CoreMind

An Alternative Architecture for Explainable, Efficient, and Evolving Artificial Intelligence

Knowledge Representation and Organisation, Adaptive Learning, and Continuous Updating Without Full Retraining

Architectural Foundations, Experimental Prototype, and Research Perspectives

Version 1.0 — 31 August 2026

© 2026 SANHAJI Youssef — CC BY-NC-ND 4.0
Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International



SANHAJI Youssef
sanhaji@outlook.com

Abstract

Recent advances in artificial intelligence have largely relied on deep learning architectures, particularly Transformer-based architectures [1], which are capable of learning distributed representations and leveraging context at scale. Despite their remarkable performance, several challenges remain, particularly in terms of computational and energy costs [15], explainability, the continuous integration of new knowledge, and the ability of models to evolve without full retraining.

This White Paper presents CoreMind, an alternative architecture designed as a foundation for the development of explainable, efficient, and evolving artificial intelligence systems. CoreMind is based on a structured and sparse tensor representation of knowledge, combined with contextualisation and adaptive learning mechanisms that enable acquired knowledge to be organised, exploited, and progressively enriched. The architecture also incorporates its own *Self-Attention* and *Multi-Head Attention* mechanisms, distinct from the attention mechanism based on Query, Key, and Value (QKV) matrices used in Transformer-based architectures [1]. Its design notably aims to enable the integration of new knowledge without requiring full model retraining and to reduce dependence on computationally intensive infrastructures.

This White Paper presents the architectural foundations of CoreMind, the principles that guided its design, and its main mechanisms. An experimental prototype was developed to evaluate the feasibility of the proposed hypotheses. The experiments show that the architecture can construct matrix-based semantic representations specific to CoreMind and exploit them across different tasks, including

similarity and neighbourhood search, analogies, automatic knowledge grouping (*clustering*), masked token prediction, *Retrieval-Augmented Generation* (RAG) mechanisms [10], exploration of semantic relationships, and an initial form of text generation. The prototype also makes it possible to observe and analyse certain internal representation and attention mechanisms.

The experiments were conducted on CPU infrastructure, without the use of a GPU, in order to investigate the feasibility of an architecture capable of learning and exploiting knowledge with controlled hardware resources. The results obtained constitute an initial experimental validation of the proposed fundamental mechanisms; however, they represent neither an exhaustive benchmark against existing architectures nor a definitive validation of the anticipated computational or energy efficiency gains.

Building on this experimental foundation, ongoing work focuses on consolidating the architecture, comparatively evaluating its performance, scaling, continuously integrating knowledge, developing multi-token generation based on semantic trajectories, and progressively exploring modalities beyond text. This research aims to assess the potential of CoreMind as an alternative architectural foundation for the development of explainable, adaptive, and efficient artificial intelligence systems.

Keywords

Artificial Intelligence Architecture · Alternative Architecture · Knowledge Representation · Sparse Tensor Representation · Matrix-Based Semantic Representation · Self-Attention · Multi-Head Attention · Adaptive Learning · Continuous Knowledge Integration · Computational Efficiency · Explainable Artificial Intelligence · Natural Language Processing · Retrieval-Augmented Generation (RAG) · Language Models

English Edition Note

> This document is the English translation of the original French version of the CoreMind White Paper, Version 1.0, published on 31 August 2026. The English edition preserves the structure, scientific content, experimental results, figures, and references of the original publication.
>
> Certain diagrams, figures, prototype interface elements, and screenshots are retained in their original French version, as they correspond directly to the experiments conducted and documented in the original publication.
>
> In the event of any discrepancy between the English translation and the original French version, the original French version shall prevail.

Original language: French
Original publication: 31 August 2026
English translation: 3 September 2026

About This White Paper

This White Paper presents CoreMind, an alternative architecture for artificial intelligence based on a specific approach to the representation, organisation, and learning of knowledge. It outlines the architectural foundations of the project, the principles that guided its design, its main mechanisms, and the initial results obtained through an experimental prototype developed to assess its feasibility.

CoreMind is presented here as a research architecture aimed, in particular, at explainability, computational efficiency, and the progressive integration of new knowledge without requiring full model retraining. The White Paper describes the current state of the architecture and the experimental

results available at the time of publication. Certain mechanisms, design choices, and parameters may evolve as research, optimisation, and validation efforts progress.

The experimental prototype is intended to provide a concrete evaluation of the proposed fundamental mechanisms and their ability to produce usable behaviours across different tasks. The results presented therefore constitute an initial experimental basis for the study of CoreMind. They do not constitute an exhaustive benchmark against existing architectures; comparative evaluations, quantitative analyses, and larger-scale validations are planned for subsequent stages of the research.

This White Paper constitutes a technical research publication rather than exhaustive implementation documentation. Certain technical details relating to the architecture's internal mechanisms are intentionally not disclosed at this stage. Nevertheless, the information presented is sufficient to describe the architectural principles, the hypotheses under investigation, the general operation of the prototype, and the experimental results obtained.

Finally, mechanisms that have been explicitly tested experimentally are distinguished from future research directions. Perspectives relating, in particular, to scaling, multi-token generation based on semantic trajectories, and extension to other modalities describe development directions that will be subject to further research and validation.

Table of Contents

Abstract 1
About This White Paper 2
1. Introduction 6
1.1 Context 6
1.2 Objectives of CoreMind 6
2. Founding Principles of CoreMind 7
2.1 Architecture, Not a Model 7
2.2 Experimental Framework and Prototype 7
2.2.1 Development Environment 10
2.2.2 Training Corpus 10
2.2.3 Current State of the Prototype 11
2.2.4 Experimental Evaluation of the Prototype (Appendix A) 12
2.3 The Four Fundamental Pillars of CoreMind 13
2.3.1 Tensor Representation 13
2.3.2 CoreMind Attention 14
2.3.3 Adaptive Learning 15
2.3.4 CoreMind Organisation and Algorithms 16
3. CoreMind Tensor Representation 18
3.1 Limitations of Classical Vector Representations 18
3.2 Principles of the CoreMind Tensor Representation 18
3.3 Contextual Horizons 19
3.4 Overall Organisation of Knowledge 20
3.5 Architectural Implications 22
4. CoreMind Attention 23
4.1 Role of Attention in CoreMind 23
4.2 CoreMind Global Attention 23
4.3 Specialised Attention and Horizons (CoreMind Multi-Head Attention) 24
4.4 Dynamic Construction of Meaning 27
4.5 Architectural Implications 27
5. Learning and Knowledge Evolution 28
5.1 Initial Knowledge Construction 28
5.2 Dynamic Construction and Organisation of Weights 29
5.3 Incremental Knowledge Evolution 30
5.4 Initialisation and Usage Modes 31
5.5 Updating Without Full Retraining 33
5.6 Architectural Implications 33

6. Capabilities Obtained Through the Exploitation of CoreMind 34
6.1 Cooperation Between the Architecture's Mechanisms 35
6.2 CoreMind Tensor Embedding 37
6.3 Semantic Similarity 39
6.4 Semantic Neighbour Search 41
6.5 Conceptual Clustering (Semantic Clustering) 43
6.6 Analogy Construction 45
6.7 Lexical Prediction (MLM) 47
6.8 Exploitation of Semantic Attention 49
6.9 Information Retrieval (RAG) [10] 51
6.10 Evaluation and Selection of Candidate Knowledge 53
6.11 Next-Token Generation 55
6.12 Current State of the Prototype and Validation Perspectives 57
7. Models Built Upon the CoreMind Architecture 58
7.1 One Architecture, Multiple Model Families 58
7.2 Principal Model Families 60
8. Perspectives for the Evolution of the Architecture 61
8.1 Towards New Approaches to Generation 61
8.2 Trajectory-Based Generation 61
8.3 Perspectives 63
9. Conclusion 63
9.1 Summary of Contributions 63
9.2 Research Perspectives 63
10. Bibliographical References 65
11. Glossary 67
12. Appendix A 70
A.1 Construction of the Tensor Embedding 71
A.2 Semantic Similarity 74
A.3 Conceptual Clustering 76
A.4 CoreMind Attention 78
A.5 CoreMind Multi-Head Attention 80
A.6 Masked Language Model (MLM) 81
A.7 Vector Analogy 83
A.8 Neural Navigator — 3D UMAP Exploration [16] 85

1. Introduction

1.1 Context

Natural Language Processing (NLP) has undergone major developments over the past decade, driven in particular by the emergence of deep neural architectures and large language models. These approaches have enabled significant advances in areas such as language understanding, information retrieval, machine translation, text generation, and conversational assistance.

Alongside these advances, the increasing complexity of models has led to substantial growth in data requirements, computational resources, and training time. Today's highest-performing models rely on specialised infrastructures involving considerable computational capacity, making their development, adaptation, and deployment difficult to reproduce outside major research centres or organisations with substantial resources [15].

This situation raises several research questions. Can architectures be designed to exploit linguistic knowledge differently? Is it possible to construct richer representations while limiting computational requirements? Can greater knowledge traceability and continuous model adaptation be achieved as data evolve?

It is within this context that the CoreMind architecture was developed.

1.2 Objectives of CoreMind

CoreMind emerged from the intention to explore an alternative approach to the representation and exploitation of linguistic knowledge in Natural Language Processing. The architecture is based on the hypothesis that the performance and capabilities observed in modern systems do not depend exclusively on the continuous increase in model size, data volume, or computational power, but may also emerge from new forms of knowledge representation and organisation.

From this perspective, CoreMind proposes a new way of representing, structuring, and exploiting linguistic knowledge. The architecture is based on the hypothesis that representation mechanisms different from those currently dominant may lead to different or complementary capabilities in understanding, inference, adaptation, or generation, depending on the tasks considered.

The objective of CoreMind is not to address a particular Natural Language Processing task. Similar to the role played by the Transformer architecture, CoreMind is positioned as a generic architectural substrate upon which different types of models and applications can be built. The capabilities demonstrated to date — such as semantic similarity, semantic neighbour search, clustering, analogies, masked lexical prediction, information retrieval, and language generation — constitute specific instantiations of this architecture rather than its sole purpose.

More specifically, CoreMind pursues several complementary objectives. The first is to investigate more structured linguistic representations capable of explicitly integrating semantic, contextual, and relational information within a single representational structure. The second is to develop inference mechanisms capable of efficiently exploiting these representations without relying on computationally intensive optimisation phases or large-scale specialised infrastructures. The third objective concerns the continuous adaptation of knowledge, enabling models built upon the architecture to evolve progressively as new information is acquired.

Finally, CoreMind places particular emphasis on the traceability of the knowledge exploited by the models derived from it. The architecture was designed with the objective of maintaining an explicit link between internal representations and the information observed in corpora or introduced during

its evolution. This property is intended to facilitate the interpretation of the results produced and to support the development of systems capable of evolving in a transparent and controllable manner.

2. Founding Principles of CoreMind

2.1 Architecture, Not a Model

A common source of confusion in Natural Language Processing is the tendency to equate underlying architectures with the models derived from them. An architecture, however, primarily constitutes a conceptual and technical framework defining how knowledge is represented, organised, and exploited. Models built upon that architecture are then specific instantiations adapted to particular tasks.

The Transformer architecture [1] clearly illustrates this distinction. It does not, in itself, constitute a translation, generation, or classification model. Rather, it provides a set of fundamental mechanisms upon which various models have been built, including BERT [2], GPT [3], T5 [4], as well as numerous specialised architectures. These models share a common architectural foundation while pursuing different objectives.

CoreMind adopts a similar philosophy. It is not a model dedicated to a particular task, but a generic architecture for the representation and exploitation of linguistic knowledge. Its objective is not to solve a specific Natural Language Processing problem, but to provide an architectural framework for building different types of models from a common set of representation, attention, and inference mechanisms.

The capabilities currently demonstrated by the prototype — including semantic similarity, semantic neighbour search, clustering, analogies, masked lexical prediction, information retrieval, and language generation — therefore do not constitute the CoreMind architecture itself. They represent specific applications built upon this architectural substrate and exploiting the properties of its internal mechanisms.

This distinction is essential to understanding the scope of the project. Just as the Transformer architecture enabled the emergence of an entire family of specialised models, CoreMind is designed as a conceptual and technical infrastructure capable of serving as a foundation for models of various kinds. The architecture thus aims to provide a unified framework for the representation, organisation, and exploitation of linguistic knowledge, independently of the downstream tasks to which that knowledge is applied.

2.2 Experimental Framework and Prototype

The CoreMind architecture is currently being developed in the form of an experimental prototype intended to validate the main concepts presented in this White Paper. This prototype constitutes a research platform for investigating the feasibility of the architecture, evaluating its fundamental mechanisms, and conducting initial experiments on real-world data corpora.

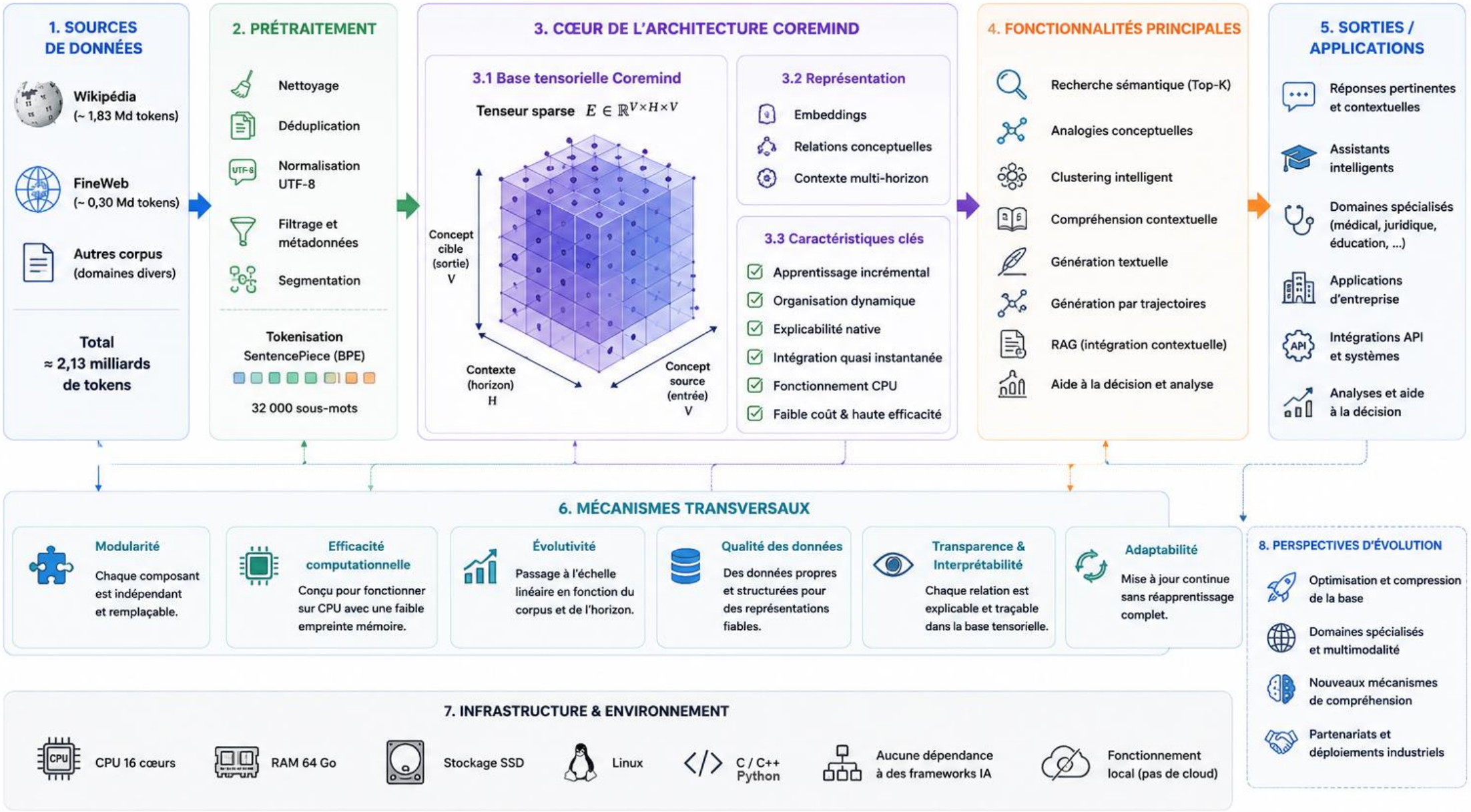
The development of the prototype has followed a progressive approach involving the design, implementation, and individual validation of the architecture's various mechanisms prior to their integration into a coherent system. This approach has made it possible to verify the operation of the main components, including knowledge representation, embedding mechanisms [5][6], semantic search, analogies, conceptual clustering, and initial generation capabilities.

The prototype presented in this document does not constitute a final product. Several mechanisms remain under investigation, optimisation, or experimental validation. The results presented in this White Paper should therefore be regarded as an initial validation of the architecture's founding principles, intended to provide a basis for future developments and subsequent research.

The following sections present the main characteristics of the experimental framework, including the development environment, the training corpus, and the current state of the prototype.

Overview of the CoreMind Architecture

A Unified Architecture for Representing, Organising, Searching, Understanding, and Generating Knowledge



2.2.1 Development Environment

The CoreMind experimental prototype was developed entirely in **Python / C / C++**, without relying on frameworks or libraries dedicated to artificial intelligence or Deep Learning. In order to maintain full control over the architecture and its internal mechanisms, a **software library specifically designed for CoreMind** was conceived and developed as part of this project. This library brings together all the fundamental components of the architecture, including mechanisms for knowledge representation, learning, semantic search, embedding, generation, and the exploitation of conceptual relationships.

The objective of this approach was to develop a fully controlled implementation, independent of existing architectures, and to validate CoreMind's fundamental concepts without relying on mechanisms provided by current artificial intelligence frameworks.

The experiments were conducted on a workstation equipped with an **Intel Core processor with 16 vCPUs (Haswell architecture)** and **64 GB of ECC memory**. Development deliberately prioritised execution on a central processing unit (CPU), without the use of dedicated hardware accelerators such as GPUs. This choice makes it possible to evaluate the operation of the architecture in a standard hardware environment and to investigate its feasibility without dependence on specialised computing infrastructure.

The prototype currently relies on a vocabulary of **32,000 tokens**, constructed using the **SentencePiece algorithm [7]**, together with a **horizon of 40**, corresponding to the maximum depth of semantic relationships taken into account during learning and knowledge exploitation mechanisms.

This configuration constitutes an initial experimental basis for validating CoreMind's fundamental principles. The selected parameters may evolve in future work to accommodate increases in corpus size, performance optimisation, and the investigation of new experimental configurations.

Element	Configuration
Development Languages	Python / C / C++
AI Framework	None
Processor	Intel Core – 16 vCPUs (Haswell)
Memory	64 GB ECC RAM
GPU Accelerator	None
Tokenisation Algorithm	SentencePiece [7]
Vocabulary Size	32 000 tokens
Horizon	40

2.2.2 Training Corpus

The experiments presented in this White Paper were conducted using several text corpora selected to progressively validate the different mechanisms of the CoreMind architecture. The initial development phases were carried out on smaller corpora, enabling each component to be tested and validated individually before its integration into a complete prototype.

The larger-scale experiments rely primarily on a corpus comprising a portion of **French Wikipedia [13]**, supplemented with data from the **FineWeb corpus [14]**. The resulting dataset represents approximately **6 GB of textual data**, corresponding to nearly **2.1 billion tokens**, providing sufficient diversity to evaluate the main mechanisms for knowledge representation, organisation, and exploitation.

Prior to training, the data underwent a preparation phase comprising text cleaning, normalisation, segmentation, and tokenisation using **SentencePiece [7]**, resulting in the construction of a vocabulary of **32,000 tokens**. This stage aims to ensure a consistent representation of the data and its compatibility with the CoreMind architecture.

The experimental strategy adopted follows a progressive scaling approach, whereby the fundamental mechanisms were first validated on smaller corpora before being evaluated on larger volumes of data. This approach enabled the prototype to be progressively consolidated while facilitating the identification and correction of limitations encountered during development.

In the current experimental configuration, the CoreMind training phase requires approximately **60 minutes** on the hardware platform described previously. This duration refers exclusively to the training of the architecture and does not include corpus preparation operations such as cleaning, normalisation, tokenisation, or vocabulary construction.

Element	Configuration
Main Corpus	Portion of French Wikipedia [13] + FineWeb [14]
Primary Language	French
Corpus Size	Approximately 6 GB
Number of Tokens	Approximately 2.1 billion
Pre-processing	Cleaning, normalisation, segmentation
Tokenisation Algorithm	SentencePiece [7]
Vocabulary Size	32 000 tokens
CoreMind Training Time*	Approximately 60 minutes
Experimental Strategy	Progressive validation on increasingly large corpora

* *The stated time refers exclusively to the CoreMind training phase on the prepared corpus. Cleaning, normalisation, tokenisation, and vocabulary construction operations are not included in this duration.*

2.2.3 Current State of the Prototype

To date, the CoreMind experimental prototype demonstrates the feasibility of the main concepts presented in this White Paper. The different development phases have resulted in a functional architecture capable of representing, organising, and exploiting knowledge from text corpora of varying sizes.

The experiments conducted have enabled the implementation of the architecture's main components, including semantic knowledge representation, the construction of embeddings [5][6], similarity measurement, semantic neighbour search, conceptual clustering, analogies, semantic search, and

initial text generation mechanisms. These results confirm the coherence of the proposed approach and constitute an initial experimental demonstration of the CoreMind architecture.

The prototype nevertheless remains a research demonstrator. Certain mechanisms are still undergoing improvement and further investigation, particularly in terms of performance optimisation, scaling to larger corpora, evaluation against established benchmarks, and the exploration of new generation strategies. Future developments will aim to consolidate these different aspects in order to progressively extend the capabilities of the architecture.

The table below presents the current status of the prototype's main functionalities.

Functionality	Current Status
Knowledge Representation	Experimentally validated
Embedding Construction [5][6]	Experimentally validated
Semantic Similarity	Experimentally validated
Semantic Neighbour Search	Experimentally validated
Conceptual Clustering [11][12]	Experimentally validated
Semantic Analogies	Experimentally validated
Semantic Search	Experimentally validated
Text Generation	Functional prototype
Evaluation on Public Benchmarks	In progress
Performance Optimisation	In progress
Trajectory-Based Generation	Research perspective

Through this experimental prototype, CoreMind demonstrates the feasibility of its main fundamental mechanisms and provides a concrete foundation for the future development of the architecture.

2.2.4 Experimental Evaluation of the Prototype (Appendix A)

The CoreMind prototype was subjected to a series of experiments aimed at evaluating the operation of the architecture's main mechanisms across different tasks. These experiments focus in particular on semantic representations, similarity and neighbour search, analogies, automatic grouping (clustering), masked token prediction, Retrieval-Augmented Generation (RAG) mechanisms, the analysis of attention mechanisms, and initial text generation capabilities.

In order to preserve the continuity of the architectural presentation, the detailed experimental protocol, test examples, results obtained, and prototype screenshots are grouped together in **Appendix A — Experimental Evaluation of the CoreMind Prototype**.

These experiments are intended to evaluate the feasibility and behaviour of the proposed fundamental mechanisms. They constitute an initial experimental basis and should not be interpreted as an exhaustive benchmark or as a definitive comparison with existing architectures.

2.3 The Four Fundamental Pillars of CoreMind

The CoreMind architecture is based on four fundamental pillars that jointly define how linguistic knowledge is represented, organised, acquired, and exploited. These pillars constitute the common foundation upon which different specialised models can be built, independently of the tasks for which they are intended.

The first pillar is the tensor representation of knowledge. It defines how lexical units, their relationships, and their contexts are structured within the architecture. This representation constitutes the fundamental basis upon which CoreMind's other mechanisms rely.

The second pillar is the architecture's own attention mechanism. Its role is to identify, weight, and exploit relevant relationships within the internal representations in order to dynamically adapt the interpretation of the linguistic context under consideration.

The third pillar is adaptive learning. Unlike approaches based on a fixed training phase followed by a separate usage phase, CoreMind was designed to enable the acquisition, integration, and continuous evolution of knowledge over time. This property constitutes one of the foundations of its capacity for progressive adaptation.

The fourth pillar encompasses the internal organisation of the architecture as well as the algorithms that coordinate the operation of its various mechanisms. These algorithms ensure the overall coherence of the system and enable representations, relationships, and acquired knowledge to be transformed into capabilities that can be exploited by models built upon the architecture.

These four pillars should not be regarded as independent components. They form a coherent whole whose interactions define the overall behaviour of the CoreMind architecture. The various capabilities demonstrated by the current prototype — such as semantic similarity, analogies, information retrieval, masked lexical prediction, and language generation — emerge from the cooperation of these fundamental mechanisms rather than from a module dedicated to a particular task.

2.3.1 Tensor Representation

Knowledge representation constitutes the first fundamental pillar of the CoreMind architecture. It defines how linguistic units, their contexts, and their relationships are organised within the system. CoreMind's founding hypothesis is that a significant proportion of a language processing system's capabilities depends directly on the richness and structure of its internal representations.

Modern approaches generally rely on dense vector representations [5][6] capable of capturing semantic properties through statistical observation of corpora. Although particularly effective, these representations condense a large amount of linguistic information into a relatively compact numerical space, part of which remains implicit and must be reconstructed during subsequent processing stages.

CoreMind explores a different approach. The architecture relies on a tensor representation designed to integrate, within a single structure, semantic, contextual, and relational information associated with lexical units. This representation is intended not only to characterise the meaning of a word or concept, but also to preserve part of the interactions it maintains with its linguistic environment.

From this perspective, lexical units are no longer considered isolated elements described solely by their semantic proximity to other terms. They are represented as entities embedded within a network of observed or inferred relationships, organised according to mechanisms specific to the architecture. This organisation makes it possible to associate each linguistic element with a richer description of its role within the corpus and the knowledge acquired by the system.

