f.webber@cortical.io
Mariahilferstrasse 4
1070 Vienna, Austria
http://www.cortical.io

ARGUMENTS FOR: SEMANTIC FOLDING AND HIERARCHICAL TEMPORAL MEMORY

Replies to the most common Semantic Folding criticisms

Francisco Webber, September 2016

Introduction

During the last few years, the big promise of computer science, to solve any computable problem given enough data and a gold standard, has triggered a race for machine learning (ML) among tech communities. Sophisticated computational techniques have enabled the tackling of problems that have been considered unsolvable for decades. Face recognition, speech recognition, self-driving cars; it seems like a computational model could be created for any human task.

Anthropology has taught us that a sufficiently complex technology is indistinguishable from magic by the non-expert, indigenous mind. Science Fiction culture, on the other hand, has nurtured the vision of machines becoming able to do what so far has only been possible for humans. One of those abilities being to fully understand the meaning behind natural language artifacts.

The majority of these computational ML models are statistical in the sense that they make strongly idealized assumptions on the data they are applied to. These a priori concepts are typically place holders for the missing conceptual model that would fully explain the phenomenon at hand.

One of the most intriguing aspects of the human brain is the fact that it does not seem to make any assumption at all about the data it gets exposed to. The brain can hear with the visual cortex and see with auditory areas, indicating that it uses model free methods.

As John Ball nicely pointed out in his blog post (A.I. is too hard for programmers):

> *"The biggest difference between computers and brains? Computer programmers define the general to store specifics, but brains store the specific to identify the general. Brains learn this way, but computers don't."*

It might well turn out that universal intelligence is more a matter of architecture than of any applied algorithm, making it intrinsically impossible for Allan Turing's "universal computing machine" to ever reach the performance levels of the "highly specialized computing machine": the human brain.

Motivation for Semantic Folding

Solving 'Hard' Problems

Natural language understanding is an AI-complete problem because it cannot be solved by an algorithm alone, but also needs *world-knowledge* to achieve human performance levels[1].
Humans, or more precisely human brains, produce language and understand language and therefore possess all necessary tooling to perform these tasks.
The common *natural science*[2] approach to investigate a non-obvious phenomenon is to collect data by measuring relevant variables associated with the phenomenon. By studying the sampled real-world data, the scientific mind can apply inductive, deductive, analogical (…) reasoning to create a *theory* that describes the *systemic atoms* involved and their *interactions* in a formal framework. If a specific phenomenon is not yet explainable by a complete theory, it is common practice to create a *computational model* that simulates the system under observation to the extent that it is able to generate data that is as similar as possible to the sampled real-world data. By lacking a sound theory, the nature of the system-atoms involved is often unclear and can only be represented using statistical descriptors, which finally results in a statistical model of the system. Although statistical modeling offers very convenient **ways** to describe complex systems and their behaviors, the precision margin of the statistical features used and the necessary conceptual simplification introduce a fundamental system-error, making extensive tuning and refining steps necessary before a model becomes useful in a real-world setting. The model-free approach of a theory is to completely avoid this model-error, at the cost of having to create a sound theory in the first place.

Example: Early pharmaceutical research has been investigating natural molecules mostly from plants or other simple organisms for their clinical use as remedies against certain diseases by making large numbers of matching-experiments and deriving a statistical model. Since these early days, a large number of molecular mechanisms that constitute the pharmacological effect have been identified and today form a theoretical framework that can be applied to molecular synthesis processes. As a result, drugs are nowadays more often designed than discovered.

The Nature of Neuro-Computation

Jeff Hawkins' declared goal is to understand how the human neocortex works or in other words to uncover the theory of cortical computation. In his approach, he is taking generic biological data as input (neuro-scientific findings), which he consolidates into hypotheses from which he derives a technical system (software) that can be used to generically measure the correctness of the hypothesis. Every confirmed hypothesis further completes the Hierarchical Temporal Memory (HTM) theory. The technical system generated by this process is implemented as the Nupic package, open-sourced and supported by the company Numenta. The resulting HTM theory describes a precise set of constraints and mechanisms that govern every cortical process.

