Machines with WebSense

Loizos Michael
Open University of Cyprus
loizos@ouc.ac.cy

Abstract

We discuss the design and development of a novel web search engine able to respond to user queries with inferences that follow from the collective human knowledge found across the Web. The engine's knowledge — or *websense* — is represented and reasoned with in a logical fashion, and is acquired autonomously via principled learning, so that the soundness of the supported inferences can be a priori guaranteed through a formal analysis of the learning method used. This engine brings together the traditional AI goal of endowing machines with common sense, and the contemporary AI goal of making sense of the Web through machine reading.

Introduction

Since antiquity, humans have sought to develop machines to facilitate physically demanding tasks and reduce the associated burden for humans. With the advent of computers, a new use of machines to alleviate human burden, that of processing information, became more prominent. It was recognized early on that certain information processing tasks would require machines to possess some sort of intelligence (Turing 1950), and this soon led to the realization that encoding and processing commonsense knowledge in a computable logic-based form could be a central aspect in the development of such intelligent machines (McCarthy 1959).

The dream of building machines that can reason and draw inferences in a manner analogous to humans has not budged from the center stage of AI since its very beginning. What has drastically changed, however, is the representation of the knowledge assumed to be the object of the reasoning process, and the type of the algorithms that are used to manipulate that knowledge and draw inferences. The change was presumably a result of the realization that the logic-based knowledge and inference algorithms that early suggestions for AI systems sought to employ would not lead to the development of full-fledged AI systems, as they lacked a reasonable process for acquiring the knowledge. Many researchers turned their attention to other types of knowledge representation (e.g., Bayesian Networks, Neural Networks, statistical information on n-grams) and to algorithms appropriate for acquiring and reasoning with that type of knowledge.

Admittedly, those approaches have borne fruits. We now have machines very competent at filtering spam emails (e.g., SpamAssassin), translating between languages (e.g., Google Translate), understanding verbal instructions (e.g., Siri), and playing games on television shows (e.g., Watson). Yet one is left with the distinct feeling that all these great results were mostly an engineering feat. This in no sense diminishes the importance and the utility derived from these accomplishments of the human intellect. But that does not change the fact that they seem to have been obtained in a way that says little about how humans reason, at least at the higher levels of the cognitive ladder, where a certain form of structured logic-like knowledge seems to play an important role.

Despite such concerns falling — at least somewhat — out of favor in mainstream AI, a large number of researchers still pursue research in good old-fashioned AI (GOFAI). Given the continuing interest, it is rather surprising that the concern that presumably "drove away" certain AI researchers from GOFAI — the lack of a reasonable and autonomous process for acquiring knowledge (contrast to Cyc (Lenat 1995)) — has not been satisfactorily addressed. It is true that even under the assumption that knowledge is somehow externally provided, there are still many fundamental and challenging problems to solve in GOFAI. Yet, if the solutions to these problems are to be deployed widely and fruitfully, analogously to how other AI technologies have been, a plausible answer to the knowledge acquisition problem needs to be found. Learning has offered a solution to other representations of knowledge, and we posit that learning will do so for the logic-based knowledge acquisition problem as well.

We suggest that both the theoretical understanding and the tools are now available for logic-based knowledge bases to be extracted by machines autonomously reading the Web. In the sequel we discuss the type of knowledge that one can hope to extract from the Web, and review relevant work on machine reading. We then present the architecture of a concrete machine that we have developed that is able to acquire and reason with the knowledge so extracted, and show how it can be used to draw commonsense-like inferences from queries given by users in natural language text. We conclude with a discussion on evaluation, and possible applications.

From CommonSense to WebSense

Much in the same way that Johannes Gutenberg's invention of the printing press — "God's highest act of grace", according to Martin Luther — led societies from the Agrarian

Age of feudalism and uneducated masses to the Information Age of global education and technology, so is the Web leading societies from the Information Age of consumption of recorded information to the Knowledge Age of distributed knowledge production and global access to the intelligence of others. For the first time in history both the opportunity and the means are present to exploit this vast knowledge source for the benefit and advancement of the human race.

To delineate it from other types of knowledge that humans typically manipulate in everyday life, we use the noun / adjective *"websense"* (contra to *common*sense) to characterize the knowledge found on the Web, which should be understood to encompass: *(i)* expert knowledge (e.g., fever near a swamp suggests malaria), as that offered in web-pages authored by medical experts; *(ii)* cultural biases (e.g., if visiting someone, bring a bottle of good wine), as that offered in movie scripts or personal blogs; *(iii)* misconceptions (e.g., correlation implies causation), as that offered in personal unaudited web-pages; *(iv)* fictional statements (e.g., the fox served soup to the crane), as that offered in stories or fables; and even *(v)* deliberate lies (e.g., heavy smoking is good for you), as that offered in certain types of advertisement.