The tensor representation thus constitutes the common foundation upon which CoreMind's other mechanisms rely. Attention, learning, and knowledge exploitation processes use this structure to identify relevant relationships, integrate new information, and produce the expected behaviours of models built upon the architecture.

The objective is not merely to increase the amount of information associated with each linguistic unit, but also to organise this information in a way that facilitates its exploitation by the architecture's internal mechanisms. CoreMind therefore explores the hypothesis that a more structured representation of knowledge may contribute to reducing the complexity of certain operations traditionally performed during processing and inference stages.

2.3.2 CoreMind Attention

The second fundamental pillar of the CoreMind architecture is its attention mechanism. As in many language processing architectures, attention plays a central role in identifying the relevant relationships required for the interpretation and exploitation of linguistic knowledge. However, CoreMind relies on its own attention mechanism, designed in accordance with its tensor representation and internal organisational principles.

Within the CoreMind architecture, attention is not limited to evaluating proximity between linguistic elements. It contributes to identifying meaningful relationships within internal representations, dynamically constructing the context of interpretation, and exploring the semantic structures involved in inference processes.

This approach is based on the hypothesis that some of the relational information required for language interpretation can be exploited directly from the knowledge organised within the tensor representation. Attention then acts as a mechanism for selecting, weighting, and activating relevant relationships according to the context under consideration and the objective pursued by the model.

The attention mechanism also contributes to the exploration of different interpretative trajectories within the knowledge space constructed by the architecture. This capability makes it possible to envisage inference processes based not only on the local analysis of relationships between linguistic elements, but also on the exploitation of broader relational structures capable of connecting multiple concepts or sequences of concepts.

In CoreMind, attention therefore constitutes a cross-cutting mechanism involved in all capabilities built upon the architecture. Whether applied to semantic search, analogies, lexical prediction, language generation, or other applications, it contributes to identifying and exploiting the information most relevant to the context under consideration.

The objective of this mechanism is not merely to assess the relative importance of the available information, but also to facilitate the construction of contextual representations adapted to the requirements of the different models derived from the architecture. Attention thus constitutes one of the principal mechanisms connecting represented knowledge with the inference processes that exploit it.

2.3.3 Adaptive Learning

The third fundamental pillar of the CoreMind architecture is its adaptive learning mechanism. It is based on an approach in which knowledge is constructed directly from observed information and then progressively enriched as the system evolves.

Unlike approaches based on successive optimisation phases requiring multiple passes over the data, CoreMind explores a mode of knowledge construction in which representations and their associated weights are established directly from observation of the corpus. Once this construction phase has been completed, models derived from the architecture become immediately usable for their intended tasks.

Adaptive learning, however, is not limited to this initial phase. The architecture was designed to enable the subsequent integration of new knowledge without requiring full model retraining. Newly acquired information can be progressively incorporated into existing structures, leading to the evolution of representations and the updating of previously acquired knowledge.

This capacity for incremental enrichment constitutes an important property of the architecture. It makes it possible to envisage models capable of evolving in step with the information provided to them, while preserving the continuity of previously acquired knowledge. New information is not simply added to the system; it can also contribute to enriching or reorganising relationships already established between knowledge elements.

Adaptive learning thus provides continuity between the initial construction of the model and its subsequent evolution. It enables the architecture to maintain the overall coherence of its knowledge while retaining the ability to progressively integrate new information without returning to an initial state.

Finally, this approach opens the way to models that can be initialised with a limited amount of knowledge and then progressively enriched throughout their life cycle. CoreMind therefore explores a vision in which the construction, exploitation, and evolution of knowledge form a continuous process rather than a succession of strictly separate phases.

2.3.4 CoreMind Organisation and Algorithms

The fourth fundamental pillar of CoreMind encompasses the overall organisation of the architecture and the algorithmic mechanisms that coordinate its various components. While the tensor representation defines how knowledge is structured, attention identifies relevant relationships, and adaptive learning enables knowledge to evolve, the internal organisation of the architecture provides the framework within which these different mechanisms can operate coherently.

The CoreMind architecture does not rely on a single component responsible for producing all of its capabilities. The observed behaviours result from coordinated interactions among several specialised mechanisms that exploit the knowledge represented within the system. The overall organisation of the architecture was designed to enable these mechanisms to cooperate in producing capabilities suited to a variety of objectives.

This organisation is intended not only to manage the knowledge stored within internal representations, but also to transform that knowledge into operations that can be exploited by the different models built upon the architecture. The internal mechanisms therefore provide the connection between representational structures, attention processes, knowledge evolution, and the expected capabilities of derived systems.

From this perspective, CoreMind algorithms should not be regarded as independent modules dedicated to particular tasks. They constitute a coherent set of mechanisms that enable knowledge to be exploited through different modes of use. Semantic search, conceptual clustering, prediction, generation, and information retrieval capabilities therefore result from specific configurations built upon the same architectural foundation.

One of the objectives of this organisation is to preserve the unity of the architecture while enabling a wide range of applications. The different models that can be built upon CoreMind therefore share the same fundamental principles of representation, attention, and learning, while employing these mechanisms in ways adapted to their specific objectives.

CoreMind organisation and algorithms thus constitute the operational link between the knowledge represented within the architecture and the observable behaviours of the models derived from it. They contribute to the overall coherence of the system and help transform a structured body of knowledge into capabilities that can be exploited across different application contexts.

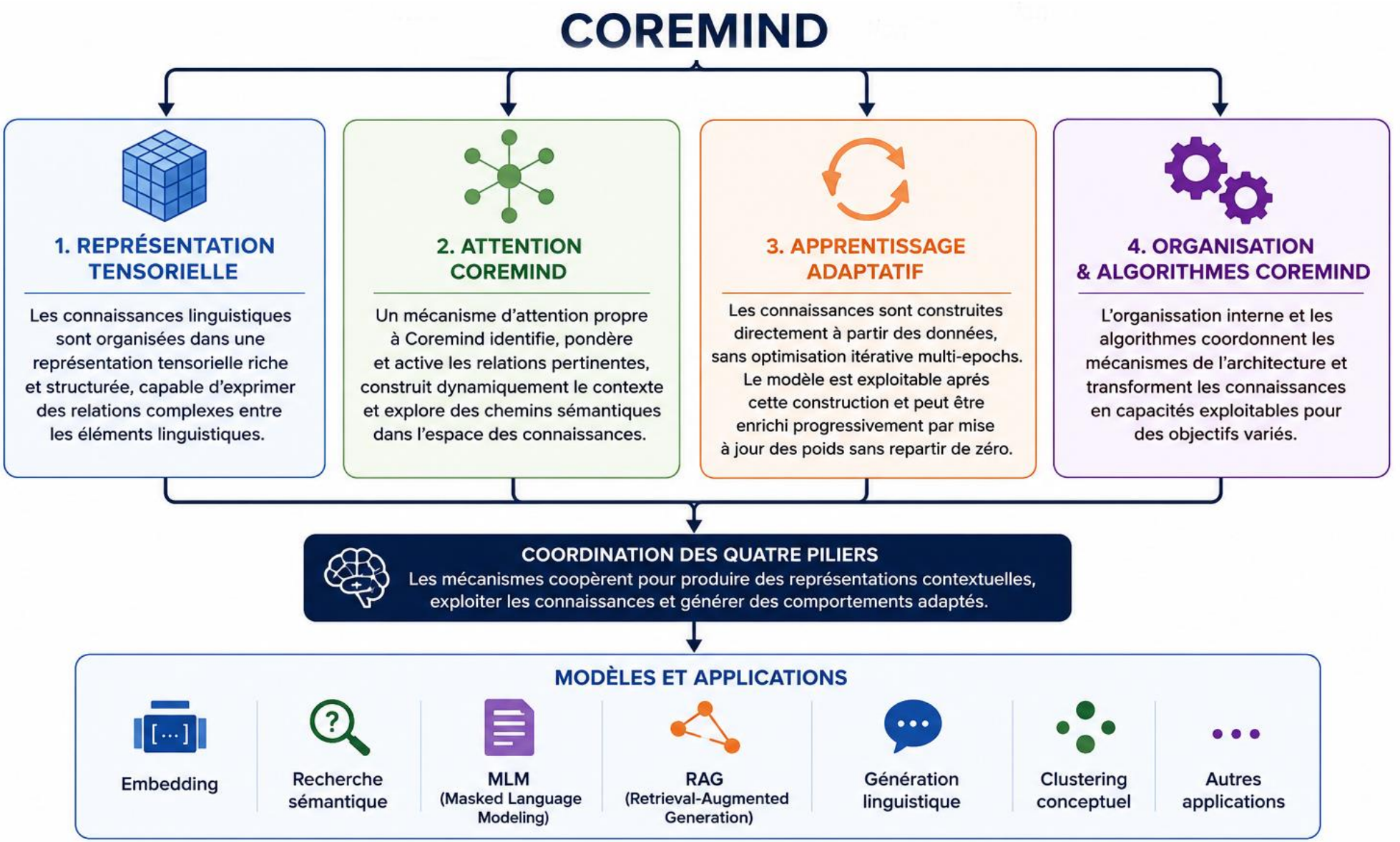


Figure 1 — *The Four Fundamental Pillars of CoreMind and Their Interactions*

3. CoreMind Tensor Representation

3.1 Limitations of Classical Vector Representations

Vector representations are currently one of the foundations of modern Natural Language Processing architectures. They make it possible to associate each lexical unit with a compact numerical representation capable of capturing certain semantic properties learned from large data corpora. These approaches have demonstrated their effectiveness across numerous tasks, including semantic search, classification, translation, and text generation [5][6].

Despite their performance, these representations generally rely on a principle of information compression. A word, concept, or linguistic unit is described by a fixed-dimensional vector in which various properties acquired during learning are condensed. This compact representation enables efficient data exploitation, but also means that a substantial amount of information remains implicit within the model.

In this context, some of the semantic, contextual, or structural relationships associated with linguistic units are not directly represented in an explicit form. These relationships must be reconstructed or reinterpreted by the architecture's internal mechanisms during inference. The performance observed in modern models therefore depends not only on the quality of the representations used, but also on the ability of the processing mechanisms to exploit and reconstruct the knowledge encoded within them.

This observation should not be regarded as a limitation in the sense of a shortcoming of existing approaches. Rather, it highlights an architectural choice: favouring compact representations in which part of the relational information remains implicit. Other strategies may nevertheless be considered in order to explore more structured or more explicit representations.

It is within this perspective that the CoreMind architecture is situated. Rather than seeking solely to increase the capacity of processing mechanisms, CoreMind explores the hypothesis that part of this complexity can be shifted towards the representation itself. The objective is to investigate the extent to which a richer representation of linguistic knowledge can facilitate its subsequent exploitation by the architecture's attention, learning, and inference mechanisms.

3.2 Principles of the CoreMind Tensor Representation

The tensor representation constitutes one of the central elements of the CoreMind architecture. It is based on the idea that a knowledge representation should not merely encode information in a form that can be processed by a machine, but should also organise that information in a way that facilitates its use by the architecture's various mechanisms.

From this perspective, CoreMind explores an approach in which the representation does not seek solely to capture the meaning of a linguistic unit. It also aims to integrate, within a single structure, information relating to its context, its relationships with other lexical units, and the interactions observed or inferred during the construction of knowledge.

The objective is not merely to increase the amount of information associated with a representation. The architecture also seeks to organise this information coherently in order to facilitate its subsequent exploitation by attention, learning, and inference mechanisms. The representation thus becomes a structured space in which knowledge can be described, connected, and mobilised according to different requirements.

This approach is based on the hypothesis that part of the complexity traditionally handled during processing stages can be shifted towards the way in which knowledge is represented and organised. A richer and more structured representation may therefore help facilitate the identification of relevant relationships, the interpretation of context, and the exploitation of acquired knowledge.

Within CoreMind, linguistic units are therefore not regarded as isolated entities associated with a simple numerical description. Instead, they form part of an organised set of relationships and interactions that contribute to the construction of their representation. This organisation constitutes one of the foundations of the architecture's ability to exploit knowledge in different forms and for different purposes.

The CoreMind tensor representation should therefore not be regarded as a simple extension of classical vector representations, but rather as the exploration of a different approach to structuring linguistic knowledge. It constitutes the foundation upon which the architecture's other mechanisms rely and plays a central role in all capabilities built upon CoreMind.

3.3 Contextual Horizons

Among the fundamental mechanisms of the CoreMind tensor representation is the concept of the contextual horizon. Horizons constitute an organisational mechanism specific to the architecture, enabling knowledge and relationships observed within the corpus to be structured according to different contextual perspectives.

Within CoreMind, knowledge is not represented as a simple collection of isolated relationships. Instead, it is organised according to a contextual structure that preserves different forms of interaction between linguistic units. Horizons contribute to this organisation by providing a framework for structuring relationships observed or constructed during the knowledge acquisition process.

The objective is not merely to represent the lexical units themselves, but also to preserve part of the relational environment in which they appear and evolve. Horizons therefore contribute to enriching knowledge representation by incorporating information relating to the contexts and interactions that contribute to their interpretation.

Horizons should not be regarded simply as a measure of proximity or as a direct representation of the position of linguistic elements. They constitute a more general mechanism of contextual organisation, enabling knowledge to be structured according to multiple levels of relationships and interactions. This organisation contributes to the richness of the tensor representation and to the architecture's ability to exploit knowledge from different perspectives.

The architecture does not impose a fixed number of horizons. Their configuration depends on the objectives of the model, the domain under consideration, or the choices made during its construction. This flexibility makes it possible to adapt the organisation of knowledge to the specific requirements of different instantiations of the architecture.

Horizons therefore play a central role in the CoreMind tensor representation. They constitute one of the mechanisms for organising knowledge beyond its mere presence in the corpus and contribute to the construction of a representational space capable of capturing a diversity of contextual relationships within a single coherent structure.

3.4 Overall Organisation of Knowledge

The CoreMind tensor representation is not limited to the individual encoding of linguistic units. It forms part of a broader organisation of knowledge aimed at structuring information, relationships, and interactions within a coherent space that can be exploited by the architecture.

From this perspective, knowledge is not regarded as a simple collection of independent elements. Instead, it contributes to the construction of an organised network of relationships in which each linguistic unit can be associated with different contexts, interactions, and levels of interpretation. This organisation makes it possible to move beyond a strictly local view of knowledge in favour of an approach based on the connections between its constituent elements.

The architecture exploits both relationships directly observed during the knowledge construction process and relationships that progressively emerge from the architecture's internal mechanisms. The represented knowledge is therefore not limited to recording information observed in the corpus; it may also incorporate relational structures resulting from the exploitation and organisation of this information.

This capability contributes to the formation of a knowledge space in which concepts can be connected through different forms of interaction. The represented relationships are not necessarily homogeneous and may simultaneously contribute to multiple levels of knowledge structuring. This organisation facilitates the exploration of existing connections between concepts as well as the identification of new connections that can be exploited by the different models built upon the architecture.

One of the objectives of this organisation is to enable flexible exploitation of knowledge. Attention, learning, and inference mechanisms can therefore mobilise different parts of this relational space according to the context under consideration and the task at hand. The representation thus becomes not only a means of storing knowledge, but also an active space for organising and exploiting relationships.

This perspective leads to knowledge being regarded as an evolving structure rather than a fixed collection of information. As new knowledge is integrated, the overall organisation can be enriched, reorganised, and give rise to new relationships that may be exploited by the architecture. The CoreMind representational space thus constitutes a dynamic environment in which observed knowledge, inferred relationships, and interactions between concepts jointly contribute to the construction of linguistic understanding.

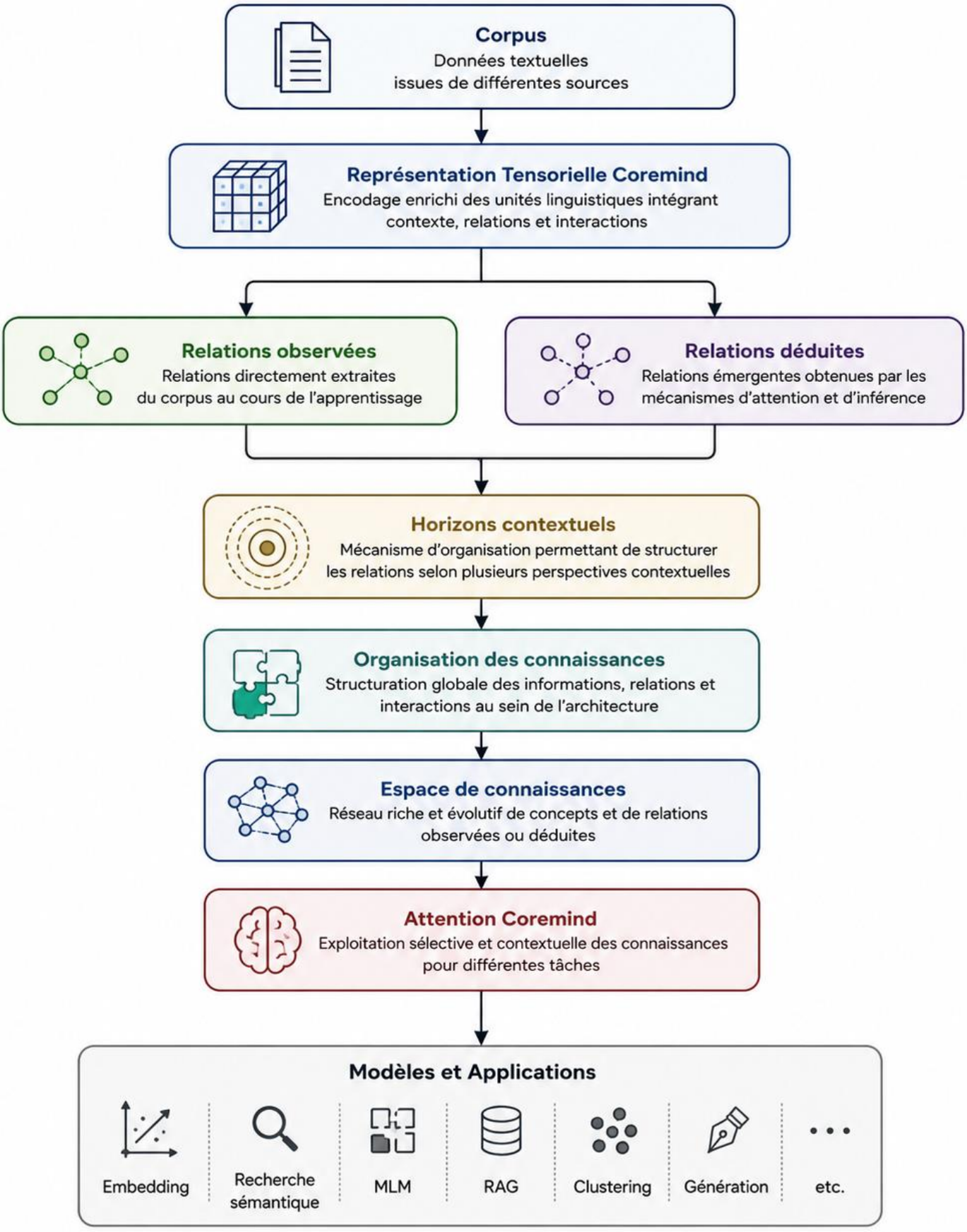


Figure 2 – *Conceptual Organisation of Knowledge within the CoreMind Architecture*

Knowledge is represented, organised, and exploited through a set of complementary mechanisms designed to support a wide variety of models and applications.

3.5 Architectural Implications

The tensor representation, contextual horizons, and overall organisation of knowledge jointly constitute the foundation upon which the CoreMind architecture is built. Beyond their role in structuring information, these mechanisms directly influence how knowledge can be exploited within the system.

One consequence of this organisation is the ability to exploit linguistic relationships already integrated into the knowledge representation. Rather than treating lexical units as isolated elements whose relationships must be systematically reconstructed during processing, the architecture explores an approach in which part of the contextual and relational information is directly available within the representation itself.

This property helps facilitate the identification of relevant relationships and the exploitation of acquired knowledge. Attention mechanisms can therefore rely on a previously organised knowledge structure, while inference processes can mobilise observed or inferred relationships across different levels of context.

The organisation of knowledge also supports the progressive evolution of the model. New information can be integrated into existing structures while preserving previously acquired knowledge. This capacity for incremental enrichment enables the architecture to adapt to evolving corpora and knowledge domains without requiring full retraining of the system.

Another important consequence lies in the flexibility of the models that can be built upon the architecture. The same fundamental mechanisms for knowledge representation, attention, and organisation can be employed to address a variety of objectives. This approach makes it possible to envisage the construction of specialised models while retaining a common architectural foundation.

Finally, the knowledge organisation explored by CoreMind opens the way to different strategies for language processing and exploitation. These include semantic search, conceptual clustering [11][12], lexical prediction, and language generation mechanisms. The architecture also enables the exploration of approaches based on the evaluation of relational structures or complete linguistic trajectories rather than solely on the local prediction of successive elements.

These different implications should not be regarded as independent functionalities added to the architecture. They result from the interaction between the mechanisms for representing, organising, and exploiting knowledge that constitute the very foundation of CoreMind. They illustrate how an approach centred on knowledge structuring can lead to the emergence of diverse capabilities within a unified architectural framework.

4. CoreMind Attention

4.1 Role of Attention in CoreMind

Attention constitutes one of the central mechanisms of the CoreMind architecture. While the tensor representation organises knowledge and adaptive learning enables its evolution, attention ensures that this knowledge is mobilised according to the context and the objectives pursued by the model.

In a language processing system, not all available information is equally important at any given moment. Understanding a sentence, interpreting a concept, or performing a particular task generally requires certain relationships, contexts, or knowledge to be prioritised among all the information represented within the system.

CoreMind attention is designed to identify and weight the elements most relevant to a given situation. It contributes to the construction of the interpretative context by selecting relationships that are likely to contribute to the understanding or exploitation of the knowledge being mobilised.

This function is not limited to the evaluation of local relationships between linguistic units. Attention also contributes to the exploration of broader relational structures within the knowledge space constructed by the architecture. It therefore enables information distributed across different levels of organisation within the representation to be exploited.

One of CoreMind's objectives is to establish a close relationship between knowledge representation and knowledge exploitation. From this perspective, attention is not conceived as a mechanism independent of the representation, but as a process that relies directly on the organisation of knowledge constructed by the architecture. The relationships, contexts, and interactions present within the representation therefore constitute the primary material exploited by the attention mechanisms.

Attention also plays a key role in the architecture's ability to dynamically adapt its interpretation of language. Depending on the context under consideration, the knowledge being mobilised may vary, allowing the system to explore different analytical perspectives while maintaining overall coherence with the represented information.

Thus, within CoreMind, attention constitutes the mechanism that connects the knowledge organised within the tensor representation with the exploitation processes that arise from it. It contributes directly to the construction of meaning, the identification of relevant relationships, and the contextual adaptation of models built upon the architecture.

4.2 CoreMind Global Attention

Global attention constitutes the first level of knowledge exploitation within the CoreMind attention mechanism. Its objective is to construct a coherent view of the linguistic context from the relationships represented between the different units that make up a sentence or text sequence.

From this perspective, linguistic units are not analysed independently of one another. Each element participates in a set of interactions that contribute to the construction of the overall context. Global attention aims to exploit these interactions in order to establish a coherent interpretation of the linguistic structure under consideration.

This global representation does not result solely from the individual analysis of the different elements present in the sentence. It also emerges from the organisation of the relationships connecting them. Global attention can therefore be regarded both as a synthesis of the local interactions observed

between linguistic units and as a unified representation of the context constructed from these interactions.

One of the roles of this mechanism is to identify the relationships that contribute most strongly to understanding the sequence under consideration. Depending on the context, certain interactions may be prioritised while others play a secondary role. Global attention therefore contributes to establishing a dynamic hierarchy of the information mobilised during interpretation.

Within CoreMind, this construction of the global context relies directly on the knowledge organised within the tensor representation. Observed relationships, represented interactions, and available contextual structures constitute the elements from which global attention constructs its overall view of the sequence.

Global attention therefore plays a central role in understanding complex linguistic structures. It makes it possible to move beyond a strictly local analysis of lexical units and construct a coherent interpretation that simultaneously integrates multiple levels of relationships and interactions.

This global view constitutes the starting point for CoreMind's more specialised attention mechanisms. It provides the general contextual framework from which different forms of analysis can subsequently be developed to exploit specific characteristics of the represented knowledge.

4.3 Specialised Attention and Horizons (CoreMind Multi-Head Attention)

Beyond the construction of a global view of the context, the CoreMind architecture relies on specialised attention mechanisms that enable different characteristics of the represented knowledge to be explored. These mechanisms complement global attention by providing multiple analytical perspectives capable of highlighting particular properties of linguistic relationships.

Specialised attention relies on the contextual horizons introduced in the previous chapter. Horizons are not merely a mechanism for organising knowledge; they also provide a framework through which different attention processes can explore represented information from complementary perspectives. This relationship between horizons and attention contributes to the richness of the contextual analysis performed by the architecture.

Each specialised mechanism can therefore prioritise certain characteristics of the represented knowledge. These specialisations should not be interpreted as fixed rules or predefined categories. They arise both from principles specific to the architecture and from the progressive organisation of the knowledge constructed throughout its evolution.

This diversity of perspectives enables CoreMind attention to examine several dimensions of the relationships present within internal representations simultaneously. The different specialised mechanisms therefore contribute to identifying structures, interactions, or contexts that may be relevant to understanding the sequence under consideration.

The objective is not merely to multiply the available analyses, but to foster the emergence of a richer understanding of the knowledge being mobilised. The information produced by the different specialised mechanisms can therefore contribute collectively to the construction of a coherent interpretation of the linguistic context.

This organisation also enables the architecture to retain considerable flexibility in the exploitation of knowledge. Depending on the requirements of the model or the nature of the task under consideration, certain analytical perspectives may play a more significant role than others. Specialised attention

therefore constitutes a mechanism complementary to global attention, enriching the interpretation of language without compromising the coherence of the overall representation of the context.