The Representational Problem

Since the early days of Natural Language Processing (NLP), a central problem has remained: How to represent symbolic natural language units, like words, sentences, paragraphs or books, as directly computable items in a similar way to numbers. What is

---

[1] Natural Language Understanding – "Encyclopedic knowledge is required to understand natural language. Therefore, a complete natural language system will also be a complete intelligent system."
FROM: "AI-Complete, AI-Hard, or AI-Easy: Classification of Problems in Artificial Intelligence", Roman V. Yampolskiy

[2] Scientific fields are commonly divided into two major groups: Natural sciences, which study natural phenomena (including biological life), and social sciences, which study human behavior and societies. (Wikipedia)

obviously missing is a representational theory for language[3]. As language is computed in the human neocortex, this representational theory should also comply with HTM theory and its constraints.

On Statistical Modeling

This fundamental lack of a representational theory has resulted in NLP relying heavily on statistical feature modeling.
While statistical language models have in general achieved a high degree of sophistication, complexity and usecase specialization, they are still subject to the systemic restrictions of statistical modeling:

- Statistical language models need large amounts of training data, orders of magnitude more than humans would need.
- Statistical language models are oversimplified compared to real human language.
- The oversimplification induces semantic noise, which leads to a high false positives rate, requiring humans for correction.
- The internal data structures of statistical language models often consist of very large floating point matrices, requiring a large computational effort for complex semantic processing.

Frequent Criticisms of Semantic Folding

"Semantic Folding is JUST Word Embedding"

This argument is like dismissing 'Tesla' for being JUST a car company. Semantic Folding forms the word vectors using distributional encoding of explicit features, leading to a high dimensional sparse representation, while traditional word embeddings use dimensionality reduction to create low-dimensional, dense word representations.
Here are the top four advantages of Semantic Folding over statistical word embeddings:

Higher Semantic Payload of Features

A central aspect of Semantic Folding's first encoding step is the degree to which semantic information is captured during the sampling process. This corresponds to the achievable signal to noise ratio of a general purpose sensor intended to record a physical value in a natural (noisy) environment. The overall achievable resolution directly depends on the sensor's dynamic range.
Word embeddings like Word2Vec generate dense vectors of several hundred features, while Semantic Folding easily accommodates several tens-of-thousands of features (the current reference implementation uses 16384 features).
Example:

- Word2Vec can be used to disambiguate the word "APPLE" to a computer company and a fruit.
- Semantic Folding disambiguates the word "APPLE" into the contexts: hardware, software, mobile device, agriculture, cooking, botany, record label

(Both systems trained exclusively on Wikipedia data.)
The topological distribution of the features reduces contextual cross-talk as it allows the simultaneous representation of all contextual meanings of a word, for the given semantic space.

[3] "Much work in traditional artificial intelligence has ignored the process of high-level perception by starting with hand-coded representations. In this paper, we argue that this dismissal of perceptual processes leads to distorted models of human cognition" - High-Level Perception, Representation, and Analogy: **A Critique of Artificial Intelligence Methodology**; *David J. Chalmers, Robert M. French, Douglas R. Hofstadter*, Center for Research on Concepts and Cognition, Indiana University, CRCC Technical Report 49 — March 1991

Universal Compositionality of the Representation

Because all atomic word representations use the same distributional grid of the underlying, clustered, semantic space, they are all directly comparable to each other by means of a binary overlap. By making a binary union of all word representations of a given sentence, a representation for a whole sentence can be built without losing any information content[4]. This compositional approach can be used for any larger text elements (sentences, paragraphs, documents books etc.) always leading to a single, unique and topologically compatible representation.
In contrast, the statistical distribution of words does not directly correspond to the frequencies of larger structures like word-tuples, triplets or whole sentences, thereby prohibiting a direct comparison between words, groups of words, sentences, paragraphs etc., essential for higher order semantics.

Transparent and Explicit Features

Due to the fact that all word vector features are explicit, it is possible to inspect their content and, therefore, tune the semantic space to get the best performance for a given use case. The tuning is done by selecting the appropriate documents to be part of the document collection used to generate the semantic space. The selection of appropriate content is obviously a skill that topic experts (most of the time the owners of the data) can do. In contrast, the tuning of statistical systems must be realized on the algorithmic level by data scientists who are knowledgeable in the mathematics of the algorithm used.
Moreover, having the actual utterances behind every position on the semantic map makes it possible to align the topological structure of the features for multiple languages, enabling the direct comparison of a word vector representation in one language to one in a different language. This is not efficiently achievable with a statistical encoding approach, where no topology is encoded in the semantic space.

