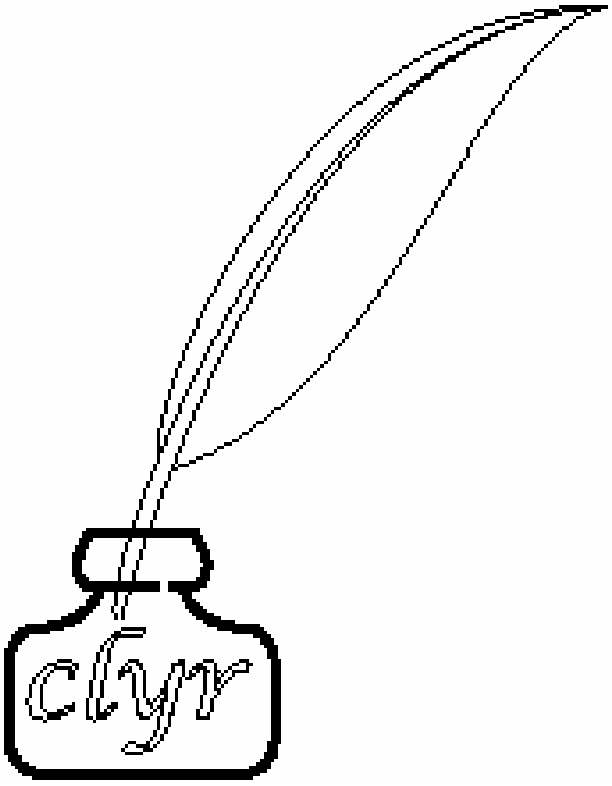
Clyr's Technology Distinction

(Document W004)



Clyr's Technology Distinction

Table of Contents

1 Introduction

The **problem** is simple. Natural language processing (*NLP*) does a poor job of pretending to use human language, and must do more than just transfer information packets between Point A and Point B; the use of *language* as the specific medium of exchange leads Point A, the human user, to expect that Point B, a machine, will not only *respond* to a person's query, but will answer with *understanding*. While this response does not have to fool the user into thinking that the machine is human, the answer must still be *relevant*; that is to say, it must address the query, but it must not be cluttered up with a load of unrelated garbage. Until now, NLP applications have failed to meet either of these criteria, because they do not foster a sense of a *meeting of the minds*. Such meetings will be exceedingly few and far between as long as the mind of a machine is, at best, a necessarily incomplete replica of a human's *natural* linguistic model of the world.

The **solution** is difficult. If a machine were to learn language in a more human fashion, its mind would be more similar to that of a human, and understanding would result more often. Ideally, a sensate automaton would symbolize sets of stimuli gathered from its environment, just as a human does. The direct sensations drawn from exposure to flowers (e.g. scents and colors) would be associated with the indirect sensations drawn from the word 'flower' (e.g. sounds and signs), creating the equivalent of a mental image in the machine. This process of symbolization could use Clyr's ***ymage lattice*** to grow language on naturally, analogous to the mental image framework drawn upon by humans as they categorize stimuli.

This lattice represents an **innovation** tantamount to a paradigm shift. Rather than trying to build an exhaustive model of the world and handing it over to a machine on a silver platter (which is a pragmatic impossibility), we are creating a seed from which the machine will grow language on its own. The natural language model learned by the machine will quickly outstrip the usefulness of artificial efforts to date. NLP applications relying upon such natural models will no longer simply shuffle arbitrary word shapes around, but will process machine-palatable representations of the *meaning* with which those shapes are associated.

The **significance** of this innovation is profound, in that machines will process meaning, and more closely portray a true understanding of human language. Even the earliest forms of this lattice will improve the performance of applications that rely on sophisticated NLP technologies. An ability to process meaning will allow data mining applications to find patterns in behavior that would otherwise have been hidden by arbitrary differences in word shapes. Machines operated by command-and-control systems will respond to the operator's intent with what feels like greater understanding. Search engines will display improved rates of recall and precision, fostering a stronger sense in the user that the machine *understands* what the user is looking for. In general, NLP applications will take a step closer to human reasoning.

There is a clear, immediate call for this kind of language-independent program in the marketplace, and the **benefits** of this project reach much farther into the future than this one initial product.