Websense is inherently distributed in the strong sense that even an individual piece of knowledge might not be stated explicitly in a single web-page, but be implicitly encoded across the Web. For instance, the websense knowledge that heavy smoking is good for humans can be seen to be distributed across the scripts of numerous movies found on the Web, portraying their lead character as being healthy, popular, and smoking a lot. The goal, then, is the development of machines able to extract this implicitly encoded websense.

A number of approaches relevant to this goal have been considered. Most prominent is the Semantic Web, where information on web-pages is tagged with meta-data not only for its formatting but also for its semantics or meaning, providing, according to W3C, "a common framework that allows data to be shared and reused across application, enterprise, and community boundaries." The feasibility of its realization through the manual tagging that is assumed has been questioned, and to date the goal has not been fulfilled.

Unlike the tagging approach where human users are expected to tag the basic blocks of data from which one may later draw inferences, Wikipedia sought to gather the human knowledge (presumably found in other places on the Web as well) in a common and semi-structured place, burdening the human users with selecting, filtering, and updating the information it holds, without, however, dealing with the explicit task of drawing inferences. OpenMind sought to have humans provide directly the pieces of knowledge used for inference, still using the usual raw text format that is prevalent on the Web as its representation, in the hope that these nuggets of knowledge could be converted in a computable form from which inferences could be drawn (Stork 1999).

More recent approaches sought to automate the acquisition of facts from the Web, circumventing to some extent the human mediators (Etzioni et al. 2005; Schoenmackers et al. 2010). IBM's Watson has also heavily exploited the Web as a source of information for the limited task for which it was designed. In general, researchers have mostly been in agreement that text is a viable and abundant source of information for the acquisition of knowledge, and evidence for this claim has been provided (Liakata 2004), even if only in a limited domain and with the active involvement of a human curator.

The goal of capitalizing the Web as a source of information has been pointed out (Mitchell 2005; Etzioni, Banko, and Cafarella 2006), without offering concrete strategies of how to develop machines that extract from the Web not only facts, but knowledge in some appropriate computable form. In particular, a prerequisite to exploiting information found on the Web, is the extraction of information implicit in text. This is related to the Recognizing Textual Entailment (RTE) task (Dagan, Glickman, and Magnini 2006), which specifies a certain property (although not a process to ensure it holds) that such an inference is expected to have: an inference is valid if it would be typically recognized as such by humans. Natural Language Processing techniques for extracting facts and entities (Etzioni et al. 2005), verbs and their arguments (Cognitive Computation Group 2006), various syntactic elements (Collins 1999), or synonyms and hypernyms (Miller 1995) are heavily used in developing tools for the RTE task.

Wolfram|Alpha exploits and draws inferences from information found on the Web. Unlike the aforementioned approaches, however, this computational search engine operates on a certain subset of knowledge, characterized as axiomatized: the type of knowledge typically found in Mathematics, Physics, and other computationally-oriented disciplines. Given appropriately curated data, along with the axioms (e.g., laws, equations) that apply in a certain field, the engine computes necessary (in a mathematical sense) inferences. For the vast majority of information found on the Web the assumption of axiomatization is not always reasonable.

Recently, a principled approach to *unaxiomatized* knowledge acquisition from text has been proposed (Valiant 2000; 2006), placing emphasis both on the acquisition of knowledge in terms of computer-readable rules, but also on the efficiency of the acquisition task, and the robustness of the acquired knowledge, by building on well-studied learning algorithms (Littlestone 1988). Although no explicit techniques for processing text were described in that work, subsequent work on extracting knowledge from text (Michael and Valiant 2008) provided experimental evidence on the feasibility of that approach on a massive scale. The work presented herein follows a similar approach to these works.

More recent work offered further theoretical underpinnings to the RTE task, extending it to the much more challenging and useful *generation* case, and allowing for a principled view of the inferences that can be drawn from sentences (Michael 2009). In that work, knowledge bases are constructed completely autonomously, without any human supervision, and the inferences that are drawn are *guaranteed to be correct*, not against the human gold standard that RTE assumes, but against a formal objective metric of soundness (Michael 2008; 2010). This work can be seen as an empirical demonstration of this earlier theoretical work.

