Available Article

Author's final: This draft is prior to submission for publication, and the subsequent edits in the published version. If quoting or citing, please refer to the proper citation of the published version below to check accuracy and pagination.

Cite as: Bergman, M. K. The Opportunity. in *A Knowledge Representation Practionary: Guidelines Based on Charles Sanders Peirce* (ed. Bergman, M. K.) 65–84 (Springer International Publishing, 2018). doi:10.1007/978-3-319-98092-8_4

Official site: https://link.springer.com/book/10.1007/978-3-319-98092-8

Full-text: http://www.mkbergman.com/publications/akrp/chapter-4.pdf

Abstract: The path to knowledge-based artificial intelligence directly coincides with a framework to aid data interoperability and responsive knowledge management. KBAI, data interoperability, and KM are the three main opportunities covered in this book. A gateway to these opportunities is to address the sources of semantic heterogeneities of information content. A knowledge graph provides the overall schema, and semantic technologies give us a basis to make logical inferences across the knowledge structure and to enable tie-ins to new information sources.

We support this graph structure with a platform of search, disambiguation, mapping, and transformation functions, all of which work together to help achieve data interoperability. KBAI is the use of large statistical or knowledge bases to inform feature selection for machine-based learning algorithms. We apply natural language processing to these knowledge bases informed by semantic technologies.

4

THE OPPORTUNITY

Charles S. Peirce, the intellectual founder of *pragmatism*, a uniquely American contribution to philosophy, advocated we obtain knowledge by balancing research effort with a likelihood of results. To do so, we should first consider all of the 'practical effects' posed by alternatives. We select what deserves more detailed attention using a mode of logical inference he called *abduction*. Wherever we have doubt, we should be open to and pursue the path of inquiry to unveil further potential 'practical effects,' enhancing our knowledge. In this way, we continually modify what we believe about the world, and therefore how we act within it.

Today, we have the ability and information to query nearly the entire storehouse of accumulated human knowledge. By combining general knowledge storehouses with representations of our organizations and domains, we have paths of inquiry leveraging computers and machine learning to test what we think we know, and to discover previously hidden anomalies or falsities to propel our knowledge further. As we have seen with earlier breakpoints in humanity's abilities to share and process information, this quest for new truth will bring significant financial benefit across the full spectrum of economic actors, from individuals and small groups to enterprises and governments.[1]

In this chapter, I discuss these opportunities under three broad tents. The first tent, more of a foundation, embraces general applications in *knowledge management* (KM). This broad tent may not be the motivating interest, but it does reside on the path to other capabilities, and it addresses important needs in their own right. The second tent, more of a process, are the approaches and applications that enable *data interoperability*. The techniques of data interoperability are essential for ingesting relevant information leading to knowledge and for unleashing the value of existing information assets across the organization. The third tent, more an expression of potential, is *knowledge-based artificial intelligence*. Via KBAI we can cost-effectively create labeled training sets (supervised learning) and training corpora (unsupervised) for machine learning to support a variety of tasks from entity and relation recognition and extraction to categorization, natural language understanding, sentiment analysis, and much more. Like the lizard eating its tail, we can also apply KBAI to our initial knowledge bases and knowledge graphs that drive these applications, producing

a virtuous cycle of knowledge expansions and better learning accuracy.

KM AND A SPECTRUM OF APPLICATIONS

Many of the problems of wasted information assets and lack of connections, some described in the prior chapter, and most with real economic costs, can be ascribed to a failure of *knowledge management.* KM is the practice of creating, sharing, finding, annotating, connecting, and extending information and knowledge for a given domain.[2] The practice includes applications and management platforms; shared workflows, vocabularies, and organizational schema; training and best practices; and roles for practitioners. The practice is managed and possibly encouraged by rewards and incentives. Software is an essential component of knowledge management, but woefully inadequate alone to accomplish it.

Some Premises

The nature of knowledge helps set some parameters for what a knowledge management system should encompass. First, knowledge is 'open' and needs an architecture and design that embraces this openness. This consideration has logical and epistemic importance that gets further treatment in *Chapter 9*. Second, knowledge is ultimately a community reality, since knowledge is what we believe and upon which we act. Because our means of communicating within the community is via symbols, we need methods for defining, clarifying, and reconciling the meanings of those symbols, such that we are effectively communicating within the community. This imperative means that we should look to semantic technologies as our representation and messaging frameworks; *Chapter 5* covers this topic. Moreover, third, we need to design our knowledge management systems to get maximum pragmatic leverage from what already exists and what we can support with such a system. We need to design our systems for knowledge *uses*, with management a contributing component to that.

Potential Applications

KM includes such applications as business intelligence, data warehousing, data integration and federation, enterprise information integration and management, competitive intelligence, workflow systems, knowledge representation, and so forth. Information management is a bit broader category and adds such functions as document management, data management, enterprise content management, enterprise or controlled vocabularies, systems analysis, information standards and information assets management to the functions of KM. Knowledge management also importantly includes pruning (deleting) dated, inaccurate, or otherwise wasteful information. An absolute essential for an effective KM system is bridging vocabulary, concept, and representation differences.

These are all important and legitimate knowledge management functions, but we often pursue them in isolation or under different databases, vocabularies, or concep-

tual approaches. In point, one could reasonably argue that much of the challenge that has faced KM has been a lack of coherence or a shared conceptual grounding to the efforts. The decades-long literature into KM supports such a view of fragmentation.

A broadly useful KM framework should support a minimum of four application areas:

- First, some form of governing conceptual and terminological schema is required by which to reference and ground disparate information sources. In KM systems based on semantic technologies, this schema takes the form of a *knowledge graph* (or ontology);
- Second, given that on average about 80% of an organization's information base resides in documents, *natural language processing* should be an integral part of the mix. NLP uses computers to extract meaningful information from natural language input or produce natural language output. NLP is one method for assigning structured data characterizations to text content; without NLP, all such assignments are manual, which does not scale;[3]
- Third, as part of these NLP capabilities, we need various extractors. *Entity recognition*, the means for identifying specific *entities* in text, is the first among equals here. Concept and relation extractors may supplement that. Extraction methods involve parsing and tokenization, and then generally the application of one or more information extraction techniques or algorithms;
- Fourth, *tagging* is a needed adjunct to extraction. The *tag* is a keyword or term we assign to a piece of information (*e.g.*, a picture, article, portion of text, or video clip). Tags describe the item and enable keyword-based classification of the information.* The resulting representation is a form of *semi-structured data*. Like extractors, we may use tags for entities, concepts, attributes, or relations. When a knowledge graph is employed, we recommend *ontology-based information extraction* (*OBIE*), which is the use of an ontology to inform this tagging process.