Efficient Binary Computation

The Semantic Folding word vector is constituted of binary features, which makes it much easier to process than the double or floating-point features resulting from a statistical encoding. In particular, Boolean operations on binary numbers have proven to be the most efficient using today's scalar microprocessors. Furthermore, the sparse binary vectors generated by the Semantic Folding approach have a substantially smaller memory footprint and can be highly compressed without loss of information.
In contrast with the costly multi-step floating-point calculus, necessary for performing complex NLP functions with statistically modeled vectors, operations on sparse binary vectors can be realized through simple Boolean operations, often in a single step.

Scientific Context

Semantic Folding is not the only approach taking advantage of the properties of sparse binary representations; there has also been work published by members of the word embedding community in an attempt to enable these benefits for their methods.
Faruqi et al. [21] transform pre-trained Word2Vec dense double vectors into sparse binary representations. Several explorations of sparsity as a useful form of inductive bias in NLP and machine learning more broadly, have been published by Kazama and Tsujii, 2003 [22]; Goodman, 2004 [23]; Friedman et al., 2008 [24]; Glorot et al., 2011 [25]; Yogatama and Smith, 2014 [26], among others.
Introducing sparsity in word vector dimensions has been shown to improve dimension interpretability (Murphy et al., 2012 [27]; Fyshe et al., 2014 [28]) and usability of word vectors as features in downstream tasks (Guo et al., 2014 [29]).

---

[4] "The Surprising Union Property", Chapter in *"Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory"*, by S. Ahmad & J. Hawkins, https://arxiv.org/pdf/1503.07469.pdf

"Why not use the open sourced Word2Vec?"

Word2Vec was created by Tomas Mikolov and his team at Google and is freely available on Google code. It is part of the family of word embedding models, which all have the goal to generate a context relative vector representation for words using a shallow neural network that is trained to capture semantic features.
Although word vectors represent the newest generation of models for analyzing (and training) semantic language features, the usefulness of the approach is not only determined by the format in which the semantic payload is formatted but also by the chosen representation.
By using dense, algebraic models, Tomas Mikolov's Word2vec, Stanford University's GloVe, Deeplearning4j and similar word embeddings have fundamental limitations, making it often hard for them to be used for many practical business cases. Here are the top arguments why these models might not be the best choice:

Dense word embeddings cannot capture ambiguity

The shallow semantics of dense word embeddings show only limited usability in contexts where ambiguity of words is an issue. The system might figure out that Italy is to Rome what France is to Paris, but it does not link Italy to Spaghetti and Leonardo da Vinci etc.
Richard Socher in a 2016 Lecture[5]:
You may hope that one vector captures both kinds of information (run = verb and noun) but then vector is pulled in different directions

Word embeddings lack semantic grounding

A practical problem of many machine learning approaches to NLP is the low predictability of the resulting quality. A specific setup and parameter set might work well in one case but not perform at all in another. Because the vectors are not semantically grounded, it is not possible to *debug* the model by tracing back into individual feature-values.
Another problem when using an algebraic model instead of explicit feature encoding is that word embeddings cannot efficiently scale a use case across different languages. Each language will need its own trained model.

Model based methods suffer from false positives

A practical NLP system is exposed to many different error sources: language ambiguity, irregular artifacts and biases in training data, oversimplification or statistical feature generation.
Even if only a small number of features in a ML word vector are wrong or omitted, the semantic vector representation can turn out erroneous and lead to an incorrect similarity measure. In contrast, when a small number of the features of a Semantic Folding representation are missing or misplaced, the overall similarity of two similar words is not substantially influenced because the majority of the features will continue to overlap. Errors in the similarity computation lead, at the application level (e.g. in a classifier), to false positives (e.g. wrongly classified documents), requiring additional, costly, human intervention in the business context.

