Phrase Detection and the Associative Memory Neural Network

Richard C. Murphy
Computer Science and Engineering Department
University of Notre Dame
rcm@cse.nd.edu

Abstract— **This paper describes the use of a novel associative memory neural network architecture to perform unsupervised phrase detection in a large, unstructured, English text corpus. To significantly increase the difficulty associated with processing the text corpus, the network is exposed to over 270 thousand web pages from the *.edu* domain with no textual substitution or alteration (for spelling, grammar, etc.). The corpus, consisting of 150M words, is represented as a string of sparse tokens and phrase detection is performed through the use of the unique information theoretic quantity of *mutual significance*.**

I. Introduction and Motivation

This work describes the construction of a novel phrase detector capable of automatically finding phrases of two to five words in an unstructured English text corpus. The detector learns solely by examining English text and requires no supervision. In this case, the corpus consists of over 270 thousand web pages sampled from the *.edu* domain and consists of 150 million words. Unlike other systems, this phrase detector is robust for very large real-world corpuses. The selection of web pages was designed to increase the difficulty of the problem given to the network.

After training, the detector can be exposed to an arbitrary string of English text and break it up into its component phrases. This is the first step in constructing a higher-level set of networks capable of abstracting the phrases into unitary tokens, and identifying the relationship between phrases (meaning, synonymy, etc.). Critically, this system runs using relatively modest computational resources.

The entire phrase detector uses only one form of information-theoretic knowledge, the *mutual significance* proposed by Robert Hecht-Nielsen. This system uses five myopic knowledge bases to identify phrases within the text corpus. The simplicity of the knowledge acquisition, combined with a straight forward method for combining knowledge from multiple knowledge bases is the basis for this system.

The rest of the paper is organized as follows: Section II gives a brief overview of the background literature; Section III describes the properties of the text corpus and associated lexicon; Section IV provides an overview of the associative memory neural network architecture used in this system; Section V details the experiment; Section VI summarizes the results; and, Section VII concludes with a summary of the impact of this work and the direction of future research.

II. Background

There are a number of proposed methods for detecting phrases within an English text corpus. Linguists have proposed rule-based methods for determining Noun and Verb phrases. Numerous methods for text summary such as [8], [4] work based on human annotation of noun and verb phrases within a document. Other systems rely on linguistic rule databases to find phrases [10], [6]. These systems are often brittle, and often require well formed text. Neural network approaches can obviously match patterns or provide categories, however a self-organizing phrase detector has yet to be specified.

III. The Text Corpus

The chosen text corpus is a set of over 270,000 web pages sampled from the *.edu* domain. The domain, while broad, was chosen to allow the network to train on relatively long documents. The use of web pages, as opposed to books or other media, is intended to expose the network to actual text with all the flaws (misspellings, grammar errors, etc.) associated with less formal writing. Even given these difficulties, the system understands the significant terms.

The corpus consists of the following:

- over 270,000 web pages,
- Approximately 1 GB of encoded data (with words replaced by pointers to a lexicon);
- a lexicon consisting of over 2.5M unique words;
- a total of 150,000,000 instances of those words to be read by the network.

Many of the words in the complete lexicon are instantiated very few times (many only once). To facilitate faster processing, the lexicon was limited to a set of the 5,000 most frequently occurring words, which covers 82.6% of the corpus.

Figure 1 shows the coverage of a lexicon of a given size. The choice to limit the size of the lexicon to 5,000 words provides for significant coverage while simultaneously allowing the simulation to occupy a relatively small footprint (running on a 32-bit uniprocessor with 1 GB of physical memory, in approximately 45 minutes). Doubling the size of the lexicon improves the coverage by less than 8%.

All punctuation in the corpus is represented by denoting a single punctuation token, and is interpreted as breaking up phrases. A more complex understanding of grammar may be necessary with significantly larger phrase construction. HTML tags are treated the same as punctuation.