Searching the Web for Inferences

As a concrete application of machines endowed with websense, we consider the design and development of a novel

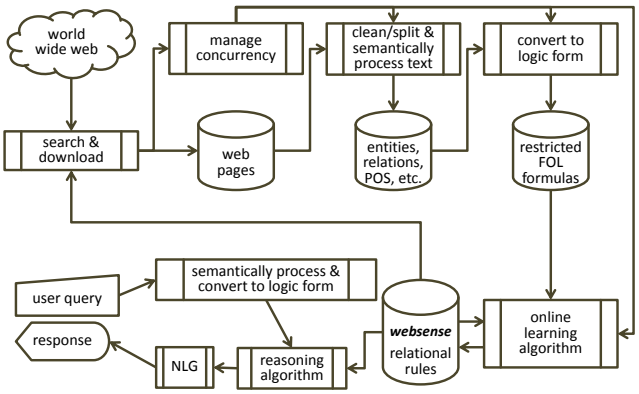

Figure 1: Architecture of a novel search engine.

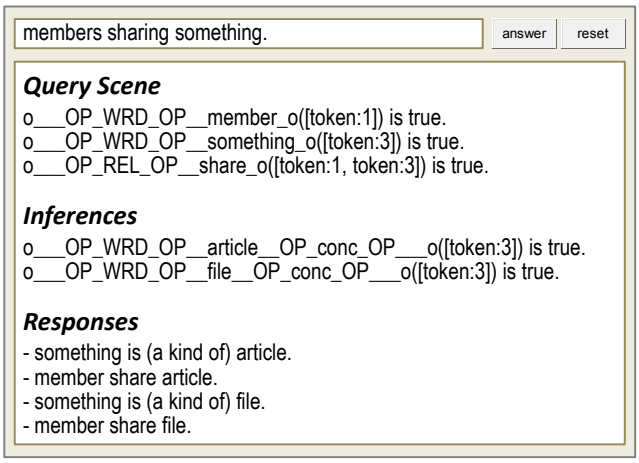

Figure 2: Interface of a search engine prototype.

web search engine able to respond to user queries provided in natural language text, with inferences that are implied by the given queries according to the collected human knowledge found across the Web. Importantly, the engine operates autonomously without human supervision, harvests the websense it uses by applying principled learning techniques, and is able to offer guarantees on the soundness of its inferences.

The architecture of our developed engine is illustrated in Figure 1. At its heart lies a relational knowledge base, with each rule encoding an equivalence that essentially provides a websense *definition* of the predicate at the head of the rule. An example of the representation of a rule is given below:

$file(x) \Leftrightarrow$	threshold(1.0)
$\exists v : scan(v, x) \wedge rogue(v)$	weight(0.962710)
$\exists v : share(v, x)$	weight(1.627098)
$\exists v : have(x, v) \wedge program(v)$	weight(0.645691)
$\exists v : open(v, x)$	weight(1.593269)

All variables in the head of a rule are assumed to be universally quantified over that rule. Variables that appear in a line of the body but not in the head must be explicitly quantified, and their scope is the formula appearing in that line.

Roughly, the semantics of each rule is as follows: Given a set of grounded predicates, one identifies which of the formulas in the body of the rule are made true (as in logic programming). The weights associated with formulas that are determined to be true (i.e., active) are summed up. If the total weight exceeds the rule threshold, then the head of the rule (with the grounding of its variables as determined by the rule body — again, as in logic programming) is inferred to be true; otherwise the head is inferred to be false. The precise semantics can be found in earlier work (Valiant 2000).

This type of rule encodes a linear threshold, where a sufficient number of formulas in the body of the rule need to hold for the head to also hold. Linear thresholds enjoy more expressivity than rules with conjunctive or disjunctive bodies, and are known to be amenable to a rather robust and efficient learning process (Littlestone 1988; Valiant 2000), even under arbitrarily incomplete information (Michael 2010).

Our engine autonomously acquires such rules by parsing text found on the Web, and applies these rules on the parsed user queries to return responses in natural language text, as automatically constructed from the drawn inferences. The underlying learning algorithm employed provides a priori guarantees on the appropriateness of the responses (Michael 2009). The engine interface in Figure 2 illustrates a particular query given by some user, the parsing of that query into a logic-based form, the websense inferences drawn from it (by applying, among others, the rule given earlier), and the composition of the inferences into natural language responses.

