MatchPlus: A CONTEXT VECTOR SYSTEM FOR DOCUMENT RETRIEVAL

Stephen I. Gallant, Principal Investigator
William R. Caid, Project Manager

HNC, Inc.
5501 Oberlin Drive
San Diego, CA 92121

PROJECT GOALS

There are two primary goals for *MatchPlus*. First we want to incorporate into the system a notion of *similarity of use*. For example, if a query deals with 'autos' we want to be able to recognize as relevant a document with many mentions of 'cars'.

Second, we want to apply machine learning algorithms to improve both ad-hoc retrieval and routing performance. Several different algorithms come into play here:

- a "bootstrap" algorithm develops context vector representations for stems so that similar stems have similar vector representations
- neural network algorithms produce routing queries from initial queries and lists of relevant and non-relevant documents
- clustering algorithms help generate a word-sense disambiguation subsystem (being implemented)
- neural network algorithms interactively improve ad-hoc user queries (being implemented)
- clustering algorithms can also speed retrieval algorithms using a new "cluster tree" pruning algorithm (planned)

A *context vector representation* is central to all *MatchPlus* system capabilities. Every word (or stem), document (part), and query is represented by a fixed length vector with about 300 real-valued entries. For any two of these items, we can easily compute a similarity measure by taking a dot product of their respective vectors. This gives a build-in, generalized thesaurus capability to the system.

RECENT RESULTS

- We have built a system for 800,000 documents (2 GB of text). This system takes Tipster topics, automatically generates queries, and performs retrievals.
- Hand-entered queries may be given using a simple syntax. Terms, paragraphs, and documents can comprise a query (all optionally weighted), along with an (optional) Boolean filter. Documents are always returned in order by estimated likelihood of relevance.
- Documents may be "highlighted" to show hotspots, or areas of maximum correspondence with the query.
- We have implemented routing using neural network learning algorithms. This resulted in a 20–30% improvement compared with the automated ad-hoc system.
- Lists of stems closest to a given stem provide useful and interesting insight into the system's vector representations.

PLANS FOR THE COMING YEAR

We have been running many bootstrap learning experiments, and some variations have resulted in significant improvements to performance. We expect that this improvement will carry over to all aspects of the system, including routing.

Currently we are implementing word sense disambiguation. We hope that this will give performance improvements, possibly even eliminating the need for phrase processing. This module should also be able to serve as a stand-alone package, providing help for machine translation and speech understanding systems.

We plan to apply learning algorithms for automated interactive query improvement in a manner similar to our approach with routing. It seems likely that this will give a significant boost to ad-hoc query performance.

Finally, we are performing additional learning experiments to improve routing.