are the ones that we use often for the querying of our results, and by this we will reduce the time that the driver spends searching for a particular result from millions of lemmas, lower texts and part-of-speeches.

2.2. Feeding Process

At this point we have an empty database so we need to fill it with segments of texts so ,by way of explanation we download files from certain corpus then we clean the files after that we feed the table `Token` after tokenizing the sentences red from the cleaned files. So that we can later make use of the entered data to construct sets of results so that user can find his wish.

2.2.1 Downloading and Cleaning Corpus

We have two main corpus Gutenberg Project and Leipzig Library where they contains multi language files that have millions word collection of samples of written language from a wide range of sources that includes extracts from international news papers, specialist periodicals and journals for all ages and interests, academic books, university essays and many other kinds of text.

The download procedure is made by firing the command `downloadcorpus` that fetch the files using a python HTTP library called `Requests` after that `cleancorpus` command will be executed in goal of extracting the the compressed folder and then cleaning the files from impurities and make them ready to be fed.

2.2.2 Writing to Database

After making the files ready to use, it comes the procedure where entering data to `Token` table by first reading chunks from the cleaned files, then loading the natural language processing model of the language from `spaCy` to use it to tokenize the texts and then call each component on the `Doc`, in order. It then returns the processed `Doc` where we will bulk create them on the table.

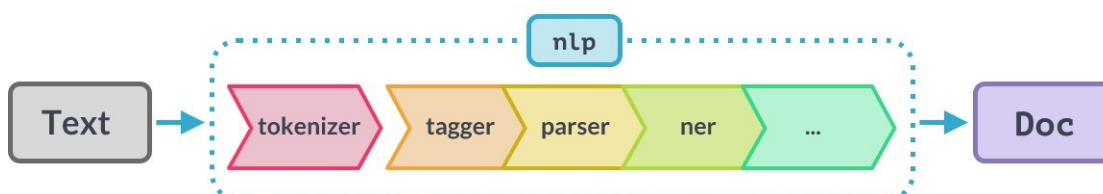


Illustration 8: SpaCy nlp pipeline

2.3. Indexing Process

The indexing process is where we summarize then gramize the tokens to group data and eliminate redundant data to increase the performance of reading from database.

Summarize section is the part where we group the tokens by the input base lemma and the result lemma that have a collocational relation, so that we have an example of every group of results that contains the input word and its collocation. After that ,it comes the step where we gramize the tokens so that we fill all 5 grams tables by

n-gram of every table, first table called **unigram** we create only one gram ,second table **bigram** we create the sequence of each 2 words and so on until the 5th gram.

2.4. Information Retrieval

This part is responsible of receiving the input word then calculate the collocations and send the result back to user.

Django framework work with an architecture called the Model-View-Template ,the user write his word and then search for its collocations. So what happens ?

First of all when the user requests for a resource, Django works as a controller and check the available resource in URL, if URL maps, the result view will be called and create a Query object then calculate for the collocation by executing a function that fetch for the results from the model then rearrange them by creating results objects for each retrieved object. At this point when the calculation is finished the Result View will render them to the template to be displayed to the user by certain criterion that can be changed.

If a user searches for a type of collocations, a `calculate_collocation` function will filter the lemmas that have the wanted pattern from the table summary and responds back to the request. Although when a user search in the advanced or the phrase mode with its three types static, dynamic and versus, the `calculate_phrase` function will fetch the n-gram sequence written and return the matched contexts.

However if a user want any type of thesaurus a `calculate_thesaurus` function will calculate the results in a different way, which is scarping the results from other external resources and check if the words obtained are in our database, then renders what does exist. Last but not least when a user search for a wordplay pattern another function will be executed with a regular expressions lookups to find the matched words.

2.5. User interface and User experience

The point of user interface and user experience (ux/ui) is to create a product that will be both visually appealing and highly pleasant to interact with.

A widespread fallacy is that UI and UX are interchangeable and that they are one field, although this is not totally accurate. They are distinct disciplines that concentrate on various aspects of a user's interaction with a digital product.

However, they are so closely related that they have largely fused into one profession. Before we get into what the UI/UX design process entails and why it matters to us, let's have a look at the primary distinctions between UX and UI.

We made sure the user's interaction with our product is fast, easy and efficient as possible. We did map out the user journey and strategise to solve user's pain points and meet their needs as much as possible.

Ultimately, our product is meant to be pleasant-looking, functional and give users a fantastic experience in terms of performance and navigation.

Our interface is available in 13 languages and that because of the `i18n django` module . Translation depends on the target language, and formatting usually depends on the target country. This information is provided by browsers in the `Accept-Language` header.

2.6. Data Analysis

Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.

We treat our data as a business asset and actively looking for ways to turn it into a competitive advantage. Success in analyzing data depends on its quality. We collect data by tracking the users behavior on the platform by implementing Google Analytic API and configure it to work with every action we want to monitor.

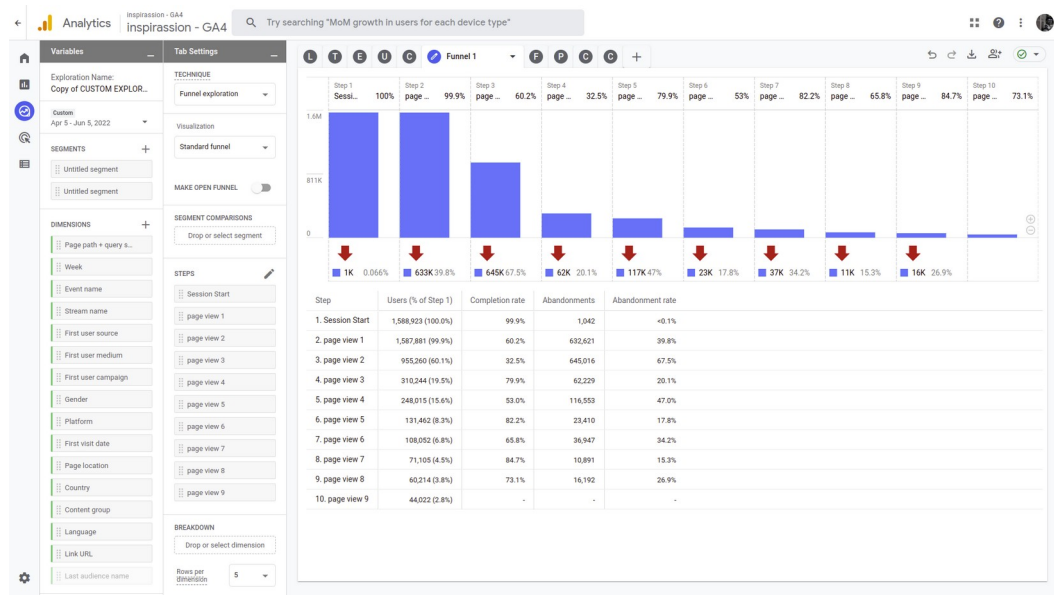


Illustration 9: Google Analytics - Funnel

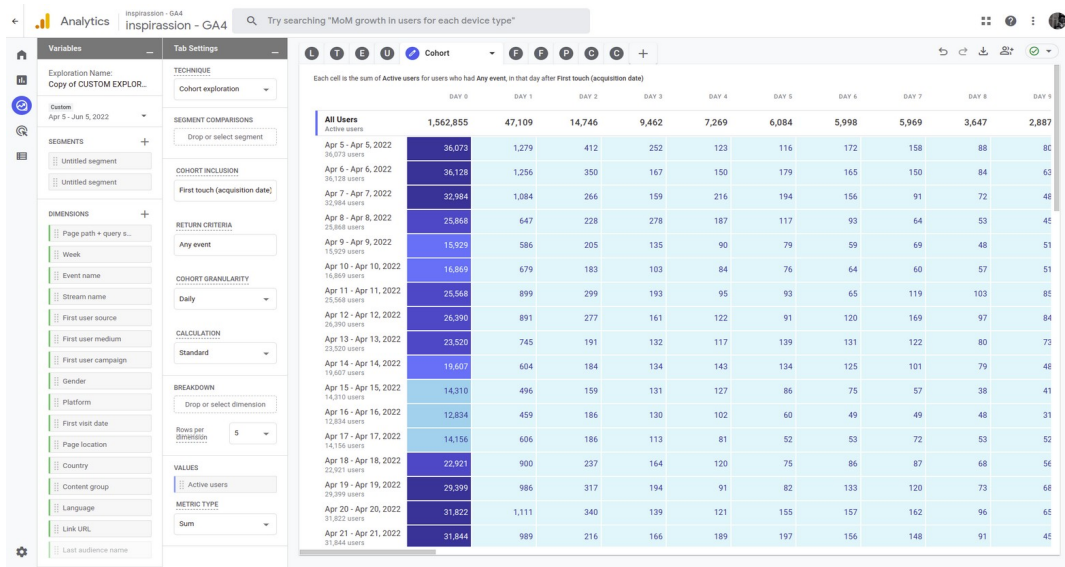


Illustration 10: Google Analytics - Cohort

Conclusion : In this chapter, we pass the groundwork for several fundamental ideas and principles that will guide how we define the layout and functionality of this platform.

Chapter 3: Search Engine Optimization

The third chapter discusses how to make Google and other search engines more effective.

3.1. The Beginning of SEO

Webmasters and content providers began optimizing websites for search engines in the mid-1990s, as the first search engines were cataloging the early Web. Initially, all webmasters only needed to submit the address of a page, or URL, to the various engines which would send a web crawler to crawl that page, extract links to other pages from it, and return information found on the page to be indexed.

The process involves a search engine spider downloading a page and storing it on the search engine's own server. A second program, known as an indexer, extracts information about the page, such as the words it contains, where they are located, and any weight for specific words, as well as all links the page contains. All of this information is then placed into a scheduler for crawling at a later date.

The benefit of a high position and exposure in search engine results was recognized by website owners,[10] offering an opportunity for both white hat and black hat SEO practitioners. According to Danny Sullivan, an industry researcher, the term "search engine optimization" first appeared in 1997. Bruce Clay, according to Sullivan, was one of the first to popularize the phrase.[11]

In engines like ALIWEB, early versions of search algorithms depended on information given by webmasters, such as the keyword meta tag or index files. Meta tags serve as a pointer to the content of each page. The webmaster's choice of keywords in the meta tag might possibly be an incorrect depiction of the site's real content, hence using metadata to index pages was proven to be less than dependable. Pages might be miscategorized in unrelated searches due to faulty data in meta tags, such as those that were not exact, full, or erroneously attributed. Web content providers also manipulated some attributes within the HTML source of a page in an attempt to rank well in search engines.[12] By 1997, search engine designers

recognized that webmasters were making efforts to rank well in their search engine, and that some webmasters were even manipulating their rankings in search results by stuffing pages with excessive or irrelevant keywords. Early search engines, such as Altavista and Infoseek, adjusted their algorithms to prevent webmasters from manipulating rankings.