Horizons and specialised attention mechanisms thus form a closely interconnected system within the CoreMind architecture. Their interaction contributes to the exploration of knowledge from multiple complementary perspectives and directly supports the system's ability to construct contextual representations adapted to different usage scenarios.

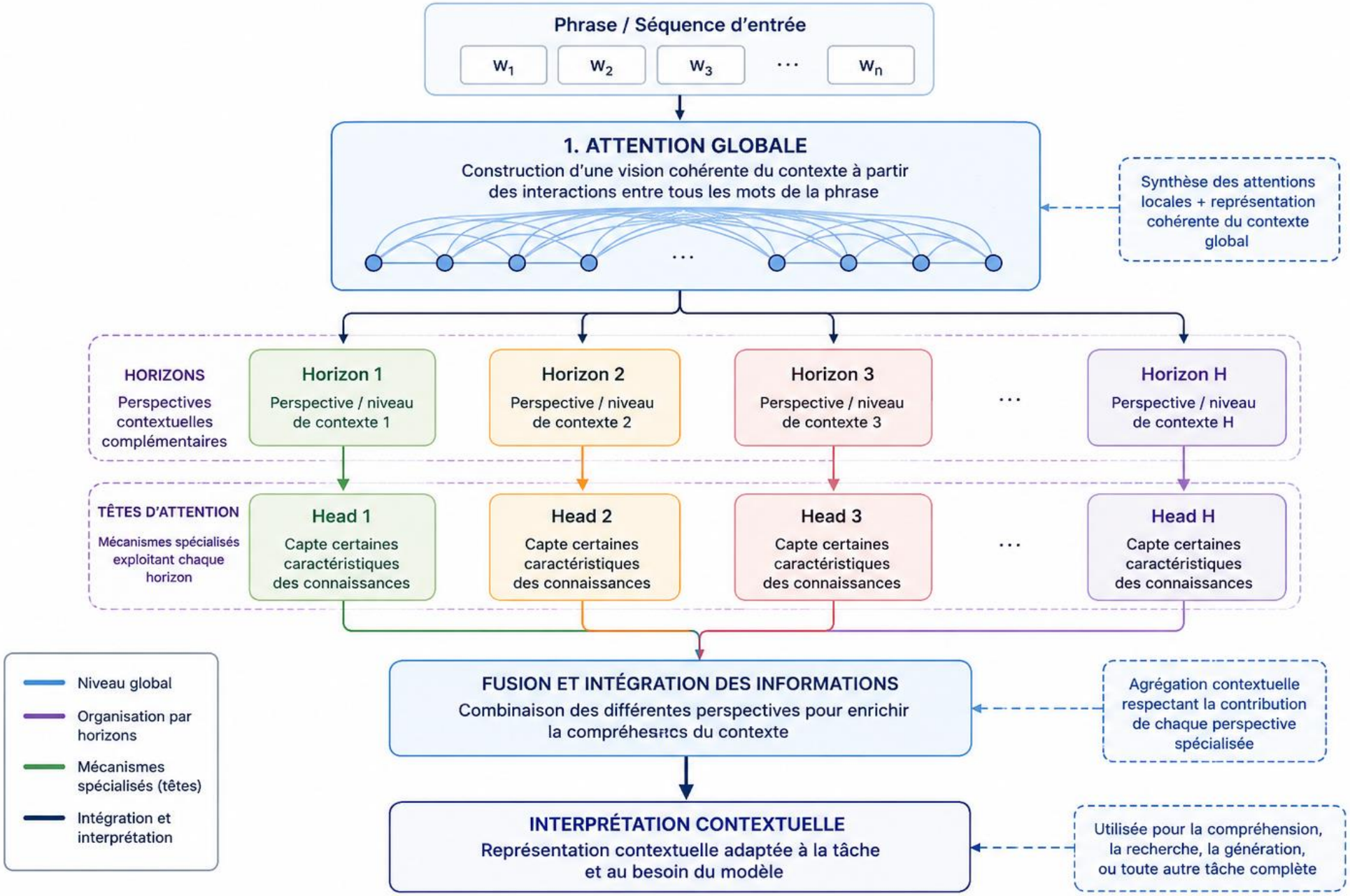


Figure 3 – *Conceptual Overview of the CoreMind Attention Mechanism*

4.4 Dynamic Construction of Meaning

One of the fundamental roles of CoreMind attention is to contribute to the dynamic construction of meaning from the knowledge represented within the architecture. In this approach, meaning is not regarded as a fixed property definitively associated with a linguistic unit. Rather, it results from the interaction between available knowledge and the context in which that knowledge is mobilised.

The representations constructed by the architecture contain semantic, contextual, and relational information organised within a coherent knowledge space. However, the mere presence of this information is not sufficient to produce an interpretation. Interpretation emerges through the contextual exploitation of this information by the attention mechanisms.

Global attention contributes to constructing an overall view of the sequence under consideration, while specialised mechanisms explore different characteristics of the knowledge being mobilised. The interaction between these different levels of analysis enables a contextual representation adapted to the situation under consideration to emerge.

This construction of meaning depends directly on context. The same knowledge can contribute to different interpretations depending on the relationships activated, the knowledge mobilised, and the interactions considered during the attention process. Meaning is therefore not contained solely within the linguistic elements themselves, but also within the relational structures that connect them within the knowledge space.

This approach enables the architecture to take multiple levels of information into account simultaneously. Local relationships, broader contexts, and organisational structures present within the knowledge can therefore jointly contribute to the understanding of the same linguistic sequence.

The dynamic construction of meaning also constitutes one of the mechanisms enabling CoreMind to adapt its interpretation to different tasks or situations. The knowledge mobilised and the relationships prioritised may vary according to the objectives pursued, while maintaining overall coherence with the organisation of the represented knowledge.

Thus, within CoreMind, meaning emerges as a property resulting from the interaction between knowledge representation, its contextual organisation, and the attention mechanisms that exploit it. This perspective places relationships and contexts at the heart of the linguistic interpretation process and constitutes one of the conceptual foundations of the architecture.

4.5 Architectural Implications

CoreMind attention constitutes one of the principal mechanisms for exploiting knowledge within the architecture. Its integration with the tensor representation and the overall organisation of knowledge produces several architectural implications that directly influence the capabilities of models built upon CoreMind.

A first implication lies in the architecture's ability to construct dynamic contextual representations adapted to each analytical situation. The interpretation of a linguistic sequence depends not only on the knowledge represented, but also on how that knowledge is mobilised and organised during the attention process. This property enables the system to adapt its analysis to the specific characteristics of the context under consideration.

The combined organisation of global attention and specialised mechanisms also makes it possible to exploit multiple analytical perspectives simultaneously. Different characteristics of the knowledge can therefore be mobilised in complementary ways to enrich the understanding of relationships present

within linguistic data. This diversity of perspectives contributes to the robustness of the interpretations produced by the architecture.

CoreMind attention also facilitates the exploitation of complex relational structures within the knowledge space. Attention mechanisms can mobilise observed or inferred relationships across different levels of contextual organisation, thereby enabling the exploration of interactions that extend beyond local associations between linguistic units.

This capability contributes to the emergence of new relationships that can be exploited by models built upon the architecture. Represented knowledge can be analysed from different perspectives, revealing connections or structures that are not necessarily explicit in the information initially observed.

Attention also plays an important role in the generality of the architecture. The same fundamental mechanisms can be mobilised across a variety of contexts, including semantic search, information retrieval, relational analysis, classification, conceptual clustering, and language generation. This property contributes to CoreMind's ability to serve as the foundation for a family of specialised models sharing the same fundamental principles.

Finally, the contextual organisation of knowledge opens the way to the exploration of inference strategies based on the analysis of relational structures broader than simple local dependencies. This property constitutes one of the elements enabling advanced mechanisms for knowledge exploitation to be envisaged, particularly in the areas of semantic search, conceptual exploration, and language generation.

Thus, the capabilities observed in models built upon CoreMind do not result from an isolated attention mechanism, but from the interaction between knowledge representation, its contextual organisation, and the attention processes that exploit it. Attention constitutes the operational link that enables a structured knowledge space to be transformed into capabilities adaptable to a wide variety of tasks.

5. Learning and Knowledge Evolution

5.1 Initial Knowledge Construction

Learning within the CoreMind architecture is based on an approach centred on the direct construction of knowledge from the information available in the corpus. This phase constitutes the starting point for the organisation of the representations, relationships, and structures subsequently exploited by the attention and inference mechanisms.

From this perspective, the primary objective is not merely to progressively adjust a set of parameters through successive optimisation phases. Above all, the architecture seeks to construct a coherent knowledge space from observed information, identified relationships, and organisational structures produced during the construction process.

Initial knowledge therefore results from the combined exploitation of relationships observed in the corpus and organisational mechanisms specific to the architecture. Acquired information is not simply recorded in a static form; it contributes to the construction of a structured knowledge network integrating contexts, interactions, and relationships.

This organisation also enables the emergence of relationships that are not necessarily explicit in the observed data. Knowledge structuring mechanisms contribute to revealing new exploitable

connections between represented elements, progressively enriching the knowledge space constructed by the architecture.

The initial construction of knowledge therefore constitutes a foundational stage in the operation of CoreMind. It provides the basis upon which the tensor representation, contextual attention, and the capabilities of models derived from the architecture rely. The weights and internal structures produced at this stage reflect not only the information observed in the corpus, but also the relational organisation constructed by the system.

This approach leads to the initial learning phase being regarded as a process of knowledge construction rather than simply a phase of parameter adjustment. The objective is to produce an exploitable representation of knowledge from its initial organisation, while retaining the possibility of enriching and evolving it subsequently.

5.2 Dynamic Construction and Organisation of Weights

Within the CoreMind architecture, weights are not regarded simply as numerical parameters resulting from an optimisation process. They constitute elements that directly contribute to the organisation of represented knowledge and evolve jointly with it throughout the model's life cycle.

During the initial construction phase, weights emerge from the combined exploitation of relationships observed in the corpus, organisational mechanisms specific to the architecture, and inferred relationships produced within the knowledge space. They therefore reflect not only the information acquired, but also the way in which this knowledge is structured and interconnected.

From this perspective, weights do not represent a fixed memory of knowledge. They actively contribute to its internal organisation and its exploitation by the architecture's various mechanisms. Knowledge representation, its organisation, and the weights that support it therefore form a coherent whole whose different components evolve interdependently.

This dynamic organisation enables the architecture to progressively integrate new knowledge without restricting weights to a simple storage role. Depending on the nature of the information introduced, its integration may lead to a local evolution of certain structures or to a broader reorganisation of the represented relationships. The weights then evolve jointly with the organisation of knowledge in order to preserve the overall coherence of the system.

This capacity for evolution does not concern only the addition of new information. The architecture's updating mechanisms also make it possible to adapt representations when existing knowledge is modified, enriched, or removed. The weights evolve alongside these changes in order to maintain a coherent organisation of knowledge without calling the entire constructed structure into question.

The CoreMind architecture therefore regards weights as a dynamic component of its knowledge space rather than as a set of parameters definitively established after a training phase. Their role extends beyond the simple memorisation of acquired information: they contribute to the structuring, evolution, and exploitation of the knowledge mobilised by models built upon the architecture.

This conception constitutes one of the foundations of CoreMind's adaptive learning. It contributes to bringing knowledge construction, its organisation, and its continuous evolution together within a single architectural process, enabling the model to maintain overall coherence while progressively adapting to new information provided to it.

5.3 Incremental Knowledge Evolution

The CoreMind architecture was designed to enable knowledge to evolve progressively throughout its life cycle. Unlike approaches based on a strict separation between a training phase and a usage phase, CoreMind regards knowledge evolution as a continuous process accompanying the exploitation of the model.

The integration of new knowledge does not consist solely of adding additional information to an existing structure. Each new contribution may interact with knowledge already represented, reinforce certain relationships, give rise to new ones, or lead to a partial reorganisation of the knowledge space when necessary.

This evolution nevertheless remains progressive. The architecture's objective is not to systematically reconstruct the entire body of knowledge whenever a modification occurs, but to preserve the continuity of existing structures as far as possible while integrating new information coherently. Depending on the nature of the knowledge introduced, changes may remain localised or extend to a broader organisation of the represented relationships.

CoreMind's incremental approach also applies to existing knowledge. Information may be enriched, corrected, or removed during the model's life cycle. The architecture is designed to accommodate these changes while maintaining the coherence of the overall organisation of knowledge and its associated weights.

This capacity for evolution helps preserve the relevance of knowledge in constantly changing information environments. Models built upon CoreMind can therefore progressively integrate new information without calling into question the entire body of previously acquired knowledge.

Incremental evolution therefore constitutes a natural extension of the initial construction of knowledge and the dynamic organisation of weights. These different mechanisms cooperate to maintain a coherent, evolving, and continuously exploitable knowledge space for the architecture's attention and inference processes.

This approach reflects CoreMind's general philosophy, according to which knowledge, its organisation, and its exploitation are not independent stages, but components of a single evolving system capable of progressively adapting to changes in its information environment.

5.4 Initialisation and Usage Modes

The CoreMind architecture was designed to operate according to different initialisation and usage modes, depending on the objectives pursued and the constraints specific to each application. This flexibility is a direct consequence of the separation between the architecture itself and the knowledge it represents.

A first mode of use consists of constructing the model from a large corpus. Knowledge is then organised during the initial construction phase, producing a model that is immediately exploitable for the intended tasks. This approach is particularly suited to general-purpose or specialised models requiring a substantial knowledge base from the outset.

The architecture can also be initialised with a limited or even empty knowledge base. In this configuration, all representation, attention, and organisational mechanisms are available from initialisation, while the knowledge space is progressively enriched as information is integrated into the system. This capability makes it possible to envisage models whose knowledge is progressively constructed according to the requirements of the application.

Between these two approaches, CoreMind also enables the progressive enrichment of an already constructed model. New knowledge can be integrated during its use in order to extend its domain of competence, update existing information, or adapt the model to a new usage context. This evolution builds upon the adaptive learning mechanisms described previously, without altering the underlying architecture.

Finally, the architecture can be used without evolving its knowledge when a stable knowledge base meets the requirements of the application. In this case, the model operates as a system whose knowledge remains unchanged during use, similar to many models currently deployed in production.

These different scenarios do not correspond to distinct architectures, but rather to different modes of using the same architecture. The fundamental mechanisms of representation, attention, organisation, and learning remain identical; only the knowledge management strategy changes according to the requirements of the application.

This flexibility constitutes one of CoreMind's important properties. It enables the architecture to be adapted to a variety of contexts, ranging from models previously constructed on large corpora to systems capable of progressively acquiring their knowledge, while maintaining a single, coherent architectural framework.

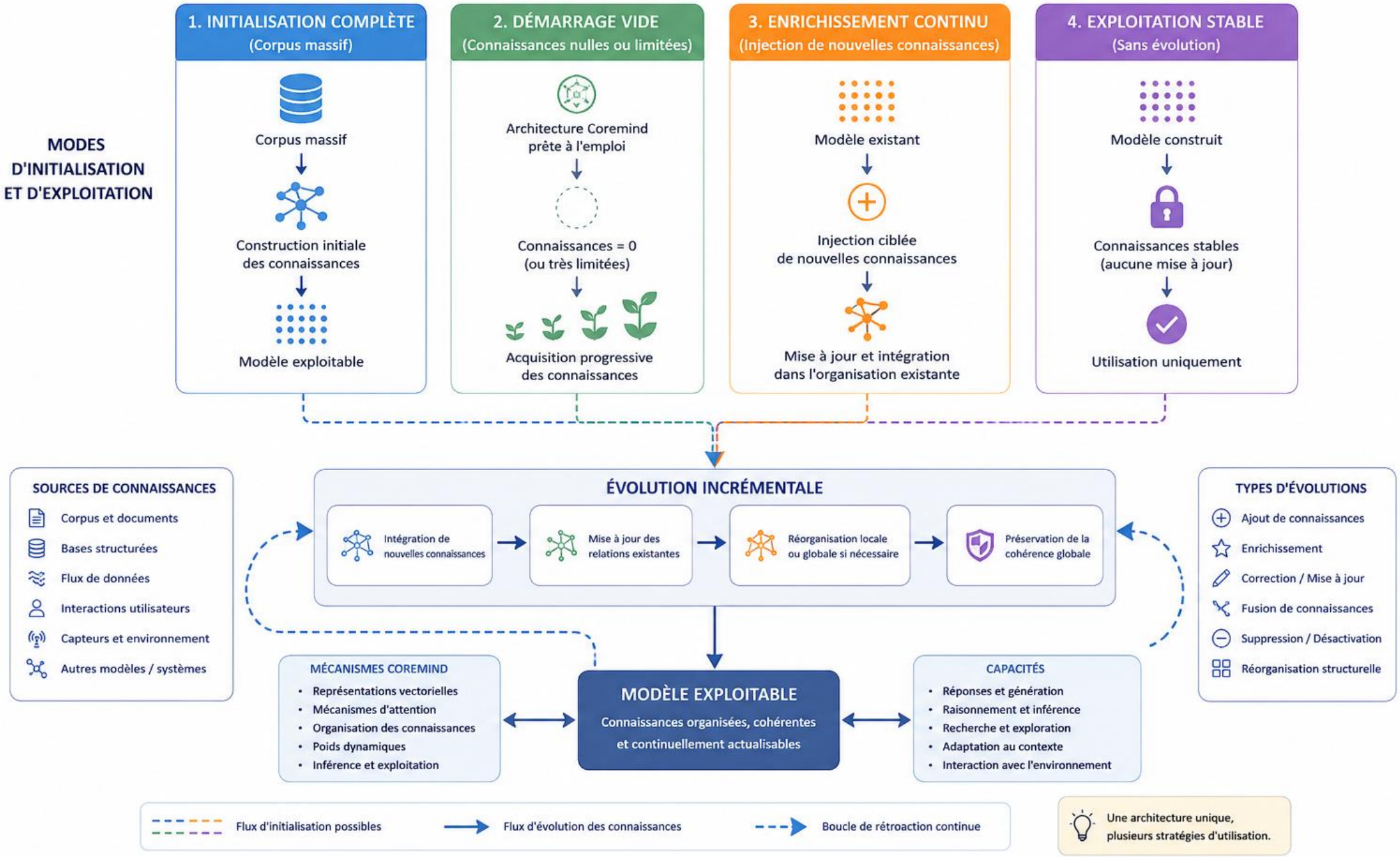


Figure 4 – *CoreMind Initialisation and Evolution Modes*

One Architecture, Multiple Strategies for Knowledge Initialisation and Evolution

5.5 Updating Without Full Retraining

The evolution of knowledge within the CoreMind architecture is based on a principle of continuity. When new information is integrated, existing knowledge is modified, or an element becomes obsolete, the objective is not to reconstruct the entire knowledge space, but to ensure its coherent evolution while preserving established structures as far as possible.

This approach enables the architecture to accommodate the natural evolution of knowledge throughout the model's life cycle. Updating mechanisms are not limited to the addition of new information; they also take into account corrections, enrichments, removals, or adaptations made necessary by changes in the domain under consideration.

Depending on the nature of the changes introduced, modifications may remain localised or lead to a broader reorganisation of certain parts of the knowledge space. This capacity for adaptation makes it possible to preserve the overall coherence of the architecture while progressively integrating new information.

The architecture does not prescribe a single knowledge management policy. The rules governing whether certain information should be retained, replaced, allowed to coexist, archived, or removed are determined by the model built upon CoreMind and by the specific requirements of the application under consideration. The architecture therefore provides a framework capable of supporting different knowledge management strategies without imposing a single behaviour.

The updating mechanisms concern all components involved in the organisation of knowledge. Representations, relationships, weights, and organisational structures evolve jointly in order to preserve the coherence of the system. This co-evolution constitutes one of the fundamental principles of the architecture and ensures continuity between previously acquired knowledge and knowledge integrated subsequently.

Through this approach, CoreMind can accommodate the progressive evolution of a knowledge domain without requiring systematic reconstruction of the model. The architecture therefore favours continuous adaptation of the knowledge space rather than a succession of independent full-retraining phases.

This updating capability constitutes a natural extension of the adaptive learning proposed by CoreMind. It makes it possible to envisage models capable of evolving sustainably in environments where knowledge is progressively enriched, corrected, or reorganised over time, while maintaining a single, coherent architectural framework.

5.6 Architectural Implications

The learning principles proposed by CoreMind naturally extend the representation and attention mechanisms described in the previous chapters. By placing the construction, organisation, and evolution of knowledge at the heart of the architecture, CoreMind proposes a framework in which knowledge constitutes a living component of the system rather than a fixed outcome of a training phase.

One of the principal implications of this approach lies in the architecture's ability to accommodate the progressive evolution of knowledge without systematically calling existing structures into question. Knowledge can be enriched, adapted, or reorganised over time, enabling models built upon CoreMind to maintain their coherence while adapting to changes in their information environment.

This approach also provides considerable flexibility in how the architecture can be used. A model may be constructed from a large initial corpus, begin with a limited knowledge base, or evolve progressively through the integration of new information. These different scenarios rely on the same fundamental mechanisms of representation, organisation, attention, and learning.

The architecture also distinguishes between the fundamental mechanisms it provides and the knowledge management policies implemented by the models derived from it. Strategies for retaining, evolving, replacing, allowing the coexistence of, or removing certain knowledge are determined by the model built upon CoreMind and by the specific requirements of its application domain. This separation contributes to preserving the generic nature of the architecture.

By bringing together representation, organisation, attention, and adaptive learning within a single framework, CoreMind opens the way to a family of models capable of exploiting, enriching, and evolving their knowledge according to a variety of requirements. The capabilities observed in these models therefore result from cooperation between these different mechanisms rather than from the isolated action of a particular component.

Thus, learning within CoreMind does not constitute an independent phase preceding the exploitation of the model. It forms part of a continuous process of knowledge construction, organisation, and evolution, enabling the architecture to maintain overall coherence throughout its life cycle.

6. Capabilities Obtained Through the Exploitation of CoreMind

Chapter Introduction

The previous chapters presented the fundamental principles of the CoreMind architecture, from knowledge representation and organisation to its exploitation through attention mechanisms and its evolution throughout the model's life cycle. Together, these mechanisms constitute a coherent architectural framework, independent of the applications that may be built upon this architecture.

This chapter describes the principal capabilities currently obtained through the exploitation of the CoreMind architecture. These capabilities do not result from an isolated component, but from cooperation between the architecture's various fundamental mechanisms, including the tensor representation, knowledge organisation, attention mechanisms, learning strategies, and exploitation algorithms developed upon this foundation.

It is important to distinguish the foundations of the architecture from the capabilities that arise from them. The tensor representation, attention mechanisms, and adaptive learning are not functionalities in themselves; they constitute the fundamental building blocks for constructing models capable of performing different language processing tasks.

The following sections present the principal capabilities currently demonstrated by the CoreMind prototype. They illustrate how the architecture's various mechanisms can be exploited jointly to perform different tasks, without prejudging other applications that may subsequently be developed upon the same architecture.

6.1 Cooperation Between the Architecture's Mechanisms

The capabilities provided by CoreMind result from the cooperation of several fundamental mechanisms within the architecture. No individual component, considered in isolation, is sufficient to produce the various capabilities presented in this chapter. These capabilities are obtained through the joint exploitation of knowledge representation, its organisation, attention mechanisms, adaptive learning, and the algorithms developed upon this architecture.

The tensor representation provides the structural foundation for knowledge. Attention mechanisms enable this organisation to be exploited according to the context under consideration. Learning processes ensure the construction and evolution of knowledge over time, while exploitation algorithms mobilise all of these components to address a given task.

This distinction between fundamental mechanisms and capabilities constitutes an important principle of CoreMind. The same architecture can therefore be exploited in different ways according to the objectives pursued, without modifying its operating principles. The resulting capabilities depend on the mechanisms mobilised and their cooperation rather than on a single component of the architecture.

The tensor embedding occupies a central position within this organisation, as it constitutes the entry point to the represented knowledge. However, it does not by itself perform the processing tasks presented in this chapter. Its role is to provide structured access to the knowledge organised by the architecture so that the different mechanisms can exploit it coherently.

The following sections illustrate this cooperation through the principal capabilities currently demonstrated by the CoreMind prototype. Each relies on a specific combination of the architecture's fundamental mechanisms, adapted to the nature of the task under consideration.

