Class Three Neurosemantics in Freeform Data Extraction

David C. Taylor
Technical Director
Neurogy Corporation
Provo, UT
dct@crystalcanyon.com

Abstract

This research provides a novel method for associating and classifying segmented text tokens in a graphical free form text field. The goal of the combined algorithms is to closely model human data extraction processes, thereby facilitating rapid training over specific application domains. The primary method is to build an acyclic hierarchical neural model of a document, where semantic content increases with altitude in the neural tree.

There are three primary tasks in the construction of a neural model of a scanned document. First, each blob on the document is classified as either being a graphical object, a line, or a letter. Second, a horizontal blob aggregator groups letters or pieces of letters into single lines of text and passes the result to an OCR engine. Finally, the output of the OCR engine is tokenized into strict alphanumeric blocks, separated by spaces and non-alphanumeric characters. Block tokenization is also applied, where contiguous areas of text containing one or more tokens with similar alignment are grouped together. Blocks are divided into segments, where each segment consists of one or more contiguous (reading order inside the block) tokens of identical type. At this point, tokens are either of type word (alphabetic characters only) or value (contains at least one numeric character). This segment set represents an initial hypothesis over the block, which can be modified at a higher semantic level.

Three types of neurons are used to build a neural tree. The leaves are Value Neurons, each of which measure the strength of a single semantic hypothesis for a single token (word) on a page, independent of all other tokens. Some interior nodes are SuperValue neurons, which combine adjacent Value Neurons to form compound Value Neurons. Higher level interior nodes are Attribute Neurons, which combine position-related Value and SuperValue Neurons to form relationships, or semantic hypotheses. Attribute Neurons can also use other Attribute Neurons to create increasing semantic complexity.

A three-pass algorithm is used to form the initial neural model of the document, First, the tokens are used to activate all available Value Neurons at all relevant locations. Once all the tokens have activated the Value neurons, the block lists are used to activate SuperValue Neurons within each block. Finally, two-dimensional recursion is used to activate all relevant layers of Attribute Neurons. Certain Attribute Neurons are user-designated as Output Neurons, and once the recursion process is complete, all non-inhibited Output Neurons dump their data with appropriate semantic relationships to a file or a process pipeline.

A positional assertion grammar (PAG) is built into the Attribute Neurons. This PAG allows the neurons to form relationships based on relative positions on a printed page, and also allows relationship strength and inverse Value Neuron activation to be affected by graphical elements on the page such as dividing lines, shaded areas, tables, and boxes. The PAG is sufficiently flexible to force precise matches in order and content, or to allow certain elements to be optional or arranged in a vertical, horizontal, or multi-line format. The elements of the PAG are simple, consisting of the relationships {Adjacent, GraphicalCell, Locale, Left-Right, Up-Down, Block, Column, GraphicalLineSeparator, SpaceSeparator, NoiseSeparator, Order}. Attribute relationships that span multiple blocks allow the relationship weight to be affected by the block separators. Relationship penalties can be set in each neuron for most of these elements, and the rest are implicit in the structure of the algorithm.

Each neuron, when properly constructed, should encapsulate as much semantic behavior as possible without requiring a discriminator function. To this end, all SuperValue and Attribute neurons are allowed to take input both from other neurons in their "official" trees and from intervening noise (unclassified tokens and neurons not in the activation tree). This allows concepts such as "InvoiceTotal" to be excited by the word "Total" or the compound word "Somma de la Factura", but inhibited by the word "before" or the set of all time/space prepositions without need for an exhaustive list in the neuron itself, for example. This greatly reduces the number of neurons needed to represent a document, and allows concept representations to focus on the most important distinguishing features.

This multi-layer model interacts well with humans in a training situation. It is sufficiently similar to the way humans perceive information hierarchies that a human can directly edit the neurons if needed. It is also relatively easy to propagate corrections down from incorrect outputs and have them "stick" at the appropriate place in the tree, since every correction has an associated spatial component as well as a semantic component and most corrections involve an inappropriate attribute-value relationship. The main weakness of the algorithm in its current state is its inability to spontaneously create new neurons to resolve a situation where discrimination is insufficient. Research is ongoing in that area, however, the existing algorithm is sufficiently strong to power commercial applications.

Current applications include free-form invoice processing and extraction of data from genealogical documents such as World War 1 draft cards. For invoice processing, most clients require the extraction of the fields invoice total, invoice date, invoice number, sender, terms, and tax. Some require the extraction of line-items from the invoice. This problem becomes significant in invoices with more than one column of items, or where each entry consists of multiple rows of differing type. Properly configured column neurons are capable of detecting weak alignment and Value Neuron patterns in tabular data, and a special table module can utilize those patterns to impose a top-down classification of every token in the table once an area containing a table is identified.

Ongoing research focuses on ways to propagate discovered semantic relationships (such as the “table”) back down through the layers of the document model, even down to the graphical segmentation and OCR layer, to correct errors in lower levels and “oscillate” to a complete document solution.