The word2vec feature vector is not compositional

Representations generated through Semantic Folding are all relative to a normative semantic space. Any two Semantic Folding vectors can be directly compared, independently of the content they represent (words, phrases, sentences, paragraphs, documents etc.). Dense embeddings, on the other hand, are tightly associated with the granularity into which the training material has been segmented and do not, for example,

[5] Richard Socher, CS224d - Deep Learning for Natural Language Processing Lecture 3: More Word Vectors, (https://cs224d.stanford.edu/lectures/CS224d-Lecture3.pdf"

allow a generic comparison of a word2vec vector with a paragraph2vec vector. Many real world use cases, however, need to compare words to documents (search), paragraphs to documents (data mining) etc.

"Semantic Folding has no comparative evaluation"

Even within academic circles, the evaluation of algorithms has turned out to be a very difficult task (e.g. Faruqui 2016 [30] and Hill 2015 [31]). In hardly any other scientific domain do published experiments have such a low reproducibility as in the field of machine learning-based natural language processing.
A majority of the papers describe a specific algorithmic approach that is tightly coupled to a specific evaluation framework. The targeted improvement needed to make the results publishable is often achieved by optimizing the method to the evaluation set. As a result, the same algorithm might be the top performer for one evaluation while having bad results in another. In addition, the evaluation sets are often not representative of real world data (Marelli 2014 [32]).

But in any case, Cortical.io is a business not a research organization and, as in all businesses, the evaluation that really counts is customer satisfaction.
Faruqui 2016 [30] concludes:

"Word vector models should be compared on how well they can perform on a downstream NLP task."

Cortical.io uses a collection of over 35+ evaluation sets for quality control in the software building process. Whenever the build-process for a new version of the Cortical.io Retina Engine is triggered, 35+ different evaluations are executed. This evaluation data is accessible for Cortical.io customers. An interesting observation is that while Cortical.io is never the highest scoring in any of the evaluation sets it scores highly in all cases. Most other approaches only perform well in a small fraction of the 35+ tests.
While Cortical.io is not actively participating in academic competitions, academic cooperation is very welcome. So far, a paper by a research group from Toulouse [17] has been published and Cortical.io has played host to several research interns.

"Semantic Folding has *ONLY* logical/philosophical arguments"

The use of the word *only* in this criticism seems inadequate. A logical proof can sometimes be even stronger than an experimental proof as the latter could just point to a local maximum, whereas the associated theory allows the prediction of an absolute, experimentally testable maximum. Pure experimental research without a fundamental theory can often become a stochastic endeavor. But the creation of a sound theory, on the other hand, calls for enough experimental data to allow inductive reasoning.
Another advantage of developing a technology along a theoretical framework is that extensions and new functional features can very often be predicted theoretically, making it much more efficient than mass trial–and-error experimentation.
The Semantic Folding theory does not solely rely on neuro-scientific findings but also has strong mathematical foundations in the work of researchers like Pentti Kanerva [10], Teuvo Kohonen [11], Hinrich Schütze [12] or Magnus Sahlgren [13].

Frequent Criticisms of HTM Theory

"Jeff Hawkins' work has been mostly philosophical and less technical"

This is exactly right: Jeff's declared goal is to develop a theory about cortical processing. Similar arguments to the same criticisms directed at Semantic Folding also apply here. A sound theory might be worth a thousand experiments. In the context of Jeff Hawkin's work it is the actual goal to work out the theoretical framework. The technical implementation is used for verification. There are not many companies in the field who have a similar degree of transparency and openness. Every part of the theory and its implementation are openly accessibly and every improvement is continuously reported. The constantly evolving Nupic [18] implementation is a very technical way to understand and investigate the HTM approach.

"There are no mathematically sound proofs or validations of the HTM assertions"

Mathematical proofs for Deep Learning are founded on a very unrealistic and mathematically simplistic interpretation of the fundamental unit: The neuron. Real neurons are, of course, very complex creatures with many physiological and molecular mechanisms working together. But in the context of finding a *theory of operation* it is not necessary to simulate neurons at the implementation level but on the functional level. And it is this realistic functional level that turns out to be substantially more complex than the one in the traditional neural network model ([5] Ahmad & Hawkins).
Other authors have also started to mathematically explore aspects like *sparse coding* (Olshausen & Field [7]) or *spatial pooling* (Mnatzaganian et al. [9])

"There is no real-world success"

Hierarchical Temporal Memory theory is of course, first of all, a theory. A successful theory is generally able to make predictions within the area of action that are able to be confirmed by empirical means.