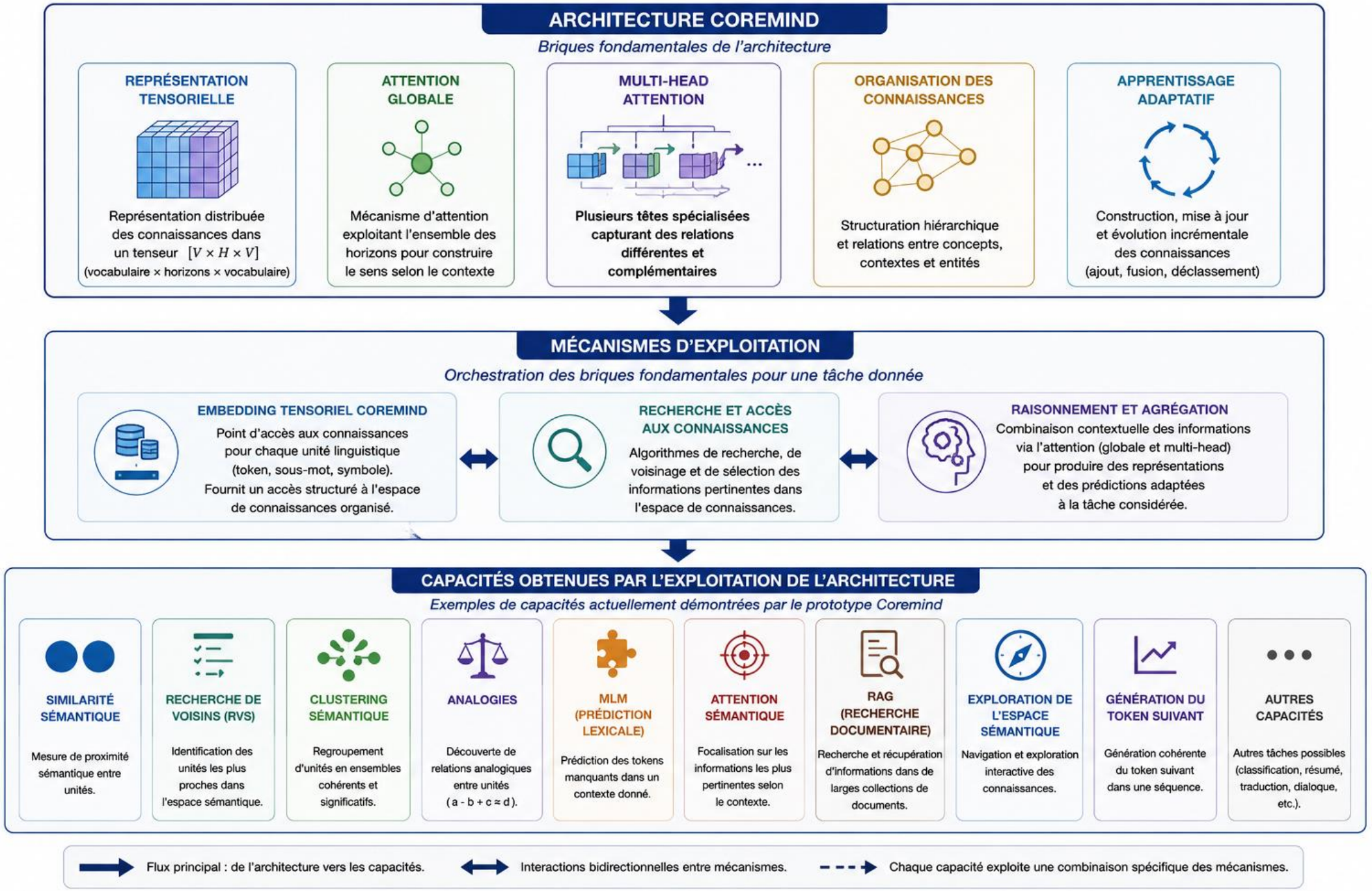


Figure 5 – *CoreMind Architecture: From Fundamental Mechanisms to Resulting Capabilities*

6.2 CoreMind Tensor Embedding

The CoreMind tensor embedding constitutes the entry point to knowledge within the architecture. Its objective is not merely to represent a linguistic unit in an exploitable form, but to provide structured access to the knowledge associated with it.

Unlike a representation limited to the description of a word or token [5][6], the CoreMind embedding is designed to make accessible the information organised during the construction of knowledge. The semantic, contextual, and relational aspects associated with a linguistic unit can therefore be mobilised by the architecture's various mechanisms according to the requirements of the task under consideration.

This representation provides a common foundation for all processing performed by models built upon CoreMind. Similarity mechanisms, neighbour search, conceptual clustering, analogy, lexical prediction, information retrieval, attention, and generation all rely on this same point of access to knowledge, while implementing their own exploitation strategies.

The objective of this approach is not merely to increase the descriptive richness of representations, but to make an already constructed organisation of knowledge directly exploitable. Subsequent processing can therefore mobilise this organisation without having to systematically reconstruct certain information required for its operation.

The CoreMind embedding therefore does not constitute a capability in itself. It represents one of the fundamental components of the architecture, providing the various mechanisms with coherent access to the organised knowledge space. The capabilities presented in the following sections result from the exploitation of this representation in combination with CoreMind's other fundamental mechanisms.

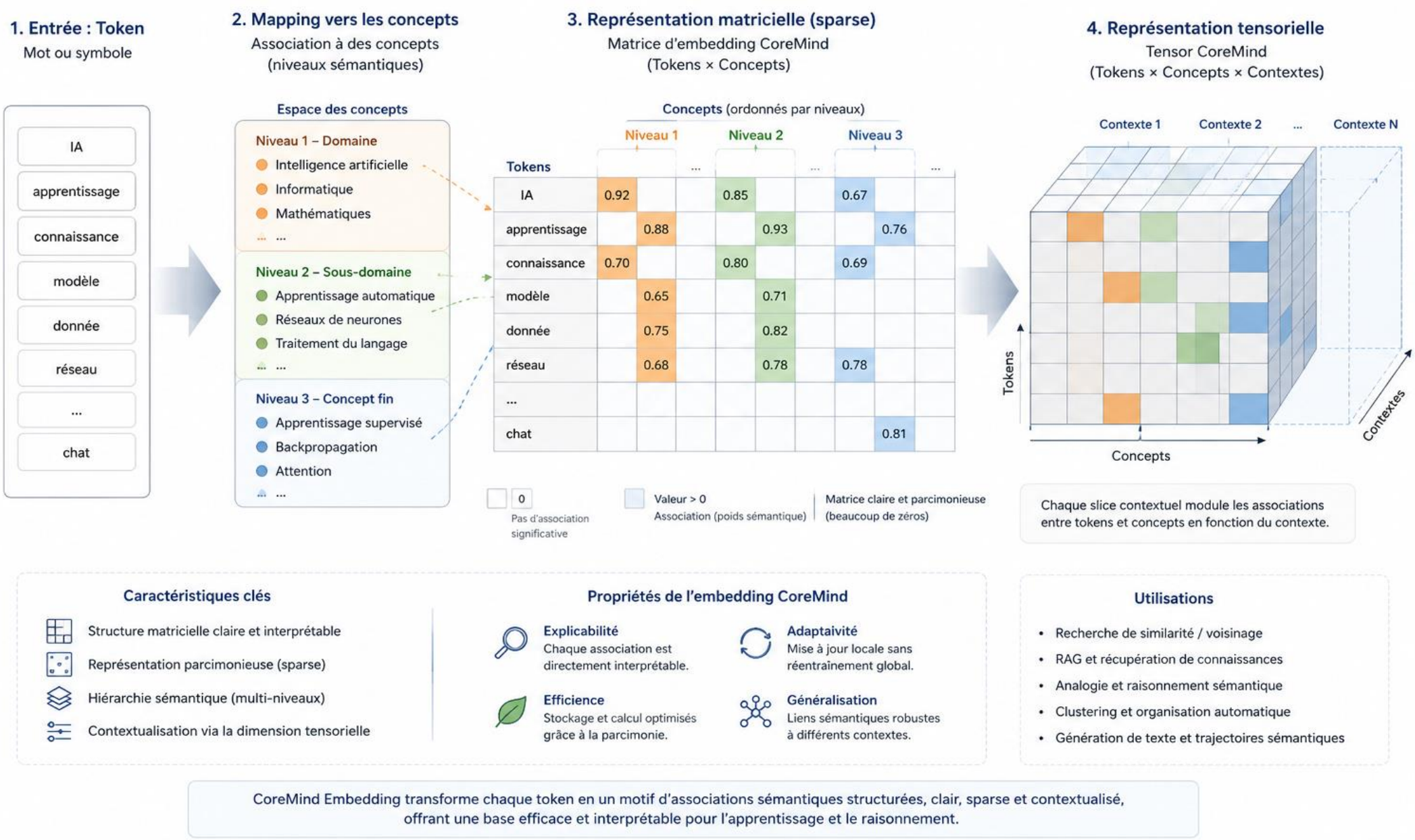


Figure 6 – *CoreMind Embedding: Matrix-Based and Tensor-Based Semantic Representation of Knowledge*

6.3 Semantic Similarity

Semantic similarity constitutes one of the first capabilities obtained through the exploitation of the CoreMind architecture [5][6]. It makes it possible to evaluate the degree of proximity between different linguistic units by jointly mobilising the architecture's fundamental mechanisms, including knowledge representation, its organisation, attention mechanisms, and the exploitation algorithms developed upon this foundation.

Unlike an approach based exclusively on the comparison of numerical representations, CoreMind considers similarity as the result of exploiting an organised knowledge space. The evaluation of proximity between two linguistic units depends not only on their respective representations, but also on the knowledge associated with them, the relationships constructed within the architecture, and the context in which the comparison is performed.

This approach makes it possible to produce a similarity measure based both on the semantic proximity of the units under consideration and on the organisation of the knowledge represented within the architecture. Two concepts may therefore be considered similar not only because their representations share common characteristics, but also because they share contexts, relationships, or coherent knowledge structures.

Semantic similarity does not constitute an end in itself. It represents a fundamental mechanism exploited by many other capabilities of the architecture. Semantic neighbour search, conceptual clustering, analogy construction, exploration of the knowledge space, information retrieval, and certain generation mechanisms rely directly on this capability to identify, compare, and select the most relevant knowledge according to the objectives pursued.

The reuse of the same similarity mechanism across different contexts illustrates one of CoreMind's fundamental principles. Rather than developing independent mechanisms for each task, the architecture favours the exploitation of a common foundation of organised knowledge upon which the different processing operations performed by models built upon CoreMind can rely.

Experiments conducted on the prototype show that this capability constitutes one of the architecture's essential building blocks. It provides the foundation for several capabilities currently demonstrated and naturally supports the mechanisms presented in the following sections, particularly semantic neighbour search, conceptual clustering, and analogy construction.

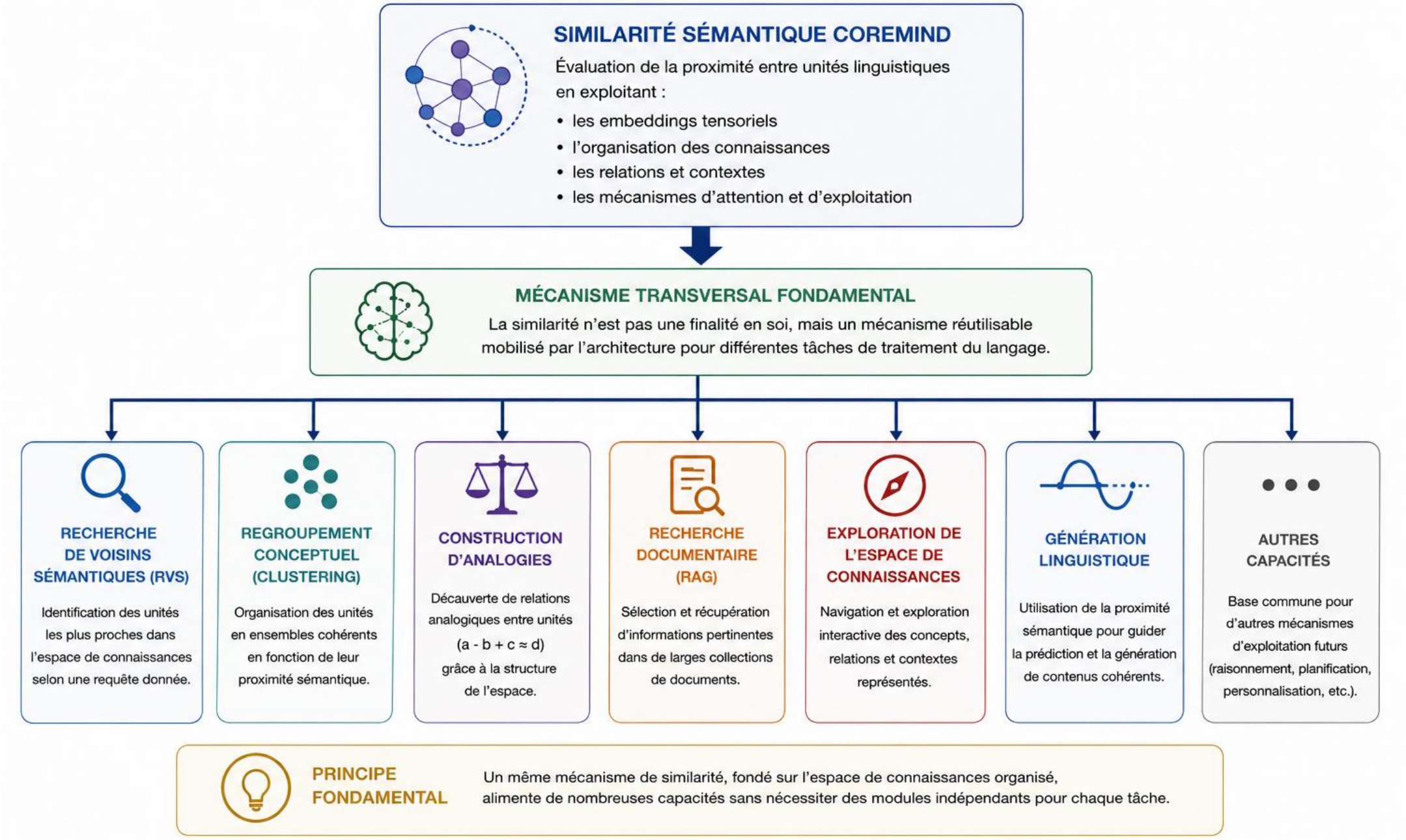


Figure 7 *– Similarity as a Mechanism*

6.4 Semantic Neighbour Search

Semantic neighbour search [5][6] constitutes a natural extension of the similarity mechanism presented previously. Its objective is to identify, within the knowledge space organised by CoreMind, the linguistic units exhibiting the most relevant proximity to a given unit. This capability makes it possible to directly exploit the proximity relationships constructed by the architecture.

Unlike a search based solely on the comparison of numerical representations, neighbour search within CoreMind exploits the organisation of knowledge constructed by the architecture. The selected units are not merely those whose representations appear close, but those whose knowledge, relationships, and contexts exhibit coherence with the query under consideration.

The architecture does not prescribe a single method for neighbour search. Strategies for exploring the knowledge space are determined by the exploitation mechanisms developed upon CoreMind and can be adapted to the constraints specific to each model or application domain. This separation between the architecture and the exploitation strategy preserves a generic framework while allowing considerable freedom in the design of specialised algorithms.

The results of this search can be interpreted in different forms according to the requirements of the application. They may consist of an ordered list of linguistic units exhibiting the greatest proximity, a coherent set of knowledge associated with a query, or a working basis for other mechanisms within the architecture. Neighbour search therefore constitutes an entry point for broader exploration of the knowledge space.

This capability plays an important role in numerous tasks built upon CoreMind. It can be mobilised for semantic search, concept recommendation, knowledge exploration, retrieval of relevant information, or as a preparatory stage for more complex mechanisms such as conceptual clustering, analogies, or information retrieval.

Experiments conducted on the prototype show that this capability constitutes a fundamental mechanism for exploiting the knowledge space. It illustrates how knowledge representation, its organisation, and similarity mechanisms cooperate to make the most relevant information directly accessible according to the context under consideration.

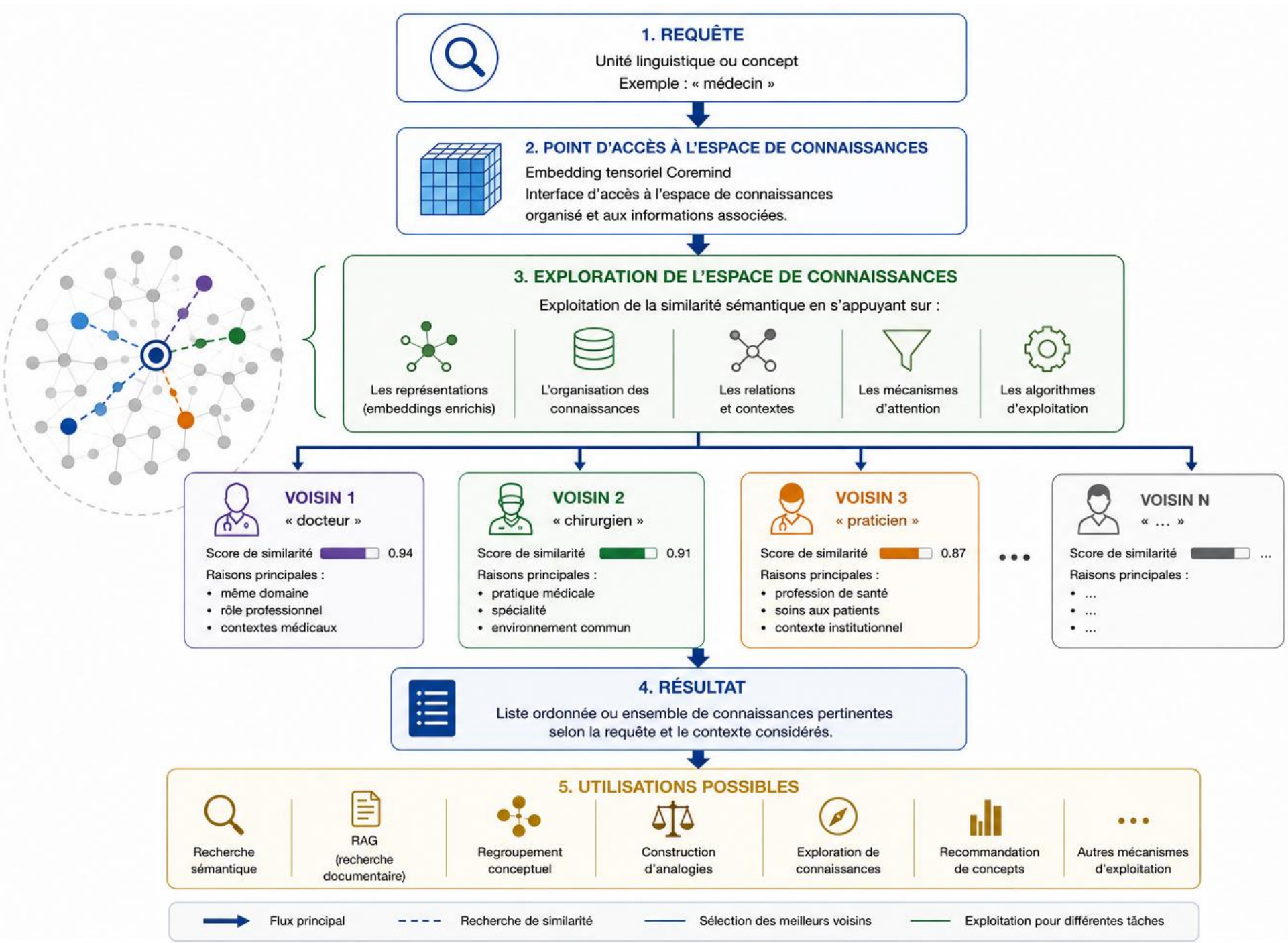


Figure 8 - *Semantic Neighbour Search within the CoreMind Knowledge Space*

6.5 Conceptual Clustering (Semantic Clustering)

Conceptual clustering [11][12] constitutes a capability obtained through the combined exploitation of knowledge representation, similarity mechanisms, and semantic neighbour search. Its objective is to automatically organise linguistic units and knowledge into coherent groups, reflecting their semantic proximity as well as the relationships constructed within the CoreMind architecture.

Unlike an approach based solely on geometric or statistical criteria, conceptual clustering within CoreMind relies on the organisation of the knowledge space. The resulting groups represent not merely neighbouring linguistic units, but sets of concepts sharing coherent knowledge, contexts, and relationships. This approach makes it possible to reveal conceptual structures that extend beyond simple numerical proximity between representations.

Conceptual clustering is not merely an analytical or visualisation tool. It represents an organisational mechanism that can be exploited across numerous tasks built upon CoreMind. The resulting groups can, in particular, facilitate the structuring of a knowledge base, navigation through the semantic space, information retrieval, the discovery of relationships between concepts, or provide a foundation for other exploitation mechanisms developed upon the architecture.

The CoreMind architecture does not impose a fixed organisation of these groupings. Clusters can evolve alongside the knowledge they represent. The addition of new information, the modification of existing knowledge, or its reorganisation may lead to the progressive evolution of the clusters in order to preserve the overall coherence of the knowledge space. This dynamic reflects CoreMind's general philosophy, according to which representations, relationships, weights, and organisation evolve jointly throughout the model's life cycle.

This capability once again illustrates the distinction between the architecture and its exploitation mechanisms. CoreMind provides an organised knowledge space together with the mechanisms that enable it to evolve, while clustering strategies can be adapted according to the objectives pursued by each model or application domain.

Experiments conducted on the prototype show that this capability constitutes an important means of organising and exploring the knowledge represented by the architecture. It naturally provides a foundation for the mechanisms presented in the following sections, particularly the discovery of analogies and the exploitation of more complex relationships between concepts.

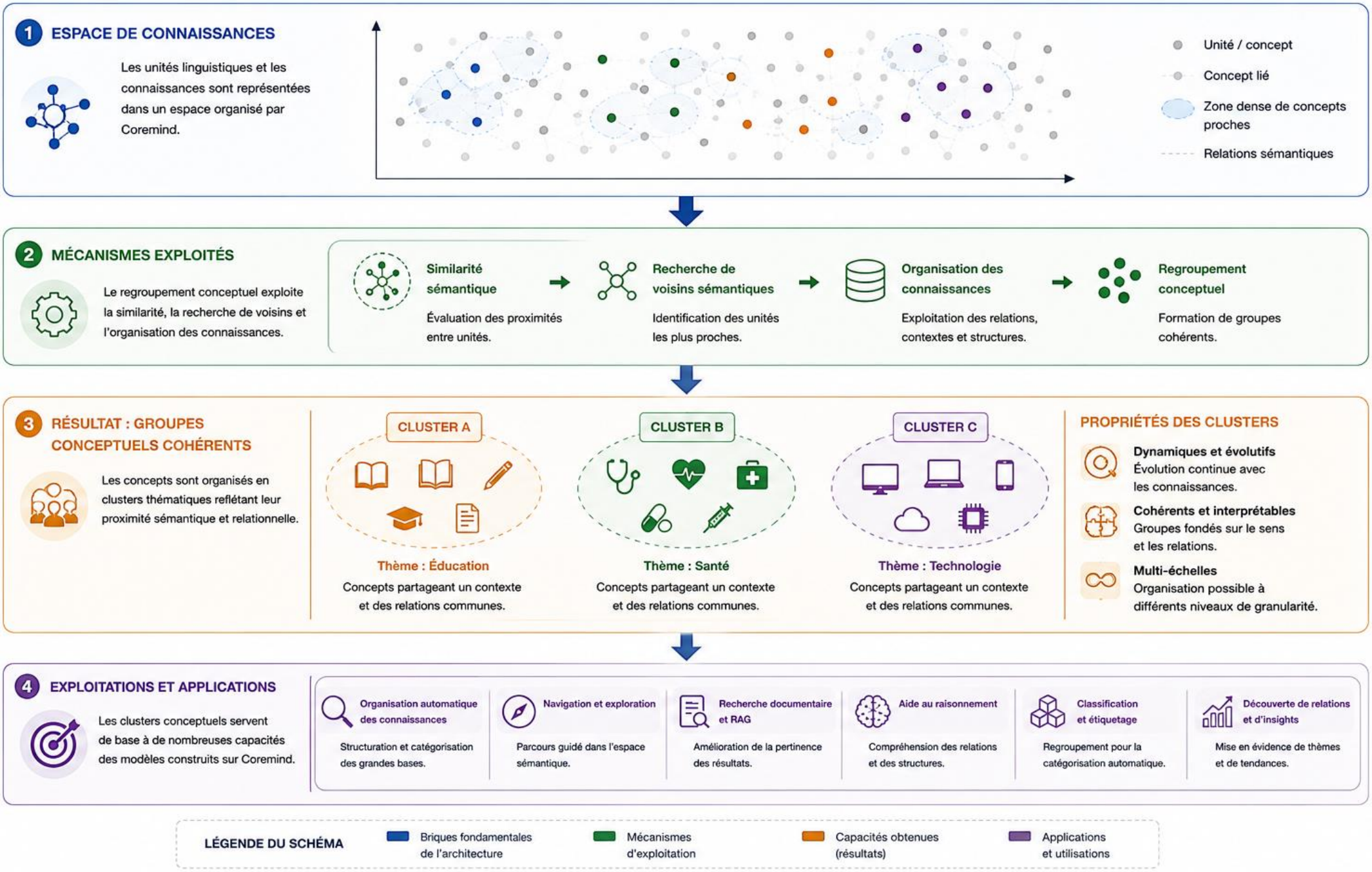


Figure 9 – *Conceptual Clustering within the CoreMind Knowledge Space*

6.6 Analogy Construction

Analogy construction, widely studied in distributed representation spaces [5][6], constitutes within CoreMind a capability obtained through the combined exploitation of knowledge representation, its organisation, similarity mechanisms, and the relationships established within the architecture. It enables the identification of correspondences between concepts, situations, or knowledge structures that share comparable relationships, while drawing upon the overall organisation of the knowledge space.

Unlike an approach based exclusively on vector operations or geometric transformations, CoreMind considers analogies as the result of exploiting the knowledge organised within the architecture. Analogical relationships are therefore identified not only from the representations of linguistic units, but also by taking into account the relationships, contexts, and knowledge structures that connect them.

This approach makes it possible to establish correspondences between concepts belonging to the same domain or to different domains whenever their relationships exhibit a comparable organisation. The resulting analogies are therefore not limited to exercises of the form “A is to B as C is to D”, but constitute a more general mechanism for discovering and exploiting relationships between knowledge elements.

This capability can be mobilised across numerous tasks built upon CoreMind. It can contribute to the discovery of implicit relationships, the progressive enrichment of a knowledge base, support for reasoning, exploration of the semantic space, or the preparation of more complex inference mechanisms. The strategies used to construct and exploit these analogies are determined by the mechanisms developed upon the architecture and can be adapted according to the objectives pursued by each model.

The CoreMind architecture does not impose a single method for constructing analogies. They can be obtained through different strategies for exploiting the knowledge space, depending on the nature of the available information, the constraints of the application, and the objectives of the model under consideration. This flexibility reflects CoreMind’s general philosophy, which distinguishes the architecture’s fundamental mechanisms from the exploitation strategies implemented by models built upon it.

