cortical.io

WHAT IS SEMANTIC FOLDING?

Why converting text into semantic fingerprints enables highly efficient natural language understanding applications

WHITE PAPER

OVERVIEW

ADVANTAGES OF SEMANTIC FOLDING

Accuracy

1 Rich semantic feature set of 16K features allows a fine-grained representation of concepts.

2 By drastically reducing the vocabulary mismatch, far less false positive results are generated.

Transparency & Explainability

1 All semantic features are self-learned, thus reducing semantic bias in the language model used.

2 The descriptive features are explicit and semantically grounded and can be inspected for the interpretation of any generated results.

Efficiency

1 Less annotated examples to train the models (a few 100's versus a few 1,000's).

2 Sparse binary vectors provide orders of magnitudes of speed improvement over traditional methods using usual floating point matrices.

3 All Semantic Fingerprints have the same size, which allows for an optimal processing pipeline implementation.

4 The semantic representations are pre-calculated and therefore don't affect the query response time.

Flexibility & Scalability

1 Training of the semantic space is fully unsupervised.

2 Tuning of the model is purely data driven and only requires domain experts and no AI experts.

3 The applications can be easily integrated into larger systems by incorporating the API over REST.

4 Calculations take place at the semantic fingerprint level (without involving the whole semantic space) and are therefore easy to scale to any performance needed.

INTRODUCTION

Human language has been recognized as a very complex domain for decades. No computer system has so far been able to reach human levels of performance. The only known computational system capable of proper language processing is the human brain.

While we gather more and more data about the brain, its fundamental computational processes still remain obscure. The lack of a sound computational brain theory also prevents a fundamental understanding of Natural Language Processing (NLP). As always when science lacks a theoretical foundation, statistical modeling is applied to accommodate as much sampled real-world data as possible.

A fundamental yet unsolved issue is the actual representation of language (data) within the brain, denoted as the *Representational Problem*. Taking *Hierarchical Temporal Memory* (HTM) theory, a consistent computational theory of the human cortex, as a starting point, Cortical.io has developed a corresponding theory of language data representation: The Semantic Folding Theory.

Semantic Folding describes a method of converting language from its symbolic representation (text) into an explicit, semantically grounded representation called a semantic fingerprint. This change in representation can solve many complex NLP problems by applying Boolean operators and a generic similarity function like Euclidian Distance.

Many practical problems of statistical NLP systems and, more recently, of Transformer models, like the necessity of creating large training data sets, the high cost of computation, the fundamental incongruity of precision and recall, the complex tuning procedures, etc., can be elegantly overcome by applying Semantic Folding. This white paper will show how Semantic Folding makes highly efficient Natural Language Understanding (NLU) applications possible.

The process of encoding words, by using a topographical semantic space as a distributional reference frame into a sparse binary representational vector, is called **Semantic Folding** and is the central topic of this document.

PART 1

THEORETICAL BACKGROUND

The Semantic Folding theory is built on top of the Hierarchical Temporal Memory theory. Both theories aim to apply the newest findings in theoretical neuroscience to the emerging field of Machine Intelligence.

Hierarchical Temporal Memory

The Hierarchical Temporal Memory (HTM) theory is a functional interpretation of practical findings in neuroscience research. HTM theory sees the human neo-cortex as a 2D sheet of modular, homologous microcircuits that are organized as hierarchically interconnected layers. Every layer is capable of detecting frequently occurring input patterns and learning time-based sequences thereof.

The data is fed into an HTM layer in the form of Sparse Distributed Representations (SDRs).

SDRs are large binary vectors that are very sparsely filled, with every bit representing distinct semantic information. According to the HTM theory, the human neo-cortex is not a processor but a memory system for SDR pattern sequences.