Building a Novel Search Engine

We continue to describe the design of the individual components of the search engine, and discuss certain choices made.

Web Crawling

Web crawling is depicted in the top-left part of Figure 1. In the simplest case, this is a typical crawler that downloads arbitrary web-pages. In our current design the crawler is externally provided with certain keywords (e.g., "spyware"), and downloads only pages containing those keywords. In particular, the web crawler uses a typical search engine to identify a seed web-page that includes the given keywords, and adds that web-page in a queue. It then repeatedly removes a web-page from the queue, probes all the web-pages that are linked from the former, and adds those with more than α occurrences of the keywords in the queue. If the removed web-page contains at least β occurrences of the keywords (where $\beta > \alpha$), it is sent for further processing. Thus, α is the eligibility threshold for web-pages to be further explored through their links, whereas β is the (more strict) eligibility threshold for web-pages to be parsed for knowledge extraction. Values for α and β were determined empirically.

The ultimate goal here is to download web-pages that are most useful given the websense already acquired. This can be done by allowing the engine to internally determine the keywords that guide the crawler. For instance, if the current websense includes the rule "if it is lunch time, then humans eat", it would be useful to learn more about the notion of "lunch time", and the conditions under which this notion is activated. This process corresponds to an abductive type of

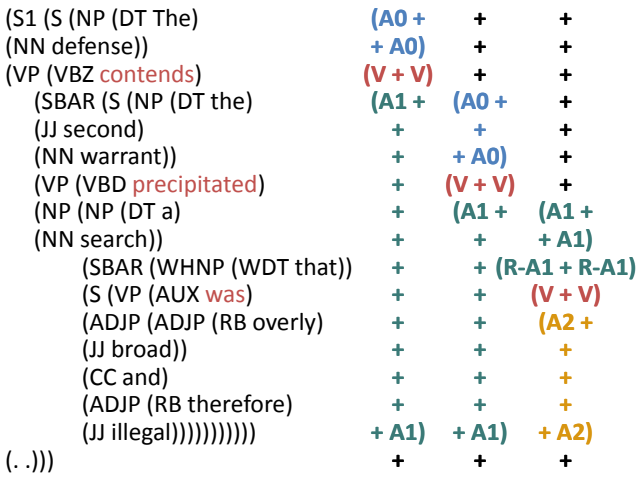

Figure 3: Parsed sentence using NLP tools.

reasoning, with the engine trying to acquire knowledge that would help it justify to draw a particular inference ("humans eat", in this case). The interaction of abduction and induction has been investigated (Ray 2009), and the techniques found therein could be employed to further improve the crawler.

If a crawler tends to favor certain pages (perhaps because they have more incoming links than others), then the knowledge acquired will likewise be biased. This is of no concern, and is well-accommodated by the definition of "websense", as long as the bias is due to the structure of the Web itself, and not due to a serendipitous design bias in the crawler.

Text Parsing

Text parsing is depicted in the top-center and top-right parts of Figure 1. These two components strip web-pages from any non-textual information, parse each individual sentence (using parsers, lemmatizers, POS taggers, semantic analyzers, etc.), and convert a subset of that information (most notably, verbs, nouns, and parts-of-speech) into certain existentially quantified conjunctions over predicates, in the spirit of earlier work (Valiant 2000; Michael and Valiant 2008).

Figure 3 illustrates the parsing of a single sentence, showing a syntactic tree that encodes the structure of the sentence, and semantic information indicating which parts of the sentence correspond to the arguments of each verb. From the parsed output the engine proceeds to extract facts that hold in the particular sentence, such as: $defense_{wrd}(2)$, $NN_{pos}(2)$, $precipitate_{sbj}(6)$, $precipitate_{sbj,obj}(6,9)$, $the_{wrd,+2}(2)$, $precipitate_{wrd,-2}(9)$, $JJ_{pos,-1}(6)$, indicating the presence of certain entities in the sentence (here represented by the numbers in the parentheses), and certain relations that hold between these entities. Such collections of relations over entities are called "scenes" (Valiant 2000; Michael and Valiant 2008). Thus, entity 2 is characterized by the word "defense" and the part-of-speech "NN", entity 6 is the subject of the verb "precipitate", and the word "precipitate" appears two places before the position of entity 9 in the given sentence.