Early search engines were vulnerable to misuse and ranking manipulation since they relied significantly on characteristics like keyword density, which were solely under the control of the website. Search engines had to adjust to guarantee that their results pages displayed the most relevant search results, rather than unrelated pages loaded with multiple keywords by unethical webmasters, in order to deliver better results to their users. This entailed shifting away from a significant dependence on phrase density and toward a more comprehensive approach to semantic signal assessment. Because a search engine's success and popularity are driven by its ability to deliver the most relevant results for any given search, low quality or irrelevant search results may cause users to seek out alternative search engines. In response, search engines created more complicated ranking algorithms that took into consideration other characteristics that were more difficult for webmasters to influence.

Companies that employ overly aggressive techniques can get their client websites banned from the search results. In 2005, the Wall Street Journal reported on a company, Traffic Power, which allegedly used high-risk techniques and failed to disclose those risks to its clients.[13] Wired magazine reported that the same company sued blogger and SEO Aaron Wall for writing about the ban.[14] Google's Matt Cutts later confirmed that Google did in fact ban Traffic Power and some of its clients.

Some search engines have extended out to the SEO community, and they are frequent sponsors and speakers at SEO conferences, webchats, and seminars. Major search engines give website optimization advice and instructions. [15] [16]Google offers a Sitemaps software that helps webmasters figure out whether Google is having issues indexing their site and also gives data on Google traffic. Bing Webmaster Tools allows webmasters to upload sitemaps and web feeds, as well as assess the "crawl rate" and track the status of web pages in the index.

In 2015, it was reported that Google was developing and promoting mobile search as a key feature within future products. In response, many brands began to take a different approach to their Internet marketing strategies.[17]

To lessen the influence of link manipulation, search engines have included a wide range of unknown criteria into their ranking algorithms by 2004. According to Saul Hansell of the New York Times, Google ranks websites using more than 200 distinct signals. [18] Google, Bing, and Yahoo, the three most popular search engines, do not reveal the methods they employ to rank pages. Some SEO experts have researched various ways to search engine optimization and given their personal perspectives.

[19] Patents relating to search engines can give insight into how search engines work.
[20] Google started tailoring search results for each user in 2005. Google customized results for logged-in users based on their past search history.

Google launched a battle against sponsored connections that pass PageRank in 2007.
[21] Google said on June 15, 2009 that they have used the nofollow feature on links to lessen the impact of PageRank sculpting. To prohibit SEO service providers from exploiting nofollow for PageRank sculpting, Matt Cutts, a well-known Google software developer, said that Google Bot will no longer consider any nofollow links the same way. [22] The use of nofollow as a result of this modification caused PageRank to vanish. To prevent this, SEO developers devised alternate strategies that allow PageRank sculpting by replacing nofollowed tags with obfuscated JavaScript. Several alternatives have also been proposed, including the use of iframes, Flash, and JavaScript. [23]

Google launched the Panda update in February 2011, which penalizes pages that duplicate material from other websites and sources. Historically, websites have copied information from one another and benefitted from this behavior in terms of search engine results. However, Google has adopted a new method that penalizes websites with duplicate material. [24] The Google Penguin update from 2012 sought to punish websites that employed search engine manipulation tactics to enhance their results. [25] Although Google Penguin is advertised as a web spam-fighting algorithm, it really concentrates on spammy links [26] by assessing the quality of the sites from which the links originate.

Google stated in October 2019 that it will begin using BERT (Bidirectional Encoder Representations from Transformers) models for English language search queries in the United States. BERT was Google's next attempt to improve their natural language processing, this time to better comprehend their consumers' search queries. [27] In terms of search engine optimization, BERT aimed to make it easier for users to find relevant information and improve the quality of traffic to websites that rank high in the SERPs.

3.2. What is Search engine Optimization

The art and science of getting sites to rank better in search engines like Google, is known as search engine optimization (SEO). Because search is one of the most common methods for individuals to get information online, ranking better in search engines can result in more visitors to a website.

Paid advertising frequently appear at the top of the results page in Google and other search engines, followed by ordinary results, or what search marketers refer to as

"organic search results." To distinguish organic search traffic from sponsored search traffic, SEO traffic is typically referred to as "organic search traffic." Paid search is also known as search engine marketing (SEM) or pay-per-click advertising (PPC).

Because search is one of the major ways that consumers traverse the web, SEO is an important aspect of online marketing. Search results are provided in an ordered list, and the higher a site can rank on that list, the more visitors it will receive. For example, the number one result will receive 40-60% of overall traffic for a normal search query, whereas the second and third results will receive much less traffic. Only a few percent of searchers go past the first page of results. As a result, even little improvements in search engine rankings can result in increased visitors and, ultimately, business for a website.

Crawlers are used by major search engines like Google, Bing, and Yahoo! to discover pages for their algorithmic search results. Sites that are linked from other search engine indexed pages are located automatically and do not need to be uploaded. Manual submission and human editorial review were necessary for the Yahoo! Directory and DMOZ, two prominent directories that closed in 2014 and 2017, respectively. [28] In addition to their URL submission console, Google offers Google Search Console, where an XML Sitemap feed may be developed and uploaded for free to guarantee that all pages are found, especially those that are not discoverable by automatically following links[29]. [30]Yahoo! used to provide a paid submission service that guaranteed crawling in exchange for a payment per click; however, this service was phased off in 2009.

When crawling a site, search engine crawlers may consider a variety of variables. Search engines do not index every page. The distance between pages and the site's root directory may also influence whether or not they get crawled.[31]

3.3. How SEO works

For each given query, search engines like Google utilize an algorithm or set of rules to choose which sites to show. These algorithms have become incredibly complicated, taking into consideration hundreds, if not thousands, of distinct ranking parameters to decide the SERP ranks. However, search engines use three key indicators to establish a site's quality and where it should rank:

- **Links** : Links from other websites are extremely important in determining a site's rating in Google and other search engines. The reason for this is that a link from another website might be interpreted as a vote of quality, since website owners are reluctant to connect to low-quality sites. Sites that receive links from a large number of other sites gain authority (referred to by Google as "Page`Rank") in the eyes of search engines, especially if the sites linked to them are also authoritative.
- **Content** : In addition to looking at links, search engines examine a webpage's content to see if it is relevant to a specific search query. A big component of SEO is writing content that is tailored to the keywords that users of search engines are looking for.
- **Page structure** : is the third and final fundamental component of SEO. Because webpages are written in HTML, the structure of the HTML code might affect how a search engine evaluates a page. Site owners may boost their SEO by include important keywords in the title, URL, and headers of their pages, as well as ensuring that their site is crawlable.

The search engine optimization process involves optimizing each of these core components of search engine algorithms in order to rank higher in the search results.

3.4. Techniques Used in Optimizing the Platform

3.4.1 Keyword Research

Keyword research is the act of locating and evaluating search phrases that people type into search engines with the intention of utilizing the information for a specific purpose, such as SEO or general marketing. Keyword research may reveal queries to target, as well as their popularity, ranking difficulties, and more.

Keyword Research provides to us a lot of information about what our target audience is looking for on Google. The knowledge we gain from these genuine search phrases increased our content strategy and our overall marketing plan.

When performing internet research, people employ keywords to identify solutions. When a result, if our content succeeds in getting in front of our audience as they do searches, we will see a potential growth in the page view per user. As a result, we should concentrate on such searches.

In addition, in the inbound methodology, we don't create content around what we want to tell people; we are creating content around what people want to discover. In other words, our audience is coming to us.

The process begins with keyword research.

We did use some tools to find the keywords people are searching for, and then improve the quality of the content that is specifically designed to solve those searches, answer those questions, and provide solutions to the issues the searches have.

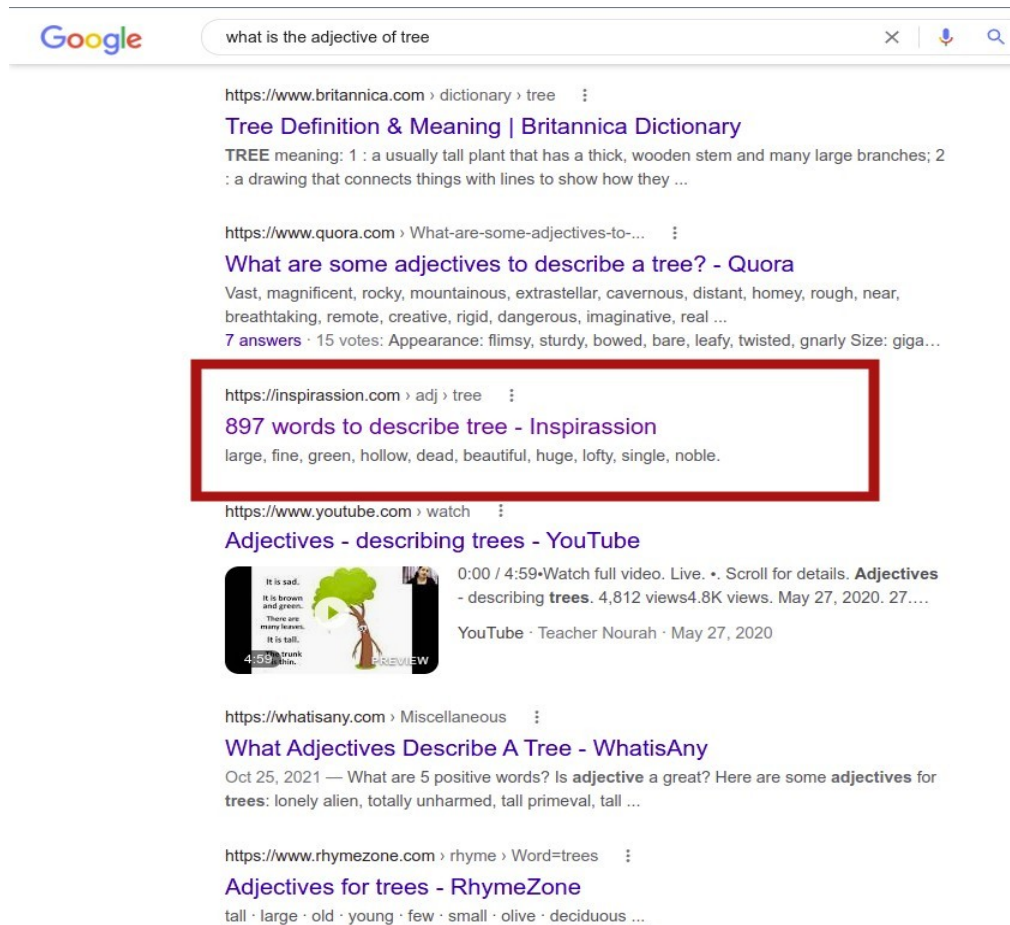


Illustration 11: Illustration 11: Example of a user search

We did pay attention to three main elements when we were searching for the most suitable and conducting keyword research.