Semantic Folding: A Brain Model of Language

By taking the HTM theory as a starting point, Semantic Folding proposes a novel approach to the representational problem, namely the capacity to represent meaning in a way that it becomes computable. **According to the HTM theory, the representation of words has to be in the SDR format, as all data in the neo-cortex has this format.**

The primary acquisition of a 2D-semantic space as a distributional reference for the encoding of word meaning is called Semantic Folding.

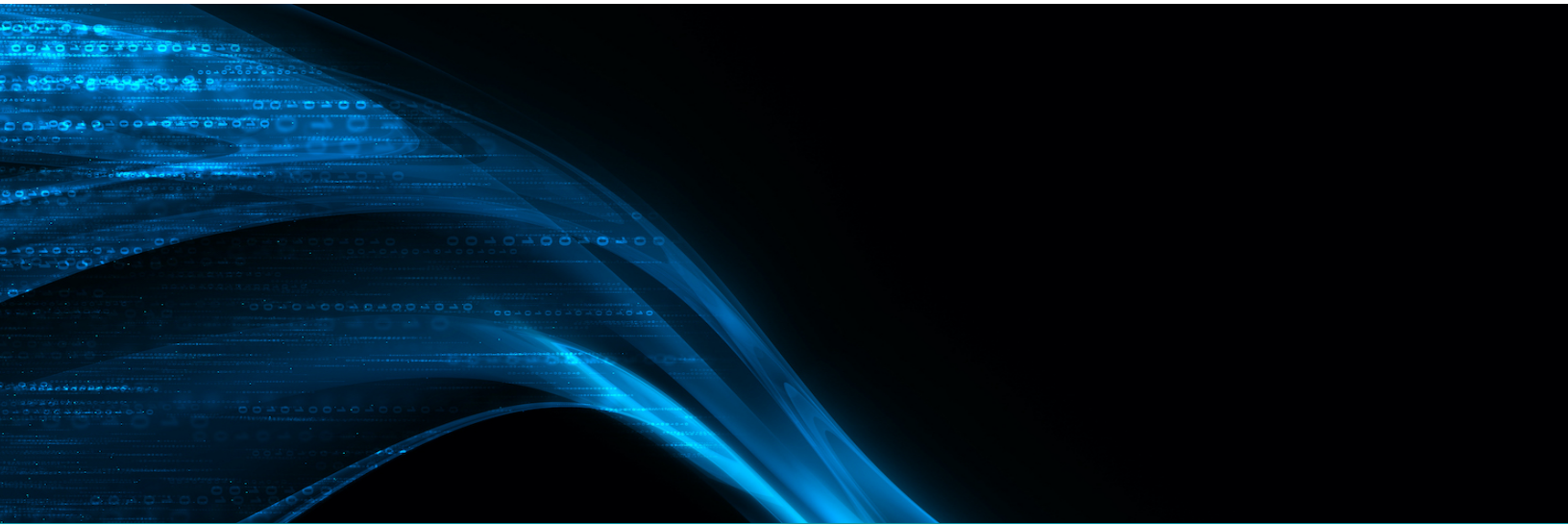
Every word is characterized by the list of contexts in which it appears. Technically speaking, the contexts represent vectors that can be used to create a two-dimensional map in such a way that similar context-vectors are placed closer to each, using topological (local) inhibition mechanisms and/or by using competitive Hebbian learning principles.

This results in a 2D-map that associates a coordinate pair to every context in the repository of contexts. This mapping process can be maintained dynamically by always positioning a new context onto the map.

This map is then used to encode every single word by associating a binary vector with each word, containing a “1”, if the word is contained in the context at a specific position and a “0” if not, for all positions in the map.

After serialization, we have a binary vector that possesses all advantages of a SDR:

- Every bit in a word SDR has semantic meaning.
- If a set bit shifts its position (up, down, left or right), the error will be negligible or even unnoticeable because adjacent contexts have a very similar meaning. This means that word SDRs are highly resistant to noise.
- Words with similar meanings look similar due to the topological arrangement of the individual bit-positions.
- The serialized word-SDRs can be efficiently compressed b only storing the indices of the set bits. The information loss is negligible even if subsampled.
- Several serialized word-SDRs can be aggregated using a bitwise OR function without losing any information brought in by any of the union’s members.