Experiments conducted on the prototype show that this capability constitutes a relevant indicator of the coherence of knowledge organisation. The quality of the analogical relationships obtained reflects the richness of the representations, the relevance of the relationships constructed, and the architecture’s ability to organise knowledge in an exploitable manner. It naturally provides a foundation for the mechanisms presented in the following sections, particularly lexical prediction and more advanced forms of knowledge exploitation.

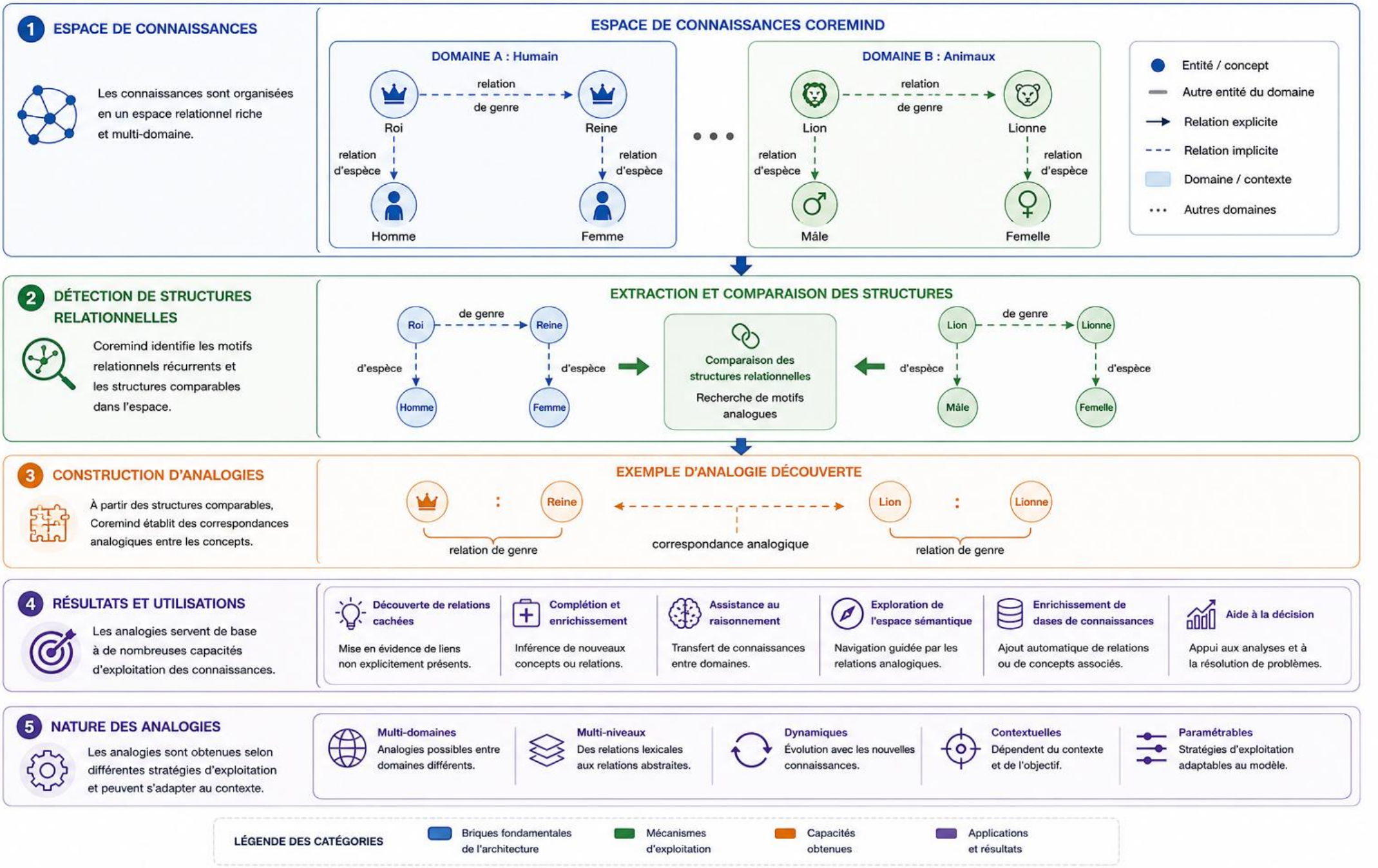


Figure 10 – *Analogy Construction through the Exploitation of the CoreMind Knowledge Space*

6.7 Lexical Prediction (MLM)

Lexical prediction [2] constitutes a capability obtained through the coordinated exploitation of the various mechanisms of the CoreMind architecture. It consists of identifying the most coherent linguistic units for completing incomplete information, drawing upon the knowledge organised within the knowledge space as well as the exploitation mechanisms developed upon this architecture.

Unlike an approach based solely on the statistical prediction of a word from its immediate context, CoreMind jointly mobilises knowledge representation, its organisation, similarity mechanisms, relationships between concepts, attention mechanisms, and the exploitation strategies implemented by the model. Prediction therefore results from a coherent analysis of all relevant information available within the knowledge space.

This capability is not limited to completing a missing linguistic unit. It can also be mobilised to complete incomplete knowledge, assess the coherence of a sequence, support certain information retrieval mechanisms, progressively enrich a knowledge base, or contribute to evaluating the quality of the representations constructed by the architecture.

The CoreMind architecture does not impose a single prediction strategy. Depending on the requirements of the model or application, exploitation mechanisms may produce a single proposal, an ordered list of hypotheses, or any other form of output suited to the context under consideration. This flexibility reflects the general principle of separating the architecture's fundamental mechanisms from the exploitation strategies developed upon this foundation.

The quality of the resulting predictions depends directly on the coherence of the organisation of knowledge represented within the architecture. A relevant prediction therefore constitutes an indicator of the richness of the representations, the quality of the relationships established between concepts, and the ability of the exploitation mechanisms to mobilise the most appropriate knowledge for a given situation.

Experiments conducted on the prototype show that this capability represents an important stage in the exploitation of the knowledge organised by CoreMind. It naturally provides a foundation for the more advanced mechanisms presented in the following sections, particularly semantic attention, information retrieval, and more complex forms of content generation.

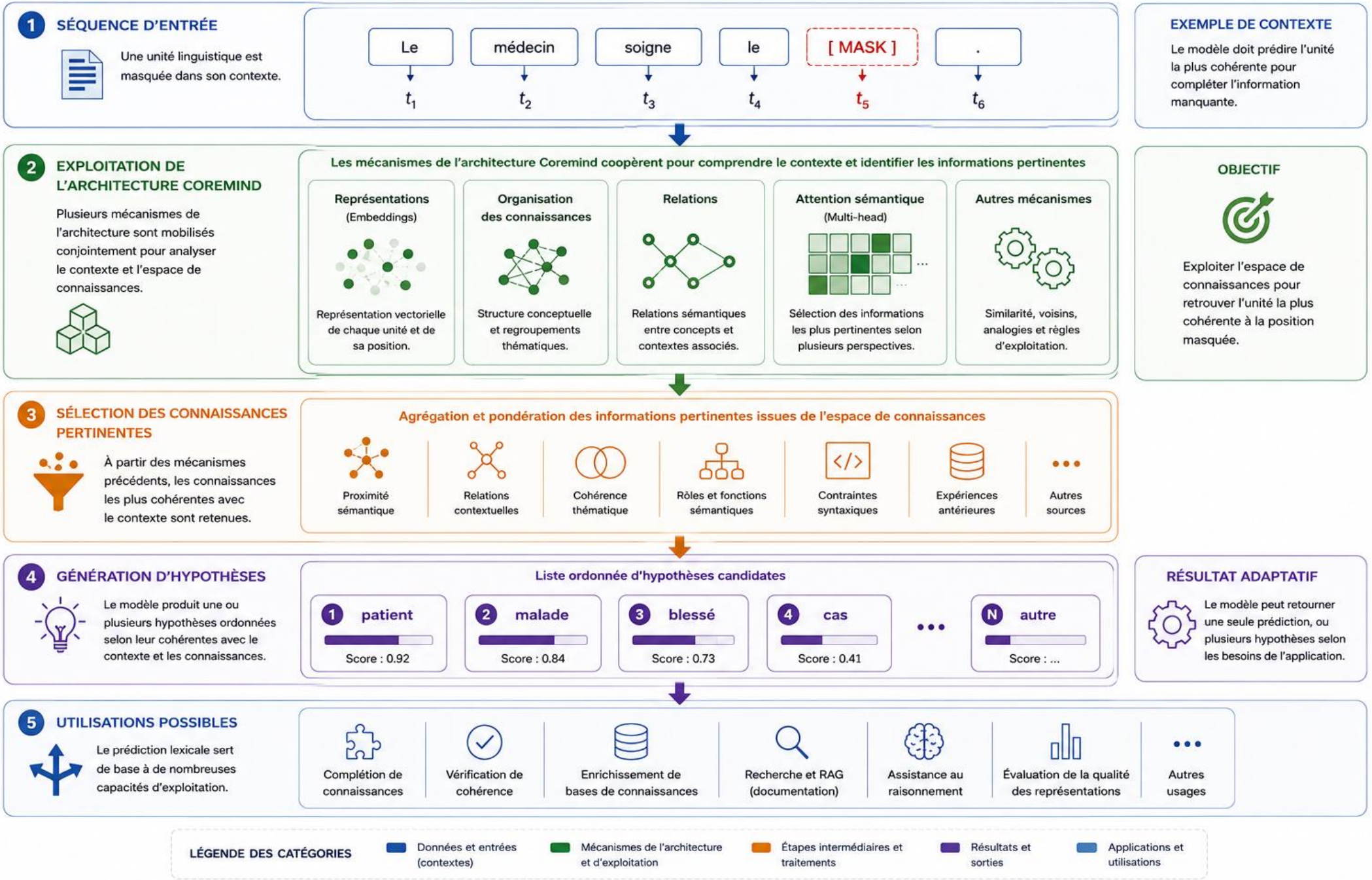


Figure 11 – *Lexical Prediction through the Exploitation of the CoreMind Knowledge Space*

6.8 Exploitation of Semantic Attention

Semantic attention constitutes one of the principal exploitation mechanisms of the CoreMind architecture. Its role is not to construct the knowledge represented within the architecture, but to identify, select, and weight the information most relevant to a given objective. It therefore enables the organised knowledge space to be exploited efficiently according to the context, the query, and the task under consideration.

Unlike an approach in which all available knowledge is mobilised uniformly, CoreMind uses attention mechanisms to focus processing on the information of greatest relevance to the situation being analysed. This selection relies on the combined exploitation of representations, relationships between concepts, knowledge organisation, and the various attention strategies developed within the architecture.

Specialised multi-head mechanisms contribute to this exploitation by enabling multiple characteristics of the same situation to be examined simultaneously. Each attention head can prioritise a particular perspective, a specific relationship, or a different family of knowledge, thereby providing several complementary perspectives on the same query. This diversity of analysis makes it possible to construct a representation that integrates multiple complementary perspectives on the relevant information, without being limited to a single selection strategy.

Semantic attention constitutes a cross-cutting mechanism reused across many of the capabilities presented in this chapter. It is involved, in particular, in lexical prediction, information retrieval, exploration of the knowledge space, analogy construction, and generation mechanisms. In each of these cases, its role is to direct processing towards the most relevant knowledge without modifying the fundamental structure of the knowledge space.

This selection remains entirely dynamic. The knowledge mobilised varies according to the nature of the query, the context of use, the objectives pursued, and the exploitation mechanisms specific to the model under consideration. The same knowledge base can therefore be exploited differently depending on the situation, while maintaining a common and coherent representation of the available information.

Experiments conducted on the prototype show that semantic attention contributes to the exploitation of the knowledge organised by CoreMind. It ensures coordination between the architecture's various components by directing processing towards the information most appropriate to each task and naturally provides a foundation for the mechanisms presented in the following sections, particularly information retrieval and content generation.

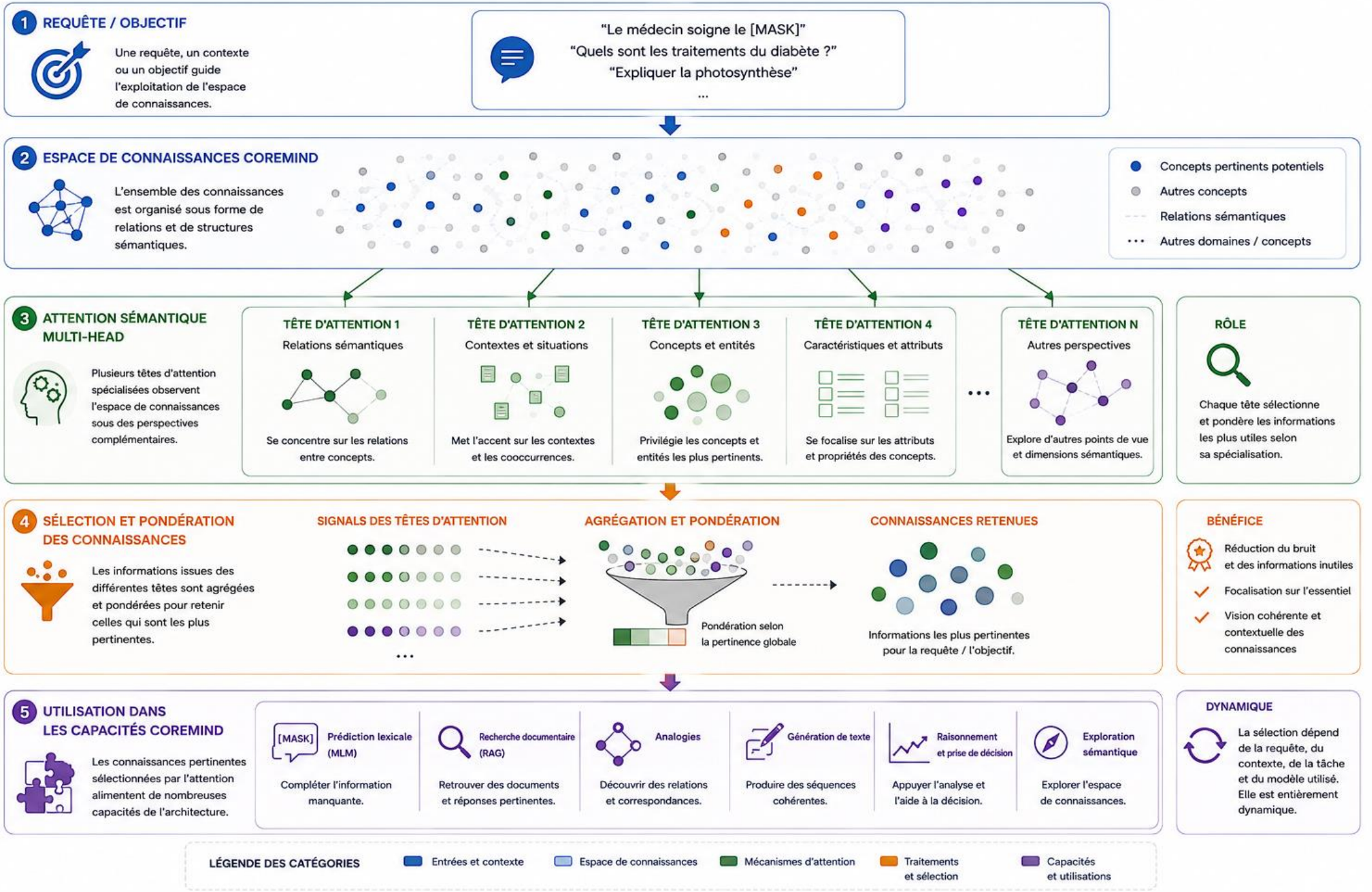


Figure 12 - *Exploitation of Semantic Attention within the CoreMind Architecture*

6.9 Information Retrieval (RAG) [10]

Information retrieval [8][9] constitutes a capability obtained through the exploitation of the CoreMind architecture to identify the information most relevant to a given query. Unlike an approach limited to retrieving similar documents [8][9], CoreMind prioritises the exploitation of the organised knowledge space in order to identify the knowledge most coherent with the objective pursued, before retrieving the documents, passages, or sources associated with it.

This capability jointly mobilises knowledge representation, similarity mechanisms, semantic neighbour search, conceptual clustering, relationships between concepts, and attention mechanisms. All of these components cooperate to direct the search towards the most relevant information according to the query, the context, and the objectives of the model.

In this approach, information retrieval does not rely exclusively on proximity between representations. It also exploits the organisation of knowledge constructed by the architecture, existing semantic relationships, conceptual groupings, and attention mechanisms that enable the most useful information to be selected. This cooperation makes it possible to move beyond simple similarity-based retrieval by drawing upon a more comprehensive understanding of the knowledge space.

The CoreMind architecture does not impose a single form for returning results. Depending on the requirements of the application, the exploitation mechanisms may return complete documents, targeted passages, sets of relevant knowledge, or any other structure suited to the context under consideration. This flexibility enables the same architecture to be used across various domains, such as information retrieval, specialised assistants, legal search engines, medical systems, or decision-support platforms.

This capability once again illustrates the separation between the architecture and its exploitation mechanisms. CoreMind provides an organised knowledge space together with the mechanisms required to explore it efficiently, while information retrieval strategies can be adapted according to the constraints of each model, domain, or application.

Experiments conducted on the prototype show that this approach makes it possible to exploit organised knowledge in order to identify information relevant to the query and its context. Information retrieval therefore becomes a direct consequence of the coordinated exploitation of the architecture's various mechanisms rather than a simple search based on proximity between representations. It naturally provides a foundation for the final capability presented in this chapter: content generation from selected knowledge.

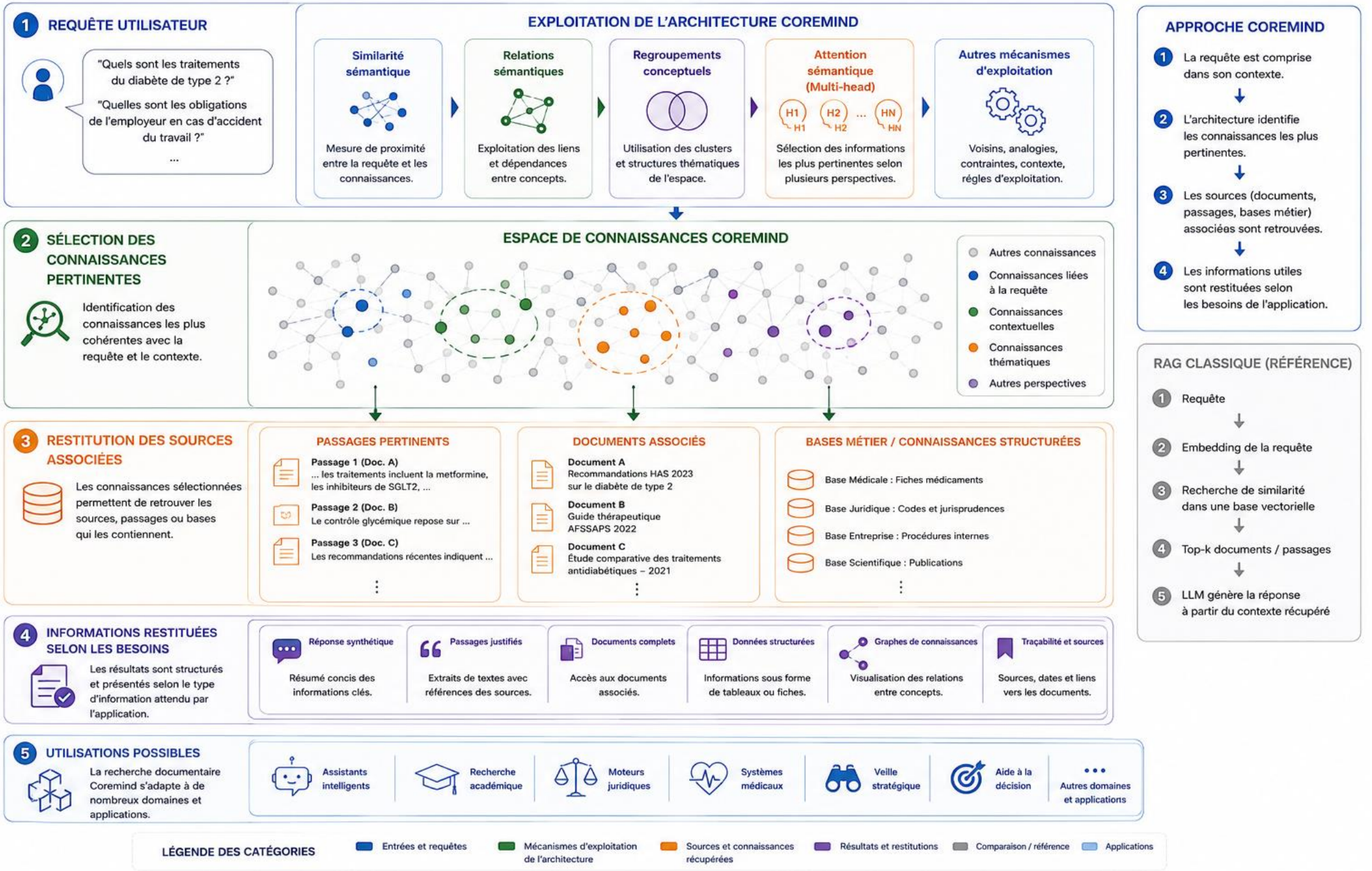


Figure 13 – *Information Retrieval through the Exploitation of the CoreMind Knowledge Space*

6.10 Evaluation and Selection of Candidate Knowledge

The evaluation and selection of candidate knowledge constitute a capability obtained through the coordinated exploitation of the various mechanisms of the CoreMind architecture. Its objective is to analyse multiple competing hypotheses, proposals, or knowledge elements in order to identify those exhibiting the highest level of coherence with the query, the context, and the knowledge available within the architecture.

Rather than relying solely on an approach that directly produces a response, CoreMind is also capable of evaluating multiple available candidates before selecting those that appear most relevant. This capability jointly mobilises knowledge representation, similarity mechanisms, relationships between concepts, the organisation of the knowledge space, and attention mechanisms in order to establish a coherent ranking of the different proposals.

The CoreMind architecture does not impose the nature of the candidates being evaluated. Depending on the model and application domain, this mechanism could be applied to different types of candidates, such as multiple-choice answers, hypotheses, documents, text passages, courses of action, or other information that can be compared against the available knowledge. This generality enables the same evaluation mechanism to be used across a wide range of domains and applications.

This capability constitutes a fundamental decision-making mechanism. It does not merely measure semantic proximity, but exploits the entire body of organised knowledge in order to assess the overall coherence of each candidate. Selection strategies may naturally vary according to the constraints of the model, the objectives pursued, or the application domain, while relying on the same architectural foundations.

Candidate evaluation may also constitute a stage preceding more complex processing. In the current prototype, it can notably contribute to selecting relevant knowledge prior to text generation or to retaining useful information during information retrieval. In the longer term, the same principle could be explored in other selection or decision-support mechanisms. This reuse across different contexts once again illustrates the cross-cutting nature of CoreMind's fundamental mechanisms.

Experiments conducted on the prototype show that this capability makes it possible to obtain a coherent ranking of multiple competing knowledge elements based on their compatibility with the knowledge space constructed by the architecture. It therefore constitutes an intermediate stage between the retrieval of relevant information and the production of a response, naturally providing a foundation for the generation mechanisms presented in the following section.

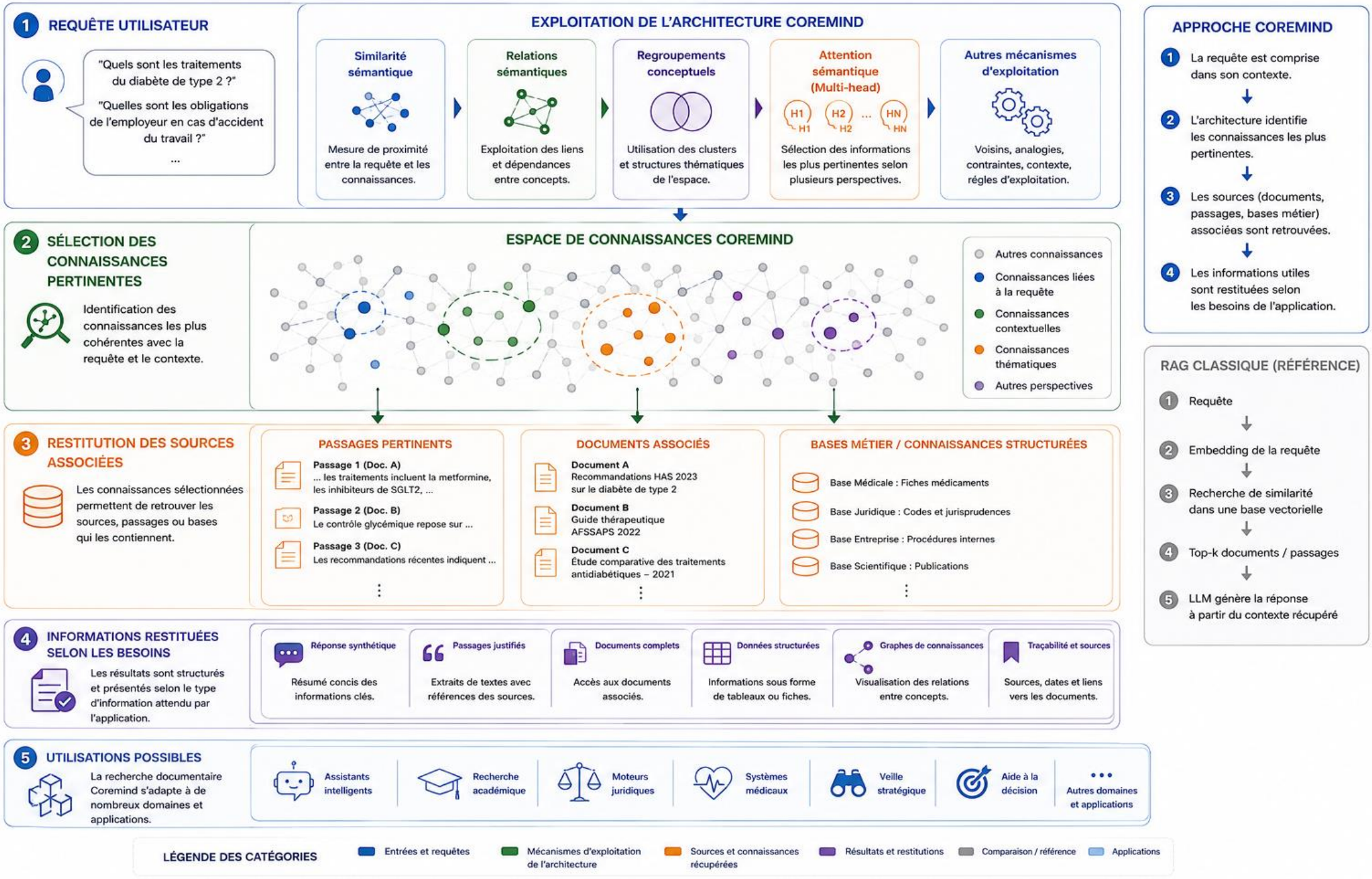


Figure 14 – *Evaluation and Selection of Candidate Knowledge within CoreMind*

6.11 Next-Token Generation

Next-token generation [3] constitutes the culmination of the various exploitation mechanisms presented in this chapter. It consists of producing the most appropriate linguistic unit from the knowledge selected and organised by the CoreMind architecture in order to extend a sequence coherently with the context and the objectives of the model.