1. Relevance: Google uses relevancy to rank content. This is when the term "search intent" comes into play. Because our content matches the demands of the most linguistically proficient writers and searchers, it helped to rank for that term. Furthermore, our materials need many future improvements to be the most relevant resource for the inquiry.

2. Authority: Google will give authoritative sources a higher weighting. That means we must do everything we can to establish ourself as an authoritative

source by adding useful, informative material to our site and promoting it to get social signals and backlinks. Unless our material is extraordinary, we have a decreased chance of ranking if we are not viewed as an authority in the industry, or if a keyword's SERPs are stacked with heavy sources we can't compete with (like Forbes or The Mayo Clinic).

3. Volume: If no one searches for a phrase on the first page, our site will not receive any traffic. As a result, we guarantee to select the most favorable term for each search title for each service offered by our platform. The number of times the term is searched per month across all audiences is indicated by MSV

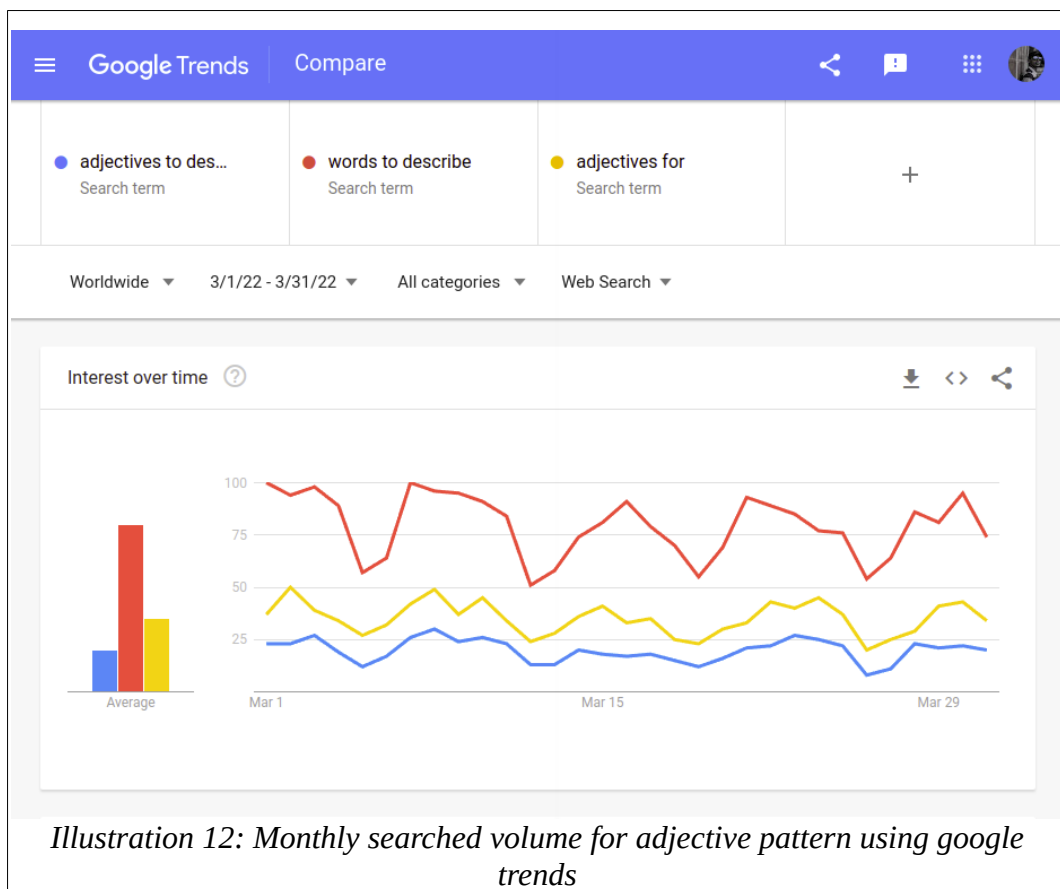


Illustration 12: Monthly searched volume for adjective pattern using google trends

(monthly search volume).

3.4.2 Link Building

The activity of obtaining links to our website from other websites is known as link building. These links are referred to as backlinks in SEO. Obtaining backlinks from high-quality websites can help our site gain authority and rank better in search engine results pages (SERPs).

According to Google, "knowing if other significant websites link or refer to the material" is a crucial aspect of determining which pages are relevant and trustworthy.

As a result, link building should be considered while developing an SEO strategy.

Let's say we have a fantastic piece of content, but our website is still relatively young and hasn't yet established any authority. Obtaining connections from other reputable websites is an excellent strategy to help Google perceive our site as more authority.

Backlinks are essentially endorsements of our website.

Because sites with more backlinks tend to rank better, it's critical to:

1. provide high-quality content that's worth linking to, and
2. employ link-building tactics to assist attract more backlinks.

Because link building takes time and work, we started as soon as we possibly could to gain an advantage over our competitors and provide significant long-term benefits to for INSPIRASSION.

When planning our link building strategy, we did use:

1. **Natural links From Other Blogs or Sites:** Readers that come upon our website naturally provide a link on their blog or website without being asked. We don't have to ask other webmasters or users for a backlink using this link type.

Natural links are not found in sponsored or paid material, and they are also not tracked. They are frequently included in the material of another blogger or webpage in order to bring value to our viewers.

This implies that natural links are more likely to appear in videos, blog posts, photos, and other product listings on a website.

Check the figures below taken from Google Search Console:

Illustration 13: All backlinks for INSPIRASSION

Total external links ⓘ
15,413

Top linking sites

Contains Filter by Site 1 ☰
DONE

Site	Linking pages	Target pages
blogspot.com	1,726	48
sigurros.de	1,085	693
olevanolomellina.info	1,064	931
indonesiakuno.com	728	682
behzatsenikalbimegomdum.com	681	613
powiat-pucki.info	503	411
wildhorses-wa.com	417	345
digitech-group.cloud	381	280
thailandanthem.com	378	360

Illustration 14: twitter backlinks for INSPIRASSION

Total external links ⓘ
15,413

Top linking sites

Filter by Site
Contains **twi** 1 ☰
DONE

Site	Linking pages	Target pages
twitter.com	5	1
milton-twin.com	2	2
investwikipw.com	1	1

Rows per page: 100 1-3 of 3 < >

3.4.3 On-page Optimization

In addition to off-page factors such as links, improving the actual structure of the page can have tremendous benefits for SEO, and is a factor that is entirely in the control of the webmaster. Common on-page optimization techniques include optimizing the URL of the page to incorporate keywords, updating the title tag of the page to use relevant search terms, and using the alt attribute to describe images. Updating a page's meta tags (such as the meta description tag) can also be beneficial-- these tags don't have a direct impact on search rankings, but can increase click-through rate from the SERPs.

There is no standard, universally-recognized workflow for onpage optimization. However, analysis and measures for implementation should be as comprehensive as possible, to ensure that every opportunity is exploited for improving search engine rankings (or other KPIs).

The following list attempts to cover the majority of the most common elements and techniques used in INSPIRASSION for on-page optimization:

Technical optimization

- 1. Server Speed :** Because search engines include website load times as part of their ranking algorithm, improving server response times is an important aspect of onpage SEO.
- 2. Source Code:** Improved website speed can be aided by efficient source code. To make it easy for the Google bot to index the site, unnecessary functionalities or code portions can typically be deleted or other aspects combined.

Content

- 1. Structural text elements:** This includes using paragraphs or bulleted lists, h- header tags, and bolding or italicizing specific text components or phrases.
- 2. Graphics:** All photos may be optimized as significant content pieces. They can improve the content's relevancy, and well-optimized photos can rank in Google's image search on their own. At the same time, they may make a

website more appealing to visitors. Users will spend more time on our site if we have appealing picture galleries. Picture file names are one aspect of image optimization.

- 3. Meta-tags:** Onpage SEO includes meta titles, which are vital for rankings, and meta descriptions, which are an indirect aspect that affects the CTR (Click-Through Rate) on search engine results pages. Even though consumers cannot see them right away, they are still regarded part of the content since they should be optimized alongside the texts and images. This helps to guarantee that the keywords and themes addressed in the text and those used in the meta tags are closely related.

Core Web Vitals

Core Web Vitals are the subset of Web Vitals that apply to all web pages, should be measured by all site owners, and will be surfaced across all Google tools. Each of the Core Web Vitals represents a distinct facet of the user experience, is measurable in the field, and reflects the real-world experience of a critical user-centric outcome.

The metrics that make up Core Web Vitals will evolve over time. The current set for 2020 focuses on three aspects of the user experience—*loading*, *interactivity*, and *visual stability*—and includes the following metrics (and their respective thresholds):

- **Largest Contentful Pain (LCP):** LCP should occur within 2.5 seconds of the page begins loaded to give a pleasant user experience.
- **First Input Delay (FID):** measures *interactivity*. To provide a good user experience, pages should have a FID of 100 milliseconds or less.
- **Cumulative Layout Shift:** measures *visual stability*. To provide a good user experience, pages should maintain a CLS of 0.1 or less.

The 75th percentile of page loads, divided across mobile and desktop devices, is a useful threshold to measure for each of the above metrics to verify we're achieving the

suggested target for most of our consumers.