In the short term, the most visible benefits of this project will be those that derive from developing a natural language processing technology that defines the state of the art, namely: 1) improved retrieval and precision for concept-based and language-independent search engines; 2) increased accuracy in data mining and information categorization; 3) greater efficiency in command-and-control systems; and 4) a closer approach to human reasoning. While the commercialization of this technology will focus on search engines (e.g. internet and transcript) and data mining (e.g. threat identification and fraud detection), its development is particularly flexible, and can easily accommodate a shift to the prototyping and production of command-and-control systems for stationary or mobile operations centers, or universal translation devices, or any other product that relies on sophisticated natural language models.

In the long term, this technology will support advances allowing automata to draw upon a much broader range of language-based reasoning capabilities, which will help to unlock the full potential of cyborganic intelligence, making possible such elusive products as Cyborganic Agents, the Intelligent Home, and so on into the future.

We will now discuss our R&D program's background (§2), and then present our proposed solution to the problem of improving NLP understanding (§3).

2 Background

Clyr is developing an ymage lattice for initial implementation in a dual-language, concept-based, ymage search engine. In this section, therefore, we will discuss the state of the art specifically in regards to the performance of search technologies.

2.1 Search Engine Performance in General

To begin with, every search balances ***recall*** (i.e. finding all of the relevant information) against ***precision*** (i.e. avoiding the inclusion of irrelevant material), as measured relative to the information that the user wants to find. The relation between these parameters is illustrated in the following chart, which plots typical comparative results for four basic types of search engine (with our projected results in plotted in parentheses):

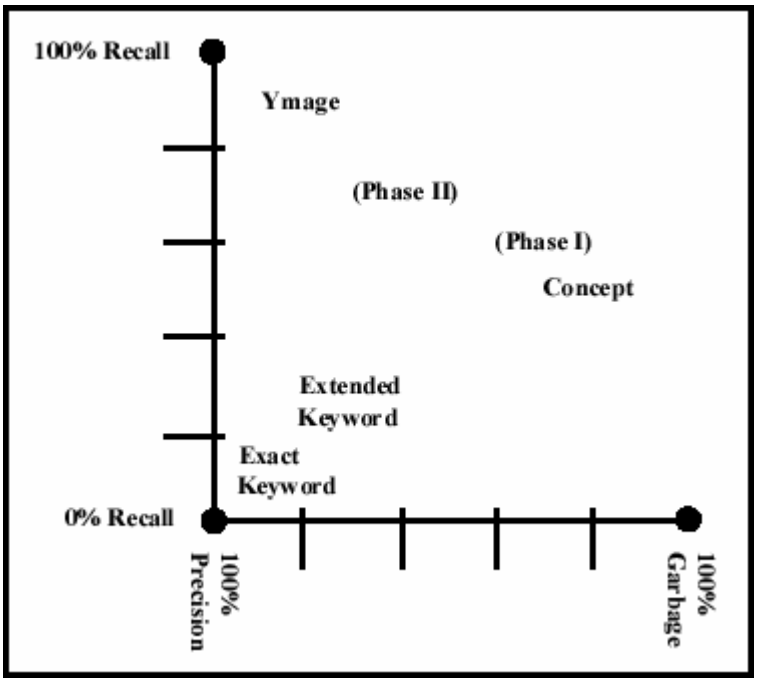


Figure 1 The State of the Art

An ***exact keyword*** search will target a very small, specific set of information, such as the word shape ‘pit’, but it will ignore shape variations such as ‘pits’ and ‘pitted’. This has the advantage of decreasing the garbage retrieved, but it decreases the likelihood of finding all of the relevant information hidden in the target text. This performance is

located in the lower left-hand corner of the chart. When such searches are extended to include word shape variations such as 'pits' and 'pitted' (i.e. ***extended keyword***), recall improves somewhat, but there is an increase in garbage as well. Keyword search engines of this sort already exist, and are evaluated in more detail below (§2.2).

In contrast, the broadest kind of ***concept*** search will create a very tolerant filter, such as when the word shape 'pit' is expanded to include concepts as distantly related as [HOLE], [STONE OF A FRUIT], and [INTERMISSION], just so long as these concepts are all associated with the word shape 'pit'. This type of search is less likely to leave relevant information hidden in the target text, but it is much more likely to pick up a lot of extra garbage. The user's answer will be buried in irrelevant material. This behavior is located in the upper right-hand corner of the chart. Products of this sort already exist, albeit in rudimentary form, as discussed below (§2.3).