PART 2

SEMANTIC FOLDING: HOW DOES IT WORK?

The process of Semantic Folding encompasses the following steps:

1 Definition of a reference text corpus of documents that represents the *Semantic Universe* the system is supposed to work in. The system will know all vocabulary and its practical use as it occurs in this Language Definition Corpus (LDC). By selecting Wikipedia documents to represent the LDC, the resulting Semantic Space will cover general English. If, on the contrary, a collection of documents from the PubMed archive is chosen, the resulting Semantic Space will cover medical English.

2 Every document from the LDC is cut into text snippets with each snippet representing a single context. The size of the generated text snippets determines the *associativity bias* of the resulting Semantic Space. If the snippets are kept very small (1-3 sentences), the word *Socrates* is linked to *synonymous* concepts like *Plato, Archimedes or Diogenes.* The bigger the text snippets are, the more the word Socrates is linked to associated concepts like *philosophy, truth* or *discourse.* In practice, the bias is set to a level that best matches the problem domain.

3 The reference collection snippets are distributed over a 2D matrix (e.g. 128x128 bits) in a way that snippets with similar topics (that share many common words) are placed closer to each other on the map, and snippets with different topics (few common words) are placed more distantly to each other on the map. This produces a 2D semantic map.

4 In the next step, a list of every word contained in the reference corpus is created.

5 By going down this list word by word, all the contexts a word occurs in are set to “1” in the corresponding bit-position of a 2D mapped vector. This produces a large, binary, very sparsely filled vector for each word. **This vector is called the Semantic Fingerprint of the word.**

NOTE: the Semantic Folding process is described in this short video explainer: https://vimeo.com/manage/videos/670161429

Tuning a semantic space means selecting relevant representative training material. This content selection task can be best carried out by a domain expert, as opposed to the optimization of abstract algorithm parameters that traditionally requires the expertise of computer scientists.

Word-SDR – Sparse Distributed Word Representation

With Semantic Folding, it is possible to convert any given word (stored in the Semantic Space) into a word-SDR, also called a Semantic Fingerprint. The Semantic Fingerprint is a vector of 16,384 bits (128x128) where every bit stands for a concrete context (topic) that can be realized as a bag of words of the training snippets at this position.

Let's consider the Semantic Fingerprint of the word jaguar (Fig. 1). It contains all the different meanings associated with this term, like the animal, the automobile and the airplane contexts. The main contexts form clusters, that are easily recognizable and help disambiguate words with several meanings.