See the results of the INSPIRASSION home page :

https://inspirassion.com/en/

Analyser

Mobile

Bureau

Découvrez l'expérience de vos utilisateurs

Cette URL Origine

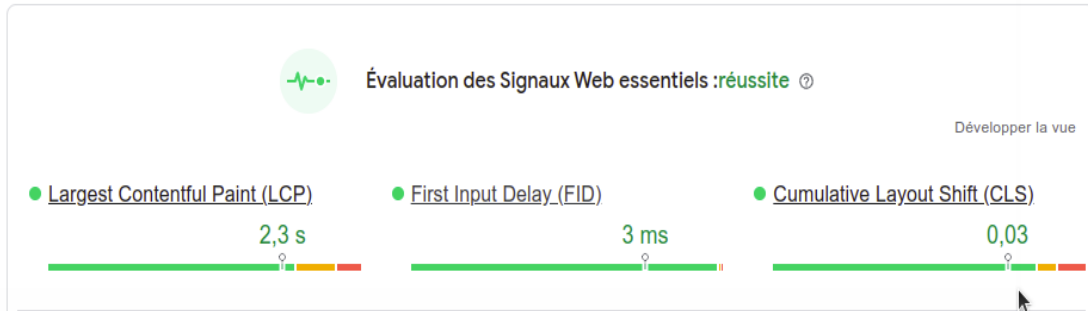


Illustration 15: Desktop LCP Score

https://inspirassion.com/en/

Analyser

Mobile

Bureau

Découvrez l'expérience de vos utilisateurs

Cette URL Origine



Illustration 16: Mobile LCP Score

Site architecture optimization

Internal links (links within a website) are also important for SEO. As a result, a search engine optimizer may boost a website's SEO by ensuring that vital pages are connected to and that appropriate anchor text is utilized in those links to increase a page's relevancy for certain searches. Creating an XML sitemap for bigger pages might also aid search engines in discovering and crawling all of the site's pages.

The search engine will find all listed pages in a sitemap with one visit to the XML sitemap file. On the website, it will have to jump through all internal links to find all the pages .

This ability of an XML sitemap to assist crawlers in faster indexation is especially important for our websites that:

- Have thousands of pages and a deep architecture.
- Frequently add new pages when applying a new feed or a new gramizing or summarizing.

We often update the sitemap links of every language pattern to see if google crawler change the rank of the updated links and if we get more clicks and page views for the new added links.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-en.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-fr.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-de.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-es.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-it.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-pt.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-pl.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-da.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-nl.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-zh.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-ar.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-ru.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adj-ja.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adv-en.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adv-fr.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adv-es.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adv-de.xml</loc>
  </sitemap>
  ▼<sitemap>
    <loc>https://inspirassion.com/sitemap-adv-it.xml</loc>
  </sitemap>
  ▼<sitemap>
```

Illustration 17: Sitemap links of all patterns of all languages

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  ▼<url>
    <loc>https://inspirassion.com/en/adj/miracle</loc>
    <changefreq>weekly</changefreq>
    <priority>0.5</priority>
  </url>
  ▼<url>
    <loc>https://inspirassion.com/en/adj/flower</loc>
    <changefreq>weekly</changefreq>
    <priority>0.5</priority>
  </url>
  ▼<url>
    <loc>https://inspirassion.com/en/adj/water</loc>
    <changefreq>weekly</changefreq>
    <priority>0.5</priority>
  </url>
  ▼<url>
    <loc>https://inspirassion.com/en/adj/fire</loc>
    <changefreq>weekly</changefreq>
    <priority>0.5</priority>
  </url>
  ▼<url>
    <loc>https://inspirassion.com/en/adj/ocean</loc>
    <changefreq>weekly</changefreq>
    <priority>0.5</priority>
  </url>
</urlset>
```

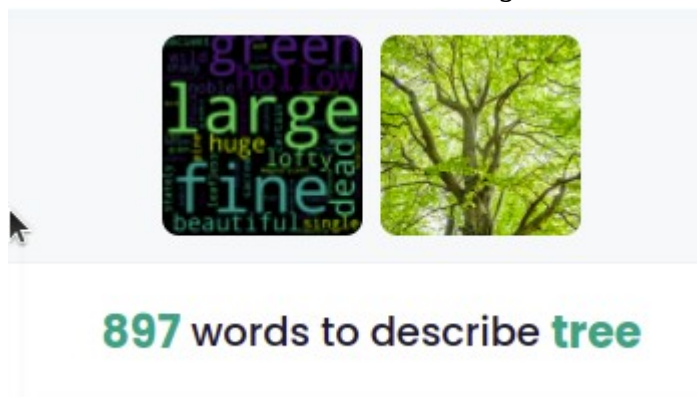
Illustration 18: sitemap links of Adjective pattern for English Language

3.4.4 Semantic Markup

Semantic markup is used to define the meaning of material on a website, such as identifying who wrote a piece of content or the topic and kind of content on a page. Rich snippets, such as more text, review ratings, and even photos, can be presented in the search results page by using semantic markup. Rich snippets in the SERPs have no effect on search rankings, but they can raise search CTR, resulting in more organic visitors.

In our case we used two types of images to improve click through rate to our platform ,Worldcloud image that contains the result words compressed within it and the subject image that auto generated from the input query of user than it will be saved for future same subject searches that can lead to our platform.

Illustration 19: Generated image in website



conclusion: we saw in The Third Chapter how to make Google and other search engines more efficient on our platform.

Chapter 4: Tools and Implementation

The goal of this chapter is to explain the tools used in develop the platform. We will talk about the programming languages used to create both the back-end and the front-end , beside the tools used in systems development life cycle and provide continuous delivery with high software quality.

4.1. Programming Languages

4.1.1 HTML 5

HTML is a markup language that we use to structure and provide meaning to our web content, such as establishing paragraphs, headers, and data tables, or embedding pictures and videos.[32]

4.1.2 CSS 3

CSS is a set of style rules that we use to apply styling to HTML content, such as changing background colors and fonts, and arranging material in numerous columns. [33]

4.1.3 Python

Python is a dynamically semantic, interpreted, object-oriented high-level programming language. Its high-level built-in data structures, together with dynamic typing and dynamic binding, making it ideal for Rapid Application Development and as a scripting or glue language for connecting existing components. Python's concise, easy-to-learn syntax promotes readability, which lowers software maintenance costs. Python facilitates program flexibility and code reuse by supporting modules and packages. The Python interpreter and its substantial standard library are free to download and distribute in source or binary form for all major platforms.[34]

4.1.4 JavaScript

Every time a web page does more than just sit there and display static information for us to look at — presenting timely content updates, interactive maps, dynamic 2D/3D visuals, scrolling video jukeboxes, and so on — we can guarantee JavaScript is involved. It's scripting or programming language and the third layer of a three-layer cake of standard web technologies, the first two of which (HTML and CSS) have been explored extensively elsewhere in the Learning Area [35]

4.2. Frameworks

4.2.1 Django

Django is a Python web framework that encourages rapid development and a simple, practical chevalier. It was created by professional developers to handle much of the web development work so we can focus on developing our app rather than recreating the wheel. It's free and open source.[36]

Django helps us write software that is:

Complete

Django adheres to the "batteries included" mentality, which means it comes with practically everything a developer might desire out of the box. Because everything we need is included in one "product," it all functions together smoothly, adheres to the same design principles, and comes with comprehensive and up-to-date documentation.

Versatile

Django has been used to create nearly every form of website, from content management systems and wikis to social networks and news sites. It can integrate with any client-side framework and serve material in nearly any format (including HTML, RSS feeds, JSON, XML, etc).

Secure

Django provides a framework that has been built to "do the right things" to defend the website automatically, which helps developers avoid many typical security blunders. Django, for example, avoids typical pitfalls like keeping session information in cookies (instead, cookies merely hold a key, and the real data is saved in the database) and directly storing passwords rather than a password hash.

Scalable

Django employs a "shared-nothing" design based on components (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Because the various components are clearly separated, it can grow to accommodate additional demand by adding hardware at any level: cache servers, database servers, or application servers. Django has successfully grown to meet the demands of some of the biggest websites (e.g. Instagram and Disqus, to name just two).

Maintainable

Django programming is created with design concepts and patterns in mind, resulting in code that is easy to maintain and reuse. It uses the Don't Repeat Yourself (DRY) philosophy in particular to avoid needless repetition and reduce the amount of code. The organization of comparable functionality into "applications" and, at a deeper level, related code into modules (following the Model View Controller (MVC pattern)).

4.2.2 Spacy

spaCy is a Python library for sophisticated Natural Language Processing (NLP). It is free and open-source.

spaCy was created with the goal of assisting developers in creating programs that analyze and "understand" vast amounts of text. It may be used to create data extraction and natural language comprehension systems, as well as to pre-process text for deep learning.[37]

It also supports pipelines trained on more than one language. This is especially useful for named entity recognition. The language ID used for multi-language or language-neutral pipelines is `xx`. The language class, a generic subclass containing only the base language data[38]

spaCy can be used for a wide variety of tasks related to NLP projects. It allows for example Tokenization, Lemmatization, POS tagging, sentence or entity recognition, dependency analysis, word/vector transformation and other normalization and cleaning techniques.

- **Tokenization:** Tokenization consists of breaking down a portion of text into words, spaces, symbols, punctuation and other elements to make "tokens". This is a fundamental step for most NLP tasks.
- **Lemmatization:** Lemmatization is directly related to tokenization, and allows a word to be reduced to its basic form. Suffixes, prefixes and other past participles can be removed to find the root of the term. This process is very useful, especially for Machine Learning and more specifically for text classification.
- **Part-of-speech (POS) :** Part-of-speech tagging is a process for assigning grammatical properties such as nouns, verbs, adverbs or adjectives to words. Words that share the same POS tags generally follow the same syntactic structure and are useful for rule-based processes;
- **Entity recognition:** Entity recognition is a process to classify named entities in a text into different predefined categories. It can be, for example, people,

places, or even dates. spaCy's statistical model makes it possible to classify a wide variety of entities, including people, entities, works of art or even nationalities.

- **Dependency parsing:** Dependency parsing is a method for driving the dependency parsing of a sentence. This reveals its grammatical format. This technique sheds light on the relationships between main words and their dependencies.