Within CoreMind, the selection of the next token jointly exploits the architecture's various fundamental mechanisms. Knowledge representations, similarity mechanisms, semantic neighbour search, conceptual clustering, analogical relationships, attention mechanisms, information retrieval, and candidate knowledge evaluation processes all contribute to identifying the most relevant information before the next token is produced.

In this approach, generation does not constitute an independent mechanism, but rather the consequence of the coordinated exploitation of the architecture as a whole. The token produced results from the knowledge mobilised, its organisation, and the selection mechanisms implemented by the model in order to satisfy the objectives pursued within a given context.

The CoreMind architecture does not impose a single generation strategy. Models developed upon this architecture may adopt different methods for selecting the next token according to application constraints, quality requirements, or the objectives pursued. Generation strategies are therefore determined by the exploitation mechanisms built upon CoreMind, while the architecture provides the knowledge and mechanisms required for their exploitation.

This capability constitutes an essential stage in the production of textual content and can be exploited in applications such as conversational assistants, document generation, automated response systems, or any other application requiring the progressive production of a linguistic sequence. It once again illustrates the distinction between the architecture's fundamental mechanisms and the generation strategies developed by the models that exploit it.

The generation presented in this section is based on a conventional sequential generation principle [3], in which each new linguistic unit is determined from the knowledge selected at the preceding stage. Perspectives for evolution towards more global generation mechanisms, capable of simultaneously exploiting multiple generation trajectories, will be presented in Chapter 8, which outlines the research directions currently being explored within the CoreMind architecture.

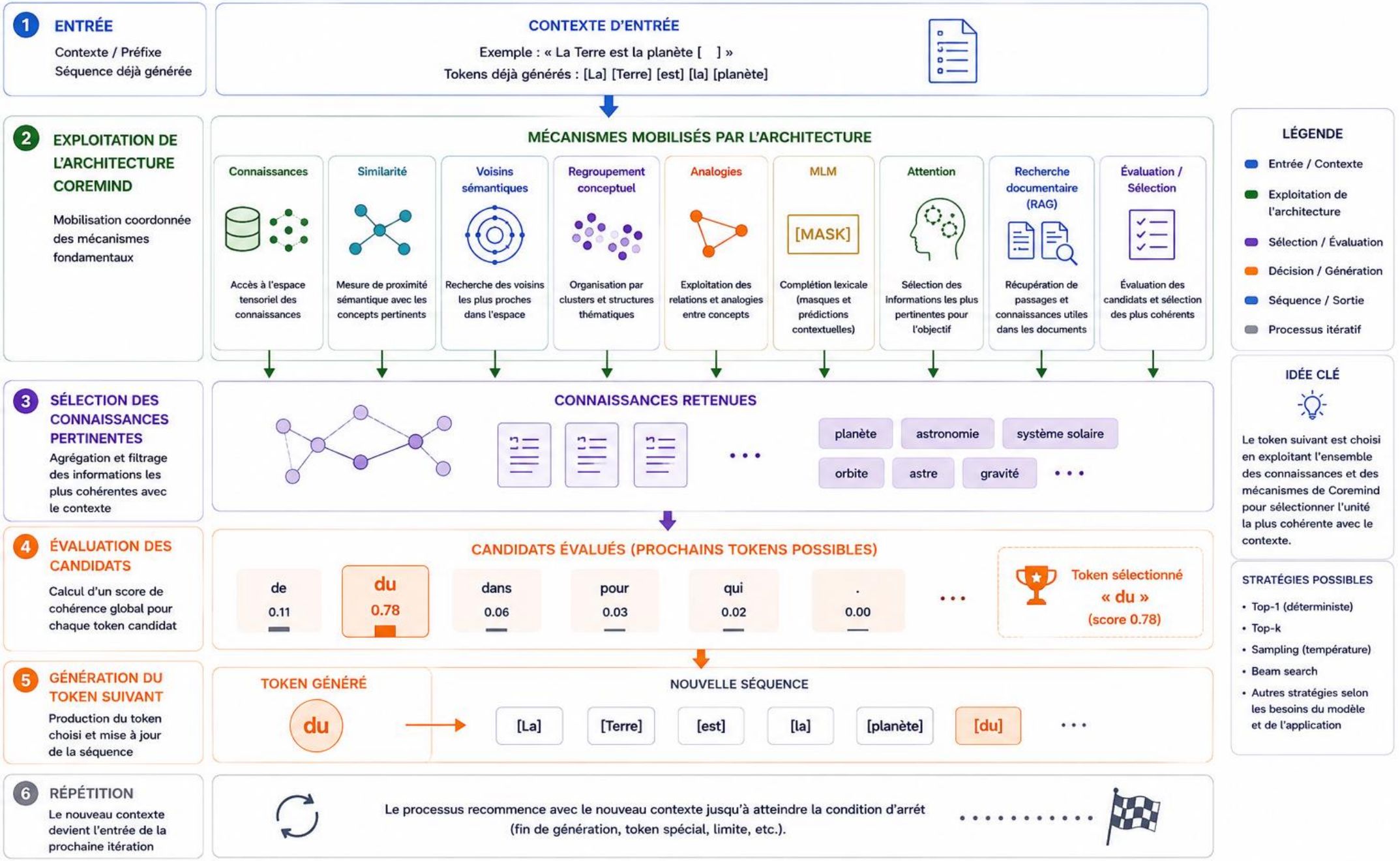


Figure 15 – *Next-Token Generation through the Exploitation of the CoreMind Architecture*

6.12 Current State of the Prototype and Validation Perspectives

The current CoreMind prototype was developed as part of a research approach aimed at experimentally validating the principal mechanisms proposed by the architecture. Its objective is to demonstrate the feasibility of the concepts presented in this document, evaluate their coherence, and observe their behaviour under different usage scenarios. At this stage, the prototype constitutes a scientific and technological demonstrator intended to support the progressive development of the architecture, rather than a fully optimised industrial implementation.

The experiments conducted to date have enabled several fundamental CoreMind mechanisms to be validated. The principles of tensor-based knowledge representation, embedding, semantic similarity mechanisms, neighbour search, conceptual clustering, analogy construction, lexical prediction, attention mechanisms, information retrieval, candidate knowledge evaluation, and sequential generation have all been implemented and evaluated through the current prototype. These results constitute an initial experimental validation of the coherence of the proposed mechanisms and show that they can be integrated within a functional prototype based on the principles presented in this White Paper.

The development of the prototype has deliberately prioritised the validation of scientific hypotheses over the pursuit of maximum performance. Implementation choices have primarily been guided by the objective of observing the behaviour of the different mechanisms, verifying their interactions, and evaluating their ability to cooperate within a single architecture. The performance currently achieved should therefore not be interpreted as the maximum performance that an optimised implementation of CoreMind could attain.

Several areas for improvement have already been identified for the next phases of development. These include computational performance optimisation, reduction of the memory footprint, scaling to very large corpora, evaluation on benchmark datasets, improved robustness in specialised domains, and the progressive industrialisation of the architecture. This work will enable CoreMind's performance to be evaluated more extensively across a variety of usage contexts.

The CoreMind architecture was designed to evolve progressively as the research advances. The mechanisms currently validated constitute a solid foundation upon which new knowledge exploitation strategies and new reasoning capabilities can be developed. The following chapters present several research directions intended to progressively extend the capabilities offered by the architecture.

The results obtained at this stage confirm the feasibility of the principal mechanisms proposed in this document and constitute an initial step towards a general architecture for the organisation and exploitation of knowledge. Future work will aim to continue their experimental validation, optimisation, and extension in order to progressively broaden the scope of CoreMind's applications.

7. Models Built Upon the CoreMind Architecture

7.1 One Architecture, Multiple Model Families

The CoreMind architecture was designed as a generic infrastructure for the representation, organisation, and exploitation of knowledge. It provides a set of fundamental mechanisms that can be reused within models designed to address different objectives.

The mechanisms presented in the previous chapters — tensor-based knowledge representation, organisation of the semantic space, attention, similarity, neighbour search, conceptual clustering [11][12], analogies, information retrieval [8][9][10], candidate knowledge evaluation, and generation — constitute a common foundation upon which several families of specialised models can be built. Each model exploits the same architectural foundations, while activating, adapting, or specialising certain mechanisms according to the requirements of the intended application.

This separation between the architecture and the models constitutes one of CoreMind's fundamental principles. The architecture defines the generic mechanisms for representing and exploiting knowledge, while the models determine how these mechanisms are combined to address a particular task. This approach promotes component reuse, facilitates the development of new models, and makes it possible to maintain coherence across applications belonging to very different domains.

From this perspective, CoreMind is not limited to the development of language models. The same architecture can be exploited to build embedding models, generative models, information retrieval engines, recommendation systems, intelligent agents, multimodal systems, or specialised models intended for domains such as healthcare, law, industry, robotics, or finance. The differences between these models lie primarily in their knowledge exploitation strategies, the data used during their learning, and the functional objectives assigned to them.

The following sections present the principal families of models that can be built upon the CoreMind architecture, together with the mechanisms they mobilise. This classification illustrates the architecture's ability to serve as a common foundation for the development of a wide variety of artificial intelligence systems.

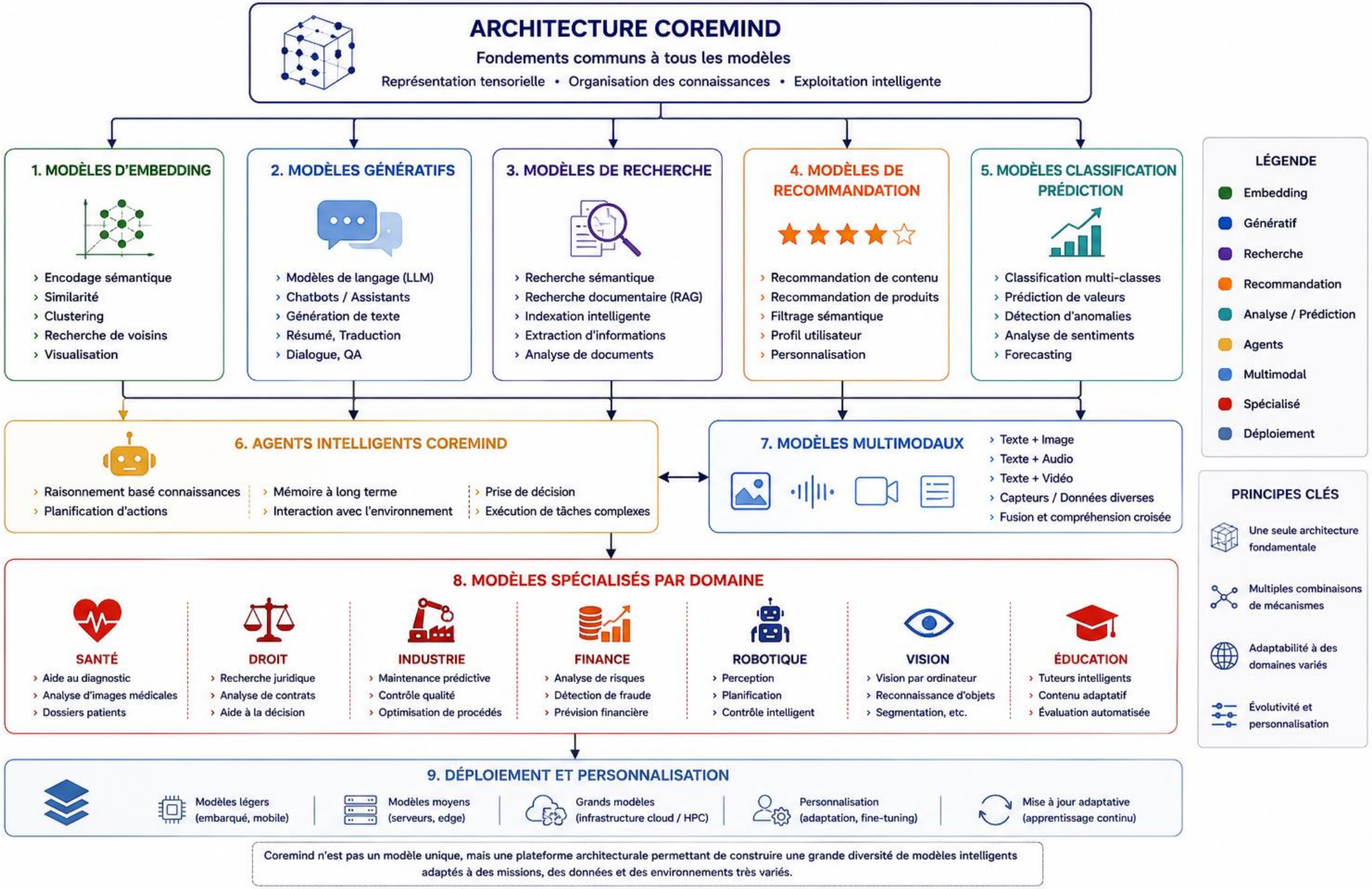


Figure 16 – *Examples of Models Built upon the CoreMind Architecture*

7.2 Principal Model Families

The CoreMind architecture was designed to serve as a common foundation for different families of artificial intelligence models. Through its mechanisms for knowledge representation, organisation, and exploitation, it can be adapted to a variety of objectives while retaining the same architectural principles. The differences between models lie primarily in the mechanisms they mobilise, the data they exploit, and the tasks they are intended to perform.

The principal families of models that can be built upon the CoreMind architecture include embedding models, generative models, semantic search models, and specialised models intended for particular application domains. Extension towards multimodal models constitutes a research perspective for the architecture.

Embedding models are dedicated to the semantic representation of knowledge. They can be used, in particular, to measure similarity between concepts, search for semantic neighbours, organise knowledge spaces, or build search and recommendation systems.

Generative models exploit the knowledge organised by CoreMind to progressively produce content adapted to the context. They can be used for text generation, conversational assistants, automatic summarisation, translation, reformulation, or any other task requiring the production of linguistic sequences.

Semantic search models specialise in exploring knowledge bases and retrieving the most relevant information. They can be used for information retrieval, specialised search engines, decision-support systems, or knowledge management platforms.

Multimodal models aim to exploit multiple information modalities simultaneously, such as text, images, audio, or other types of data, in order to construct a unified representation of knowledge and facilitate interactions between different information sources.

Finally, **specialised models** correspond to adaptations of the architecture tailored to a particular domain, such as healthcare, law, finance, industry, education, or scientific research. They exploit the same fundamental mechanisms while integrating the knowledge and constraints specific to their application domain.

This diversity of models illustrates CoreMind's aim to constitute a generic artificial intelligence architecture. Rather than addressing a single task, it provides a coherent set of mechanisms that can be reused to build models serving a variety of objectives, while sharing the same representation and organisation of knowledge.

8. Perspectives for the Evolution of the Architecture

8.1 Towards New Approaches to Generation

Content generation is currently one of the principal applications of artificial intelligence models [3]. Current approaches predominantly rely on sequential generation, in which each token is progressively produced from the available context and the elements generated previously [3]. This strategy has demonstrated its effectiveness across a wide range of tasks and currently constitutes the reference approach for generative models [3].

The CoreMind architecture, based on an explicit and structured representation of relationships between knowledge elements, nevertheless opens the possibility of exploring alternative generation strategies. Indeed, the semantic organisation of knowledge is not limited to next-token prediction, but makes it possible to envisage multiple generation paths that can be constructed, compared, and evaluated before selecting the most relevant solution.

This approach is not intended to replace existing methods, but rather to explore a new research direction made possible by the mechanisms specific to the CoreMind architecture. It forms part of the architecture's progressive evolution, with the objective of investigating new generation strategies capable of exploiting semantic relationships between knowledge elements more effectively.

The following section presents one of these research perspectives, based on the concept of semantic trajectory-based generation.

8.2 Trajectory-Based Generation

One of the research perspectives currently being considered within the CoreMind architecture is the exploration of a generation approach based on semantic trajectories. Unlike generation based exclusively on a succession of local predictions [3], this approach aims to construct multiple coherent generation paths from the knowledge organised within the architecture.

Each trajectory represents a sequence of concepts connected through their semantic relationships. Multiple trajectories can therefore be explored simultaneously in order to produce different generation hypotheses. These hypotheses are then analysed and compared using CoreMind's evaluation mechanisms, making it possible to identify the trajectory most coherent with the context and the objective of the generation process.

This approach is based on the idea that the quality of a generation can be evaluated not only at the level of the next token, but also at the level of a sequence of concepts forming a broader semantic trajectory. The organisation of knowledge proposed by CoreMind provides a framework for exploring this approach by drawing upon relationships between concepts rather than solely on the local succession of predictions.

At this stage, this approach constitutes a research direction whose principles will need to be studied, experimented with, and validated during future developments of the architecture. Its objective is to explore new generation strategies that fully exploit the knowledge representation and organisation mechanisms specific to CoreMind.

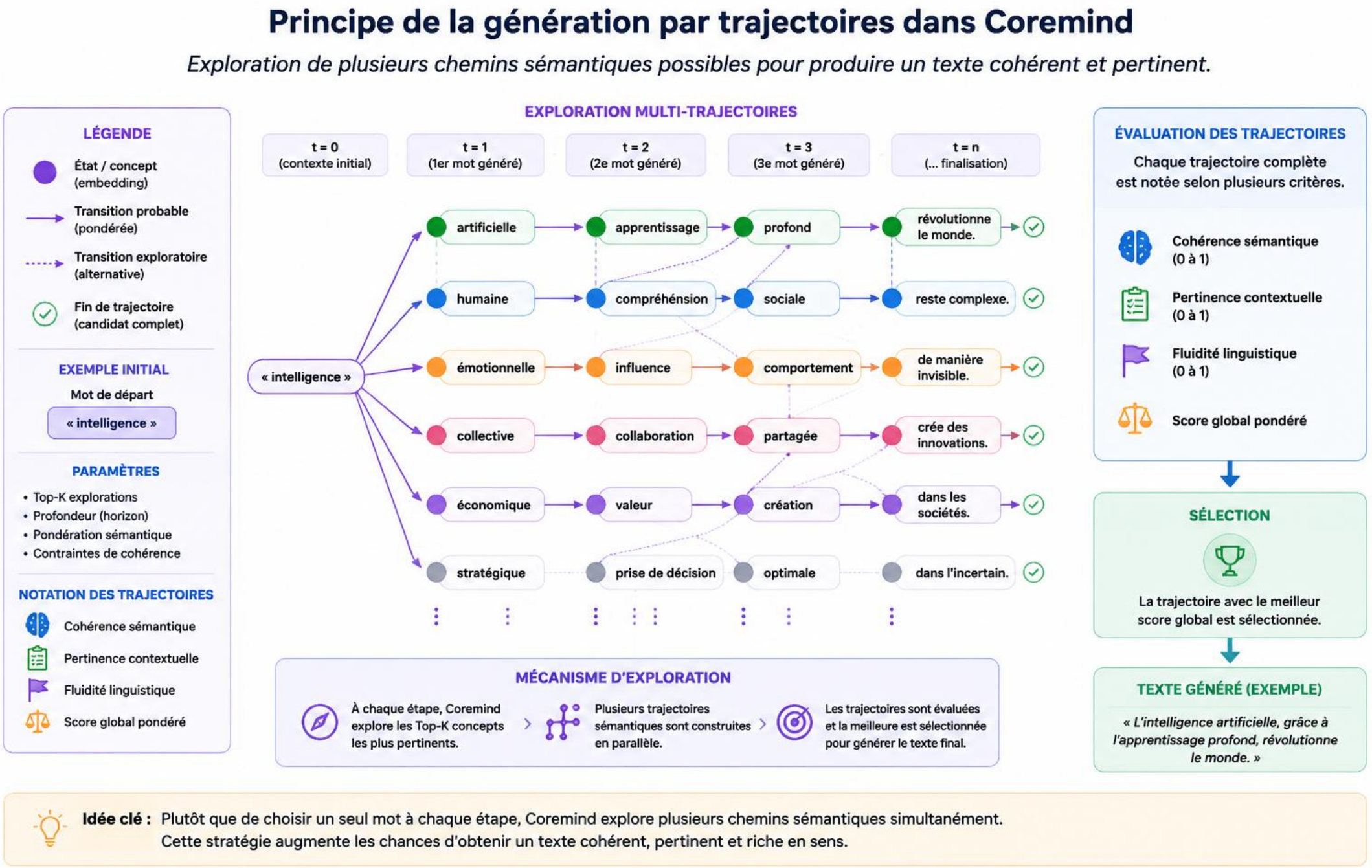


Figure 17 *– Principle of Trajectory-Based Generation within CoreMind*

8.3 Perspectives

The perspectives presented in this chapter explore the possibility of evolving the CoreMind architecture towards new generation strategies that exploit the organisation of knowledge more directly. The mechanisms currently validated constitute a solid foundation upon which new approaches can be progressively developed and evaluated.

Trajectory-based generation represents one such perspective. Its objective is to explore generation methods based on a more global view of the semantic relationships between concepts, complementing the sequential approaches widely used today. The principles presented in this chapter will require further research, experimental validation, and comparative analysis in order to assess their benefits and limitations.

More generally, the CoreMind architecture was designed as an evolving research platform, enabling new mechanisms to be progressively integrated while maintaining a coherent representation of knowledge. This modularity provides a favourable framework for exploring new ideas and for the continuous evolution of the architecture in line with scientific advances.

9. Conclusion

9.1 Summary of Contributions

This White Paper has presented CoreMind, an artificial intelligence architecture proposing an alternative approach to the representation, organisation, and exploitation of knowledge. CoreMind was conceived as a generic foundation for building different families of models sharing the same fundamental mechanisms.

The architecture is based on an explicit organisation of relationships between knowledge elements, combined with a set of mechanisms dedicated to their representation, retrieval, comparison, evaluation, and exploitation. This approach enables a variety of tasks to be performed, including semantic representation, knowledge retrieval, content generation, and the construction of specialised models.

Experiments conducted on the prototype have provided an initial experimental validation of the principal mechanisms investigated and enabled their feasibility to be evaluated within the prototype. Although several perspectives for further development remain to be explored, the results obtained show that the adopted approach constitutes a coherent foundation for pursuing the development of new artificial intelligence methods.

CoreMind is therefore presented not as a fixed system, but as an evolving architecture intended to be progressively enriched through future research, experimental validation, and applications across different domains.

9.2 Research Perspectives

The work presented in this White Paper constitutes an initial stage in the development of the CoreMind architecture. The results obtained with the prototype provide an initial experimental validation of the principal concepts investigated and open the way to new research perspectives aimed at enriching and consolidating the architecture.

Future work will focus in particular on improving existing mechanisms, validating them on larger corpora, evaluating performance on established benchmarks, and investigating new strategies for exploiting knowledge. Particular attention will also be given to analysing the explainability, robustness, and adaptability of the architecture across different application domains.

The perspectives presented in this document, such as the exploration of new generation approaches, form part of this progressive research process. Their development will require further theoretical and experimental validation before they can be definitively integrated into the architecture.

In the longer term, CoreMind is intended to evolve as an open architecture capable of progressively integrating new mechanisms while retaining the fundamental principles presented in this document. This approach will enable the architecture to evolve alongside advances in the field of artificial intelligence while providing a coherent framework for the development of future models and applications.

This White Paper therefore constitutes the starting point of a broader research programme whose objective is to continue exploring new approaches to the representation, organisation, and exploitation of knowledge in support of explainable, adaptable, and efficient artificial intelligence systems.

10. Bibliographical References

The references below bring together the principal scientific works, corpora, and technical resources cited or discussed in this White Paper. They situate the CoreMind architecture within the context of the current state of the art in knowledge representation, Natural Language Processing, and semantic search. Their inclusion does not imply that CoreMind is based on these architectures or implementations; rather, these references provide a scientific foundation for contextualising the concepts presented.