These are the essential functions required to 'ingest' new content and provide a shared vocabulary via the schema for placing content onto a common footing. This shared representation is the basis for a series of specific KM format conversions from multiple external sources, and in functions such as search, retrieval, analysis, and visualization. As we add multiple input sources to the system, we assign *metadata* by source (such as title, provenance, workflow dates, formats, and such) to the content, providing still additional means for searching, filtering, and aggregating the content.

If the KM system is also a precursor to more knowledge- and intelligence-oriented tasks, we advise including reasoners and mappers. *Reasoning* is one of many

* Tagged information is one of the main sources of *semi-structured data*; see *Chapter 5.*

logical tests using *inference* rules as commonly specified using an ontology language, and often a description logic. Many reasoners use first-order predicate logic to perform reasoning. Inference commonly proceeds by forward chaining or backward chaining (see *Chapter 8*). *Mapping* connects *objects* in two different sources to one another, using a specific *property* to define the relation. A linkage is a subset of possible mappings where the connections may be traced and followed. Mappings are the means by which we bring in multiple information sources leading to a federation of sources, so that we may use, analyze, or reason over all of them. It is the central task of *data federation*. Pairwise mappings result in a combinatorial explosion of connections as the number of sources increases. A hub-and-spoke design is the only practical architecture to overcome this problem since it scales linearly, with a reference set of concepts, such as KBpedia, providing the hub.

A Minimal Scaffolding

We could stop with this initial configuration and merely deploy the knowledge management system for generic KM tasks. This basis, the minimal scaffolding, is sufficient to address the lost opportunities and waste described in the prior chapter. However, we have our sights set higher than recovering lost opportunities.

The general development path this book recommends is to first address these lost opportunities, perhaps on a small or departmental basis (see 'pay as you benefit' in *Chapter 13*). As we gain confidence and climb the learning curve, it is then appropriate to bridge out to encompass more departments and to begin deploying machine learning to develop bespoke extractors and classifiers, tuned for the relevant nature of your growing knowledge base.

We introduce the role of KBpedia here as a lead-in to later chapters. KBpedia is an open source knowledge graph with maps to leading knowledge bases. *Parts III* and *IV* cover design and deployment topics in detail. *Appendix B* is a broad overview of KBpedia. *Appendix C* discusses the features available in KBpedia for machine learning.

DATA INTEROPERABILITY

Data integration is the bringing together of data from heterogeneous and often physically distributed data sources into a single, coherent view. Sometimes this is the result of searching across multiple sources, in which case it is called federated search. However, it is not limited to search. Data integration is a crucial concept in business intelligence and data warehousing and a driver behind master data management (MDM). Data integration first became a research emphasis within the biology and computer science communities in the 1980s.[4] [5] At that time, extreme diversity in physical hardware, operating systems, databases, software, and immature networking protocols hampered the sharing of data. Data *interoperability* extends beyond integration to add unified views for analysis and reasoning across its sources.

By its nature, data integration means that we combine data across two or more

datasets. Such integration brings to light the myriad aspects of semantic heterogeneities, precisely the kinds of issues why we use semantic technologies. However, resolving semantic differences, which we probe in detail in *Chapter 5*, cannot be fulfilled by semantic technologies alone. While semantics can address the basis of differences in meaning and context, resolution of those differences or deciding between differing interpretations (that is, ambiguity) also requires many of the tools of artificial intelligence or natural language processing (NLP). By decomposing this content into its various sources of semantic heterogeneities — as well as the work required to provide for such functions as search, disambiguation, mapping, and transformations — we can begin to understand how all of these components can work together to help achieve data interoperability.

The Data Federation Pyramid

It is easy to forget just how far data federation has progressed in the last four or five decades. Before the introduction of the IBM personal computer in 1981, the hardware landscape was diverse and fragmented. There were mainframes from weird 36-bit Data General systems to DEC PDP minicomputers to the PCs themselves. Even on PCs, there were multiple operating systems, and many then claimed that CP/M was ascendant, let alone the upstart MS-DOS or the gorilla threat of IBM's OS/2 (in development). Hardware differences were manifest, and operating systems were diverse; nothing worked with anything else.

'Data federation' at that time needed to first look at issues at the iron or silicon or OS level. Those problems were pretty daunting, though the clever folks behind Ethernet and Novell with PCs were about to show one route around the traffic jam. Client-server and all of the 'N-tier' networking speak soon followed. It was an era of progress, but still, one of costly and proprietary answers to get devices to talk to one another. That is where the Internet, specifically the Web protocols of HTTP and HTML and the Mozilla (then commercially Netscape) browser came in. Within five years (actually less) from 1994, the Internet took off like a rocket, doubling in size every 3-6 months.

In the early years of trying to find standards and conventions for representing semi-structured data (though not yet called that), the primary emphasis was on data representation and transfer protocols. In the financial realm, one standard dating from the late 1970s was electronic data interchange (EDI). In information and library science, the MARC communications format for sharing catalog metadata arose in the 1960s and remains well-used in many countries today. In science, there were tens of exchange formats proposed with varying degrees of acceptance. Notable examples are the abstract syntax notation (ASN.1), TeX (a typesetting system created by Donald Knuth and its variants such as LaTeX), hierarchical data format (HDF), CDF (common data format), and the like, as well as commercial formats such as PostScript, PDF (portable document format), and RTF (rich text format). One of these formats was the 'standard generalized markup language' (SGML), first published in 1986. SGML was flexible enough to represent either formatting or data exchange.