4.2.3 Vue.js

Vue.js is a user interface framework written in JavaScript. It's built on top of conventional HTML, CSS, and JavaScript, and it's based on a declarative and component-based programming approach that makes it easy to create basic or complicated user interfaces.[39]

Vue js has many Features

- **Components :** Vue components contain reusable code by extending basic HTML elements. Components, at their most basic level, are custom elements to which the Vue compiler applies functionality. Components in Vue are just Vue instances with pre-defined parameters.[40] A Vue component is demonstrated in the code sample below. The component displays a button and displays the number of times it has been clicked.
- **Templates :** Vue employs an HTML-based template syntax for connecting the displayed DOM to the data of the underlying Vue instance. All Vue templates are valid HTML that can be processed by browsers and HTML parsers that follow the standard. The templates are compiled into virtual DOM render methods by Vue. Vue uses a virtual Document Object Model to render components in memory before sending them to the browser. When the app state changes, Vue may use the reactivity system to compute the smallest number of components to re-render and the smallest amount of DOM operations.
- **Reactivity :** Vue has a reactivity mechanism that makes use of simple JavaScript objects and fast re-rendering. During the render, each component maintains track of its reactive dependencies, so the system understands when to re-render and which components to re-render.[41]
- **Transitions :** When things are inserted, changed, or deleted from the DOM, Vue provides a range of techniques to apply transition effects. This includes:

Apply classes to CSS transitions and animations automatically, Third-party CSS animation libraries should be included, During transition hooks, use JavaScript to directly alter the DOM. Third-party JavaScript animation libraries should be integrated.

4.2.4 Webpack

Webpack is a JavaScript module bundler that is free and open-source. It's mostly for JavaScript, but if the appropriate loaders are present, it can also modify front-end components including HTML, CSS, and pictures. Webpack takes modules with dependencies and creates static assets that represent them.

It can perform many operations:

- helps to bundle resources.
- watches for changes and re-runs the tasks.
- can run Babel transpilation to ES5, allowing us to use the latest JavaScript features without worrying about browser support.
- can transpile CoffeeScript to JavaScript
- can convert inline images to data URIs.
- allows to use require() for CSS files.
- can run a development webserver.
- can handle hot module replacement.
- can split the output files into multiple files, to avoid having a huge js file to load in the first page hit.
- can perform tree shaking.

4.3. Deployment

4.3.1 Amazon Web services (AWS)

Amazon Web Services, Inc. is an Amazon company that offers metered pay-as-you-go cloud computing platforms and APIs to consumers, businesses, and governments.

AWS server farms provide distributed computing processing capability and software tools through these cloud computing web services.

Amazon EC2 offers the most comprehensive and comprehensive instance selection to fit our workload's requirements.

There are instance kinds for general purpose, compute optimized, memory optimized, storage optimized, and accelerated computing that give the best balance of computation, memory, storage, and networking for our workloads. These instance classes are powered by Intel, AMD, NVIDIA, and AWS processors, which bring further performance and cost improvements.

Local storage and increased networking options offered with instance types assist to further optimize performance for disk or network I/O constrained applications.

Many instance types also provide bare metal instances, which give our applications direct access to the underlying server's CPU and memory for non-virtualized settings or applications that require their own hypervisor. Visit the EC2 instance types page to discover the correct instance for our workload.

The AWS Compute Optimizer receive suggestions for the best AWS Compute resources for workloads, which can help for save money and improve performance. [42]

Global Infrastructure :

- Multiple Locations
- High Precision Time with Amazon Time Sync Service
- Choice of operating systems and software

Cost and Capacity Optimization :

- Pay for what we use
- Scale seamlessly with Amazon EC2 auto scaling
- Optimize compute performance and cost with Amazon EC2 Fleet
- Optimized CPU configurations
- Pause and resume our instances

Storage :

- Optimal storage for every workload

Networking:

- High Packet-Per-Second Performance and Low Latency with Enhanced Networking
- Run High Levels of Inter-Node Communications with Elastic Fabric Adapter
- Manage Dynamic Cloud Computing Services with Elastic IP Addresses

- High Throughput and Low Latency with High Performance Computing (HPC) Clusters
- Access Services Hosted on AWS Easily and Securely with AWS PrivateLink

Operating Systems and Software : Amazon Machine Images (AMIs) are prepackaged operating systems that include Microsoft Windows and Linux distributions including Amazon Linux 2, Ubuntu, Red Hat Enterprise Linux, CentOS, SUSE, and Debian. We collaborate with our partners and the community to provide us as many options as possible. The AWS Marketplace offers a variety of commercial and free software from well-known suppliers that may be installed on our EC2 instances.[42]

4.3.2 Git

Git is a version control system for recording changes in any group of files. It's typically used to coordinate work among programmers who are working on source code together during software development. Speed, data integrity, and support for dispersed, non-linear processes are among its objectives.

4.3.3 GitLab

GitLab is an open-source business that makes GitLab, a DevOps tool that integrates the capacity to build, protect, and run software in one application.

Gitlab has an important features:

- Perform code reviews and enhance collaboration with merge requests
- Complete continuous integration (CI) and continuous deployment/delivery (CD) pipelines to build, test, and deploy our applications
- Each project can also have an issue tracker, issue board, and a wiki
- Used by more than 100,000 organizations, GitLab is the most popular solution to manage Git repositories on-premises
- Completely free and open source

4.3.4 Docker

Docker is a free and open platform for building, deploying, and operating applications. Docker allows us to decouple our applications from our infrastructure, allowing us to swiftly release software. We can manage our infrastructure the same way we control our apps with Docker. We may drastically minimize the time between developing code and executing it in production by leveraging Docker's approaches for shipping, testing, and deploying code fast.[43]

Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allows us to run many containers simultaneously on a given host. Containers are lightweight and contain everything needed to run the application, so we do not need to rely on what is currently installed on the host. You can easily share containers while we work, and be sure that everyone we share with gets the same container that works in the same way. [44]

Docker provides tooling and a platform to manage the life cycle of our containers:

- Develop our application and its supporting components using containers.
- The container becomes the unit for distributing and testing our application.
- When we are ready, deploy our application into our production environment, as a container or an orchestrated service. This works the same whether our production environment is a local data center, a cloud provider, or a hybrid of the two.

4.3.5 Docker Compose

Docker Compose is a tool that was created to assist in the definition and sharing of multi-container applications. With Compose, we can specify the services in a YAML file and then spin everything up or down with a single command.[45]

The main benefit of utilizing Compose is that we can describe our application stack in a file, save it at the root of a project repository (where it can now be version controlled), and simply allow others to contribute to the project. Someone may just clone the repository and open the compose app.

4.3.6 Nginx

NGINX is an open source web server, reverse proxy, caching, load balancing, media streaming, and other applications. It began as a web server that was optimized for speed and stability. NGINX may also operate as an email proxy server (IMAP, POP3,

and SMTP), as well as a reverse proxy and load balancer for HTTP, TCP, and UDP servers.[46]

The original purpose of NGINX was to produce the fastest web server possible, and the project continues to strive towards that goal. In web server performance benchmarks, NGINX routinely outperforms Apache and other servers. Websites have evolved from basic HTML pages to dynamic, complex content since the initial release of NGINX. NGINX has evolved with it, and currently supports all of the modern Web's features, including WebSockets, HTTP/2, gRPC, and multi-format video streaming (HDS, HLS, RTMP, and others)[47]

4.3.7 Gunicorn

Gunicorn is one of several WSGI server implementations, but it's particularly essential since it's a reliable, widely used component of web app deployments that's powering some of the world's largest Python-powered online apps, like Instagram.

Gunicorn conforms to the PEP3333 WSGI server standard, allowing it to run Python web applications that implement the application interface. If it is used in a web framework like Django, Flask, or Bottle to create a web application, the application will follow the WSGI definition.[48]

4.3.8 Cloudflare

Cloudflare is a free CDN that protects websites from attacks like SQL injection and identity theft. It also enhances site performance and speeds up loading times by utilizing their global network of data centers. The Cloudflare network functions as a massive VPN.

This implies that a site visitor will be physically routed to the nearest server in order to view our website. Simply said, the closer the data center is to our website, the faster it will load for the user. This is a Cloudflare-only feature.

Another advantage is that bot crawls are reduced, which can be beneficial or dangerous depending on the source. Typically, a bot crawl is used to visit websites and "crawl" for information on our page. While they can be useful for discovering content thieves who are utilizing our website design without our permission, they can also be used to generate fake ad impressions, insert spam and malware, and steal website material and information through file backdoors. A nice way to think of Cloudflare is as a web proxy. Cloudflare can filter suspicious information and keep us secure from hazardous viruses, spambots, and detrimental traffic flow since a cache of our website is available to visitors.

4.3.9 Sentry.io

Sentry.io is an open-source full-stack error tracking solution that supports PHP, Node.js, Python, Ruby, C#, Java, Go, React, Angular, Vue, JavaScript, and other server, browser, desktop, and native mobile languages and frameworks. Dropbox, AirBnB, PayPal, Uber, reddit, Mozilla, MailChimp, and Microsoft all utilize the system to keep track of thousands of apps.