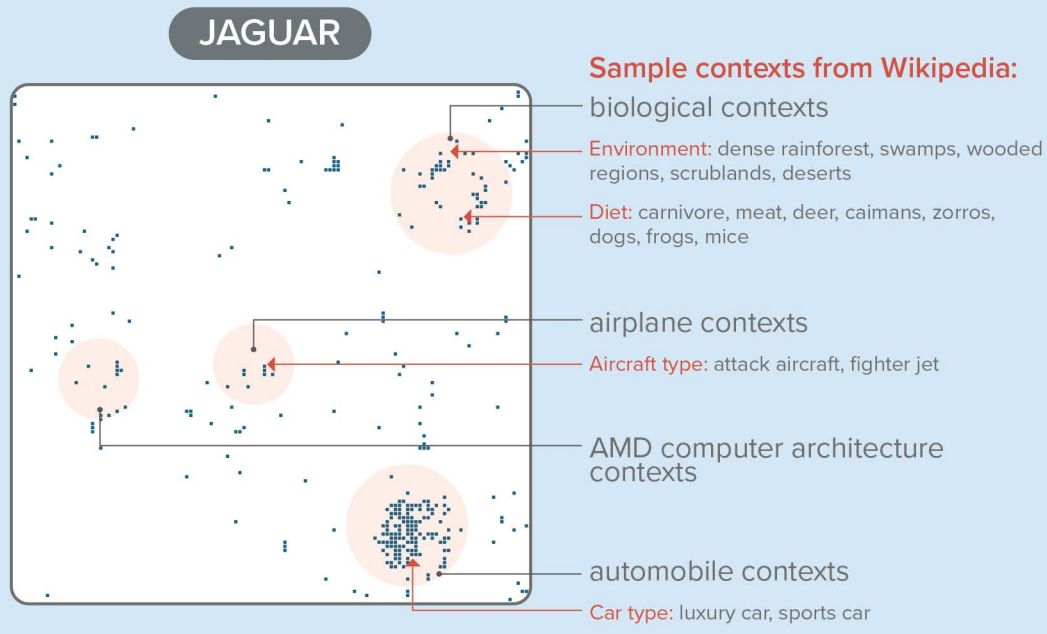


Fig. 1 Semantic Fingerprint of the word "Jaguar"

Document-SDR – Sparse Distributed Document Representation

The word-SDRs represent atomic units and can be aggregated to create document-SDRs (Document Fingerprints). Every constituent word is converted into its Semantic Fingerprint. All these fingerprints are then stacked and the most often represented features produce the highest bit stack.

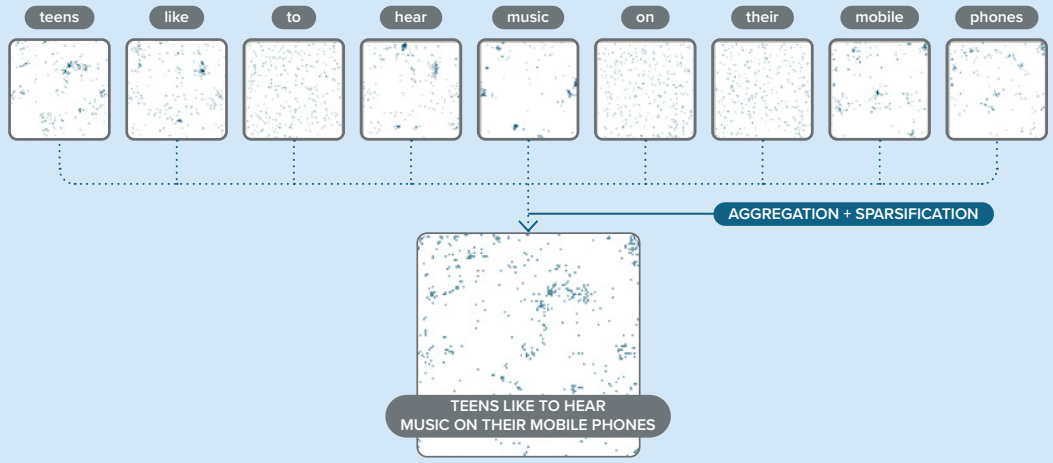


Fig. 2 Aggregation of word-SDRs into a text-SDR

The bit stacks of the aggregated fingerprint are now cut at a threshold that keeps the sparsity of the resulting document fingerprint at a defined level.