- HTM theory has already made several prognoses that were later able to be confirmed. For example, one very specific prediction of the temporal memory is that overall cell activity becomes sparser during a continuous predictable sensory stream and was confirmed in [19]. Another example is the case where the HTM has generated a prediction pattern that proves correct learning should occur only in the specific distal dendritic segment which became active previously. Some initial evidence for this is in [20].
- In practical terms, Nupic program code has proven to be particularly well-suited for anomaly detection products like *HTM for Stocks* or *GROK for IT-Analytics*. This domain of anomaly detection in complex systems is currently underserved by machine learning vendors.
- From the Cortical.io perspective, the biggest success of HTM theory has been to enable the development of Semantic Folding technology, which is rapidly gaining traction with global NLP customers.

However, the biggest form of success for a scientific theory remains as that of coherently explaining a given observable phenomenon in its entirety.

"There is no comparison with more popular algorithms of Deep Learning"

Comparisons of algorithmic methods are quite problematic as described earlier. The purely experimental comparison is highly complex to realize when maintaining high scientific standards. So, in practice, a combination of theoretical arguments with adequate supporting experiments is a much more realistic approach for small scale research organizations.
In the Numenta research paper, "Continuous online sequence learning with an unsupervised neural network model" [2], the HTM-sequence memory performance is compared with other sequence learning algorithms, (including statistical methods: autoregressive integrated moving

average (ARIMA), feed-forward neural networks: online sequential extreme learning machine (ELM), and recurrent neural networks: long short-term memory (LSTM) and echo-state networks (ESN)), on sequence prediction problems with both artificial and real-world data. The outcome of the comparison has two very relevant results:

- The HTM system achieved a comparable accuracy with state of the art approaches. This means that even with its tight biological constraints, HTM theory seems to have reached the precision of the current state of the art computational models.
- The evaluation has demonstrated that HTM networks do actually exhibit certain collateral advantages as predicted by HTM theory. Abilities like *continuous online learning*, *multiple simultaneous predictions*, *noise resistance* and *fault tolerance,* among others, often turn out to be key factors when it comes to implementing the algorithm in the context of real-world challenges.

Considering that HTM theory is still formally incomplete, substantial improvements can be expected in the foreseeable future.

"Google/Deep Mind present their performance metrics, why not Numenta"

Google and other big players in the AI arena indeed publish example code, and performance metrics relative to them. As mentioned earlier, it is often very hard to reproduce these results on a practical level, due to resource limitations for smaller organizations. When comparing the amount of openly published data and tools to the number of active researchers working at these big players, it still seems very likely that the large majority of work in the domain remains in the dark. Numenta, in contrast, has, since its inception published every aspect of their work, be it theoretical or technical in nature.

fMRI support for Semantic Folding Theory

Semantic Folding theory argues that language is internally represented by a *sparse binary, topologically distributed pattern of explicitly observable features* [14].
As early as 2008, the pioneering work of Tom Mitchell et al. [15] has shown that there is a strong correlation between the sparse binary distributed patterns of cortical activity captured via fMRI (functional magnetic resonance imaging) and language content like words. This correlation is strong enough to be able to train a classifier capable of detecting a word based on the fMRI-activation pattern.
More recently, Huth & Gallant et al. [16] were able to create a map of the distributed word representations across the neocortex of actual test persons. This representational map even turned out to be consistent across individuals, as T. Mitchell could show in his experiment. This intercortex consistency of the topology of the map is one of the first biological validations for the concept of a single semantic map, used to semantically ground the neural word representations, a key element of the Semantic Folding theory. Very few language machine learning techniques have ever been able to get such direct support from biological findings.