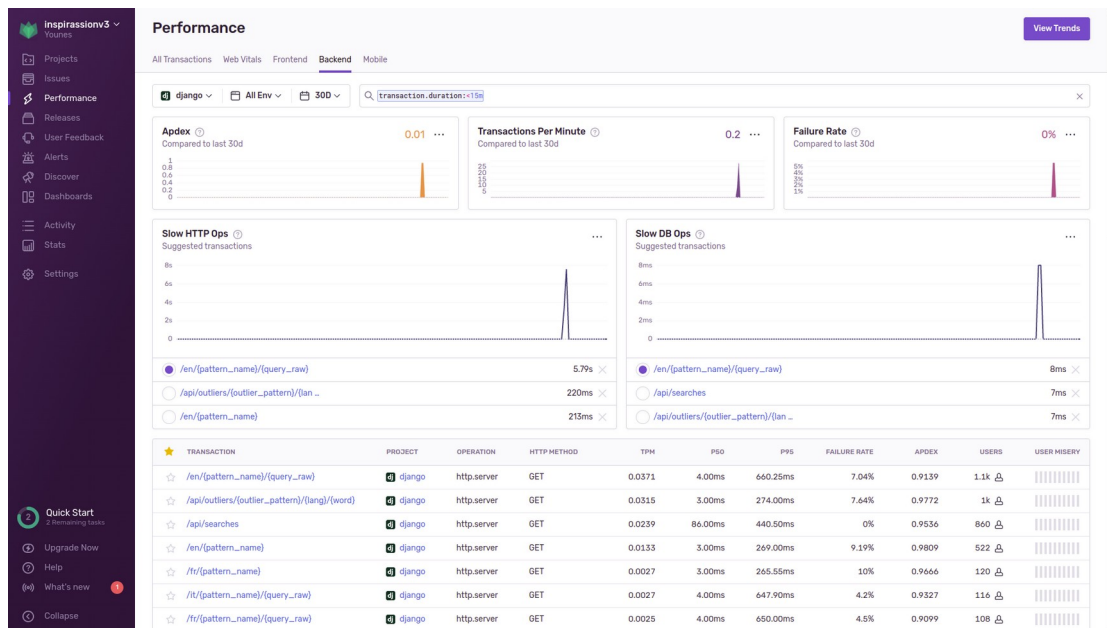


Illustration 20: Sentry Performance Monitoring

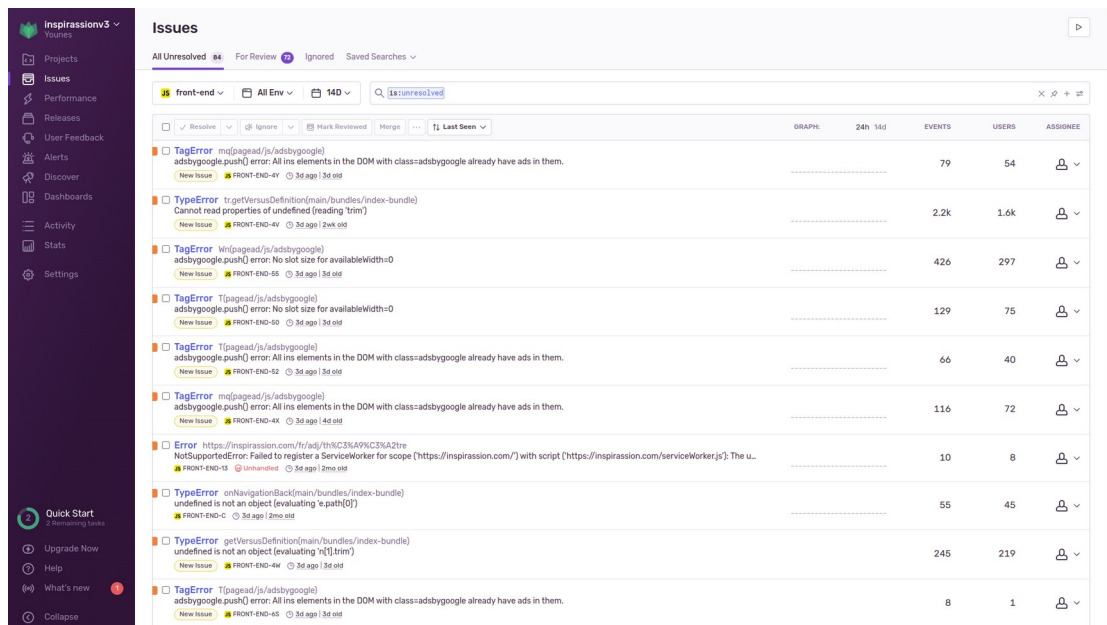


Illustration 21: Sentry Errors

4.4. Implementation

We can see different pages of the platform and her design with features available in multiple languages through his captures.