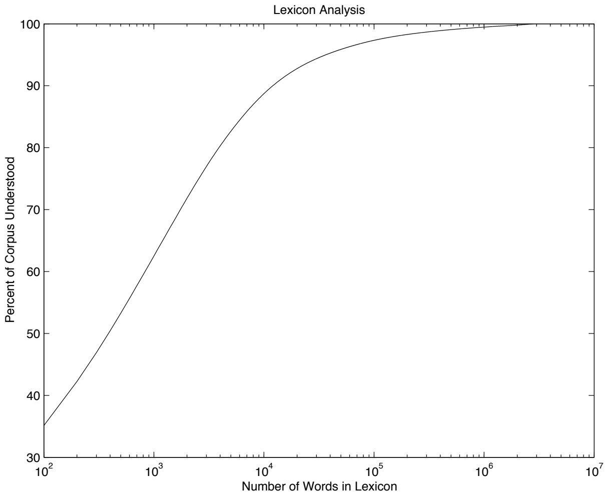

Fig. 1. Coverage of the Lexicon

IV. The Architecture

The system is constructed using the basic cortronic architecture proposed by Hecht-Nielsen. [3] Inspired by the associative memories proposed (independently) by Steinbuch and Willshaw [9], [11], Hecht-Nielsen's architecture significantly extends the simple associative model by providing differing representations for tokens and a *mutual significance* weighting for the relationship between two tokens.

The neural network architecture operates symbolically, using *tokens* that consist of a set of m neurons chosen to represent information (in this case words) uniformly at random from a neural region of n neurons. While the choice of language for this particular experiment eases the token-based representation of information, this type of symbolic processing extends to any type of information processing operation that can be quantized, and is inspired by the information processing that may be employed by the thalamacortex.

The system is designed to examine up to five words simultaneously and attempt to construct phrases. George Miller proposed that humans examine between 5 and 9 words when reading[7]. These words are given to the system by expressing them on a *neural region*, consisting of n neurons. Each token (representing a word) consists of a set of m neurons chosen uniformly at random. The neural regions are very sparse, and neurons which are never expressed are not stored. Numerous n and m parameters have been used, with the same result. Small experiments used $n = 10,000$ and $m = 25$ to represent words, however that proved computationally challenging (for commodity hardware). Consequently, the parameters $n = 1,000,000,000$ and $m = 1$ were substituted to eliminate redundant mutual significance storage and simplify the feature attractor component of the network (rather than a more brain-like neural representation). See Section IV-B The choice of these parameters is more for experimental convenience than statistics. Maintaining the (extreme) sparseness is all that is required for the system to function correctly.

It should be noted that analysis of token versus neuron representations in more complex systems is an open research question. Trade-offs such as fault tolerance, information representation capacity, etc., are well beyond the scope of this work.

A. Mutual Significance Evaluation

The fundamental operation of given architectures is mutual significance evaluation. Two tokens (i and j) are understood in terms of the defined, information theoretic quantity of mutual significance:

$$S(i,j) = \frac{p(i,j)}{p(i)p(j)}$$

That is, significance is simply the ratio of the joint probability and the *a priori* probabilities. Mutual significance is the number of times chances two tokens appear together in the regions of interest. That is, it is known from statistics that $p(i,j) = p(i)p(j)$ when tokens i and j occur independently. Thus a mutual significance ≤ 1.0 occurs when two tokens are independent.

This network resembles a number of other critical quantities. The mutual information given by the two tokens is $log_2 S(i,j)$. There are obvious connections with *relative entropy* or *Kullback-Leibler divergence* [2], [1].

The Mutual Significance is the only knowledge learned by the network, and is contained in a series of *fascicles* which link two regions in a pairwise fashion. All these fascicles can be used when performing information processing. In fact, phrases in this architecture are represented as a set of tokens containing high mutual significance between **each region** in the network.

The significance computation by the network is an approximation of the function given above, determined by counting joint (and independent) occurrences of tokens as exposed to each mutual significance *fascicle*. Thus, values less than one can represent non-random co-occurrences.

