Learning Ontology by Reading: Scoring Candidate Knowledge

Names
Location

Abstract

This paper describes the configuration and evaluation of the scoring component of a system that learns ontological concepts, properties and value sets from unconstrained text. The experiment reported in this paper sought to determine the optimum combination of automatic and manual tasks in knowledge acquisition that maximizes the quality of knowledge acquired. This experiment was a follow-up on our earlier work where we sought to determine what quality of results could be expected from a fully automatic knowledge acquisition system. We briefly describe the system architecture, the experimental setup, results and evaluation. The paper concludes with an extensive discussion of complexities of ontology acquisition, whether carried out by people or systems, and a program of work that addresses these complexities.

Introduction

Automatic population of static knowledge resources (SKRs) holds promise for overcoming the so-called knowledge bottleneck of language processing systems. Both the process of developing such capabilities and the end resources are of great interest to our semantically-oriented NLP group, particularly since we have two types of enabling technologies that can be brought to bear: (1) large, deep, manually crafted SKRs (lexicon, ontology and fact repository) from which to bootstrap and (2) a semantic analysis engine that interprets input text such that meanings extracted from text, rather than text strings, can be learned.

The process of automatically enhancing the ontology in our environment, called XXXX (XXXX), is comprised of the following main steps:

1. Select words/concepts to be learned.
2. Compile a corpus.
3. Create text meaning representations (TMRs) for the corpus, which are written an unambiguous, ontologically-grounded metalanguage and contain the results of word sense disambiguation, semantic dependency determination and reference resolution.
4. Extract candidate property-value pairs from TMRs.
5. Score each property-value pair for utility and confidence.
6. Evaluate the learned knowledge.

To clarify what we mean by ontology (for a discussion of ontology classification and a review of automatic ontology learning efforts see Biemann 2005), the XXXX ontology is a hierarchically ordered inventory of concept frames in which the hierarchy reflects the IS-A relation and each concept belonging to the OBJECT and EVENT subtrees is described by an inventory of, on average, 16 properties, each of which can have multiple values. Property values are, by default, inherited from parents to children, though this inheritance can be overridden whenever necessary for differentiating concepts.

Ideally, all of above-mentioned stages of ontology learning would be carried out fully automatically, with results mirroring those achieved by a person carrying out the same task: the system would independently determine which information could be learned at a given time based on the current state of the SKRs and the content of the available corpora, and the depth and complexity of information would grow as the knowledge core for bootstrapping grew (XXXX and XXXX). However, achieving both full automation and fully acceptable results from the outset is beyond the current state of the art, making the organization and prioritization of the work over time both centrally important and quite challenging.

We have recently carried out two experiments that relaxed different aspects of the ideal, fully automatic, configuration. The first experiment, reported in XXXX, chose high automation at all stages over quality of results – and, indeed, the results were not stellar. In that experiment, the engine sought to learn all necessary lexical and ontological information about unknown words, including the number of senses of the word and the best position for each sense in the ontological hierarchy. Much effort was devoted to automatic sense discrimination; much noise was created by errors in the automatically generated TMRs that served as input for learning (carrying out word sense disambiguation and semantic dependency determination to perfection fully automatically is arguably the central long-term challenge for the field); and as a result it was difficult to evaluate the knowledge learned.

In the experiment reported here, we focused on optimizing the latter stages of the overall process – the scoring of candidate ontological knowledge and its evaluation – and agreed to manually supply certain prerequisites, such as ensuring that the TMRs that served as input to learning were correct and that they contained at least some knowledge that was sufficiently relevant to learn. Of course, the overall long-term objective is to introduce more automation – and successfully deal with greater amounts of ensuing noise – to different stages of the learning process, attempting to always optimize human-computer collaboration in SKR compilation, with the balance of effort shifting over time toward the automatic component.