Entities in this scene have a dual interpretation. They correspond both to objects pointed to by words (e.g., entity 6 in the third relation is treated as a concrete real-world object, namely the warrant mentioned in the sentence), but also to words as objects themselves (e.g., entity 6 in the last relation is treated as the word "warrant" at the sixth position in the sentence, and the relation $JJ_{pos,-1}(6)$ states that "warrant" is preceded by a word whose part-of-speech is "JJ"). This treatment of entities has been shown to be useful in acquiring knowledge from text (Michael and Valiant 2008).

Finally, composite expressions of the facts in each scene are created, such as: $warrant_{wrd}(6) \wedge precipitate_{wrd,+1}(6)$, $precipitate_{sbj,obj}(6,9) \wedge warrant_{wrd}(6) \wedge RB_{pos,+3}(9)$, $\exists v : precipitate_{sbj,obj}(6,v)$, $\exists v \exists u : precipitate_{sbj,obj}(v,u)$. Collections of such composite expressions correspond to the inputs available to the engine's learning module, and such composite expressions are what end up being the formulas in the bodies of the rules found in the knowledge base.

Rule Learning

Rule Learning is depicted in the bottom-right part of Figure 1. Our current design adopts the Winnow algorithm for learning linear thresholds (Littlestone 1988), appropriately extended to a relational setting (Michael and Valiant 2008).

For each one predicate designated as a learning target, the learning process followed proceeds roughly thus: To create negative learning examples for the target (which do not occur naturally / often in the constructed scenes), randomly selected grounded instances of the target are assumed to be false, in a manner that roughly balances the number of positive and negative instances across all scenes. For each (positive or negative) instance of the target, all remaining predicates and composite expressions in the scene are considered as learning features. To ensure that the learned rule's body does not include free variables other than those in the rule's head, only features that do not reference entities other than those in the target instance are kept as active. Following the Winnow algorithm, the current version of the rule for the target is applied on the active features to make a prediction. The weights associated with the active features in the rule are demoted or promoted by a multiplicative constant, if the prediction is a false positive or a false negative, respectively.

Despite the expressivity of linear thresholds as rules, and the noise-resilient and attribute-efficient algorithm used for learning them (Littlestone 1988), experience has shown that learning produces rule bodies with many formulas that have rather small weights. Pruning and retraining the rules (to avoid affecting their accuracy) leads to significant logistic overheads. It is natural, then, to consider other classes of rules that would support more succinct representations. A natural candidate for this investigation would be the class of formulas in Disjunctive Normal Form (DNF). Although the general class is intractable to learn (Valiant 1985), its (still rather expressive) subclass with constant-sized terms is known to be learnable. Using DNF formulas will also lead to knowledge bases whose rules are syntactically and semantically closer to the type of knowledge considered in many works in logic-based knowledge representation and reasoning, and such rules could be readily used by existing reasoning tools, without the need for new reasoning algorithms.

Rule Reasoning

Rule reasoning is depicted in the bottom-center part of Figure 1. This component draws inferences by applying a relational knowledge base on a set of input predicates. That is, each rule in the engine's knowledge base is applied exactly once directly on the set of input predicates determined by the user query, and the set of the rule heads that are found to be true comprises the inferences of the engine on that input.

If the example rule of the preceding section were applied on a scene comprising $share(1, 2)$, $open(3, 4)$, $scan(5, 6)$, $roque(5)$, then the reasoning process followed would infer $file(2)$ and $file(4)$. Note that even though the scene does not explicitly include $\exists v : scan(v, 6) \wedge rogue(v)$, this composite expression is still considered as an active feature in the scene. Therefore, the given scene would have triggered the rule to infer $file(6)$, had the weight 0.962710 in the first line of the rule body been higher than the rule's threshold.

It was stated earlier that a linear threshold rule infers its head to be true if sufficiently many formulas in its body are made true in a scene, and infers its head to be false otherwise. Note, however, that the "otherwise" part is ill-defined, or at best ambiguous, when a formula's truth value may remain undetermined in a scene. Indeed, a grounded predicate not appearing in a scene is not the same as its negation appearing in the scene, and a three-valued logic is more appropriate in the context we consider than a binary-valued logic (Michael 2009). Under such a three-valued logic, then, while a scene may often offer sufficient information for a rule to infer its head to be true (as in our example above), it is rare that it will offer sufficient information (i.e., enough negated predicates) for a rule to infer its head to be false. Typically, thus, rules end up either predicting positively, or abstaining.

Their logic-based representation notwithstanding, linear threshold rules cannot be directly handled by typical reasoning algorithms. As a result, we have developed a new Prolog interpreter able to cope with this more expressive representation when reasoning. Each rule is encoded in Prolog as an implication, and since at most one such rule is available for each predicate, Prolog's closed world assumption effectively yields the desired treatment of the rule as an equivalence.