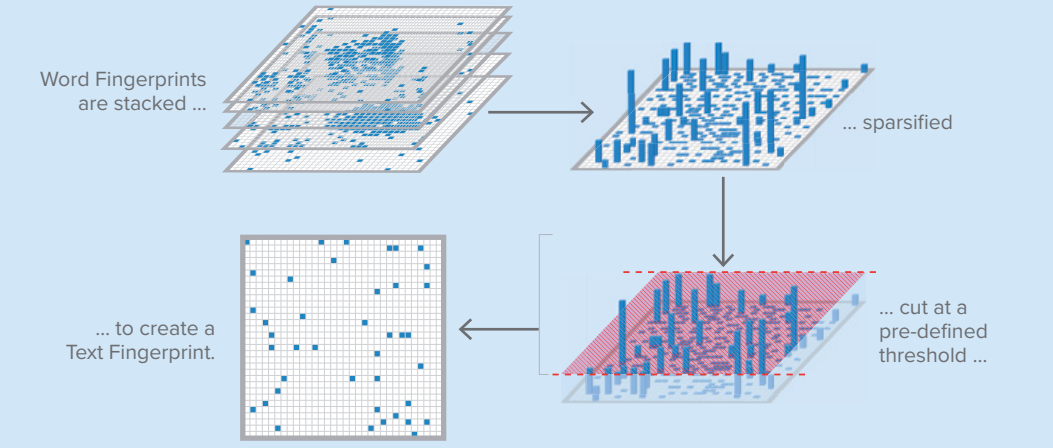


Fig. 3 Aggregation & Sparsification of text-SDRs into a document-SDR

The representational uniformity of word-SDRs and document-SDRs makes semantic computation easy and intuitive for documents of all sizes.

Applying Similarity as the Fundamental Operator

Due to the topological arrangement of the Semantic Fingerprints, similar words or texts do actually have similar Semantic Fingerprints. The similarity is measured in the degree of overlap between the two representations.

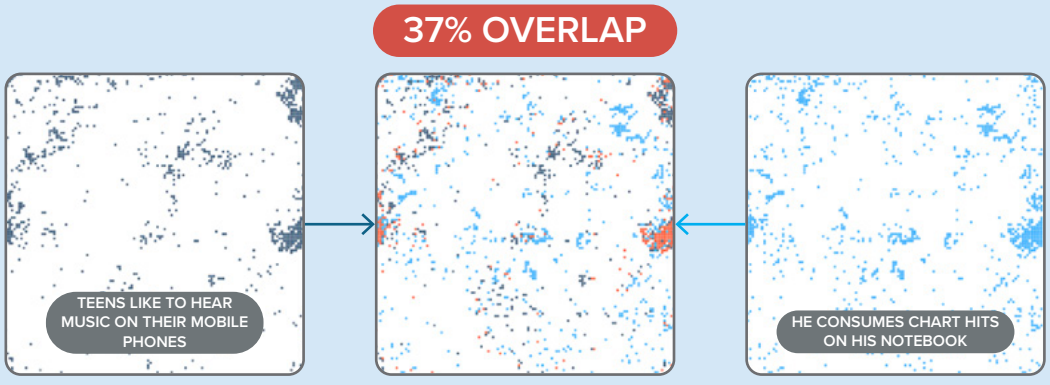


Fig. 4 Similar text snippets result in similar fingerprints

There are two different semantic aspects that can be detected while comparing two Semantic Fingerprints:

1 The absolute number of bits that overlap between two fingerprints describes the semantic closeness of the expressed concepts.