However, with its flexibility came complexity. Only when its two simpler progeny arose, namely HTML (HyperText Markup Language) for describing Web pages and, later, XML (eXtensible Markup Language) for data exchange, did variants of the SGML form emerge as widely used common standards. JSON has also now joined this group as a leading data representation. The Internet and its TCP/IP and Web HTTP protocols and XML standards, in particular, have been major contributors to overcoming respective physical and syntactical and data exchange heterogeneities.

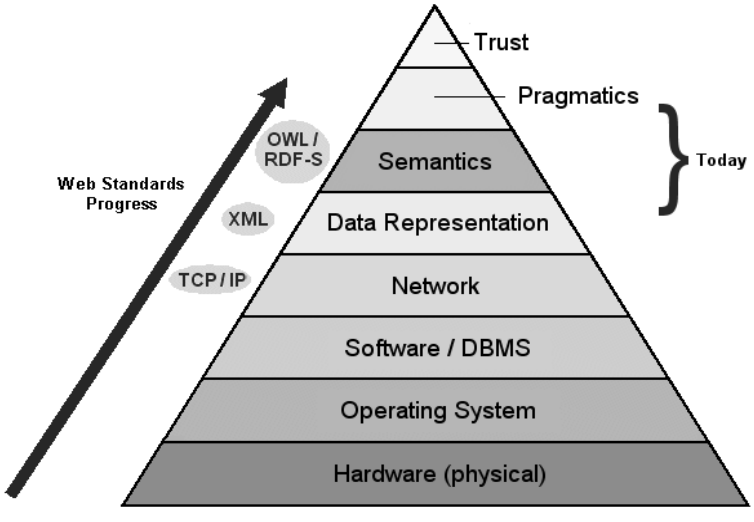

Figure 4-1: Climbing the Data Federation Pyramid

I illustrate this historical progression over the decades, from the bottom up, using the data federation pyramid in *Figure 4-1.* Current progress and adoption place us, today, with a stack that has boundaries at the data (and knowledge) representation, semantics, and pragmatics layers in the *Figure 4-1* pyramid. We show only a part of the progress in Web standards. The TCP/IP and HTTP protocols were essential to overcome the network bottlenecks; we show OWL and RDF due to their importance to our story and their role in addressing semantics issues. We can not integrate information as knowledge until we overcome the semantic challenges. Pragmatics covers understanding the kinds of practical needs and implications resulting from our integration of information. Trust refers to the ability to identify and track the provenance of our information to judge whether we use and integrate it or not. These upper layers of the stack are some of the unresolved issues we attend to over the rest of this book.

Benefits from Interoperability

Data interoperability should be one of the chief emphases of a knowledge management initiative because of these challenges, many of which have remained un-

solved for decades:

- 80% of all available information is in text or documents (unstructured);
- 40% of standard IT project expenses are devoted to data integration in one form or another, due to the manual effort needed for data migration and mapping;
- Information volumes are now doubling in fewer than two years;
- Current information resides in individual stores, stovepiped from one another, with few or no connections to external knowledge sources;
- Other trends including smartphones and sensors are further accelerating information growth; and
- Effective business intelligence requires the use of quality, integrated data.

The problem is creating a focus and then beginning to implement a data interoperability initiative. We know it promises to deliver some key, measurable benefits:

- *Efficiency* — trillions of dollars are spent each year globally in the research, creation, re-use, publishing, storing and browsing of information. Relevant information is hard to find, and sometimes we overlook useful but obscure information. The lack of reuse of prior good content because it is not discoverable is unconscionable given today's technologies;
- *Cost* — missed information or lack of awareness of relevant information leads to increased time, increased direct costs (labor and material), and increased indirect costs to re-create it. Awareness, understanding, and re-use of existing information would save millions or more for large firms annually if we could overcome these interoperability gaps;
- *Insight* — drawing connections between previously unconnected things and enabling discovery are essential inputs to innovation, itself the overall driver of productivity (and, therefore, wealth) gains. The reinforcing leverage of interoperability resides in its ability to bring new understandings and insights; and
- *Capture* — we benefit by capturing the many fields, data streams, APIs, mappings, DBs, datasets, Web content, on-the-fly discoveries, and device sensors available through the connectedness of the Web and the Internet of things (IoT).

For decades, the vision of data interoperability has mostly remained unfulfilled. Though significant progress has occurred in climbing the data federation pyramid, only when one is at the very topmost layers can we achieve actual data interoperability. The semantics are an absolute threshold. A few practitioners and a few exemplary organizations have demonstrated the worth of semantic technologies to leverage this next step. Doing so adheres to Peirce's *pragmatic maxim*, the understanding of a topic or object by an apprehension of all of the practical consequences potentially arising from it.

Adopting knowledge graphs is a prerequisite for applying semantic technologies to the fullest. Once adopted with the graph mindset embraced, it is then straightfor-

ward to extend the scope of the graphs a bit to encompass labels for user interfaces and calls to small, external Web applications. We discuss these so-called *administrative ontologies* in *Chapter 11*. These practical techniques cost little to implement and can be a useful adjunct to standard knowledge maintenance.

Material progress on the data interoperability challenge will bring us one step closer to self-service information management. The benefits and flexibilities from doing so will extend from creating data and content to publishing and deploying it. The fact that any source — internal or external — or format — unstructured, semi-structured and structured — can be brought together with semantic technologies is a qualitative boost over existing KM approaches. Further, since we represent all information via simple text formats, we can readily manipulate and manage that information with easy to understand tools and applications. Reliance on open standards and languages by semantic technologies also leads to greater use and availability of open source systems. In short, self-service information management could be one of the great benefits from interoperability. These are the kinds of opportunities that will enable knowledge management to fulfill its vision.

A Design for Interoperating

Ultimately, since we express all of our content and information with human language, we need to start there to understand the first sources of semantic differences. Like the differences in human language, we also have differences in worldviews and experience. These differences are often conceptual and reveal distinctions in real-world perspectives and experiences. From there, we encounter differences in our specific realms of expertise or concern, or the relevant domain(s) for our information and knowledge. Then, as we probe details, we give our observations and characterizations data and values to specify and quantify our observations. The attributes of these data are subject to the same semantic vagaries as concepts. Attributes also pose challenges in how we measure and express units.