The approximation of mutual significance by counting the co-occurrences is given by the following formula:

$$S(i,j) \approx \frac{\frac{C(i,j)}{L}}{\frac{C(i)}{L}\frac{C(j)}{L}}$$

Where $C(i,j)$ represents the co-occurrence of tokens i and j, $C(i)$ and $C(j)$ the total occurrences of i and j respectively, and L the total number of learning events processed by the mutual significance evaluator.

There are a number of critical properties of the mutual significance:

- The co-occurrence approximation will converge to the mutual significance once the network has sampled enough data.
- The mutual significance has limited dynamic range. Given that it represents the number of times chance that two tokens occur together. Thus, even in practice, the dynamic range is about 0-250, and can be further restricted.
- The gradation between different levels of mutual significance is relatively coarse-grain. When the mutual significance is used to compare two tokens, very close values represent equally valid answers.

Given enough data, exposure to new information will leave the mutual significance unchanged. Consequently, the network can be continuously learning. Furthermore, given the properties of the mutual significance stated above, after training the network's significance values can be stored as a small, integer value rather than requiring floating point computation. (It is further reasonable that neurons could implement a similar quantity.)

B. Feature Attractor

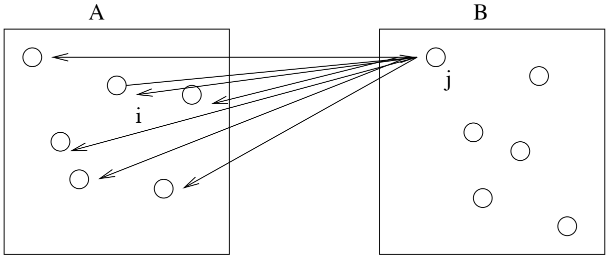

Fig. 2. A Simple Feature Attractor

The feature attractor learns only the canonical representation of all tokens in the lexicon. The feature attractor learns by expressing a token T_k onto region A and another token T_l onto region B (in the case of words, T_k, and T_l can, in fact, be the same token). The network then learns a set of neural connections between neurons in T_k (ie, i) and neurons in T_l (ie, j). The connection from A to B reinforces every connection between j and all neurons expressed on A. That is, there is a connection between all neurons expressed on A and all neurons expressed on B, and visa versa. Figure 2 shows the relationship between two neurons, i expressed on region A and j, expressed on region B.

At a later time, neurons can be partially expressed on A, and by traversing the connections from A to B and back (B to A), a canonical token can be reconstructed by selecting the m most activated neurons. This forces any expressed token (even a partial or damaged one) to return to the closes actual token. Tokens expressed using mutual significance weights (or a combination of many such tokens being expressed) leads to the closest token being selected in all but cases of sever degradation. This is similar to the adaptive resonance theory and the work of Kohonen [5].

The chosen parameters of $n = 1B$ and $m = 1$ serve to simply the feature attractor. Although the simulation allows for a parameterized feature attractor, the parameters selected for the purpose of the experiment greatly simplify the feature attractor operation (a single neuron is activated in each region). This reduces both the time and space complexity of the simulation. Because the input tokens are pristine (that is, they are words, rather than visual or other potentially noisy input), the chosen parameters are feasible.

V. Experimental Architecture

This experiment consists of two parts: first, phrases are extracted from the corpus by learning the mutual significance between five regions representing a window into the text. Secondly, after reading the corpus, phrases are extracted using the mutual significance gathered from the first pass.