Unfortunately, standard Prolog implementations seem unable to computationally cope with the number and size of the available rules. Thus, we found it necessary to re-implement the reasoning component of the engine in a typical procedural language, using advanced data structures and techniques to speed the process up. More work in this direction will clearly be necessary as the engine scales to larger contexts with more predicates. The consideration of DNF formulas as the underlying representation of rules is expected to alleviate considerably the computational cost of reasoning.

Of interest is our choice to apply each of the rules populating the knowledge base only once. At first it may seem utterly obvious that the repeated application of rules, so that they take into account the conclusions drawn by other rules (as done in Prolog programs), will lead to improved performance (in terms of the completeness of the inferences). This is not, however, direct. The rules in this context are not arbitrary, but are learned as definitions of their head predicates. It is not, therefore, at all clear why each rule would not already encode the necessary and sufficient conditions under which its head predicate should hold, in a way that would render the chaining of rules superfluous (Valiant 2006).

Despite the argument above, it is possible to formally establish (Michael 2008; 2009) that chaining is indeed beneficial. But to properly reap the benefits of chaining without sacrificing the guarantees provided by the principled learning process, one has to interleave learning and reasoning in a computationally demanding manner. This interleaving effectively boosts the completeness of the predictions, in the sense that the set of learned rules as a whole abstains less often than without chaining. Although the implementation of our learning and reasoning components supports such an extension (Michael and Valiant 2008), we have found it simpler to do without it in this first version of the engine.

Text Generation

Text generation is depicted in the bottom-left part of Figure 1. Our current design produces very simple sentences by combining a single predicate that was inferred, with at most two predicates from the user query. By way of illustration: *(i)* if $chase(x, y)$ and $cat(y)$ hold in the query, and $dog(x)$ is inferred, then the sentence "dog chase cat" is returned; *(ii)* if $student(x)$ holds in the query, and $clever(x)$ is inferred, then the sentence "student is (a kind of) clever" is returned.

Obviously one could do much more in terms of the natural language generation task that is considered here (cf. Figure 2), both in terms of making the already returned sentences more natural sounding (e.g., by properly choosing the word tenses, pluralities, etc.), and in terms of producing more complex sentences that take into account more predicates in the query and the inferences. Work in Natural Language Generation can play an important role in this direction (Reiter and Dale 2000; The Open Cognition Project 2010).

Evaluating the Engine's Inferences

The inferences returned by the search engine are elaborative, in the sense that they make claims about "implied" truths, and not claims about concrete and specified truths. How can one possibly evaluate such inferences fairly, since one does not have (and cannot have) access to some ground truth?

We have already mentioned that the engine acquires its knowledge through principled learning methods, building on the Probably Approximate Correct semantics (Valiant 1984; Michael 2010). In a sense this particular design choice already addresses, or rather side-steps, the problem of evaluation, since one can formally prove the appropriateness of the drawn inferences in an objective manner (Michael 2009).

Of course, there are many reasons why one would not be satisfied with this answer. For one thing, learning-theoretic models that offer formal guarantees do so under certain assumptions (e.g., that each training instance is drawn independently from some underlying probability distribution), which are not necessarily true in a given real-world setting. Furthermore, even if such assumptions were to hold, it is not clear why one would expect that the actual implementation of the learning algorithm faithfully (i.e., without any bugs) captures the workings of the algorithm. It seems, then, that the eventuality of an empirical evaluation cannot be avoided.

An empirical evaluation of the inferences of the engine would require the presentation of pairs of sentences — a user query and a returned response — and the rating of the latter as an appropriate inference of the former. Much like what is done in the Recognizing Textual Entailment (RTE) task (Dagan, Glickman, and Magnini 2006) and the Winograd Schema Challenge (WSC) task (Levesque 2011), human participants can be employed for this evaluation. Unlike what is done in the RTE and WSC tasks, where both the query and the response are available up front, our engine is given as input only the first of the sentences and is expected to generate the second one. Hence, human rating can be done only a posteriori, after the engine produces its responses.

A second distinction compared to the evaluation settings used for the RTE and WSC tasks is that the responses of our engine should be interpreted as *websense* inferences, and not as commonsense ones. Although in many cases the engine's inferences will also be commonsense inferences, the distinguished nature of the former may necessitate that the human evaluators be given precise instructions in this regard.