To put this work in context, few, if any, extant ontologies are rich in property value descriptions; most essentially represent a subsumption hierarchy. Accordingly, as discussed by Biemann 2005, most ontology learning pertains exclusively to learning IS-A hierarchies with the occasional inclusion of meronymic (PART-OF) relations. XXXX (pp. 30-36) provides an overview of past ontology learning experiments, including those that go beyond the IS-A hierarchy. The work that is closest in spirit to ours is that being pursued by James Allen's group, who have recently begun a program of ontology learning using deep semantic analysis. In Allen et al. 2011, they report on an experiment designed to learn lexicon and ontology from glosses in WordNet. Their contribution, like ours, reflects as much an analysis of challenges as a report of results; but they, like us, come to the conclusion that this direction of work remains both necessary and potentially fruitful, despite those challenges.

The next section of the paper describes the second experiment mentioned above, and the final section discusses in some detail lessons learned from this pair of experiments as well as the overall place of, and contribution to, AI of this program of study.

The Learning Experiment

Coverage. This experiment addressed two words: *vaccine* and *coffee*, which were manually determined to have one and three senses worthy of inclusion in the ontology, respectively. The appropriate place in the ontological hierarchy for each new sense was also predetermined manually, as follows:

VACCINE	*is-a*	MEDICAL-PREPARATION
COFFEE-FOODSTUFF	*is-a*	PLANT-DERIVED-FOODSTUFF
COFFEE-CROP	*is-a*	CROP-PLANT
COFFEE-BEVERAGE	*is-a*	HOT-BEVERAGE

Each concept in the XXXX ontology is richly described by property values – an average of 16 per concept. A concept by default inherits the full inventory of property values from its parent, which is why its position in the hierarchy is of key importance from the point of view of economy of acquisition effort. The goal of this ontology learning experiment was to modify at least some property values of the newly posited children such that each child differed from its parent (and siblings) in correct, distinguishing ways. This experimental setup was actually quite close to the real-world scenario we seek to support: most of our existing concepts, although manually acquired, are not as well specified as they could be due to lack of acquirer time, meaning that they are not optimal *as defined by their inventory of property-value pairs*.

Corpus. For each word of interest – *vaccine* and *coffee* – we complied a corpus of sentences, each of which contained at least one description that might be of interest to the learner (e.g., *Many people like coffee* gives no useful information about the meaning of *coffee* since one can like just about any object or event in the world). The *vaccine* corpus contained 26 sentences and the *coffee* corpus, 58 sentences. The size of the corpora was small and the content carefully selected because the experimental design involved manually creating gold-standard TMRs for each sentence, which is a labor-intensive process, even given that we benefited from the availability of a convenient tool environment, XXXX. For orientation, the TMR for the toy input *Hot coffee bought in a cafe is delicious but expensive* is as follows (values of abstract scalars are on the scale {0,1}; numerical suffixes on concepts indicate instances):

COFFEE-1	
TEMPERATURE	.8
GUSTATORY-ATTRIBUTE	1
COST	.8
THEME-OF	BUY-1
BUY-1	
THEME	COFFEE-1
LOCATION	CAFE-1
CAFE-1	
LOCATION-OF	BUY-1

For the *vaccine* texts, the TMRs contained 28 property-value pairs, 18 of them unique; for the *coffee* texts there were 138 property-value pairs, 63 of them unique. Since only one sense of VACCINE was posited, all vaccine-related property-value pairs were tested against this sense. By contrast, since three senses of *coffee* were posited, each coffee-related property-value pair was tested against each posited sense of "coffee": COFFEE-FOODSTUFF, COFFEE-CROP, COFFEEE-BEVERAGE. To put a finer point on it, the TMRs for "coffee" texts contained an unspecified concept called COFFEE, and it was necessary to automatically determine which of the three actual senses was being described in each case.

After the inventory of TMRs was prepared, property-value pairs were extracted from each frame headed by

VACCINE or COFFEE. These served as input to the learner. A sampling of property-value pairs extracted from TMRs concerning "coffee" is shown in the first two columns of Table 1 (Columns 3 and 4 will be described later).