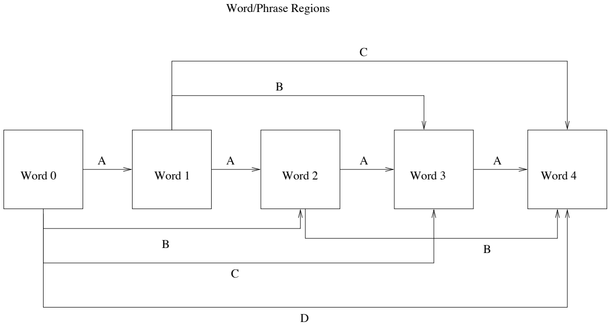

Fig. 3. The Experimental Architecture

Figure 3 shows the experimental architecture. Four knowledge bases are used during phrase recognition: A, B, C, and D in the figure. The A fascicle learn pre-word, post-word associations and are applied pairwise between words (0, 1), (1, 2), (2, 3), and (3, 4). The B fascicle learns pre/post-post-word associations and is applied between word regions (0, 2), (1, 3), and (2, 4). Similarly, the C fascicle, learns the associations between one word, and the word three words past it, and is applied between regions (0, 3), and (1, 4). Finally, the D fascicle learns the association between the first and last word in a phrase. It should be noted that a null or blank token is a valid part of a phrase (that is, three word phrases have two blank tokens at the end if no significant words appear there).

Each of the word regions in Figure 3 contains two parts: a neural region, upon which tokens can be expressed, and its associated feature attractor that chooses the most strongly expressed token on the region. The feature attractor component is not depicted in the picture.

A. Feature Attractor Training

After choosing the appropriate word lexicon a single word feature attractor is constructed to be shared among all neural regions. The structure is relatively large, and the sharing reduces the space complexity of the simulation.

Each word in the lexicon has two random token representations (which can be the same to reduce lexicon storage requirements). The shared feature attractor network is used in each step to snap incomplete or partial tokens into a pristine token within the lexicon.

B. Mutual Significance Evaluator Initialization

The experiment begins by marching all words in the corpus through the five word window represented by word regions 0 through 4. Any pairwise mutual significance association (via the A-D fascicles) between two non-blank words is learned. This represents the initial knowledge base used throughout the experiment. These fascicles represent relatively simple, limited knowledge, but when combined prove to yield meaningful higher-level information.

Table I summarizes the total number of learning events for each phrase detection fascicle in the architecture.

Fascicle	Number of Learning Events
A	98,706,035
B	77,429,644
C	62,078,406
D	50,203,717

TABLE I

FASCICLE LEARNING EVENTS

C. Phrase Detection

After the positional associations are understood, they are used to extract phrases from the corpus. This step is the most computationally intensive. A phrase is identified when **ALL** pairwise mutual significance are greater than a given threshold, for the purposes of this experiment 1.0. Theoretically a significance greater than 1.0 represents non-random co-occurrences. For example, when presented with the window: `university of iowa and the`, the following mutual significance evaluations are made:

	of	iowa	and	the
university	10.45	76.42	0.743	0.594
of		1.532	n/a	n/a

TABLE II

EXAMPLE MUTUAL SIGNIFICANCE EVALUATION OF A PHRASE

Table II shows the full mutual significance expansion of the phrase using all knowledge fascicles. The extracted phrase is `university of iowa`. On the first pass, “and” and “the” are eliminated from the phrase due to their low mutual significance with “university” as expressed through fascicles C and D respectively. The remaining tokens expressed in word regions $0, 1,$ and 2 are used to form a hierarchical token whose mutual significance is learned by repeated exposure throughout the corpus.

Other identified phrases beginning with `university of` are:

- university of california
- university of southern california
- university of notre dame
- university of chicago
- university of pennsylvania
- university of pennsylvania library

Naturally some phrases are not as clean because the network's exposure to the terms, even in a very large corpus, is small. For example, the network, when exposed to `university of texas at austin` identified `university`, `university of university of texas`, and `university of texas at` as phrases, but failed to pick up the word `austin` because $S(of, austin) = 0.562$ through the use of fascicle C. The occurrence in the corpus of the full term is very low.

Some other example phrases include:

- department of art
- department of chemistry
- department of germanic and slavic
- department of computer science
- department of justice
- department of state
- department of religion and philosophy
- department of mechanical engineering
- department of internal medicine

Naturally the addition of a sixth neural region would have undoubtedly given `department of germanic and slavic languages`. (The “university of” and “department of” samplings were chosen simply to make the search for examples tractable. All examples are taken from the raw early learning output of the network.