To date we have evaluated our engine on a small number of pre-selected user queries, and with a small number of human participants. The engine was trained only on web-pages that included the word "spyware", and this context was made clear to the evaluators. To mitigate the effect that the performance of the (currently rather simplistic implementation of the) text generation component would have on the evaluation of the inferences, we manually grammatically corrected the engine's responses, without affecting their essence. The outcome of this informal proof-of-concept evaluation was encouraging, with the evaluators acknowledging that the inferences were appropriate for the given queries and context.

Below are two example sets of queries and associated responses as given in the aforementioned evaluation study:

User Query A: *"members share something"*
Response A.1: *"something is an article"*
Response A.2: *"members share files"*

User Query B: *"people who steal from others"*
Response B.1: *"people are hackers"*
Response B.2: *"thieves steal from others"*

Given the first query as input (cf. Figure 2), the engine recognizes the presence of two entities, one having the property "member" and one having the property "something", and the existence of a relation "share" that holds on those entities. When rules are applied on the scene constructed from the query, the engine infers that the second entity also has the properties "article" and "file". Combining those inferences with the scene information, the engine constructs a first sentence to the effect that the entity having the property "something" also has the property "article", and a second sentence to the effect that the entity having the property "member" relates through "share" to an entity that has the property "file". Given the engine's training context, "member" is likely to refer to members of online forums or download servers, and thus the inference that they are sharing articles or files is a reasonable one. The engine treats the second query analogously, except now inferences are drawn for the first entity in the scene. Like before, the inference that those who steal are hackers is a reasonable one, given the training context.

Our next step is a large-scale systematic evaluation, staying, however, within a specific context (such as "spyware"). In one direction, we will seek to employ crowdsourcing tools like Amazon's Mechanical Turk to allow humans to evaluate pre-selected user queries and responses, or even to actively choose queries and evaluate the responses that are returned. In another direction, we will consider whether standardized corpora of stories or existing WSC instances can be used as a source of queries and expected responses, and investigate how our engine can be extended so that its inferences can be directly evaluated against what is already available.

Conclusions

The goal of developing machines able to employ common sense (McCarthy 1959) has been one of the early and continuing driving forces behind research in Artificial Intelligence. We have argued in this work that the means are now available to develop such machines in a purely autonomous manner, by allowing them to extract a commonsense-like type of knowledge from the Web, and to represent such *websense* knowledge in a logical form amenable to formal reasoning. A working prototype of an engine endowed with such abilities was discussed, through the lens of a novel web search engine able to respond to user queries with inferences that follow from the knowledge encoded collectively on the Web.

The benefits of building a machine able to extract and aggregate websense in a computable form are numerous:

(i) Research in Good Old-Fashioned AI (McCarthy 1959) assumes that knowledge in a computational form is already available, and seeks to investigate how such knowledge is to be reasoned with. Providing the means to actually acquire this knowledge in a robust and scalable manner will aid in fulfilling the goal of building machines with AI. Research in Multi-Agent Systems and Human-Computer Interaction seeks to build machines that interact naturally with humans, and endowing machines with websense would be a step in that direction. Research in Computational Models of Narrative seeks to understand stories computationally, and the availability of websense would offer an important resource in doing so (Michael 2013; Diakidoy et al. 2013). Progress on the Winograd Schema Challenge (Levesque 2011), or the full-fledged Turing Test itself (Turing 1950), could be made possible by allowing machines to draw websense inferences close (if not indistinguishable) to those drawn by humans.

(ii) Conceptually, building machines able to acquire and process websense would constitute a significant step towards realizing the goal of the Semantic Web, in a manner completely devoid of human intervention and the currently employed approach of manual tagging with meta-data, and with the financial gains that this automated approach may suggest. It would amount to a paradigm shift in viewing the Web no longer as a collection of information, but as a source of collective knowledge, representing the experiences, beliefs, biases, fears, and hopes of humans across the world.

(iii) The outcome of the websense acquisition task would lead to a massive knowledge base, which may reveal information about the collective human beliefs, prejudices, and

preferences (as stated across the Web), along with the sociological, philosophical, or economical implications that such information may have. This would facilitate large scale social studies, which, in the spirit of Unity of Science, would employ the scientific method to analyze, hypothesize, and evaluate using the available computable knowledge.