[1] Vaswani, A., Shazeer, N., Parmar, N., et al. *Attention Is All You Need.* Advances in Neural Information Processing Systems (NeurIPS), 2017.
https://arxiv.org/abs/1706.03762

[2] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. NAACL-HLT, 2019.
https://arxiv.org/abs/1810.04805

[3] Brown, T. B., et al. *Language Models are Few-Shot Learners.* NeurIPS, 2020.
https://arxiv.org/abs/2005.14165

[4] Raffel, C., et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5).* Journal of Machine Learning Research, 2020.
https://arxiv.org/abs/1910.10683

[5] Mikolov, T., Chen, K., Corrado, G., & Dean, J. *Efficient Estimation of Word Representations in Vector Space.* 2013.
https://arxiv.org/abs/1301.3781

[6] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. *Distributed Representations of Words and Phrases and their Compositionality.* NeurIPS, 2013.
https://arxiv.org/abs/1310.4546

[7] Kudo, T., & Richardson, J. *SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer.* EMNLP, 2018.
https://arxiv.org/abs/1808.06226

[8] Robertson, S., & Zaragoza, H. *The Probabilistic Relevance Framework: BM25 and Beyond.* Foundations and Trends in Information Retrieval, 2009.
https://doi.org/10.1561/1500000019

[9] Karpukhin, V., et al. *Dense Passage Retrieval for Open-Domain Question Answering.* EMNLP, 2020.
https://arxiv.org/abs/2004.04906

[10] Lewis, P., et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.* NeurIPS, 2020.
https://arxiv.org/abs/2005.11401

[11] MacQueen, J. *Some Methods for Classification and Analysis of Multivariate Observations.* Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967.
https://digicoll.lib.berkeley.edu/record/113015

[12] Ward, J. H. *Hierarchical Grouping to Optimize an Objective Function.* Journal of the American Statistical Association, 1963.
https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845

[13] Wikimedia Foundation. *Wikipédia — édition francophone, corpus de données (dump).* Wikimedia Downloads
https://dumps.wikimedia.org

[14] Penedo, G., Kydlíček, H., allal, L. B., et al. *The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale*. NeurIPS 2024 Datasets and Benchmarks Track, 2024.
https://arxiv.org/abs/2406.17557

[15] Strubell, E., Ganesh, A., & McCallum, A. *Energy and Policy Considerations for Deep Learning in NLP*. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2019, pp. 3645–3650.
https://aclanthology.org/P19-1355/

[16] McInnes, L., Healy, J., & Melville, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2018. arXiv:1802.03426.
https://arxiv.org/abs/1802.03426

[17] Rousseeuw, P. J. *Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis*. Journal of Computational and Applied Mathematics, Vol. 20, 1987, pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7.
https://www.sciencedirect.com/science/article/pii/0377042787901257

11. Glossary

This glossary brings together the principal technical terms used in this White Paper. The definitions provided are given within the specific context of the CoreMind architecture and are intended to facilitate understanding of the concepts presented. They do not necessarily constitute general or universally accepted definitions within the fields of artificial intelligence or Natural Language Processing. Certain terms may therefore be adapted or used with a specific meaning particular to the CoreMind architecture.

CoreMind Self-Attention

A mechanism for evaluating the relationships and interactions between the different units within the same sequence in order to construct a contextualised representation. Within CoreMind, CoreMind Self-Attention relies on mechanisms specific to the architecture and differs from the attention mechanism based on Query, Key, and Value (QKV) matrices used in Transformer architectures.

CoreMind Multi-Head Attention

An attention mechanism based on multiple specialised heads that enable different types of relationships between represented units to be analysed simultaneously. Within CoreMind, these heads can be composed modularly according to the characteristics to be analysed, and their results are combined to construct a global representation of the interactions.

Adaptive Learning

A CoreMind mechanism in which knowledge is constructed from observed information and then progressively enriched and reorganised as the system evolves. It ensures continuity between the initial construction of the model and the subsequent integration of new knowledge, without requiring full model retraining.

MLM (Masked Language Model)

A language modelling task consisting of masking a linguistic unit within a sequence and then predicting that unit from the available context. Within CoreMind, this task is used experimentally to evaluate the architecture's ability to exploit its semantic representations and learned relationships in order to identify a masked token.

Semantic Neighbourhood

The set of concepts or linguistic units exhibiting the most relevant semantic proximity to a given concept within the knowledge space. Within CoreMind, exploring this neighbourhood makes it possible, in particular, to identify and exploit the proximity relationships constructed by the architecture.

Knowledge Updating

The process of evolving the knowledge represented within CoreMind without systematically reconstructing the entire model. It may involve adding, modifying, enriching, or removing information, with the objective of preserving the overall coherence of the knowledge space.

Semantic Analogy

A relationship that makes it possible to identify conceptual correspondences between multiple elements based on their semantic representations.

CoreMind Architecture

An architecture for the representation, organisation, and exploitation of knowledge based on a semantic tensor foundation, designed to enable knowledge retrieval, analysis, and generation.

Tensor Foundation

A multidimensional data structure used by CoreMind to represent relationships between concepts, their contexts of occurrence, and their semantic interactions.

Clustering

A method for automatically grouping concepts that exhibit similar characteristics or semantic relationships.

Knowledge

Information represented and organised within the CoreMind architecture in a form that can be exploited by the various processing mechanisms.

Context

The set of information surrounding a concept and contributing to its semantic interpretation.

Corpus

A collection of textual documents used to construct the CoreMind knowledge base.

Embedding

A vector representation of a concept that captures its semantic properties and enables its proximity to other concepts to be measured.

Trajectory-Based Generation

An exploratory approach consisting of constructing and evaluating multiple candidate semantic paths before selecting the one considered most relevant for producing text. In this White Paper, this approach is presented as a research perspective.

Horizon

A dimension of the tensor foundation representing the contextual depth taken into account around a concept.

Knowledge Organisation

The process of structuring relationships between concepts in order to facilitate their exploitation, retrieval, and evolution.

Experimental Prototype

A research implementation developed to validate the principal mechanisms of the CoreMind architecture. It does not constitute a final product.

RAG (Retrieval-Augmented Generation)

An approach combining the retrieval of relevant information with a text generation mechanism. Within CoreMind, this capability relies on semantic search performed by the architecture.

Knowledge Representation

The organisation of concepts and their relationships in a form that can be exploited by CoreMind's internal mechanisms.

Semantic Representation

The modelling of concepts in a way that preserves their semantic relationships within the knowledge base.

Semantic Search

A search method based on semantic proximity between concepts rather than simple keyword matching.

Semantic Similarity

A measure of the degree of proximity between two concepts according to their semantic representation.

Sparse Tensor

A multidimensional tensor in which only relationships that are actually observed are stored, in order to avoid systematically storing absent relationships and thereby limit memory usage.

Token

An elementary unit produced by the tokenisation process and used as the basic representation when constructing the knowledge base.

Tokenisation

The process of dividing text into elementary units (tokens) before they are processed by the architecture.

Top-K

A method consisting of selecting the K concepts or results with the highest scores according to a given criterion.

12. Appendix A

A.0 Experimental Validation of the Principal Functionalities of the CoreMind Prototype

The examples presented in this appendix are taken directly from the CoreMind prototype and illustrate the principal functionalities validated at the time of writing this White Paper.

Introduction

This appendix presents a selection of results obtained with the current **CoreMind** prototype. Its objective is to illustrate, through concrete examples, the principal functionalities described in the White Paper and to demonstrate their experimental validation within the current implementation of the architecture.

The prototype was developed as a research environment enabling the principal CoreMind mechanisms to be progressively validated. The experiments presented in this appendix cover the functionalities currently implemented, including the construction of the tensor embedding, attention mechanisms, semantic similarity, neighbour search, conceptual clustering, analogies, lexical completion, multiple-choice questions (MCQs), information retrieval, text generation, and incremental learning mechanisms.

The results presented are taken directly from the CoreMind prototype. In order to provide a clear and consistent presentation, they are reproduced in the form of tables, synthetic diagrams, and representative examples rather than screenshots of the prototype interface. This approach makes it possible to emphasise the operation of the various mechanisms while maintaining a presentation appropriate for a scientific document.

It is important to emphasise that this appendix is not intended to provide an exhaustive quantitative evaluation or a comparison with existing architectures. The experiments presented are primarily intended to demonstrate the proper functioning of the principal mechanisms described in the White Paper and to illustrate the currently validated capabilities of the prototype.

Quantitative evaluations on benchmark datasets, comparisons with the state of the art, and detailed performance measurements will be addressed in future work as the architecture and its implementation evolve.

The following sections present each experimentally validated functionality in turn, accompanied by representative examples and a brief analysis of the results obtained.

Table A.1 — Experimental Configuration

Element	Configuration Used
Prototype	CoreMind Prototype
Corpus	Wikipedia + FineWeb
Vocabulary Size	32 000 tokens
Horizon	40
Representation	Sparse tensor embedding
Direction	Right & left (+k and -k)
Horizon Used	10

A.1 Construction of the Tensor Embedding

Objective

This experiment illustrates the construction of the tensor embedding by the CoreMind prototype from the training corpus. The objective is to show how knowledge is progressively organised within the knowledge space so that it can be exploited by the architecture's various mechanisms.

Principle

During learning, each occurrence observed in the corpus contributes to enriching the knowledge representation. Unlike a simple vector representation, the CoreMind tensor embedding organises relationships between tokens across multiple semantic horizons, making it possible to preserve information about the context and dependencies observed in the data.

The resulting embedding constitutes the entry point for all the mechanisms presented in the White Paper, including attention, semantic similarity, neighbour search, conceptual clustering, analogies, and generation.

Figures A.1.a and A.1.b show the results of the embedding analysis for the word "banque".

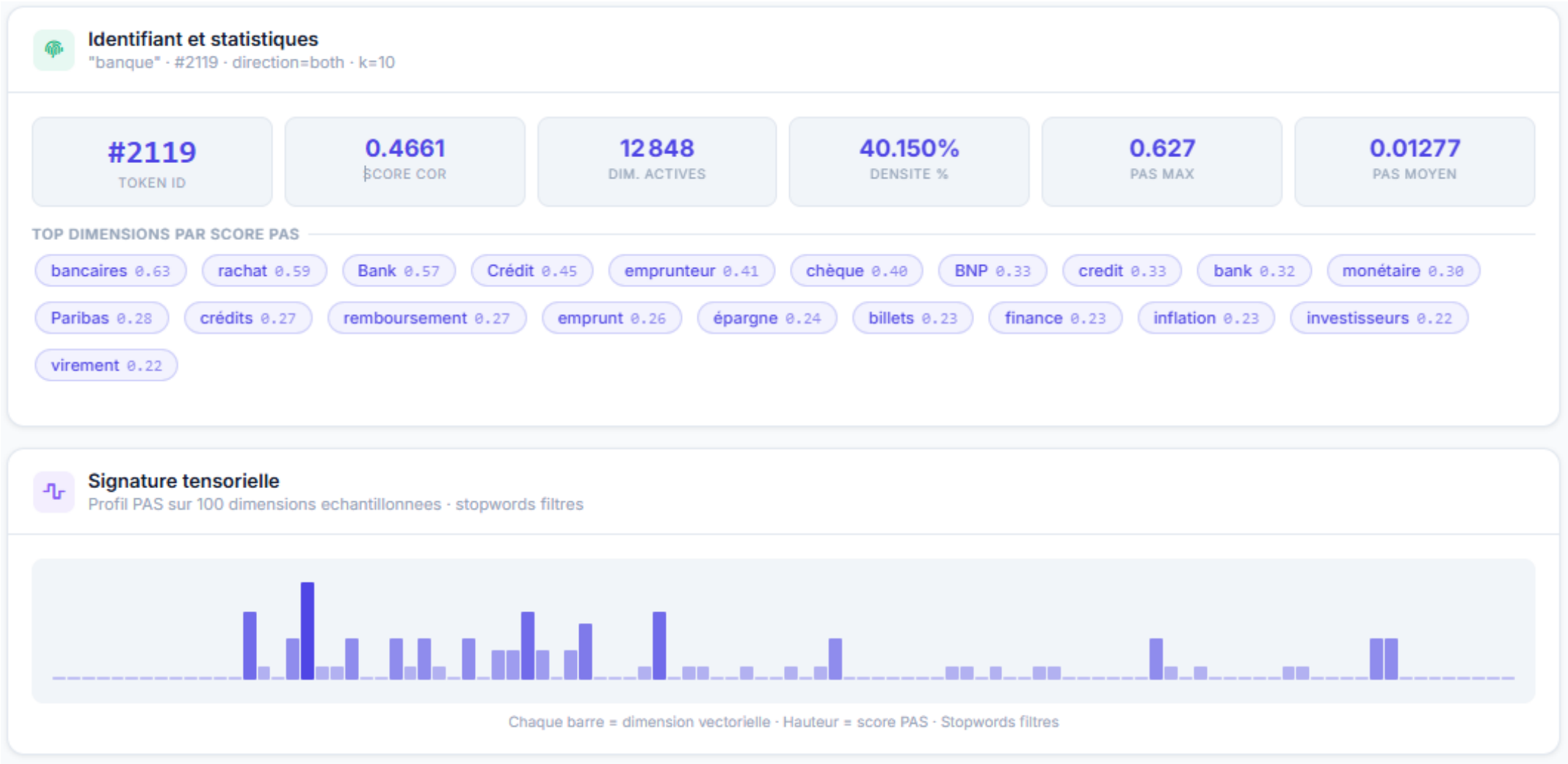


Figure A.1.a — *Analysis of a Tensor Embedding Constructed by the CoreMind Prototype for the Token "banque".*

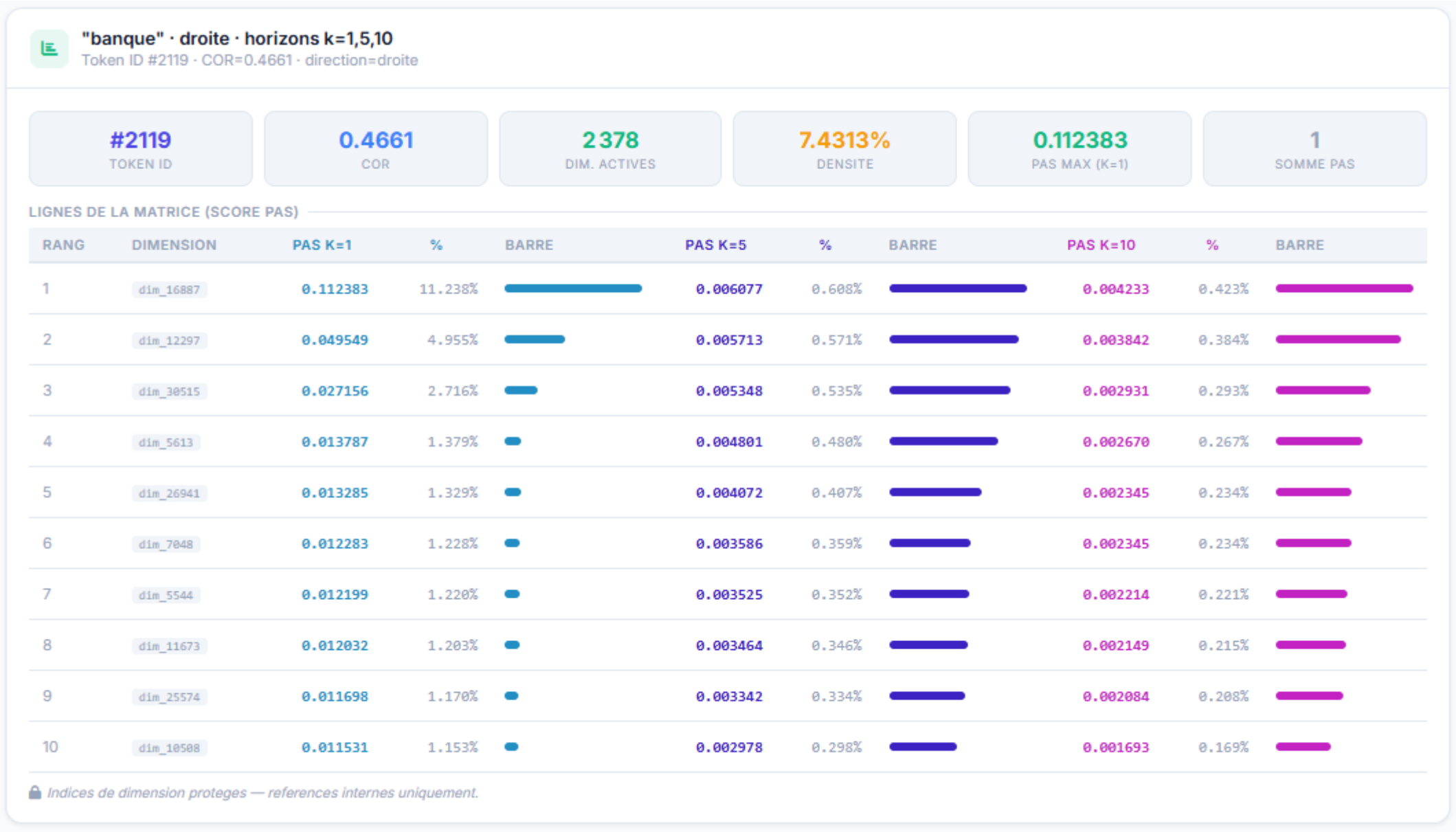


Figure A.1.b *— Extract from the Matrix Constructed by the CoreMind Prototype for the Token "banque".*

Analysis

The analysis of this token highlights the operation of the tensor embedding constructed by the CoreMind prototype. The calculated statistics (PAS score, number of active dimensions, density, and activation distribution) show that each token is represented by a structured tensor signature rather than by a simple compact representation.

The dimensions exhibiting the highest PAS scores correspond to the most active components of the representation of the analysed token. They constitute the principal points of access to the associated knowledge and are subsequently exploited by the attention, semantic similarity, neighbour search, conceptual clustering, and generation mechanisms.

Finally, the tensor signature highlights the distribution of activations within the representation space. This distribution reflects the internal organisation of the knowledge constructed by CoreMind and provides the foundation for the mechanisms presented in the following sections of this appendix.

A.2 Semantic Similarity

Objective

This experiment illustrates the semantic similarity mechanism implemented in the CoreMind prototype. The objective is to show how the architecture automatically evaluates the degree of proximity between two linguistic units by exploiting the knowledge constructed within the tensor representation space.

Principle

For a pair of linguistic units, CoreMind calculates an overall similarity score from the information available within the knowledge space. This evaluation relies on the tensor representation constructed during learning, together with several internal indicators used to estimate the semantic proximity between the two elements being compared.

The resulting score provides a measure that can be directly exploited by the architecture's other mechanisms, including neighbour search, conceptual clustering, analogies, information retrieval, and generation mechanisms.

Figure A.2 shows the results of the similarity analysis for the words "chat" and "chien":

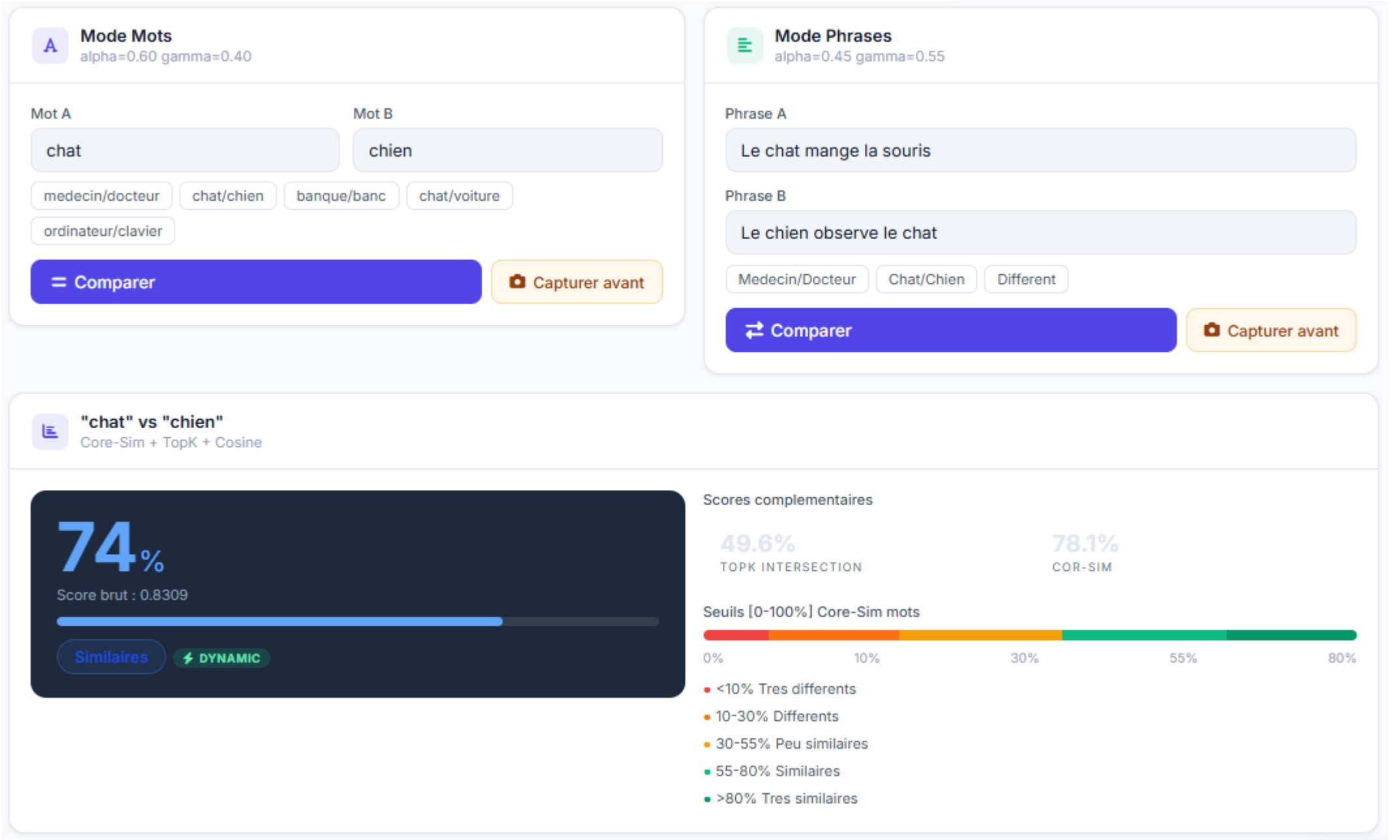


Figure A.2 — *Example of Semantic Similarity Evaluation between Two Linguistic Units in the CoreMind Prototype.*

Analysis

The experiment shows that the prototype assigns an overall similarity score of 74% between the words *chat* and *chien*, classifying them in the "Similar" category. The complementary indicators (TopK Interaction and Cosine CoreM) provide additional information on the relationships identified within the knowledge space and contribute to the final similarity evaluation.

This result illustrates the prototype's ability to identify concepts sharing numerous common semantic characteristics. The set of indicators produced provides a richer evaluation than a single score alone and constitutes a foundation that can be exploited by the search, conceptual clustering, and semantic relationship analysis mechanisms developed within the CoreMind architecture.

Conclusion

This experiment validates the proper functioning of the semantic similarity mechanism implemented in the CoreMind prototype. The results obtained show that the architecture is capable of coherently evaluating the proximity between two linguistic units and producing a measure that can be directly exploited by the system's other functionalities. This capability constitutes one of the fundamental mechanisms upon which the principal functionalities presented in the following sections of this appendix rely.

A.3 Conceptual Clustering

Objective

This experiment illustrates the ability of the CoreMind prototype to automatically group linguistic units into semantically coherent sets. The objective is to show that the representations constructed within the knowledge space make it possible to identify groups sharing common characteristics, without using predefined categories.

Principle

From a set of words belonging to different domains, the prototype calculates their semantic proximity and then applies a clustering algorithm to form homogeneous clusters. The resulting groups are derived solely from the relationships learned during training and from the organisation of knowledge within the CoreMind tensor space.

Figure A.3 Conceptual clustering obtained by the CoreMind prototype on a set of 29 linguistic units belonging to different semantic domains.

List of words analysed:
pharmacien, guerre, voiture, médecin, loup, artiste, seringue, vélo, poule, infirmier, camion, chat, paix, patient, moto, cheval, hôpital, vache, musique, renard, autobus, chirurgien, chien, médicament, lapin, docteur, malade.

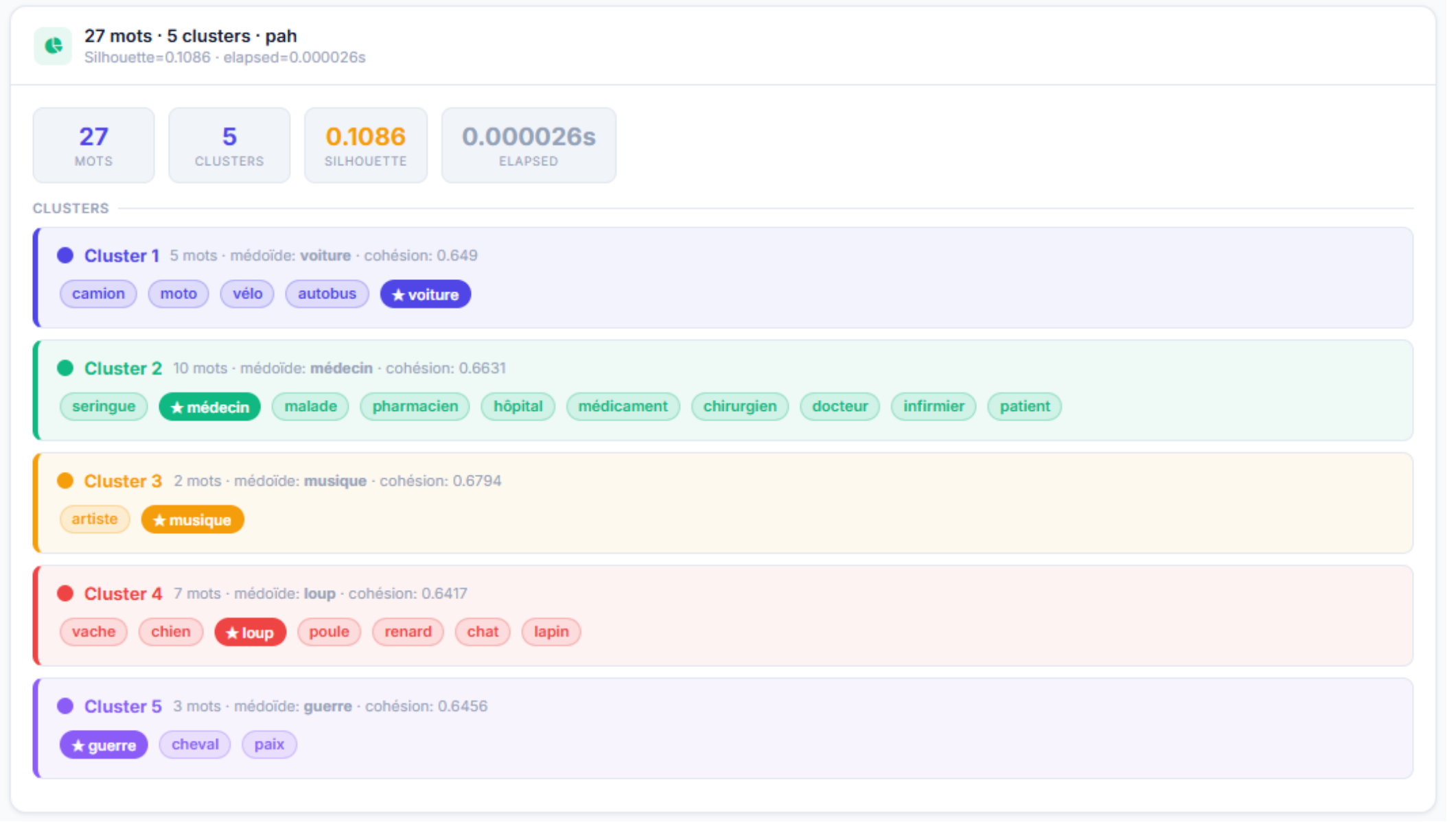


Figure A.3 *— Example of a CoreMind Clustering Result*

Analysis

The results show that the prototype automatically groups the linguistic units into sets exhibiting strong semantic coherence. The five resulting clusters correspond to clearly identifiable domains — animals, healthcare, transport, art, and war — without any categories being provided to the system during the analysis.