The current challenge is to resolve differences in meaning, or *semantics*, between disparate data sources. Your '*glad*' may be someone else's '*happy*' and you may organize the world into countries while others organize by regions or cultures. From the conceptual to actual data, then, we see differences in perspective, vocabularies, measures, and conventions. Only by systematically understanding these sources of heterogeneity — and then explicitly addressing them — can we begin to try to put different information on a common footing. Only by reconciling these differences can we begin to get data to interoperate. Some of these differences and heterogeneities are intrinsic to the nature of the data at hand. Some of these heterogeneities also arise from the basis and connections asserted between datasets, as misuse of the *sameAs* predicate showed in early linked data applications. Fortunately, in many areas, we are transitioning due to technological progress to overcome many of these sources of semantic heterogeneity. Semantic Web approaches where data items are assigned unique IRIs are another source of making integration easier. Moreover, whether all agree from a cultural aspect if it is right, we also see English becoming the *lingua*

franca of research and data.

To bring about a basis for data interoperability, John Blossom argues the importance of Web approaches and architectures; incorporation of external data; leverage of Web applications; and, use of open standards and APIs to avoid vendor lock-in.[6] Much, if not all of this, can be aided by open source software. Open source is not indispensable: commercial products that embrace these approaches can also be compatible components across the stack. Further, we need to resolve semantic heterogeneities. Though only a single layer of the pyramid in *Figure 4-1* above, resolving semantics is a complicated task and may involve *structural conflicts* (such as naming, generalization, aggregation), *domain conflicts* (such as schemas or units), or *data conflicts* (such as synonyms or missing values). Researchers have identified nearly 40 distinct types of possible semantic heterogeneities, to which we delve into more detail in *Chapter 5*.

Semantic technologies give us the basis for understanding differences in meaning across sources, specifically geared to address real-world usage and context. Semantic tools are essential for providing common bases for relating structured data across various sources and contexts. These same semantic tools are also the basis by which we can determine what unstructured content 'means,' thus providing the structured data tags that also enable us to relate documents to conventional data sources (from databases, spreadsheets, tables, and the like). These semantic technologies are thus the key enablers for making information — unstructured, semi-structured and structured — understandable to both humans and machines across sources. Such understandings are then the basis for powering the artificial intelligence applications involving human language.

An initial embrace of semantic technologies for knowledge management often naturally leads to adopting knowledge graphs. These ontologies provide a means to define and describe these different worldviews. Referentially integral languages such as RDF (Resource Description Framework) and its schema implementation (RDF-S) or the Web ontological description language OWL are leading standards among other emerging ones for machine-readable means to communicate the semantics of data. You can read more about the languages of these semantic technologies in *Chapter 8.*

Adoption of semantic technologies does not necessarily mean open data nor open source (though they are suitable for these purposes with many open source tools available). We can apply the techniques equivalently to internal, closed, proprietary data and structures. We can use these techniques as a basis for bringing external information into the enterprise. The use of 'open' here refers to the critical use of the open world assumption (*Chapter 9*). Moreover, the design practices we recommend here do not require replacing current systems and assets; they can be applied equally to public or proprietary information; and, they can be tested and deployed incrementally at low risk and cost. The very foundations of our recommended practice encourage a learn-as-you-go approach and active and agile adaptation. While embracing semantic technologies can lead to quite disruptive benefits and changes, we can do so as a layered initiative with minimal disruption. Incremental adoption is

one of the most compelling aspects of semantic technologies.

KNOWLEDGE-BASED ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is the use of computers to do or assist complex human tasks or reasoning. AI has many, broad sub-fields from pattern recognition to robotics, and sophisticated planning and optimizations. Knowledge-based artificial intelligence, or KBAI, is the use of large statistical or knowledge bases to inform feature selection for machine-based learning algorithms used in AI. Correctly expressed KBs can support creating positive and negative training sets, promote feature set generation and expression, and generate reference standards for testing AI learners and model parameters. The use of knowledge bases to train the features of AI algorithms improves the accuracy, recall, and precision of these methods. These improvements lead to better information queries, including for pattern recognition. Further, in a virtuous circle, KBAI techniques can also be applied to identify additional possible facts within the knowledge bases themselves, improving them further still for KBAI purposes. Lastly, we hope that better ways to represent knowledge (with richer feature sets) may help unlock some of the black-box aspects typical of neural nets and deep learning.

Knowledge-based artificial intelligence is not a new idea. Its roots extend back perhaps to one of the first AI applications, Dendral, more than a half-century ago in 1965. Edward Feigenbaum initiated Dendral, which became a ten-year effort to develop software to deduce the molecular structure of organic compounds using scientific instrument data. Dendral was the first expert system and set the outline for knowledge-based systems, which are one or more computer programs that reason and use knowledge bases to solve complex problems. Indeed, it was in the area of expert systems that AI first came to the attention of most enterprises. Expert systems spawned the idea of knowledge engineers, whose role was to interview and codify the logic of the chosen experts. However, expert systems proved expensive to build and difficult to maintain and tune.

The specific identification of 'KBAI' was (to my knowledge) first made in a Carnegie-Mellon University report to DARPA in 1975.[7] The source knowledge bases were broadly construed, including listings of hypotheses. The first known patent citing knowledge-based artificial intelligence is from 1992.[8] Within the next ten years there were dedicated graduate-level course offerings on KBAI at many universities, including at least Indiana University, SUNY Buffalo, and Georgia Tech. In 2007, Bossé *et al.* devoted a chapter to KBAI in their book on information fusion, but still, at that time, the references were more generic.[9] However, by 2013, as a report by Hovy *et al.* indicates, collaborative, semi-structured information stores such as Wikipedia were assuming a prominent position in AI efforts.[10] It has been the combination of KB + AI that has led to the notable AI breakthroughs for knowledge purposes of the past, say, decade. It is in this combination that we gain the seeds for sowing AI benefits in other areas, from tagging and disambiguation to the complete integration of text

with conventional data systems. Further, the structure of all of these systems can be made inherently multi-lingual, meaning that context and interpretation across languages can be brought to our understanding of concepts.