Table 1. A simplified rendering of the GUI used for manual evaluation of candidate property-value pairs.

Property	Value	Score	Status
THEME-OF	INGEST	4.69	Keep
THEME-OF	COMMERCE-EVENT	2.81	Keep
HAS-OBJECT-AS-PART	ORGANIC-CHEMICAL-COMPOUND	1.13	Keep
THEME-OF	PREPARE	.625	Edit
PART-OF-OBJECT	CROP-PLANT	.625	Delete
AMOUNT	CUP	.625	Keep

Scoring. The learner subjected each property-value pair, as applied to each candidate sense, to the following eight scoring functions, which were invented introspectively.

1. *Baseline scorer*. All property-value pairs receive a score of 0.5 (on a scale of 0-1) except those belonging to a stop list, which are penalized to .35. The stop-list members primarily represent elements of TMR that reflect *text* meaning rather than *ontological* meaning. For example, modal frames indicate speaker attitudes and set frames indicate plurality/cardinality, none of which is relevant for the ontological description of concepts.

2. *Specific-instance penalty scorer*. Information describing *generic* types of objects and events (*Lions roar*) is most useful for ontology supplementation, whereas information describing object/event *instances* (*That lion is barking*) is less desirable because it might be atypical or counterfactual. Accordingly, property-value pairs describing specific instances are lightly penalized.

3. *Wrong domain for case-role penalty*. Direct case roles (AGENT, THEME, etc.) apply to EVENTs (i.e., their DOMAIN is EVENT) whereas indirect case roles (AGENT-OF, THEME-OF, etc.) apply to OBJECTs. If the candidate property is supposed to apply to an EVENT but the target concept to which it is being applied is an OBJECT, or vice versa, a penalty is issued. For example, if [THEME: NEST] were being tested against a target concept meaning a type of BIRD, there would be a penalty since BIRDs cannot have THEMEs.

4. *"Corefer" scorer*. COREFER is a property indicating the ontological type of the entity heading a TMR frame. For example, the following TMR description reflects the meaning of a text input like *Coffee is a brown beverage*:

```
COFFEE
    COLOR      brown
    COREFER    BEVERAGE
```

When the system attempts to determine which sense of "coffee" [COLOR: BROWN] applies to, it checks to see if COFFEE-BEVERAGE, COFFEE-FOODSTUFF or COFFEE-CROP is a descendant of BEVERAGE. COFFEE-BEVERAGE *is* a descendant of BEVERAGE whereas the other two are not. Accordingly, [COLOR: BROWN] as applied to COFFEE-BEVERAGE receives a bonus, whereas [COLOR: BROWN] as applied to the other two senses receives a penalty.

5. *Higher Specification Scorer*. Given a candidate property-value pair, if the value is an ontological descendant of the target concept's initially recorded value (which is directly inherited from its parent), then a bonus is awarded. For example, if the initially recorded value of the property LOCATION is PLACE, and the candidate property-value pair is [LOCATION: FARM], then that property value will receive a bonus because FARM is an ontological descendant of PLACE.

6. *Instance Count Scorer*. The more times a given property-value pair is attested in a corpus, the bigger the bonus that property-value receives as ontological knowledge (as applied to *some* sense).

7. *Ontological Depth Scorer*. Property values that occupy a "medium-depth" position in the ontology (between 5 and 10 levels down from the root, ALL, for OBJECTS, and between 4 and 10 levels down for EVENTS) receive a bonus; those near the root of the tree (4 levels down from the root for OBJECTS; 3 levels down from the root for EVENTS) receive a penalty; those very low in the tree (very specific) have no effect on scoring. The intuition is that highly generic fillers will not be very useful in distinguishing one concept from another, whereas highly specific ones might represent idiosyncrasies of the input text rather than ontologically valid generalizations.