Reality Check: Business Applicability

As previously stated, Cortical.io regards customer feedback as the most important objective evaluation method for the usefulness of its products and the associated technological approach. For the last 18 months, Cortical.io has systematically exposed Semantic Folding technology to as many businesses in as many domains and use cases as possible. This outreach resulted in about two dozen large enterprise customers in Europe and the US, scaling from 4K to 600K employees. All of them are engaged in an exploration process with several steps: learning about Semantic Folding; identifying, discussing, prototyping and in several cases even implementing their most pressing use cases using Cortical.io's *Language Intelligence* approach.

Global Industry Under Pressure

The dominant finding was the tremendous pressure that all of these organizations have to find a way to systematically integrate the multitude of and ever increasing amounts of text-based information into their business processes.
Even the most conservative sectors seem to have realized that their future successes or even survival depends on their skill in harnessing the wealth of language data available.

- Producers or sellers of consumer products need to better understand their customers.
- Financial services need to monitor their business processes to comply with increasingly refined regulations.
- Media services need to make their content offerings findable.
- Technical support centers have to improve efficiency in solving customer requests.
- Service centers need to automate their agents to improve customer experience.
- Business and legal services need to make their contracts machine understandable.
- Internet-based businesses need to personalize their services on a global scale.

While this list is just one of use case patterns that are common in all sectors, the full truth is that, at some point, every actor in a business, be it a human or a system, will need machine-empowered *Language Intelligence* to stay efficient.

"We have tried everything …"

Considering the amount of buzz around all forms of machine learning and the ubiquitous hope that intelligence will emerge by just leveraging enough data, it was a surprise to see how little ML code has actually been deployed in production.
"We have tried everything but nothing worked out so far.", became a very common refrain. The most common reasons for failed ML-NLP initiatives have been the large number of *false positives* when trying to apply *classifiers*, the lack of *training data*, the difficulty of *sourcing gold standards*, the impossibility of tuning *language models* by analyzing the trained state in the underlying *DL network*, to name just a few.
Interestingly, it was not the promise of higher precision in benchmarks, but the systematic avoidance of these problems when applying *Semantic Folding*, that restored confidence.

References

[1] Hawkins, J., and Ahmad, S. (2016). Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex. *Front. Neural Circuits* 10, 1–13. Doi:10.3389/fncir.2016.00023. http://journal.frontiersin.org/article/10.3389/fncir.2016.00023/full

[2] Cui, Y., Surpur, C., Ahmad, S., and Hawkins, J. (2015). Continuous online sequence learning with an unsupervised neural network model. arXiv:1512.05463 [cs.NE]. Available at: http://arxiv.org/abs/1512.05463.

[3] Lavin, A., and Ahmad, S. (2015). Evaluating Real-time Anomaly Detection Algorithms – the Numenta Anomaly Benchmark. In *14th International Conference on Machine Learning and Applications (IEEE ICMLA'15)* (Miami, Florida: IEEE). Doi:10.1109/ICMLA.2015.141.

[4] Rozado, D., Rodriguez, F. B., and Varona, P. (2012). Extending the bioinspired hierarchical temporal memory paradigm for sign language recognition. *Neurocomputing* 79, 75–86. Doi:10.1016/j.neucom.2011.10.005.

[5] Ahmad, S., and Hawkins, J. (2016). How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites. arXiv:1601.00720 [q–bio.NC]. Available at: http://arxiv.org/abs/1601.00720.

[6] Ahmad, S., and Hawkins, J. (2015). Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory. *arXiv*. Available at: http://arxiv.org/abs/1503.07469.

[7] Olshausen, B. A., and Field, D. J. (2004). Sparse coding of sensory inputs. *Curr. Opin. Neurobiol.* 14, 481–487. Doi:10.1016/j.conb.2004.07.007.

[8] Kanerva, P. (1988). *Sparse Distributed Memory*. Cambridge, MA: The MIT Press.

[9] Mnatzaganian, J., Fokoué, E., and Kudithipudi, D. (2016). A Mathematical Formalization of Hierarchical Temporal Memory Cortical Learning Algorithm's Spatial Pooler. Available at: http://arxiv.org/abs/1601.06116 [Accessed January 25, 2016].

[10] Kanerva, Pentti (1988). *Sparse Distributed Memory*. The MIT Press. ISBN 978-0-262-11132-4.

[11] Kohonen, T.(1995). Self-organizing maps. Berlin, Heidelberg: Springer.