At its finest, an ***ymage*** search represents an ideal, where a broad target is *governed* by the user's intent; for example, the shape 'pit' would be expanded to include those shapes associated *in context* with the concept [HOLE], but it would ignore those which refer to [STONE OF A FRUIT]. Very little, if any, relevant information would be left hidden, and the garbage would be minimal. Such an engine does not yet exist, but its performance would be located in the upper left-hand corner of the chart. We are currently developing Clyr's ymage lattice as the first step in creating such a product, as discussed more thoroughly below (§3).

2.2 Keyword Search

Claims about the unusually strong performance of keyword search engines are misleading because they leave the level of relevance unstated; for example, if a search is conducted on the keyword 'pit', and the relevance of the answer is ignored (i.e. the precision is allowed to be 0%), then the recall should *always* be 100%. However, as soon as the user wants to find a particular *meaning* of 'pit' (i.e. a specific concept associated with the keyword 'pit'), recall drops significantly, because information about [HOLE], for example, will be returned along with material associated with [STONE OF A FRUIT]. Claims of strong performance, then, should be interpreted in terms of *relevance*.

Contemporary performance measures appeal to four broad categories of relevance, as follows:

- *irrelevant* (i.e. the answer is highly unlikely to be relevant to the user);
- *technically relevant* (i.e. the answer might be interpreted as relevant to the user);
- *possibly relevant* (i.e. the answer is just as likely as not to be relevant to the user);
- *relevant* (i.e. the answer is highly likely to be relevant to the user)

Tests run on popular internet search engines tend to deliver results as follows, with measures based on the first twenty returned documents:

Recall (relevant material returned)	**Precision (irrelevant material ignored)**	
	Median	**Best**
Irrelevant (0% relevant)	95%	100%
Technically Relevant (25% relevant)	80%	90%
Possibly Relevant (50% relevant)	40%	50%
Relevant (100% relevant)	5%	10%

Table 1 Retrieval: Recall versus Precision

These results are representative of those rendered by diverse tests, and do not tend to be disputed in the industry. Relevance is assigned as a matter of evaluation by a user (i.e. as a matter of individual human preference), and not because the keyword search had any technology in place to dictate or objectively define the relevance of the results.

This is the sense in which it is claimed that exact keyword searches are 100% accurate; they can be said to display a recall rate of 100% if relevance is ignored. There are clearly a couple of provisos associated with these claims: 1) the answers are only 10% likely to be highly relevant; and 2) morphological variants of a word will be hidden (i.e. 'pit' will not hit 'pits', 'pitting', 'pitted', 'pitter', or 'pitless', neither will it hit eccentric forms such as spelling variations, acronyms, abbreviations, and so on); therefore, this hypothetical 100% recall rate is really only about 10% to begin with. So, as the user cares *less* about the relevance of the results, the keyword search is seen as performing *better*… and this is the state of the art for exact keyword searches.

Testing reveals that an extended keyword search will register more hits than an exact keyword search at least half of the time. In those cases where it does so, it will often register three or more times as many hits. In this sense, treating extended keyword search as the standard makes the recall rate for an exact keyword search quickly drop, in comparison, from the claimed 100% to only 33%. The performance of the extended keyword engine does not depend so much upon its pattern-matching algorithms as it does upon the lexical resources used to generate the keyword extensions; that is to say, the search engine is still an "exact" matcher, but the databases generate a long list of variations on the keyword for "exact" matching.

On top of all of this, keyword searches will still leave a significant amount of information hidden in the net. An extended keyword search on 'pit' will still miss any references to the concept [HOLE] where the text uses a synonym of 'pit', such as 'hole', 'shaft', 'chasm', 'well', 'abyss', 'crater', 'divot', 'mine', or 'pothole'. This drops the recall rate for extended keyword searches precipitously. Concept searches are designed to get around such obstacles, but they still have problems of their own.

2.3 Concept Search

Some NLP technologies claim to support a concept search, the purpose of which is to ameliorate some of the disadvantages inherent in relying upon word shape alone; however, just as the meaning of "accuracy" has been manipulated to the extent that claims of accuracy have lost their value, there are some contemporary definitions of "concept" that allow a concept search to be no different than a keyword search. So, what distinguishes a trivial concept search from a valuable one?

