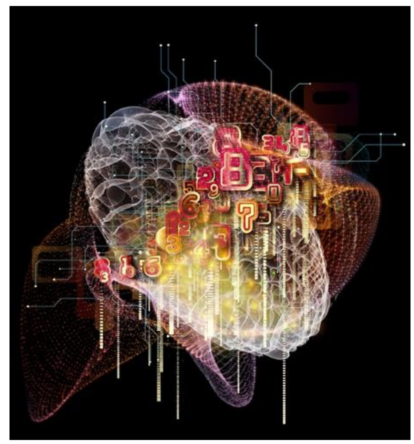
www.pipelinepub.com
Volume 18, Issue 11

Semantic Folding: A Brain Model of Language

By: Francisco Webber



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, and so on can be elegantly overcome by applying Semantic Folding. This article 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.

The Semantic Folding Theory

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 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 an SDR:

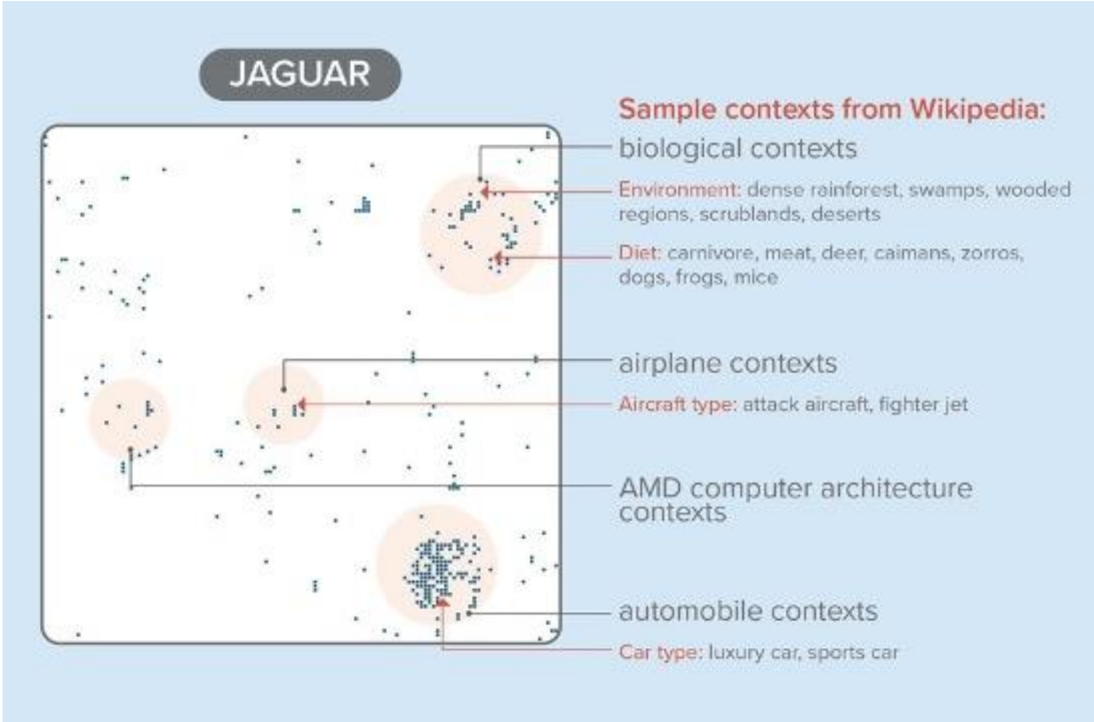


Fig. 1 Semantic Fingerprint of the word "Jaguar"

- 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 by 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.

Semantic Folding: how does it work?

The process of Semantic Folding encompasses the following steps:

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.

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.

The reference collection snippets are distributed over a 2D matrix (for example 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.

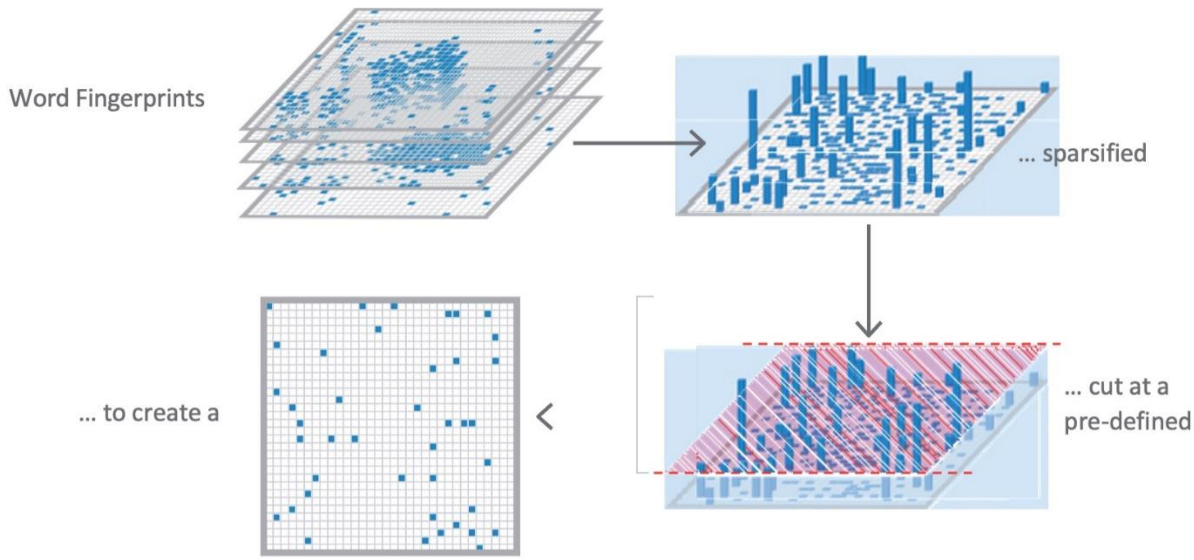


Fig. 2: Aggregation & Sparsification of text-SDRs into a document-SDR

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

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.