(iv) A solid framework for websense acquisition offers a tangible empirical task for evaluating machine learning techniques, suggesting new opportunities for the development of algorithms, with emphasis on scalability, parallelizability, and use of efficient data structures, as demanded by the explosion of web data. It would, in particular, offer an insight into how learning and reasoning with learned knowledge can fruitfully interact, with the derivation of plausible hypotheses of how this interaction is done in humans also.

One could argue that a state of affairs where the aforementioned benefits can be realized is scientifically within reach, and that it is, now, mostly an engineering task to build upon existing frameworks and prototypes, such as the engine presented herein, to reach that state. This task would likely require a well-orchestrated multi-disciplinary effort, involving researchers from Artificial Intelligence, Theoretical Computer Science, Web Technologies, Machine Learning, Natural Language Processing and Generation, Parallel Processing, and Database Management. But given the availability of the human resources, there seems to be little reason why McCarthy's vision of machines with common sense (McCarthy 1959) would not be a reality in the not-so-distant future.

References

Cognitive Computation Group. 2006. Semantic Role Labeler. University of Illinois at Urbana-Champaign, U.S.A. http://l2r.cs.uiuc.edu/ cogcomp/asoftware.php?skey=SRL.

Collins, M. 1999. *Head-Driven Statistical Models for Natural Language*. Ph.D. Dissertation, University of Pennsylvania, U.S.A.

Dagan, I.; Glickman, O.; and Magnini, B. 2006. The PASCAL Recognizing Textual Entailment Challenge. *Machine Learning Challenges* LNCS 3944:177–190.

Diakidoy, I.-A.; Kakas, A.; Michael, L.; and Miller, R. 2013. Narrative Text Comprehension: From Psychology to AI. In *Proc. of 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'13)*.

Etzioni, O.; Banko, M.; and Cafarella, M. 2006. Machine Reading. In *Proc. of 21st AAAI Conference on Artificial Intelligence (AAAI'06)*, 1517–1519.

Etzioni, O.; Cafarella, M.; Downey, D.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D.; and Yates, A. 2005. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence* 165(1):91–134.

Lenat, D. 1995. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM* 38(11):33–38.

Levesque, H. 2011. The Winograd Schema Challenge. In *Proc. of 10th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'11)*.

Liakata, M. 2004. *Inducing Domain Theories*. Ph.D. Dissertation, University of Oxford, U.K.

Littlestone, N. 1988. Learning Quickly when Irrelevant Attributes Abound. *Machine Learning* 2(4):285–318.

McCarthy, J. 1959. Programs with Common Sense. In *Proc. of Conference on the Mechanization of Thought Processes*, 75–91.

Michael, L., and Valiant, L. 2008. A First Experimental Demonstration of Massive Knowledge Infusion. In *Proc. of 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, 378–389.

Michael, L. 2008. *Autodidactic Learning and Reasoning*. Ph.D. Dissertation, Harvard University, U.S.A.

Michael, L. 2009. Reading Between the Lines. In *Proc. of 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, 1525–1530.

Michael, L. 2010. Partial Observability and Learnability. *Artificial Intelligence* 174(11):639–669.

Michael, L. 2013. Story Understanding... Calculemus! In *Proc. of 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'13)*.

Miller, G. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* 38(11):39–41.

Mitchell, T. 2005. Reading the Web: A Breakthrough Goal for AI. *AI Magazine*.

Ray, O. 2009. Nonmonotonic Abductive Inductive Learning. *Journal of Applied Logic* 7(3):329–340.

Reiter, E., and Dale, R. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

Schoenmackers, S.; Etzioni, O.; Weld, D. S.; and Davis, J. 2010. Learning First-Order Horn Clauses from Web Text. In *Proc. of 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP'10)*, 1088–1098.

Stork, D. 1999. The Open Mind Initiative. *IEEE Expert Systems and Their Applications* 14(3):16–20.

The Open Cognition Project. 2010. Natural-Language Generation Module. http://wiki.opencog.org/.

Turing, A. 1950. Computing Machinery and Intelligence. *Mind* LIX:433–460.

Valiant, L. 1984. A Theory of the Learnable. *Communications of the ACM* 27(11):1134–1142.

Valiant, L. 1985. Learning Disjunctions of Conjunctions. In *Proc. of 9th Joint Conference on Artificial Intelligence (IJCAI'85)*, 560–566.

Valiant, L. 2000. Robust Logics. *Artificial Intelligence* 117(2):231–253.

Valiant, L. 2006. Knowledge Infusion. In *Proc. of 21st AAAI Conference on Artificial Intelligence (AAAI'06)*, 1546–1551.