To begin with, creating a *rudimentary* form of a concept search engine is not really difficult, it is simply time consuming. There are publicly available lists of synonyms that allow for word shapes such as 'pit' and 'well' (as in [HOLE]) to be associated with a single arbitrary code. When this code is substituted for those words in both a query and a target text, then a search for the word 'pit' will retrieve all instances of that same substituted code in a text. The engine searches for instances of the "concept" [HOLE], no matter whether it is tied to the word 'pit' or 'well'. Relative to the keyword results given above, a concept search using simple synonymy would typically display

50% recall with 20% precision at 100% relevance. Broader forms of synonymy (e.g. associating ‘abyss’ and ‘divot’ with ‘pit’) will raise recall, but decimate precision. There are engines like this already, and they all suffer from the same set of problems.

To illustrate the first of these problems, let’s pose a sample case where the concept code ‘8899’ is associated with every word shape that can be used to mean [HOLE], such as ‘shaft’, ‘well’, and so on. This code will be substituted for these word shapes in both the query and the target text. Recall measures will go up compared to a keyword search on ‘pit’, because the search will also retrieve instances in the text where ‘shaft’ and ‘well’ are used in context to mean [HOLE]; however, the precision will drop when the search retrieves instances where ‘well’ is used as an adverb, or where ‘shaft’ refers to a pole. This problem arises from the fact that words can be ***polysemous***, that is to say, one shape attaches to many meanings.

In addition, when you substitute this sort of ***arbitrary*** code for the word forms, then you cannot appeal to any of the ***advanced cognitive relationships*** (*ACRs*) holding between concepts; for example, if you assign code ‘1234’ to the word ‘child’, and the code ‘9876’ to the word ‘family’, you lose out on membership relations, because there is nothing about the number 1234 that would give you any reason to suspect that a 1234 is a member of a 9876. You also lose out on several other ACRs between concepts, such as, but not limited to: *taxonyms* (oranges are types of citrus fruit); and *entailment* (dreaming implies sleeping).

Finally, an arbitrary code that is also ***synthetic*** (i.e. not analyzable) will only work with narrow forms of synonymy; for example, one code might be assigned to both ‘pit’ and ‘hole’, but two additional, different codes might be used for ‘abyss’ and for ‘divot’, because their meanings are more specific than the simple concept [HOLE]. ‘Pit’ would then be processed as a synonym of ‘hole’, but not of ‘abyss’ or of ‘divot’. ‘Pit’ and ‘abyss’ *could* be treated as synonyms by assigning them the same code, but then ‘abyss’ and ‘hole’ would become synonyms by association. Simply defining a set of synonyms by fiat is a waste of time, because some users will want ‘pit’ and ‘divot’ to be treated as synonyms in one search, but not in another.

3 Proposed Solution

Clyr is removing the aforementioned obstacles by developing an ***ymage lattice*** (§3.1). Where contemporary NLP models are arbitrary and synthetic, our lattice is iconic and analytic. The feasibility of this lattice is being evaluated in terms of the performance of an ***ymage search engine*** in which it is embedded (§3.2). Note that the terms used in this section have already been defined in more detail in our other white papers.

3.1 Cognitive Image Lattice

Data organization frameworks are defined by the properties of their contents. A Lego® warehouse needs to organize pieces according to their autonomous properties (i.e. by size, shape, and color, at the very least) and by their dependent ones (i.e. whether they belong to a particular kit, or are generic across kits). These same properties determine the shape of the framework and its nodes. Similarly, an ymage lattice is a data organization framework defined by the properties of the ***concept meaning definitions*** that it organizes.

Clyr technology derives definitions of a concept's meaning from diagrammatic representations of the cognitive images that the concept evokes; that is to say, meaning is a matter of hearing or seeing a word and bringing an associated image to mind. These cognitive image definition structures are the ***ymages*** described in our white papers.

There are several distinct advantages to using ymages. First, they are ***grounded*** in sensorimotor experience, which means that they are not circular like dictionary-style definitions. In a dictionary, words are defined in terms of more words. This grounding reflects the set of concepts that a human is born with (e.g. [PAIN], [PLEASURE]), and so not only breaks the pattern of circularity, but provides a better bootstrap for a machine learning language.