VI. RESULTS

Phrase Length	Number of Unique Phrases Detected
2	102,104
3	59,364
4	20,037
5	14,268

TABLE III

UNIQUE PHRASES DETECTED

After the phrase detector was constructed, it was exposed to the original text corpus to determine how many unique phrases were detected within the corpus. These results are summarized in Table III. Aside from the fact that the phrases emitted by the network are rational to human beings, the relatively large number of unique phrases discovered demonstrates the success of the system.

Although the numbers may appear relatively small, it should be noted that the phrase “university of” appears 1,305 times in the text corpus. Thus, 102,104 unique two-word phrases is, in fact, a relatively large number. By the same token, the system is also capable of detecting the phrase “real time”, which appears only 10 times in the corpus. In total, 372,787 phrase instances exist within the 150M word corpus.

Given the size of the corpus, the approximations of the mutual significance values have successfully converged. Consequently, exposing the network to new data produces phrase detections in nearly identical ratios.

VII. CONCLUSIONS AND FUTURE WORK

This work described the implementation of a phrase detection system using Hecht-Nielsen's proposed model of the thalamacortex. The system automatically extracted 195,773 unique phrases from a 150 million word text corpus. The system's learning is unsupervised, and requires only exposure to the text corpus itself. The computation and memory requirements of the system are relatively modest.

The future work in this area consists of two critical parts: first, increasing the size of the experiment; and second, increasing the functionality. In terms of increasing the experiment's size, both the size of the text corpus and the number of neural regions will be expanded. In particular, initial experimentation shows that detecting 9 or 10 word phrases using the same architecture is feasible. The use of a larger, and perhaps more well structured corpus should enhance the results.

In terms of increasing the functionality of the system, there are again two primary components: first, the same structure is capable of determining word synonymy; second, the addition of Hecht-Nielsen's proposed *hierarchical abstractor* neural network can be used to create unitary token representations for phrases. However, both of these enhancements are contingent upon increasing the size of the experimental system.

Additionally, it should be noted that the same architecture can be used to detect word synonymy. The same knowledge fascicles can be used to project word synonyms into a given region and the feature attractor can be used to select (in the correct order) the best answers.

VIII. Acknowledgments

I am deeply grateful to Robert Hecht-Nielsen and his ECE270[a-c] course sequence at the University of California, San Diego. I would also like to thanks Peter Kogge, and the University of Notre Dame's department of Computer Science and Engineering.

References

[1] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[2] Simon Haykin. *Neural Networks: A Comprehensive Foundation*.

[3] Robert Hecht-Nielsen. A Theory of the Cerebral Cortex. *Proceedings of the 1998 International Conference on Neural Information Processing (ICONIP98)*, 1998.

[4] E. Hovy and C. Lin. Automated text summarization in summarist. In *ACL Workshop on Intelligent Scalable Text Summarization*, 1997.

[5] T. Kohonen. *Self-Organizing Maps, 2ed*. Springer-Verlag, 1996.

[6] David A. McAllester and Robert Givan. Natural language syntax and first-order inference. *Artificial Intelligence*, 56(1):1–20, 1992.

[7] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

[8] Katashi Nagao and KoitiHasida. Automatic text summarization based on the global document annotation. In *COLING-ACL*, pages 917–921, 1998.

[9] Karl Steinbuch. *Automat und Mensch, 2ed*. Springer-Verlag, 1963.

[10] Frieder Stolzenburg, Stephan Höhne, Ulrich Koch, and Martin Volk. Constraint logic programming for computational linguistics. *Lecture Notes in Computer Science*, 1328, 1997.

[11] David Willshaw, O. Buneman, and H. Longuet-Higgens. Non-Holographic Associative Memory. *Nature, 222, 960-962*, 1969.