The UMAP projection [16] (A.8) highlights a clear separation between the different groups while preserving the proximity of concepts belonging to the same domain. This representation confirms that the relationships learned within the CoreMind knowledge space enable concepts to be organised according to their semantic similarity.

The silhouette score [17], together with the cohesion measures calculated for each cluster, provides complementary quantitative indicators for evaluating the quality of the resulting clustering.

Conclusion

This experiment validates the ability of the CoreMind prototype to automatically organise knowledge into semantically coherent groups based on the relationships constructed within the tensor space. Conceptual clustering can therefore be exploited for knowledge organisation, navigation through the semantic space, and information retrieval.

A.4 CoreMind Attention

Objective

This experiment aims to illustrate the ability of the CoreMind prototype to calculate attention relationships between the different units of a sentence and to identify the tokens exhibiting the greatest positional and semantic importance.

Principle

From the analysed sentence “Le petit chien noir cherche son maître dans la rue”, the prototype calculates an attention matrix connecting each token to the other tokens. The resulting weights are then normalised in order to represent the intensity of the internal relationships. A positional score H is also calculated to estimate the relative importance of each token within the overall structure of the sentence.

Figure A.4 shows the results of the analysis of the sentence “Le petit chien noir cherche son maître dans la rue”.

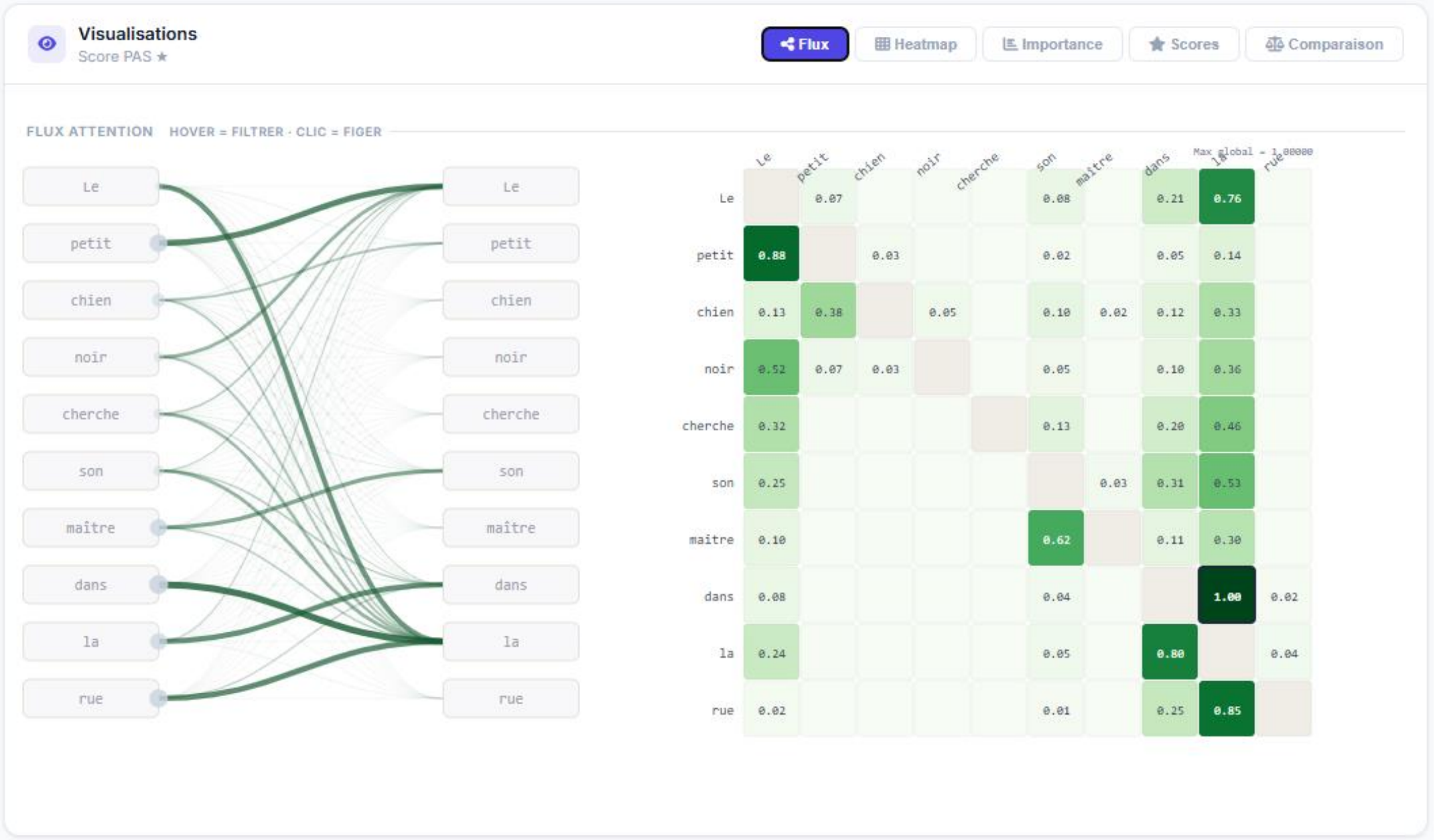


Figure A.4 *— Example of CoreMind Attention Calculation on a Sentence in French*

Analysis

The results show that the prototype assigns different weights to linguistic units according to their position and their interactions with the other tokens. The terms “dans”, “rue”, “petit”, “maître”, and “la” obtain the highest positional scores in this experiment.

The attention matrix highlights several strong relationships between the words in the sentence, particularly between “dans” and “la”, “rue” and “la”, as well as between “maître” and “son”. These relationships indicate that the prototype is capable of representing local and contextual dependencies between linguistic units.

Conclusion

This experiment validates the operation of the CoreMind attention mechanism on a complete sentence. The prototype produces both a global representation of token importance and a detailed matrix of their internal interactions.

A.5 CoreMind Multi-Head Attention

In order to capture multiple types of relationships between tokens simultaneously, CoreMind relies on a modular Multi-Head Attention architecture. CoreMind makes it possible to freely compose a set of specialised heads, each designed to analyse a particular aspect of the interactions between tokens.

The number and nature of these heads are defined during model design and can be adapted to the requirements of the application. The architecture therefore does not impose a predefined number of heads: a configuration may include multiple specialised mechanisms depending on the desired level of analysis, the application domain, and the implementation constraints.

Each head produces its own attention matrix, reflecting a specific interpretation of the relationships between tokens. The different matrices are then normalised and merged to construct a global representation of the interactions, integrating the complementary information extracted by each mechanism.

The prototype presented in this appendix illustrates this principle using a configuration composed of four attention heads, selected solely for demonstration purposes. These include PAS (CoreMind Score), VIP (Vector Importance Prediction), and SYM (Symmetric Relations), which constitute attention mechanisms developed specifically within the CoreMind architecture. This configuration represents only one example of how the architecture can be instantiated and does not constitute a limitation of the architecture itself.

Figure A.5 presents the general operation of this approach. It shows how multiple specialised mechanisms simultaneously analyse the same sentence before their respective representations are merged to produce a final attention matrix integrating the complementary information extracted by the different mechanisms. This modular architecture enables new attention mechanisms to be integrated without modifying the model’s fundamental principles, thereby providing considerable flexibility for adaptation to different domains and use cases.

A.6 Masked Language Model (MLM)

Objective

The Masked Language Model (MLM) task makes it possible to evaluate CoreMind's ability to predict a masked token using only the context provided by the other words in the sentence. This task constitutes a standard benchmark for assessing the quality of semantic representations learned by a language model [2].

Principle

Within CoreMind, the prediction of a masked token relies on the direct exploitation of the semantic relationships stored within the knowledge tensor. Each token in the context contributes to the search for the most relevant candidates according to the architecture's internal mechanisms.

Unlike approaches based exclusively on probability distributions, CoreMind constructs a ranking of candidates from the set of interactions observed between the tokens in the context. The results are accompanied by explanatory information, including the principal contributing tokens and various indicators that make it possible to interpret the model's decision.

Figure A.6 illustrates this mechanism through the sentence: *Le chat boit du [MASK] dans la cuisine.*

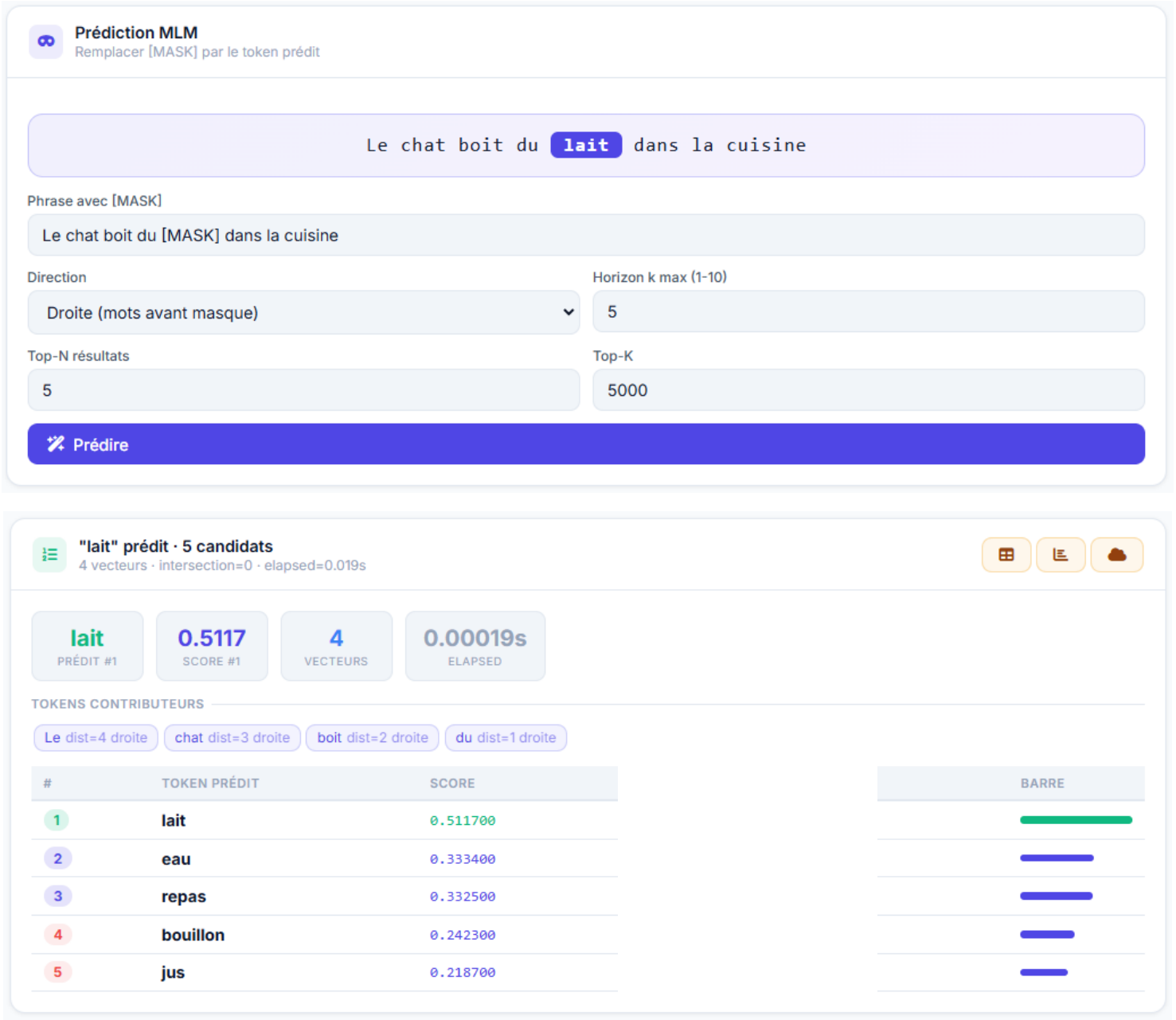


Figure A.6 — *Example of a Masked Token Prediction Result*

Analysis

In this example, CoreMind correctly identifies “lait” as the best candidate for completing the sentence. The subsequent predictions (*eau, repas, bouillon, jus*) belong to the same lexical field, showing that the model is not limited to retrieving an exact word but is capable of proposing several semantically coherent alternatives.

The figure also highlights the principal tokens that contributed to the prediction, together with the various indicators calculated by the prototype. This representation makes it possible to analyse the model’s decision-making process and identify the contextual elements that had the greatest influence on the final result.

Conclusion

This experiment shows that CoreMind is capable of performing a Masked Language Modelling task by directly exploiting the semantic relationships learned by its architecture. Beyond predicting the masked token, the prototype provides an explainable representation of the decision-making process by identifying contributing tokens and producing analytical indicators. This approach illustrates CoreMind’s ability to combine predictive capability with interpretability, two central objectives of the architecture.

A.7 Vector Analogy

Objective

Vector analogies make it possible to evaluate CoreMind’s ability to manipulate semantic relationships between concepts. The objective is to verify that the vector representations learned by the model preserve a sufficiently coherent structure to solve reasoning tasks of the form A − B + C = D, widely used as a reference in the evaluation of language models.

Principle

Within CoreMind, an analogy consists of combining several semantic vectors in order to search for the concept whose representation is closest to the resulting vector. This operation is performed directly within the vector space constructed by the architecture, exploiting the relationships learned between tokens.

The calculation mechanism relies on a combination of vector operations developed within CoreMind, making it possible to preserve semantic relationships while searching for the most coherent candidate. The results are then ranked according to their similarity score, providing a direct interpretation of the quality of the analogy.

Figure A.7 presents an example of solving the analogy:

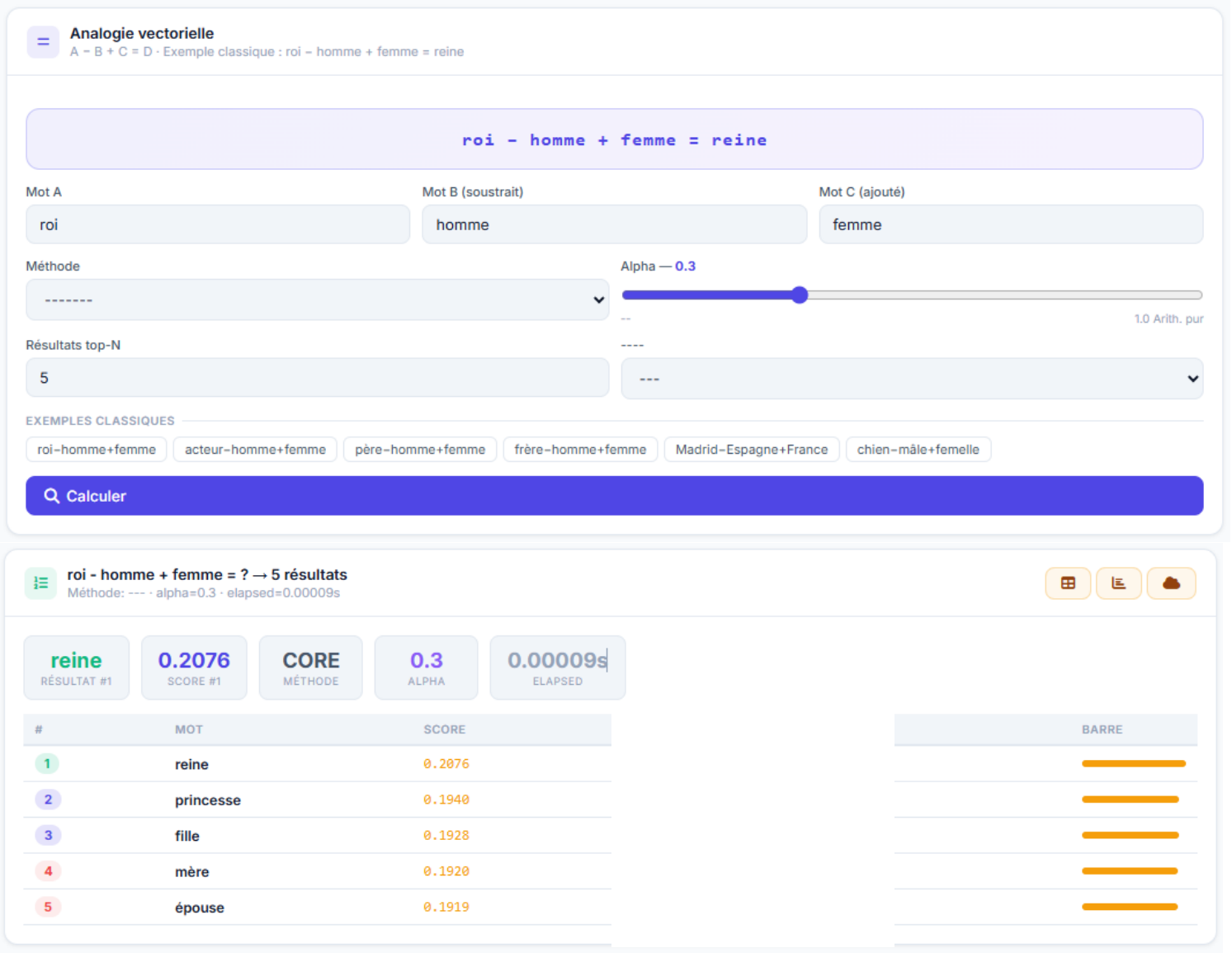


Figure A.7 *– Example of a Vector Analogy Performed by CoreMind.*

Based on the operation "roi – homme + femme", the model searches for the concept whose vector representation is most coherent with the expressed relationship. The results are ranked according to their similarity score, with the best candidate corresponding to the solution selected by the model.

Analysis

In this example, CoreMind identifies “reine” as the best result for the analogy, while several other closely related concepts appear in the ranking, such as *princesse, fille, mère,* and *épouse*. This diversity of candidates illustrates the richness of the semantic relationships learned by the model, as well as its ability to explore a coherent conceptual neighbourhood.

The figure also presents the principal indicators produced by the prototype, including the similarity score, the calculation method used, and the parameters associated with the experiment. Together, these elements make it possible to interpret the quality of the results while providing a transparent visualisation of the resolution process.

Conclusion

This experiment shows that CoreMind is capable of manipulating complex semantic relationships directly within its vector space. The resulting analogies reflect a coherent organisation of the learned representations and indicate that vector operations can preserve certain conceptual relationships between tokens. This property constitutes an important indicator of the quality of the semantic space constructed by the architecture and illustrates its ability to exploit relationships between concepts beyond simple lexical proximity.

A.8 Neural Navigator — 3D UMAP Exploration [16]

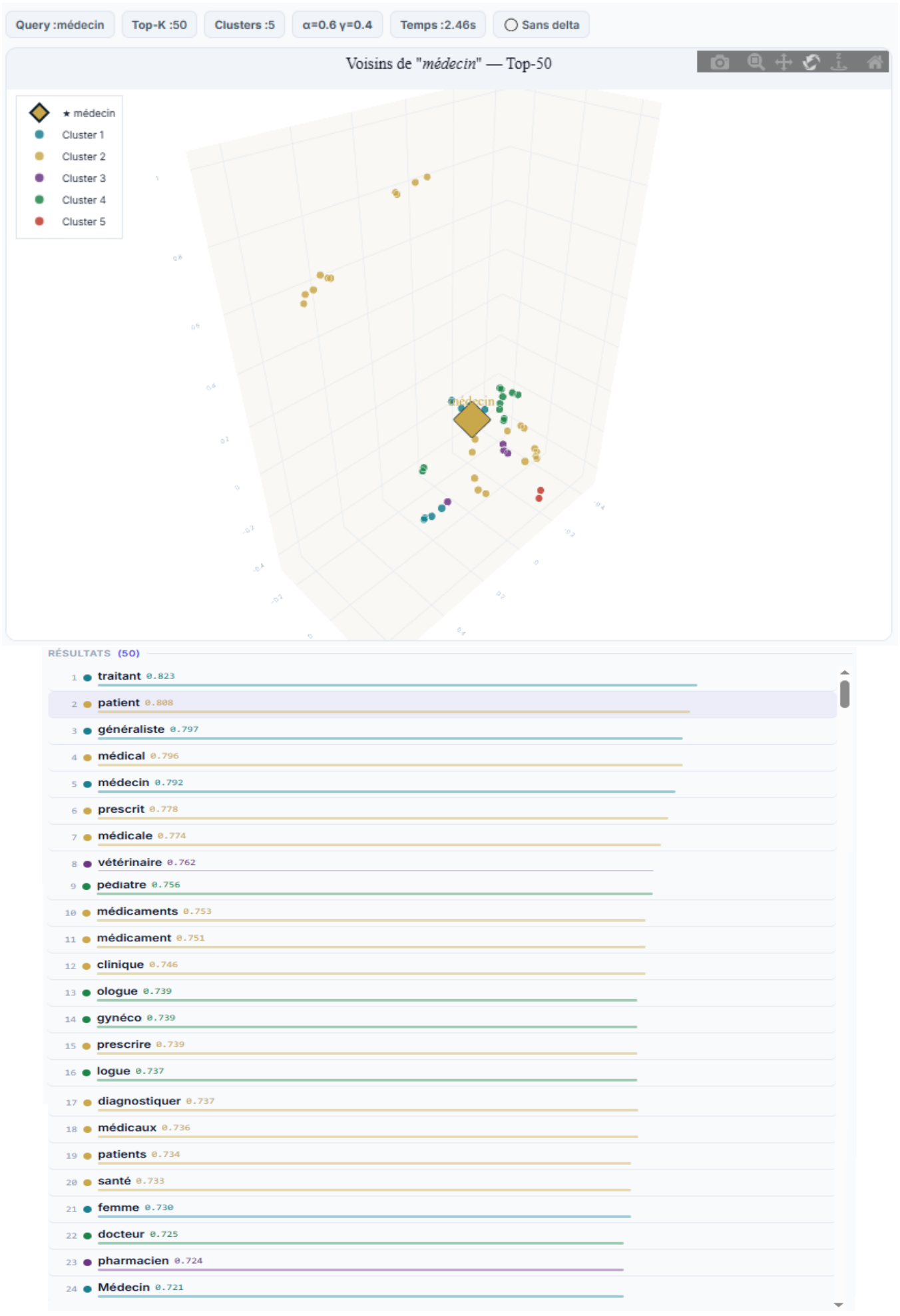
Objective

The Neural Navigator aims to visualise the organisation of the semantic space constructed by CoreMind. Through a three-dimensional projection based on UMAP (Uniform Manifold Approximation and Projection) [16], it enables the exploration of neighbourhood relationships between the vector representations learned by the model and makes it possible to observe their organisation within a low-dimensional space.

Principle

The vector representations generated by CoreMind exist within a high-dimensional space that is difficult to interpret directly. To facilitate their exploration, dimensionality reduction is applied using the UMAP algorithm [16], which projects the vectors into a three-dimensional space while preserving neighbourhood relationships as far as possible.

The Neural Navigator then makes it possible to query a given concept and visualise its nearest semantic neighbours. The projected concepts are automatically grouped according to their proximity, revealing the local structure of the semantic space as well as the different conceptual families constructed by the model.



La Figure A.8 *presents this exploration based on the concept "médecin", together with the principal neighbours identified by CoreMind.*

Analysis

The projection highlights a coherent organisation of the semantic neighbourhood around the concept “médecin”. The highest-ranked results primarily include medical professions (*généraliste, docteur, pharmacien, pédiatre, gynéco*), actors within the healthcare domain (*patient, traitant*), as well as concepts associated with medical practice (*prescrit, diagnostique, clinique, médicaments, santé*).

The three-dimensional visualisation shows that these concepts are not distributed randomly but form several local groups corresponding to different semantic substructures. The neighbour ranking confirms this organisation by presenting the closest concepts according to their similarity score. Overall, these results illustrate CoreMind’s ability to naturally structure a coherent semantic space without explicit supervision.

Conclusion

This experiment highlights a structured geometric organisation of the representations constructed by CoreMind, some of whose neighbourhood relationships can be visually explored using the UMAP projection [16].

The use of a UMAP projection [16] makes it possible to reveal the neighbourhood relationships and conceptual groupings present within the latent space, providing a particularly useful exploration tool for the qualitative analysis of the model. The Neural Navigator therefore provides a means of visually observing the structure of the knowledge learned by CoreMind and evaluating the coherence of the resulting representations.