KBAI is part of the AI branch that includes knowledge-based systems. Besides areas already mentioned, knowledge-based systems also include:

- Knowledge models — formalisms for knowledge representation and reasoning; and
- Reasoning systems — software that generates conclusions from available knowledge using logical techniques such as deduction and induction.

As the influence of expert systems waned, another branch emerged, that of knowledge-based engineering and their support for CAD– and CASE-type systems. Still, we can charitably describe the overall penetration to date of most knowledge-based systems as disappointing.

It is different today. Structured information and the means to query it now gives us a powerful, virtuous circle whereby our knowledge bases can drive the feature selection of AI algorithms, while those very same algorithms can help find still more features and structure in our knowledge bases (see *Figure 4-2*). Once we reach this threshold of feature generation, we now have a virtuous dynamo for knowledge discovery and management. We can use our AI techniques to refine and improve our knowledge bases (the top loop of *Figure 4-2*), which then makes it easier to improve our AI algorithms and incorporate still further external information (the bottom loop). Effectively utilized KBAI (knowledge-based artificial intelligence) thus becomes a generator of new information and structure.

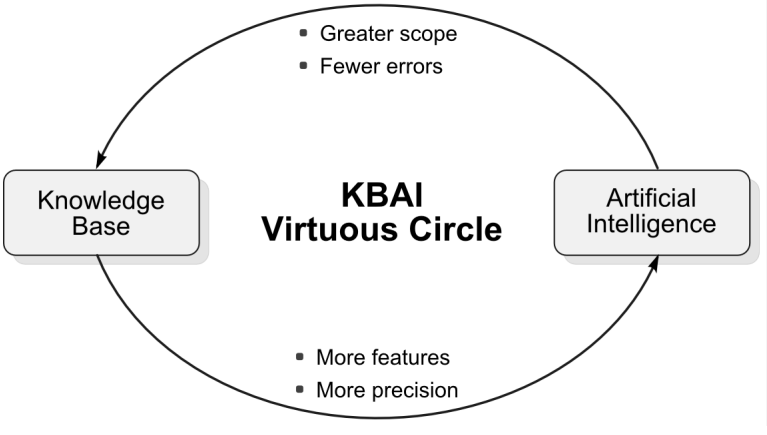

Figure 4-2: Virtuous Knowledge-based Artificial Intelligence

This virtuous circle has not been applied fully, seen mostly to date in the adding of new facts to Wikipedia or Wikidata. Importantly, we can apply these same basic techniques to the very infrastructural foundations of KBAI systems in such areas as

data integration, mapping to new external structure and information, hypothesis testing, diagnostics and predictions, and the myriad other uses to which researchers for decades hoped AI would contribute. The virtuous circle between knowledge bases and AIs does not require us to make leaps and bounds improvements in our core AI algorithms. Instead, we need only stoke our existing AI engines with more structure and knowledge fuel to keep the engine churning.

KBAI has two primary knowledge sources: recognized knowledge bases, such as Wikipedia, and statistical corpora. Knowledge bases are coherently organized information with instance data for the concepts and relationships covered by the domain at hand, all accessible in some manner electronically. Knowledge bases can extend from the nearly global, such as Wikipedia, to particular topic-oriented ones, such as restaurant reviews or animal guides. Some electronic knowledge bases are designed explicitly to support digital consumption, with defined schema and standard data formats and, increasingly, APIs. Others may be electronically accessible and highly relevant, but the data is hard to consume and requires extraction and processing before use. Hundreds of knowledge bases are suitable for artificial intelligence, most of a restricted domain nature.[9] *Chapter 11* is devoted to this topic.

The use and role of statistical corpora are harder to describe. Statistical corpora provide relationships or rankings to aid the processing of (mostly) textual information. Uses can range from entity extraction to machine language translation. Huge sources, such as search engine indexes or massive crawls of the Web, are most often the sources for these knowledge sets. The statistical corpora or databases have a precise focus. While lists of text corpora and many other things may contribute to this category, the ones actually in commercial use are huge and designed for bespoke functionality. A good example is the Web 1T 5-gram data set.[12] This data set, contributed by Google for public use in 2006, contains English word n-grams and their observed frequency counts. N-grams capture word tokens that often coincide with one another, from single words to phrases. The length of the n-grams ranges from unigrams (single words) to five-grams. Google generated the database from approximately 1 trillion word tokens of text from publicly accessible Web pages.

Another example of statistical corpora is what Google's Translate uses. According to Franz Josef Och, a former lead manager at Google for its translation activities, a solid base for developing a usable language translation system for a new pair of languages should consist of a bilingual text corpus of more than a million words, plus two monolingual corpora each of more than a billion words. Statistical frequencies of word associations form the basis of these reference sets. Google initially seeded its first language translators using multiple language texts from the United Nations.[7] If we add structure to statistical corpora, they may evolve to look more like a knowledge base. NELL, for example, contains a relatively flat listing of assertions extracted from the Web for various entities. NELL goes beyond frequency counts or relatedness but does not have the full structure of a general knowledge base like Wikipedia.[14] We thus can see that statistical corpora and knowledge bases reside on a continuum of structure, with no bright line to demark the two categories.

Created using both statistical techniques and results from machine learning, we

are using these methods to extract massive datasets of entities, relationships, and facts from the Web. Some of these efforts, like NELL, or its academic cousins such as KnowItAll or Open IE (UWash), involve extractions from the open Web. Others, such as the terabyte (TB) n-gram listings from Google, are derived from Web-scale pages or Google books. Word, sentence or graph vectors are other types. These examples are but a sampling of various datasets and corpora available. These various statistical datasets may be used directly for research on their own or may contribute to further bootstrapping of still further-refined AI techniques. Similar datasets are aiding advertising placements and search term disambiguation. In some cases, while the full datasets may not be available, open APIs may be available for areas such as entity identification or tabular data.