2 By looking at the topological position where the overlap happens, the shared contexts can be explicitly determined.

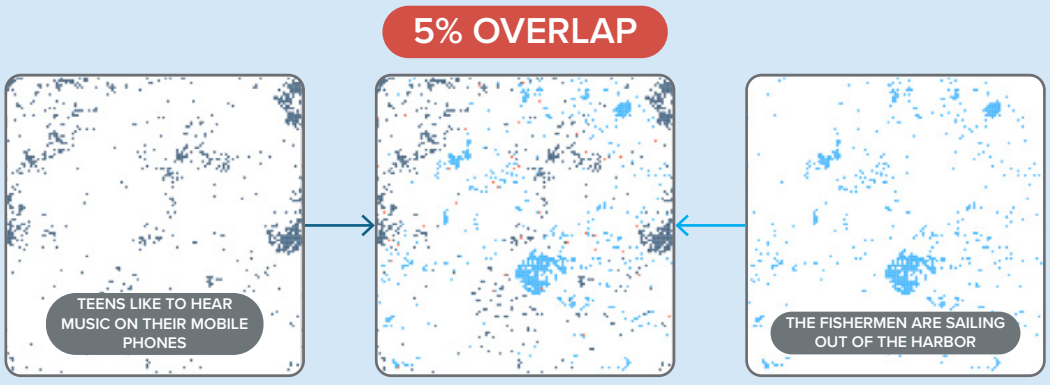


Fig. 5 Distinct text snippets result in dissimilar fingerprints

Because they are expressed through the combination of 16K features, the semantic differences captured by a Semantic Fingerprint can be very subtle.

PART 3

APPLICATIONS OF SEMANTIC FOLDING

Semantic Folding builds the basis for high-level NLP functionalities that can be integrated in many different applications

Classification of Documents

Traditionally, document classifiers are defined by providing a sufficiently large number of pre-classified documents and then by training the classifier with these documents. The difficulty of this approach is that many complex classification tasks across a larger number of classes require large amounts of correctly labeled examples. The resulting classifier quality degrades in general with the number of classes and their (semantic) closeness.

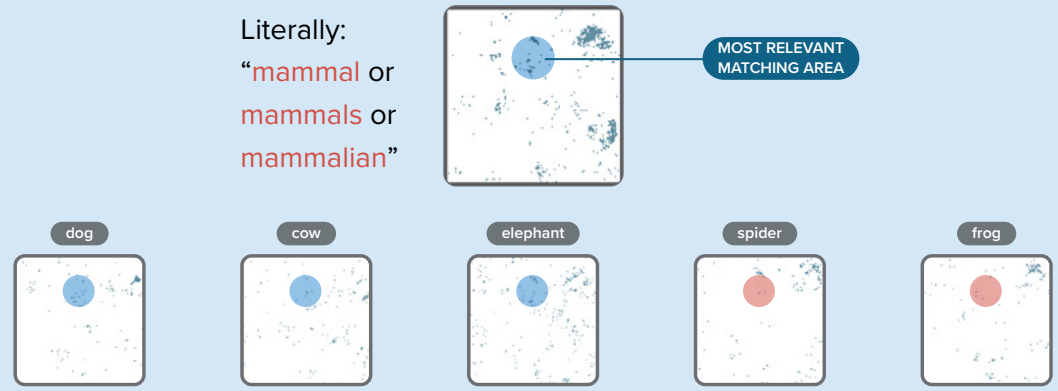


Fig. 6 Classification using Semantic Fingerprints

With Semantic Fingerprints, there is no need to train the classifier with many labeled examples. The only thing needed is a reference fingerprint that specifies an explicit set of semantic features describing a class.
This reference set or semantic class skeleton can be obtained either through direct description (enumerating a small number of generic class features and creating a Semantic Fingerprint of this list – for example the three words "mammal" + "mammals" + "mammalian", or by formulating an expression. By computing the expression: "tiger" AND "lion" AND "panther", a Semantic Fingerprint is created that specifies big cat features.

Searching Documents

Using document similarity for enterprise search has been on the agenda of many products and solutions in the field. Widespread use of the approach has not been reached mainly because, lacking an adequate document (text) representation, no distance measures could be developed that could keep-up with the more common statistical search models.

With Semantic Folding, searching is reduced to the task of comparing the fingerprints of all stored (indexed) documents with a query fingerprint that has either been generated by an example document ("Show me other documents like this one") or by typing in a description of what to look for ("Acts of vengeance of medieval kings").

After the query fingerprint is generated, the documents are ordered by increasing distance. In contrast to traditional search engines, where a separate ranking procedure needs to be defined, the fingerprint-based search process generates an intrinsic order for the result set.

Additionally, it is possible to provide personalized results by simply allowing the user to specify two or three documents that relate to his/her interests or working domain (without needing to be directly related to the actual search query). These user-selected domain documents are used to create a user-profile-fingerprint. Now the query is again executed and the (for example) 100 most similar documents are selected and are sorted by increasing distance from the profile fingerprint.