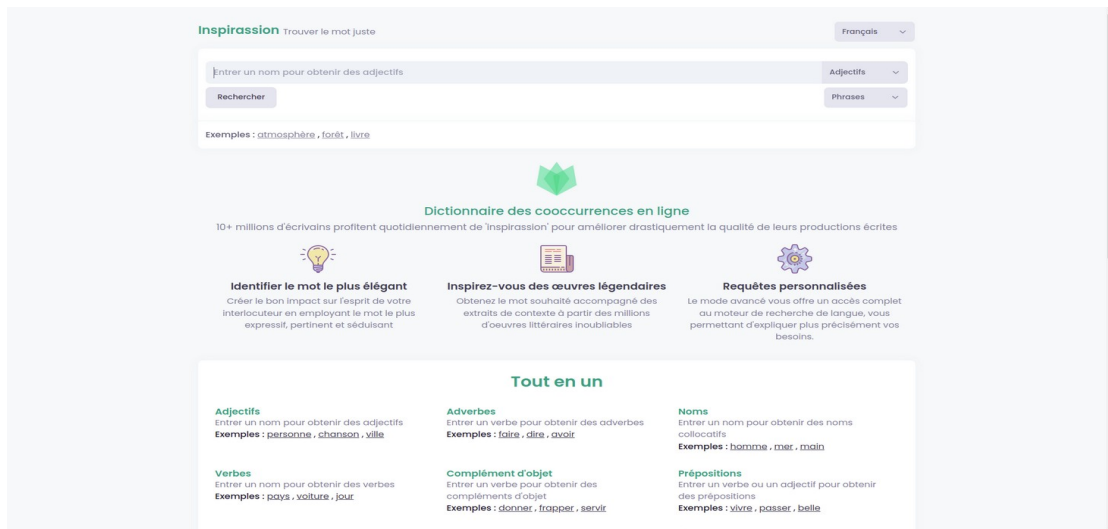


Illustration 22: Inspirasson - French Home Page

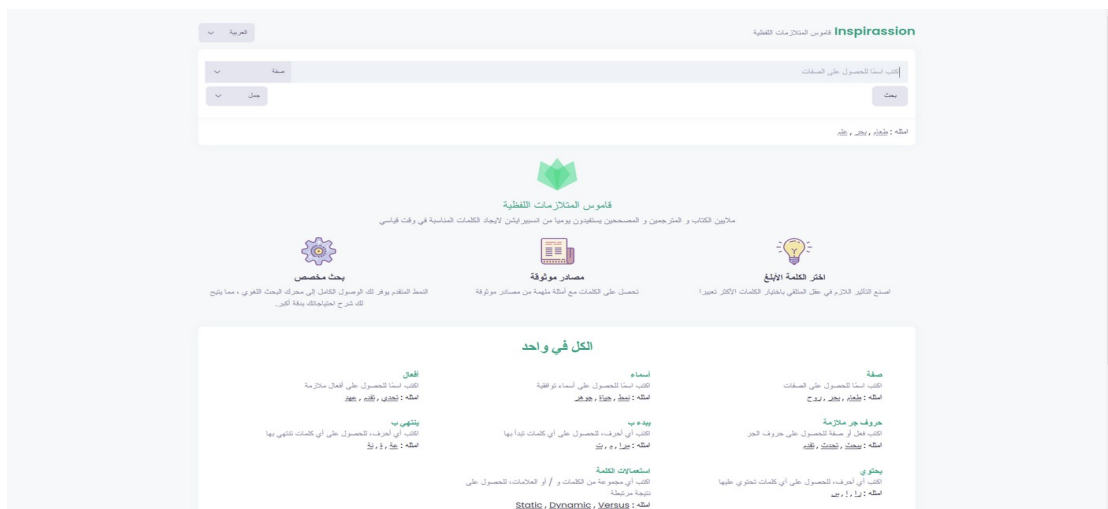


Illustration 23: Inspirasson - Arabic Home Page

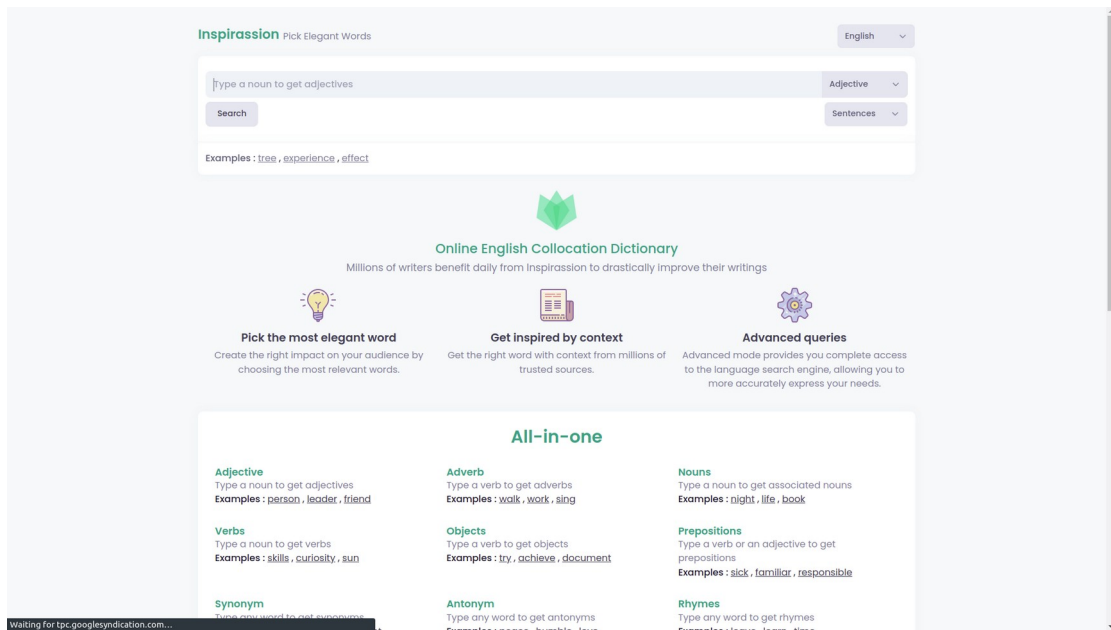


Illustration 24: Inspirasson - English Home Page

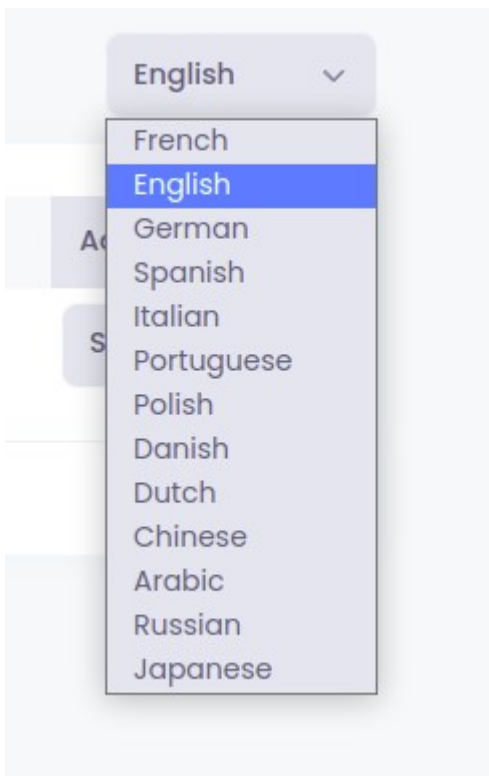


Illustration 26: Inspirasson - Languages Supported



Illustration 25: Inspirasson - Patterns Used

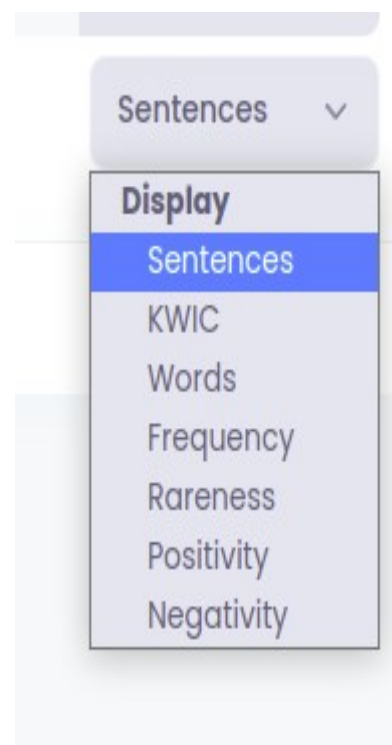


Illustration 27: Inspirasson - Displays Implemented

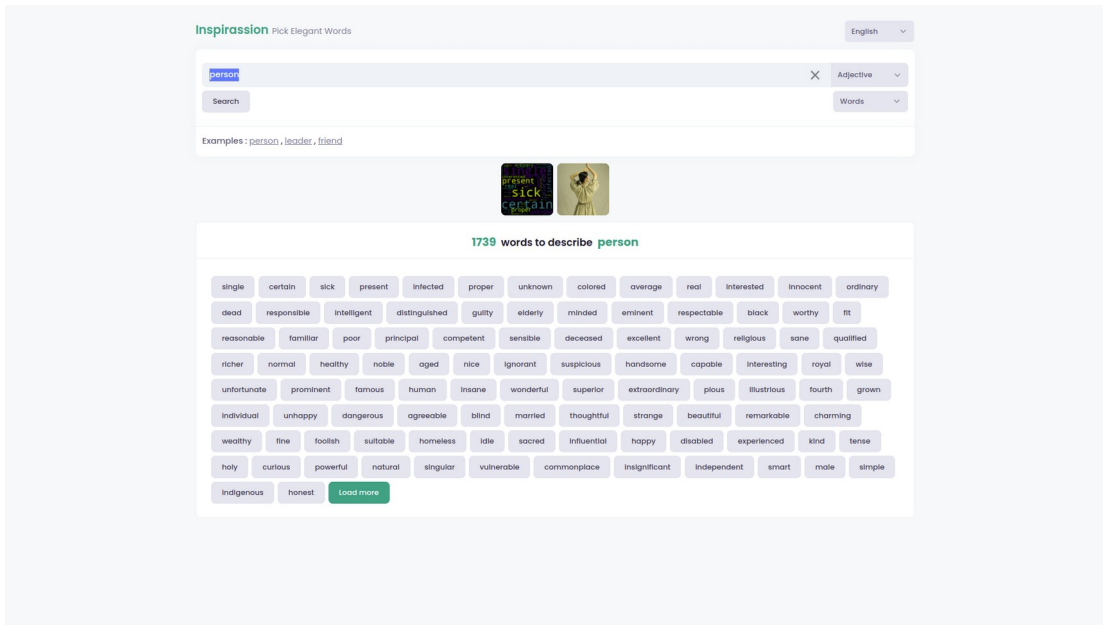


Illustration 28: Inspirasson - Words Mode

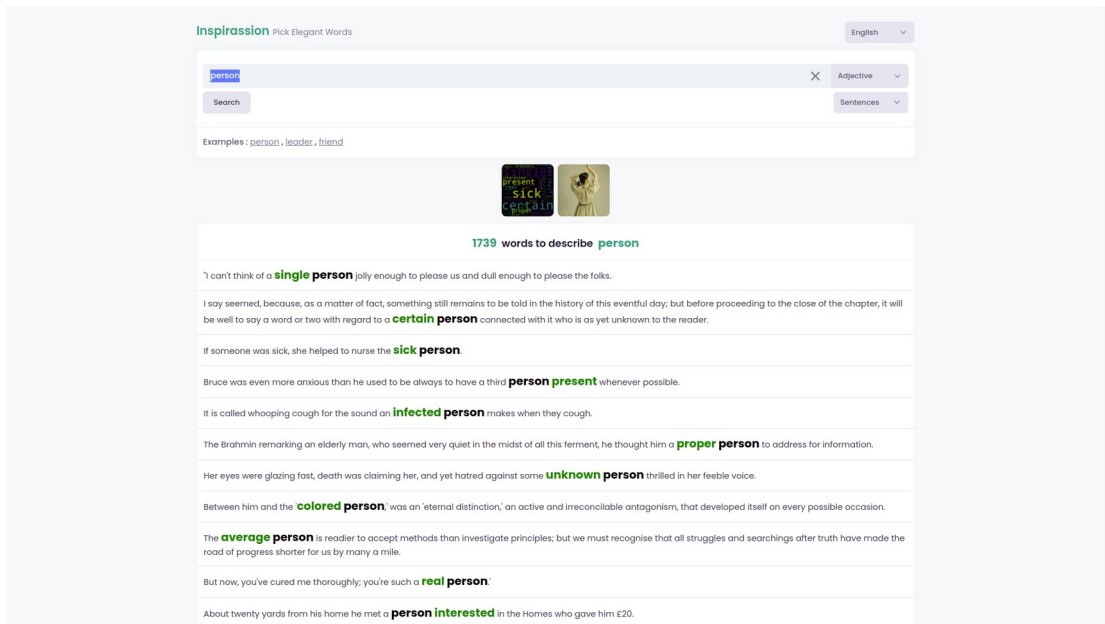



Illustration 29: Inspirasson - Sentences Mode

Inspirasson Pick Elegant Words English

× Adjective

KWIC

Examples : [person](#) , [leader](#) , [friend](#)



1739 words to describe person

"I can't think of a **single person** jolly enough to please us and dull enough to please the folks.

the close of the chapter, it will be well to say a word or two with regard to a **certain person** connected with it who is as yet unknown to the reader.

If someone was sick, she helped to nurse the **sick person**.

Bruce was even more anxious than he used to be always to have a third **person present** whenever possible.

It is called whooping cough for the sound an **infected person** makes when they cough.

an, who seemed very quiet in the midst of all this ferment, he thought him a **proper person** to address for information.

were glazing fast, death was claiming her, and yet hatred against some **unknown person** thrilled in her feeble voice.

Between him and the **colored person** was an 'eternal distinction,' an active and irreconcilable antagonism, that c

The **average person** is readier to accept methods than investigate principles; but we must reco

But now, you've cured me thoroughly, you're such a **real person**.

About twenty yards from his home he met a **person interested** in the Homes who gave him £20.

"An **innocent person** will not die if he drinks the poison," said Ekponyong.


Illustration 30: Inspirasson - KWIC Mode

Inspirasson Pick Elegant Words English

× Adjective

Frequency

Examples : [person](#) , [leader](#) , [friend](#)



1739 words to describe person

1. single	291
2. certain	276
3. sick	182
4. present	144
5. infected	141
6. proper	138
7. unknown	124
8. colored	116
9. average	108
10. real	99
11. interested	98
12. innocent	95
13. ordinary	86
14. dead	86
15. responsible	84
16. intelligent	76
17. interested in	70



Illustration 31: Inspirasson - Frequency Mode

Inspirassion Pick Elegant Words English

X Adjective

Positivity

Examples : [person](#) , [leader](#) , [friend](#)

1739 words to describe person

1. perfect	<div style="width: 100%;"></div>
2. faultless	<div style="width: 100%;"></div>
3. masterful	<div style="width: 100%;"></div>
4. legendary	<div style="width: 100%;"></div>
5. impressed	<div style="width: 100%;"></div>
6. impressive	<div style="width: 100%;"></div>
7. delightful	<div style="width: 100%;"></div>
8. excellent	<div style="width: 100%;"></div>
9. magnificent	<div style="width: 100%;"></div>
10. wonderful	<div style="width: 100%;"></div>
11. priceless	<div style="width: 100%;"></div>
12. brilliant	<div style="width: 100%;"></div>
13. incredible	<div style="width: 100%;"></div>
14. ideal	<div style="width: 100%;"></div>
15. beautiful	<div style="width: 100%;"></div>
16. splendid	<div style="width: 100%;"></div>
17. ...	<div style="width: 100%;"></div>

Illustration 32: Inspirassion - Polarity Mode

Inspirassion Pick Elegant Words English

on behalf of the Usage

Search Sentences

Examples : on behalf of the , far away , neither Learn More

on

1123 example sentences with on behalf of the

In 1088, however, he was once more in rebellion against the Crown **on behalf of the** Conqueror's eldest brother, Robert of Normandy.

Upon this Guthrum and his fellow kings and earls swore **on behalf of the** army that they would quit the King's country and give hostages.

Edwin James had proved a very strong case **on behalf of the** validity of the will.

The best method of giving practical effect to this principle is that the tutelage of such peoples should be entrusted to advanced nations who by reason of their resources, their experience or their geographical position can best undertake this responsibility, and who are willing to accept it, and that this tutelage should be exercised by them as Mandatories **on behalf of the** League.

"I'm collecting **on behalf of the** Hamburg Botanical Gardens," I answered.

"On the 18th of September, 1847, it (the building) was put up for sale by the magniloquent Mr. George Robins, and in consequence of a strong appeal to the feelings of the people, made through the public press, by which a National Subscription was raised for the purpose; this house was bought at the bidding of Mr. Peter Cunningham, for something more than 3,000 pounds sterling, and was placed under Trustees **on behalf of the** Nation."

Prussia, playing the part of Judas, pretended to welcome these reforms at first and lent the Poles its encouragement; but when Russia took up arms **on behalf of the** Polish reactionary party, and the country turned to Prussia to aid it in defending the constitution, the treacherous Frederick William not only declined to do so, but began to send his troops to occupy Polish territory.

But the protection thus given is often concerned with very important matters, and is constantly employed **on behalf of the** poorest and the most helpless.