Second, like jigsaw puzzle pieces, ymages are ***iconic*** rather than arbitrary. The rules for combining them into higher-order concepts (i.e. phrases) are an integral part of their meaning. Clyr technology is not a matter of mechanical parsing, but rather it is a process of ***cognitive linguistic ymage refinement*** (hence *Clyr*). Clyr drives the ymage search engine described below.

Finally, ymages are not synthetic; in fact, they are not just analytic, but rather are ***polyanalytic*** (e.g. a flower ymage contains a petal ymage, and it is part of the ymage used

to depict a garden, and so on). Polyanalytic ymages allow for a more natural concept expansion than the simple synonymy generated by thesaurus listings. Reference to an ymage in the lattice allows for its expansion along several dimensions defined by its ACRs to other ymages, such as the following:

		Concept	Relationship	Expansion
Genetic	twin	[PUMP]	is very much like	[TICKER]
	synonym	[PUMP]	is like	[HEART]
	cousin	[PUMP]	is kind of like	[LIVER]
	hint	[PUMP]	is distantly like	[CHEST]
	antonym	[GOOD]	is unlike	[BAD]
Taxonym	hyponym	[ORANGE]	is a type of	[CITRUS FRUIT]
	hypernym	[CITRUS FRUIT]	has types	[ORANGE, LEMON…]
Meronym	part	[WHEEL]	is a part of	[CAR]
	substance	[BANANA]	is a substance of	[BANANA SPLIT]
	member	[CHILD]	is a member of	[FAMILY]
Holonym	part	[CAR]	has parts	[WHEEL, DOOR…]
	substance	[BANANA SPLIT]	has substances	[BANANA, ICE CREAM…]
	member	[FAMILY]	has members	[PARENT, CHILD…]
Entailment	equated	[LIVING]	implies	[AGING]
	contained	[DREAMING]	implies	[SLEEPING]
	backward	[UNLOCKING]	implies	[LOCKING]
	causation	[CREATING]	implies	[EXISTING]
Eccentrics	acronym	[DVD]	equals	[DIGITAL VERSATILE DISK]
	abbrev.	[ST.]	equals	[STREET] or [SAINT]

Table 2 Advanced Cognitive Relationships (ACRs)

If a query contains a concept (e.g. [ORANGE]), then a text is likely to be relevant if it contains the associated concept expansion (e.g. [CITRUS FRUIT]). There are also domain-specific uses for each of these relations, such as where [HATCH] is a nautical twin for [DOOR], or [105] is a criminal code for [SABOTAGE].

3.2 Ymage Search

The ymage lattice is the lexical resource drawn upon by our Clyr engine. Crucially, the Clyr engine extracts *meanings from forms* (i.e. forms such as text), allowing data mining efforts to find patterns in meanings, rather than in arbitrary word shapes. This engine can support a wide variety of NLP applications, but its output will initially feed a set of pattern-matching algorithms that compose our ymage alignment engine. Together, the refinement and alignment engines define our ymage search engine. We will be using measures of the performance of the ymage search engine to evaluate the lattice's effectiveness. Because this ymage search engine relies ultimately on the lattice, it will allow the user to take advantage of the ACRs mentioned above while conducting a search.

4 Conclusion

So why use an ymage search instead of a concept search?

A concept search using simple synonymy will improve recall, but will increase garbage as well (§2.3): at 100% relevance, a state-of-the-art concept search might display 50% recall with only 20% precision. Using broader synonymy in a concept search will raise recall again, but drop precision even further. In contrast, Clyr's ymage search will draw upon a suite of ACRs that are not available to contemporary concept searches, and Clyr technology will improve recall *without* penalizing precision. The feasibility of the lattice, then, will be measured by our ability to raise recall beyond that achieved by concept engines (to 65%), while ensuring that precision does not slip below 20% (at 100% relevance).

Doubling recall while maintaining precision would result in twice as many relevant hits, which is a 100% improvement. In fact, we intend to *improve* precision to 35%. In short, our initial development efforts are pushing the performance of our ymage search engine closer to the upper edge of Table 1, without straying to the right.

Our *Technology Introduction* white paper discusses the symbolic nature of language, and explains *why* Clyr technology can do this.