Like this, two different users can cast the same search query on the same document collection and get different results depending on their topical preferences.

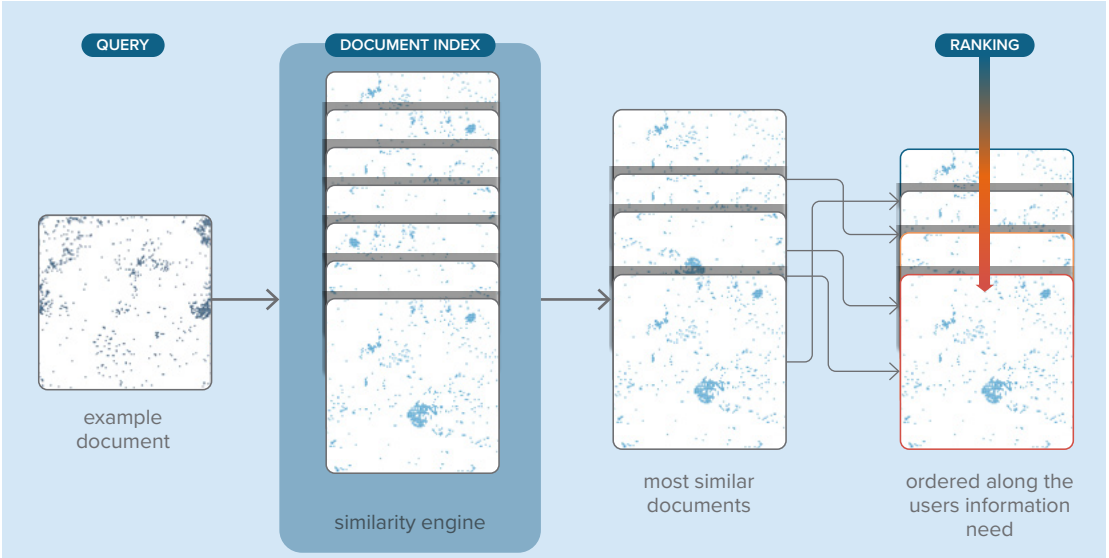


Fig. 7 A similarity based configuration of a search system

Cross-Language Ability

If aligned semantic spaces for different languages are used, the resulting fingerprints become language independent.

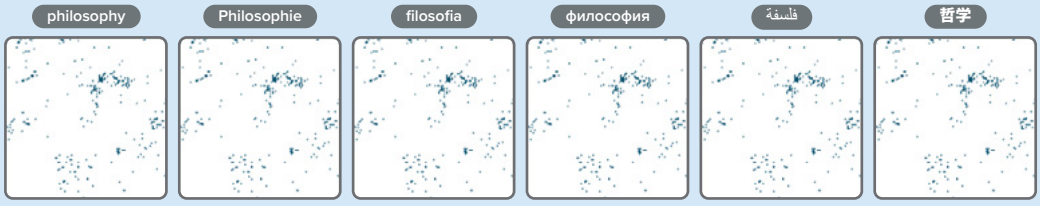


Fig. 8 Language independence of Semantic Fingerprints

This means that an English message-fingerprint can be directly matched with an Arabic message-fingerprint. When filtering text sources, the filter criterion can be designed in English while being directly applied to all other languages. An application can be developed using the English Semantic Space while being deployed with a Chinese one.

Since the publication of the original Semantic Folding white paper in 2016*, the implementation of several applications leveraging the advantages of Semantic Fingerprints to solve real-world challenges in enterprise environments has validated the viability of this method.

To learn more about applications of Semantic Folding for natural language understanding, visit www.cortical.io or contact info@cortical.io

The original white paper Semantic Folding Theory was published in 2016 on Arxiv.