On the following day, Saturday, I therefore betook myself to Norwood with a parcel of M. Zola's books, which I had received from Messrs. Macmillan & Co. **on behalf of the** Countess of Bective, who (prompted by the same spirit as Miss Loie Fuller) wished to sell these volumes at the 'bookland' stall on the occasion of the Charing

Illustration 33: Inspirassion - Advanced Static Mode

Inspirassion Pick Elegant Words English

adj tree Usage

Search Frequency

Examples : on behalf of the , far away , neither Learn More

old other

1210 example sentences with adj tree

1. other trees	170
2. old trees	161
3. tall trees	153
4. old tree	133
5. large tree	127
6. young trees	121
7. large trees	119
8. great trees	116
9. hollow tree	114
10. great tree	104
11. Small trees	93
12. many trees	91
13. big trees	88
14. few trees	79
15. green trees	75
16. big tree	71
17. tall trees	70

Illustration 34: Inspirassion - Advanced Dynamic Mode

Inspirassation Pick Elegant Words English

on december vs in december Usage

Search Sentences

Examples : on behalf of the , far away , neither Learn More

Do we say on december or in december

on december 1864 occurrences	in december 2428 occurrences
The last heavy piece of fighting on the XXth Corps' front before the attack on Jerusalem was on December 3 , when a regiment of yeomanry, which like a number of other yeomanry regiments had been dismounted to form the 74th Division, covered itself with glory.	The extracts are taken from letters written mostly in December and January last. (a) - -
An advance along the whole front was ordered to begin at six o'clock on December 29 .	XVI Saint-Evremond to Ninon de l'Enclos The Memory of Youth I was handed in December , the letter you wrote me October 14.
On December 10 , B.C. 133, Tiberius Gracchus entered upon the office of tribune, to which he had been elected, and pledged himself to the abolition of crying abuses.	Writing to Sidney Colvin in December 1887, he said, "I get along with my papers for Scribner not fast, nor so far specially well; only this last, the fourth one..."
On December 8 , 1847, the delra was stationed on Moorish territory, at Agueddin, on the left bank of the Meloula.	They are the roses in December , you remember someone said that God gave us memory so that we might have roses in December.
In order to minimize the danger, therefore, the French admiral, Du Fournet, in command of the Allied fleet, demanded the surrender to the Allies of certain guns and war material, and this demand being refused French and British marines were landed at the Piraeus on December 2 , 1816, and took possession of the Acropolis.	The old sculptor survived his illustrious patron and friend only two and a half years, declining gently into the grave, and his body was brought here in December , 1466.
	The Government of India found in December a movement that was a grave menace to the very foundations of public peace and security.

Illustration 35: Inspirassation - Versus Advanced Mode

Conclusion

In summary, we have shown in this thesis that there are many different types of collocations needed for language generation. Collocations are flexible and they can involve many words in various ways. We have described a fully implemented platform “INSPIRASSION” that automatically acquires such collocations from different large textual corpora and we have show how they can be represented in a flexible ways .We also tried to optimize our platform as much as we can so that we can reach much more users and we get level up ranked by most search engines so that we can appear at the top. By using some optimization techniques we were able to make search results appear so fast so that the user won’t get bored while surfing in our platform and we are working every day so that we can get more users and extend to other products in the near future.

Bibliography

- 1: , Knowbot programming: System support for mobile agents, , <http://www.cnri.reston.va.us/home/koe/iwoos-full.html>
- 2: Deutsch, Pete, An Internet archive server, 2017, https://groups.google.com/g/comp.archives/c/LWVA50W8BKk/m/wyRbF_IDc6cJ
- 3: , World-Wide Web Servers, 2012, <https://www.w3.org/History/19921103-hypertext/hypertext/DataSources/WWW/Servers.html>
- 4: Oppitz, Marcus; Tomsu, Peter, Inventing the Cloud Century: How Cloudiness Keeps Changing Our Life, Economy and Technology, , https://books.google.dz/books?id=vrEvDwAAQBAJ&pg=PA238&redir_esc=y#v=onepage&q&f=false
- 5: Greenberg, Andy, The Man Who's Beating Google , 2009, <https://www.forbes.com/forbes/2009/1005/technology-baidu-robin-li-man-whos-beating-google.html?sh=3a6e266bbc1b>
- 6: Altucher, James , 10 Unusual Things About Google (also: the worst VC decision I ever made), 2011, <https://www.forbes.com/sites/jamesaltucher/2011/03/18/10-unusual-things-about-google-also-the-worst-vc-decision-i-ever-made/?sh=7ba2495f12ee>
- 7: , Yahoo! And Netscape Ink International Distribution Deal, , https://web.archive.org/web/20131116112021/http://files.shareholder.com/downloads/YHOO/701084386x0x27155/9a3b5ed8-9e84-4cba-a1e5-77a3dc606566/YHOO_News_1997_7_8_General.pdf
- 8: Pursel, Bart, Search Engines, 2018, <https://psu.pb.unizin.org/ist110/chapter/2-1-search-engines/>
- 9: Gandal, Neil, The Dynamics of Competition in the Internet Search Engine Market, 2001, <https://escholarship.org/uc/item/0h17g08v>
- 10: Sew Staff, Intro to Search Engine Optimization | Search Engine Watch, 2007, <https://www.searchenginewatch.com/2007/03/12/intro-to-search-engine-optimization/>
- 11: Danny Sullivan, Who Invented the Term "Search Engine Optimization, 2004, <https://web.archive.org/web/20100423051708/http://forums.searchenginewatch.com/showpost.php?p=2119>
- 12: Pringle, G., Allison, L., and Dowe, D., What is a tall poppy among web pages?, 2007, https://users.monash.edu/~lloyd/tilde/InterNet/Search/1998_WWW7.html
- 13: David Kesmodel, Sites Get Dropped by Search Engines After Trying to 'Optimize' Rankings, 2005, <https://www.wsj.com/articles/SB112714166978744925?apl=y&r=947596>
- 14: Adam L. Penenberg (, Legal Showdown in Search Fracas, 2005, <https://www.wired.com/2005/09/legal-showdown-in-search-fracas/?currentPage=all>
- 15: , Google's Guidelines on Site Design, 2007,
- 16: , Bing Webmaster Guidelines, 2014, <https://www.bing.com/webmasters/help/webmasters-guidelines-30fba23a>
- 17: Andrew Broadbent, By the Data: For Consumers, Mobile is the Internet, 2015, <https://www.startupgrind.com/blog/mobile-is-the-internet-for-consumers/>
- 18: Zoltan Gyongyi & Hector Garcia-Molina , Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2007, <http://infolab.stanford.edu/~zoltan/publications/gyongyi2005link.pdf>

- 19: Sullivan, Danny , Rundown On Search Ranking Factors, 2007, <https://www.searchenginewatch.com/2005/09/29/rundown-on-search-ranking-factors/>
- 20: Christine Churchill, Understanding Search Engine Patents, 2007, <https://web.archive.org/web/20070207222630/http://searchenginewatch.com/showPage.html?page=3564261>
- 21: Loren Baker, 8 Things We Learned About Google PageRank, 2009, <https://www.searchenginejournal.com/8-things-we-learned-about-google-pagerank/5897/>
- 22: Matt Cutts, PageRank sculpting, 2010, <https://www.mattcutts.com/blog/pagerank-sculpting/>
- 23: Danny Sullivan , Google Loses “Backwards Compatibility” On Paid Link Blocking & PageRank Sculpting, 2009,
- 24: Amit Singhal, "Google Loses "On Paid Link Blocking & PageRank Sculpting, 2011, <https://searchengineland.com/google-loses-backwards-compatibility-on-paid-link-blocking-pagerank-sculpting-20408>
- 25: AARON ADERS,
NEWSLETTERS
SUBSCRIBE
SEARCH ENGINE OPTIMIZATION
What You Need to Know About Google's Penguin Update, 2012, <https://www.inc.com/aaron-aders/what-you-need-to-know-about-googles-penguin-update.html>
- 26: Barry Schwartz, Google Penguin looks mostly at your link source, says Google, 2017,
- 27: Pandu Nayak, Understanding searches better than ever before, 2020, <https://blog.google/products/search/search-language-understanding-bert/>
- 28: Danny Sullivan , Submitting To Directories: Yahoo & The Open Directory, 2007, <https://web.archive.org/web/20070519052103/http://searchenginewatch.com/showPage.html?page=2167881>
- 29: , What is a Sitemap file and why should I have one?, 2007, https://developers.google.com/search/docs/advanced/sitemaps/overview?topic=8514&visit_id=637889325978346596-4286135351&rd=2
- 30: , Search Console - Crawl URL, 2015, https://developers.google.com/search/docs/advanced/crawling/ask-google-to-recrawl?utm_source=wmx&utm_medium=deprecation-pane&utm_content=submit-url
- 31: Cho, J. and Garcia-Molina, Efficient Crawling Through URL Ordering , 2007, <http://ilpubs.stanford.edu:8090/347/>
- 32: , What is HTML?, , https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started
- 33: , What is CSS?, , https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS
- 34: , What is Python? Executive Summary, , <https://www.python.org/doc/essays/blurb/>
- 35: , What is JavaScript?, , https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- 36: , Meet Django, , <https://www.djangoproject.com/>

37: , What's spaCy?, , <https://spacy.io/usage/spacy-101>
38: , Multi-language support, , <https://spacy.io/usage/models>
39: , What is Vue?, , <https://vuejs.org/guide/introduction.html#what-is-vue>
40: , Components Basics, , <https://vuejs.org/guide/essentials/component-basics.html>
41: , Reactivity in Depth, , <https://vuejs.org/guide/extras/reactivity-in-depth.html>
42: , Amazon EC2 features, , <https://aws.amazon.com/ec2/features/>
43: , Docker overview, ,
<https://docs.docker.com/get-started/overview/#:~:text=Docker%20is%20an%20open%20platform,ways%20you%20manage%20your%20applications>
44: , The Docker platform, ,
<https://docs.docker.com/get-started/overview/#:~:text=Docker%20is%20an%20open%20platform,ways%20you%20manage%20your%20applications>
45: , Use Docker Compose, , https://docs.docker.com/get-started/08_using_compose/
46: , What is NGINX?, , <https://www.nginx.com/resources/glossary/nginx/>
47: , NGINX as a Web Server, , <https://www.nginx.com/resources/glossary/nginx/>
48: , Why is Gunicorn important?, , <https://www.fullstackpython.com/green-unicorn-gunicorn.html>