The Web is the reason these sources — both statistical corpora and knowledge bases — have proliferated, so the dominant means of consuming them is via Web services with the information defined and linked to IRIs. The availability of electronically accessible knowledge bases, exemplified and stimulated by Wikipedia, has been the telling factor in recent artificial intelligence advances. For example, at least a thousand different papers cite using Wikipedia for various natural language processing, artificial intelligence, or knowledge base purposes. These papers began to stream into conferences about 2005 to 2006, and have not abated since. In turn, researchers are applying the various techniques innovated for extracting more and more structure and information from Wikipedia to other semi-structured knowledge bases, resulting in a renaissance of knowledge-based processing for AI purposes. These knowledge bases are emerging as the information substrate under many recent computational advances, such as for virtual agents we command by voice. The agents use pattern recognition at the front and back end of the workflow based on statistical datasets derived from phonemes and text. The agents apply natural language processing, as informed by knowledge bases and represented by semantic technologies, to the text sandwiched between these bookends to conduct question understanding and answer formulation.

This remarkable chain of processing is now almost taken for granted, though its commercial use in virtual agents is fewer than ten years old. For different purposes with different workflows, we see useful question answering and diagnosis with systems like IBM's Watson[15] and structured search results from Google's Knowledge Graph.[16] Try posing some questions to Wolfram Alpha and then stand back and be impressed with the data visualizations. Behind the scenes, pattern recognition from faces to general images or thumbprints is further eroding the distinction between man and machine. Google's Knowledge Vault extends the Knowledge Graph using probabilistic methods to add facts gleaned from the Web.[17] Google Translate now effectively covers language translation between more than 100 human languages.[18] All major Web players are active in these areas, from Amazon's recommendation system[19] to Facebook, Microsoft, Twitter or Baidu. Unfortunately, the sponsors required significant effort to re-organize and characterize the source knowledge bases as coherent inputs to KBAI. All of the impressive advances we have seen to date in distant supervised machine learning applications result from bespoke, manually

trained efforts, repeated numerous times across providers.

We now understand how content-rich electronic knowledge bases may help power machine learning for natural language understanding and information processing. The usefulness is apparent to re-express the KBs to maximize the features available for machine learning, including disjointedness assertions to enable selection of positive and negative training sets. Specific aspects of the KBs, for which such re-organization is appropriate, include concepts, types, entities, relations, events, attributes, and statements. As we build these frameworks, they can facilitate mappings to other knowledge structures, and aid in data interoperability and information integration. We may apply these same principles for building a general structure to new domains or new knowledge bases. Three significant aspects — in machine learning, knowledge supervision, and feature engineering — intersect to re-express knowledge bases for KBAI purposes. Let's investigate each in turn.

Machine Learning

Machine learning is the construction of algorithms that can learn from and make predictions on data by building a model from example inputs. A wide variety of techniques and algorithms may be employed — such as Markov chains, neural networks, conditional random fields, Bayesian statistics, and many other options — that can be characterized by many dimensions. Some are supervised, meaning we need to train them against a standard labeled corpus to estimate parameters; others require little or no training — that is, are unsupervised — but may be less accurate as a result. Some are statistical; others use pattern matching of various forms. *Supervised learning* is a machine learning task of inferring a function from labeled training data, which optimally consists of positive and negative training sets. The supervised learning algorithm analyzes the training data and produces an inferred function that is used to determine the correct class labels for unseen instances. In supervised learning, we present positive and (often) negative training examples to the learning algorithm. *Unsupervised learning* is a different form of machine learning, in that the approach attempts to find meaningful, hidden patterns without the use of labeled data. We require no training examples in unsupervised learning. Supervised methods are more accurate than unsupervised methods, and nearly universally so in the realm of content information and knowledge.

Deep learning is a recent trend to combine multiple techniques. In this approach, the algorithm models the problem set as a layered hierarchy of distributed representations, with each layer using (often) neural network techniques for unsupervised learning, followed by supervised feedback (often termed 'back-propagation') to fine-tune parameters. While computationally slower than other techniques, this approach has the advantage of automating the supervised learning phase and is effective across a range of AI applications. The major disadvantage is that deep learning creates 'hidden' statistical features within its intermediate layers; it is impossible to interpret how the technique determines its final results. Deep learning is nonetheless producing amazing results in recognizing images, audio, video or sensory percep-

tion, and language translation. Effectiveness in knowledge areas has been less satisfactory, with the lack of explanatory power a further detriment.

In supervised learning, the main drawback is the effort and expense associated with labeling the positive or negative training examples (sets). The maximum effort occurs from constructing the training sets entirely by hand. We can reduce the effort by constructing them in a semi-automatic manner or by letting knowledge bases provide the labels. These techniques are known as semi-supervised, weak supervision, or distant supervision.[20] [21] [22] The accuracy of the eventual models is only as good as the trueness of the input training sets, with traditionally the best results coming from manually determined training sets. We call the most accurate of these sets 'gold standards.' The creation of manual training sets may consume as much as 80% of overall machine learning efforts and is always a time-consuming task whenever employed.

One way to help overcome the costs of developing manual training sets is by a sub-class of supervised learning called *distant supervision*, which is a method to use knowledge bases to label entities or other types automatically in text, which is then used to extract features and train a machine learning classifier. The knowledge bases provide coherent positive training examples and avoid the high cost and effort of manual labeling. When we use knowledge bases for distant supervision, we only use a portion of the structure as features. Still, other distant supervision efforts may be geared to other needs and use a different set of features. Indeed, broadly considered, knowledge bases have a rich diversity of possible features. These potential features arise from the text, and its content, syntax, semantics, and morphology; use vectors of co-occurring terms or concepts; categories; conventions; synonyms; linkages; mappings; relations; attributes; content placement within its knowledge graph; and, disjointedness. *Appendix C* shows just how broadly diverse these types of features may be.

State-of-the-art machine learning for natural language processing and semantics uses distant supervision and knowledge bases like Freebase[23] or Wikipedia to extract training sets for supervised learning. We can create relatively clean positive and negative training sets with much-reduced effort over manually created ones. However, as employed to date, distant supervision has mostly been a case-by-case, problem-by-problem approach, and most often applied to entity or relation extraction. The effort has heretofore not been systematic in approach nor purposefully applied across a range of ML applications. How to structure and use knowledge bases across a range of machine learning applications with maximum accuracy and minimum effort is what we call *knowledge supervision*, which I discuss more in a moment.