[12] Schütze, H.(1992). Dimensions of meaning. In Proceedings of the 1992 ACM/IEEE Conference on Supercomputing, Supercomputing'92 (pp. 787–796). IEEE Computer Society Press

[13] Sahlgren, Magnus. "The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces." (PDF). SICS. Stockholm University.

[14] Webber, Francisco. (2015). Semantic Folding Theory and its Application in Semantic Fingerprinting. arXiv. Available at: http://arxiv.org/abs/1511.08855v2

[15] Tom M. Mitchell, et al. Science 320, 1191 (2008); Predicting Human Brain Activity Associated with the Meanings of Nouns, DOI: 10.1126/science.1152876

[16] Alexander G. Huth, Wendy A. de Heer, Thomas L. Griffiths, Frédéric E. Theunissen & Jack L. Gallant, Nature 532, 453–458 (28 April 2016); Natural speech reveals the semantic maps that tile human cerebral cortex, doi:10.1038/nature17637

[17] Ibriyamova, Feriha and Kogan, Samuel and Salganik-Shoshan, Galla and Stolin, David, Using Semantic Fingerprinting in Finance (March 28, 2016). Available at SSRN: http://ssrn.com/abstract=2755585 or http://dx.doi.org/10.2139/ssrn.2755585

[18] Nupic Community: http://numenta.org

[19] K.A.C. Martin, S. Schröder, Functional heterogeneity in neighboring neurons of cat primary visual cortex in response to both artificial and natural stimuli., J. Neurosci. 33 (2013) 7325–44. http://www.ncbi.nlm.nih.gov/pubmed/23616540 (accessed April 22, 2015).

[20] A. Losonczy, J.K. Makara, J.C. Magee, Compartmentalized dendritic plasticity and input feature storage in neurons., Nature. 452 (2008) 436–41. doi:10.1038/nature06725.

[21] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith, Sparse Overcomplete Word Vector Representations, ACL 2015, http://www.manaalfaruqui.com/papers/acl15-overcomplete.pdf

[22] Jun'ichi Kazama and Jun'ichi Tsujii, Evaluation and Extension of Maximum Entropy Models with Inequality Constraints, Proceedings of the 2003 Empirical Methods in Natural Language Processing (EMNLP 2003), pp. 137-144. July, 2003, Sapporo, Japan; http://www.jaist.ac.jp/~kazama/papers/kazama_emnlp03r.pdf

[23] Joshua Goodman. 2004. Exponential priors for maximum entropy models. In Proc. of NAACL, http://www.aclweb.org/anthology/N04-1039

[24] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2008; Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9(3):432– 441, http://biostatistics.oxfordjournals.org/content/9/3/432.full.pdf+html

[25] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011; Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proc. of ICML, http://www.icml-2011.org/papers/342_icmlpaper.pdf

[26] Dani Yogatama and Noah A Smith. 2014. Linguistic structured sparsity in text categorization. In Proc. of ACL, http://www.aclweb.org/anthology/P14-1074

[27] Brian Murphy, Partha Talukdar, and Tom Mitchell, 2012; Learning effective and interpretable semantic models using non-negative sparse embedding, In Proc. of COLING, http://talukdar.net/papers/nnse_coling12.pdf

[28] Alona Fyshe, Leila Wehbe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. 2015. A compositional and interpretable semantic space. In Proc. of NAACL. http://talukdar.net/papers/naacl15_comp_nnse.pdf

[29] Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Revisiting embedding features for simple semi-supervised learning. In Proc. of EMNLP. http://ir.hit.edu.cn/~car/papers/emnlp14guo.pdf

[30] Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, Chris Dyer, Problems With Evaluation of Word Embeddings Using Word Similarity Tasks, RepEval 2016, https://arxiv.org/pdf/1605.02276.pdf

[31] Felix Hill, Roi Reichart, Anna Korhonen; SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. Computational Linguistics 41(4): 665-695 (2015) http://arxiv.org/pdf/1408.3456v1.pdf

[32] M. Marelli , S. Menini , M. Baroni , L. Bentivogli , R. Bernardi , R. Zamparelli, A SICK cure for the evaluation of compositional distributional semantic models (2014), Fondazione Bruno Kessler