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* (see Fig. 1 on previous page). 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.

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.

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 (see Fig. 2 on previous page).

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 (see Fig 3).

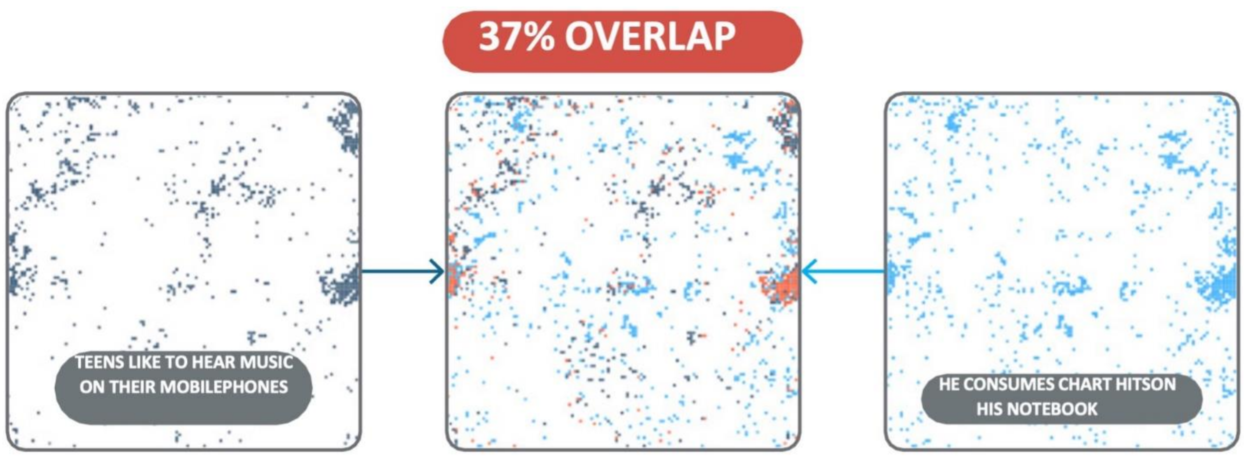


Fig. 3. Similar text snippets result in similar fingerprints

There are two different semantic aspects that can be detected while comparing two Semantic Fingerprints (see Fig 4 on next page):

- The absolute number of bits that overlap between two fingerprints describes the semantic closeness of the expressed concepts.
- By looking at the topological position where the overlap happens, the shared contexts can be explicitly determined.

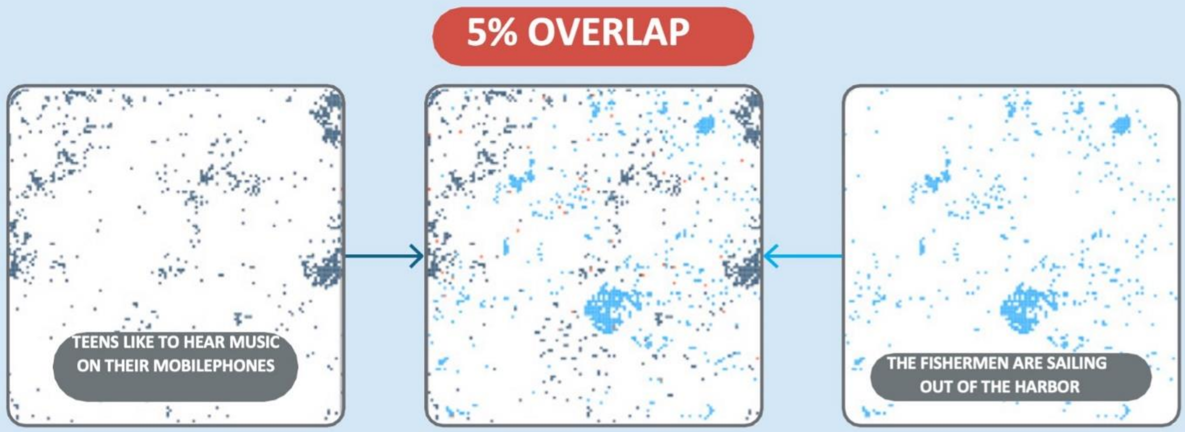


Fig. 4 Distinct text snippets result in dissimilar fingerprints

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

Semantic Folding opens new horizons for automating workflows involving large volumes of complex documents that currently still rely on human review and interpretation, like contract analysis and insurance policy review. With its ability to understand the meaning of natural language and to process data in real time, Semantic Folding offers also a great opportunity to build next generation Know Your Customer tools by extracting insights from media posts and online customer reviews.