Besides supervised and unsupervised learning, a third broad category of machine learning is *reinforcement learning*. Unlike the first two categories where prior examples are used to learn a statistical prediction for new cases, reinforcement learning focuses on the learning process itself. Reinforcement learning is an active, iterative process where rewards associated with a given set of objectives are used to select from and optimize next actions. "Although one might be tempted to think of reinforcement learning as a kind of unsupervised learning because it does not rely on examples of correct behavior, reinforcement learning is trying to maximize a reward

signal instead of trying to find hidden structure."[24] Because we have not heretofore linked knowledge bases with models of action, we have limited our use of KBs to static questions and applications. Insofar as we may be able to stage and embed our knowledge bases into a true action model, a topic of *Chapters 7* and *16*, we may be able to see them inform models of reinforcement learning as well.

Knowledge Supervision

Whatever the combination of method, feature set, or training sets, the ultimate precision and accuracy of the machine learning requires the utmost degree of true results in both positive and negative training sets. Training to inaccurate information merely perpetuates inaccurate information. As anyone who has worked extensively with source knowledge bases may attest, assignment errors and incomplete typing and characterizations are all too familiar. Further, few existing knowledge bases provide disjointedness assertions. Though early efforts in artificial intelligence understood that capturing and modeling common sense was both an essential and surprisingly tricky task — the impetus, for example, behind the thirty-year attempt of the Cyc knowledge base — what is new in today's circumstance is how these massive knowledge bases can inform and guide symbolic computing. The literally thousand research papers regarding the use of Wikipedia data alone shows how these massive knowledge bases are providing base knowledge around which AI algorithms can work. Unlike the early years of mostly algorithms and rules, AI has now evolved to explicitly embrace Web-scale content and data and the statistics that we may derive from global corpora.

The innovation of distant supervision has been to leverage knowledge bases to overcome the costs of labeling data and creating positive and negative training sets for supervised learning. Wikipedia, as noted, has been leveraged for these purposes by such players as IBM, Google, Facebook, Baidu, Microsoft, Amazon, and others. However, each of these players has done their own massaging of Wikipedia from scratch to support these purposes. None of this is free. Much purposeful work is necessary to configure and stage the data structures and systems that support the broad application of distant supervision. The idea of *knowledge supervision*, our third component to KBAI along with feature engineering and machine learning, is to take distant supervision one step further.

To achieve these aims for knowledge supervision, we purposefully stage our source knowledge bases. We structure the KBs to maximize information extraction of concepts, entities, relations, attributes, and events because we have provided such structure in the central knowledge graph of KBpedia. We use these structures for linking and mapping to still additional knowledge sources. We support this entire process with methods for codifying self-learning such that our systems continue to get more accurate. We test continuously to improve the assignments and the accuracy of the system.

- Attribute 'slot filling'
- Bespoke analysis
- Bespoke platforms
- Classifiers
  - Concept tagging
  - Document categorization
  - Entity classifiers
- Cluster analysis
  - Concept clustering
  - Data clustering
- Cognitive computing
- Converters
  - Data conversion
  - Format converters
- Disambiguators
  - Word sense disambiguation
- Duplicates removal
- Entity dictionaries
  - Gazetteers
- Information extraction
  - Attribute extractors
  - Entity recognizers
  - Event extractors
  - Relation extractors
  - Sub-graph extraction
- Knowledge base improvements
- Knowledge base population
- Machine learning
  - Deep learning
  - Distant supervision
  - Knowledge supervision
  - Supervised learning
  - Reinforcement learning
  - Unsupervised learning
- Mapping
  - Data mapping
  - Knowledge base mapping
  - Ontology mapping
- Master data management
- Natural language processing
  - Artificial writing
  - Autocompletion
  - Entity linking
  - Language translation
  - Multi-language versions
  - Phrase (n-gram) identification
  - Speech recognition
  - Speech synthesis
  - Spell correction
  - Text generation
  - Text summarization
- Ontologies
  - Ontology development
  - Ontology matchers
  - Ontology mappers
- Pattern recognition*
  - Computer vision
  - Facial recognition
  - Image recognition
  - Optical character recognition
- Reasoning
  - Inferencing
  - Question answering
  - Recommendation systems
  - Semantic relatedness analysis
  - Sentiment analysis
- Search and information retrieval
- Semantic publishing

Table 4-1: Knowledge-based AI Applications

Table 4-1 provides a listing of some of those areas to which knowledge supervision may apply; some already use distant supervision or have been shown useful in academic research, others we have not yet exploited.

Knowledge supervision is thus the purposeful structuring and use of knowledge bases to provide features and training sets for multiple kinds of machine learners that we may apply to multiple artificial intelligence outcomes. While distant supervision also uses knowledge bases, it does so passively, taking the knowledge bases as is, rather than re-expressing them in a purposeful, directed manner across multiple machine learning problems. Knowledge supervision is thus the better method to achieve KBAI.

* Not a *knowledge supervision* ML option.[28]

Feature Engineering

Feature engineering is the process of creating, generating and selecting the features used in machine learning, based on an understanding of the underlying data and choosing ones likely to impact learning results and effectiveness. A feature is a measurable property of the analyzed system. A feature is equivalent to what statistics calls an explanatory variable. The ML algorithms tend to favor features with high explanatory power independent of other features (that is, they are *orthogonal*) because each added feature adds a computational cost. Many features correlate with one another; in these cases, we need to find the strongest signals and exclude the other correlates. Tuning and refinement are also more difficult with too many features, what has sometimes been called the curse of dimensionality. Overfitting by using too many features is also often a problem, which limits the ability of the model to generalize to other data. Still, using too few features results in inadequate explanatory power.

Features and training sets are the major determinants of how successful the machine learning is. *Training sets* are a set of data used to discover potentially predictive relationships. In supervised learning, a positive training set provides data that meet the training objectives; a negative training set fails to meet the objectives. Features also pose trade-offs and require skill in selection and use. Though it is hard to find a discussion of best practices in feature extraction, many practitioners note that striking this balance is an art.[25] We might also need multiple learners to capture the smallest, independent (non-correlated) feature set with the highest explanatory power.[26]

An understanding of what features are possible within knowledge bases is the first hurdle toward more purposeful knowledge supervision. We stage the structured information as RDF triples and OWL ontologies, which we can select and manipulate via APIs and SPARQL. We also stage the graph structure and text with the support of a search engine,[27] which gives us powerful faceted search and other advanced NLP manipulations and analyses. These same features may also be utilized to extend the features set available from the knowledge base through such actions as extracting new entities, attributes, or relations; fine-grained entity typing;[28] creation of word vectors or tensors; results of graph analytics; forward or backward chaining; and efficient processing structures. *Appendix C* overviews the features available to KBAI from the KBpedia knowledge structure.

Because all features are selectable via either structured SPARQL query or faceted search, it is also possible to more automatically extract positive and negative training sets. Attention to proper coverage and testing of disjointedness assertions is another purposeful step useful to knowledge supervision since it aids identification of negative examples for the training. We get into such operational details in *Chapters 12* to *14*.

These opportunities do not exhaust those available from applying Peircean guidelines to knowledge representation, backed by knowledge bases. However, knowledge management, data interoperability, and knowledge-based AI form the leading edge

of promising new capabilities.

Chapter Notes

1. Some material in this chapter was drawn from the author's prior articles at the *AI3:::Adaptive Information* blog: "Search and the '25% Solution'" (Sep 2005); "Climbing the Data Federation Pyramid" (May 2006); "structWSF: A Framework for Data Mixing" (Jun 2009); "The Open World Assumption: Elephant in the Room" (Dec 2009)' "Changing IT for Good" (Mar 2010); "Ontology-Driven Apps Using Generic Applications" (Mar 2011); "Democratizing Information with Semantics" (Apr 2011); "Leveraging Intangible Assets Using Semantic Technologies" (May 2011); "Knowledge Supervision as a Grounding for Machine Learning" (Jun 2015); "Why the Resurgence in AI?" (Jan 2016).
2. Girard, J., and Girard, J., "Defining Knowledge Management: Toward an Applied Compendium," *Online Journal of Applied Knowledge Management*, vol. 3, 2015, pp. 1–20.
3. Even with natural language processing (NLP), we do not see fully automated systems. We need some manual inspection to remove errors in NLP processing and to ensure quality commitments to the *knowledge base*. Such managed NLP systems are best characterized as "semi-automatic" and place human editors into critical parts of the workflow.
4. Buneman, P., "Semistructured Data," *ACM Symposium on Principles of Database Systems (PODS)*, Tucson, Arizona: 1997, pp. 117–121.
5. Davidson, S. B., Overton, C., and Buneman, P., "Challenges in Integrating Biological Data Sources," *Journal of Computational Biology*, vol. 2, 1995, pp. 557–572.
6. Blossom, J., "Enterprise Publishing and the 'New Normal' in I.T. – Are You Missing the Trend?," Mar. 2010.
7. Erman, L. D., and Lesser, V. R., *A Multi-level Organization for Problem Solving Using Many Diverse, Cooperating Sources of Knowledge*, Pittsburgh, Pa: Carnegie-Mellon University, 1975.
8. See "Method for converting a programmable logic controller hardware configuration and corresponding control program for use on a first programmable logic controller to use on a second programmable logic controller," US Patent No 5142469 A, August 25, 1992.
9. Bossé, É., Roy, J., and Wark, S., eds., *Concepts, Models, and Tools for Information Fusion*, Artech House Publishers, 2007.
10. Hovy, E., Navigli, R., and Ponzetto, S. P., "Collaboratively Built Semi-Structured Content and Artificial Intelligence: The Story So Far," *Artificial Intelligence 194*, 2013, pp. 2–27.
11. Suchanek, F. M., and Weikum, G., "Knowledge Bases in the Age of Big Data Analytics," *Proceedings of the VLDB Endowment*, 2014, pp. 1713–1714.
12. The Web 1T 5-gram dataset is available from the Linguistic Data Corporation, University of Pennsylvania.
13. Och, F. J., "Statistical Machine Translation: Foundations and Recent Advances," *The Tenth Machine Translation Summit*, Phuket, Thailand: 2005.
14. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. R., Jr, and Mitchell, T. M., "Toward an Architecture for Never-Ending Language Learning," *AAAI*, 2010, p. 3.
15. IBM, "This is Watson," *IBM Journal of Research and Development*, vol. 56, 2012.
16. Singhal, A., "Introducing the Knowledge Graph: Things, Not Strings," *Official Google Blog*, May 2012.
17. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W., "Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion," *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2014, pp. 601–610.
18. Schulz, T., "Google's Quest to End the Language Barrier," *Spiegel Online*, Sep. 2013.
19. Linden, G., Smith, B., and York, J., "Amazon. com Recommendations: Item-to-item Collaborative Filtering," *IEEE Internet Computing*, vol. 7, 2003, pp. 76–80.
20. Distant supervision was earlier called self-supervision, indirect supervision or weakly-supervised.

21. Craven, M., and Kumlien, J., "Constructing Biological Knowledge Bases by Extracting Information from Text Sources," *International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 1999, pp. 77–86.

22. Mintz, M., Bills, S., Snow, R., and Jurafsky, D., "Distant Supervision for Relation Extraction without Labeled Data," *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, Suntec, Singapore, 2-7: 2009, pp. 1003–1011.

23. Freebase was retired from service by its owner, Google, in 2016. Many of its knowledge assertions have been transferred to Wikidata.

24. Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction, 2nd Edition (pre-release draft)*, Cambridge, Mass.: MIT Press, 2017.

25. Domingos, P., "A Few Useful Things to Know About Machine Learning.," *Communications of the ACM*, vol. 55, 2012, pp. 78–87.

26. A rich literature provides guidance on feature selection and feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the available features. It is also possible to apply methods -- the best known and simplest being principal component analysis, among many -- to reduce feature size (dimensionality) with an acceptable loss in accuracy.

27. We have used the open source systems Lucene and Solr in our platform work, though options such as Elasticsearch would work as well.

28. Ling, X., and Weld, D. S., "Fine-Grained Entity Recognition," *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.

28. Pattern recognition and its sub-methods are included in the table because they can be trained against labeled metadata for identification and captioning purposes. The input data differs from knowledge sources because they are digitized media. For this reason, these should not be understood as *knowledge supervised* sources.