8. *Selectional Constraint Scorer*. This scorer applies a bonus to candidate property values that corroborate (are equal to or in the subtree of) those inherited from the target concept's parent, and it penalizes candidate property values that conflict with those inherited from the parent. There are two levels of penalty. A moderate penalty is issued if the candidate filler is not identical to or within the ontological subtree of the initial filler: e.g., if initial ontological specification includes [THEME: AUTOMOBILE] but the new information in the TMR includes [THEME: INGESTIBLE] there will be a penalty because INGESTIBLE is neither identical to nor in the subtree of AUTOMOBILE. A large penalty is issued if the above condition holds *and* at least one of the originally specified ontological fillers is "broad" (near the ontological root), since broad fillers are expected to cover a wide variety of specific cases met with in text. For example, if the initial ontological specification includes [THEME: PHYSICAL-OBJECT] but the new information in the TMR includes [THEME: MOTION-EVENT], then there will be a large penalty because of the violation of a very broad inherited constraint.

Although the *baseline scorer* must be applied first, all other scorers can be applied in any order. The actual scoring function was based on introspection and tweaked somewhat during testing. It is understood to be preliminary, requiring additional evidence-based modification.

Based on output scores {0,1}, the learner assigned one of the following three statuses to each property-value pair as it was applied to each candidate sense of the root word:

Keep: a high-confidence vote that this property-value pair belongs to the candidate concept.
Edit: a high-confidence vote that this property-value pair is close to correct for the candidate concept but fails in one of two ways: either the value is an ontological sibling of the needed concept or it is up to two levels of subsumption away from the needed concept. The idea is that this knowledge might be confident enough to be included in the ontology even without amendment (as in a fully automatic, lifelong learning system), but it would be better if a person – or further machine learning – would revisit it for further optimization.
Delete: a high-confidence vote that the given property-value pair does not belong to the candidate concept.

A GUI was created to permit users to view and edit the system's recommendations, as well as add property values, if desired, to create the gold standard (more on the definition of "gold standard" below). The basic contents of the GUI, mocked up for reasons of space, is shown in Table 1.

Evaluation. System evaluation introduced experiment-motivated enhancements to the well-known measures of precision and recall.

Precision. Since two levels of correctness were delineated – "keep" and "edit" – precision was calculated as follows:

P = (#keep + (#edit x PENALTY)) / #suggested

where

- **#keep** means "System vote: Keep or Edit ~ User vote: Keep"
- **#edit** means "System vote: Keep or Edit ~ User vote: Edit"
- **PENALTY** is a static value used to penalize precision for the property / fillers that were added to #edit
- **#suggested** is the number of property-value pairs the system originally marked as Keep or Edit.

For example, our targeted word sense COFFEE-CROP contained 12 distinct property-value pairs that the system marked as Keep or Edit. Of these, 4 were marked by the user as Keep and none as Edit. The resulting precision was 0.333.

Recall. Recall can be defined in two ways, which we will refer to as basic recall (Rb) and total recall (Rt).

Basic recall (Rb) indicates how many property values were learned of the number of property-value pairs that could have been learned given the corpus. The formula, which considers the Keep and Edit statuses of property-value pairs, is:

Rb = #userandsystem / #user

where **#user** means the number of property-value pairs that the user labeled as either Keep or Edit, and **#userandsystem** indicates the number of property-value pairs that both the user and system labeled as Keep or Edit. Continuing our COFFEE-CROP example: the user labeled only 4 property-value pairs as Keep or Edit, all of which were marked by the system as Keep or Edit as well. Thus Rb for the corpus for this word sense was 1.000. This measure suggests how well the system can help a user to carry out system-aided ontology development by suggesting candidate property-value pairs.

Total recall (Rt) includes a penalty for knowledge that should be in the gold standard but the system could not have learned given the input corpus (the knowledge was absent from the texts). This is calculated using property-value pairs that the user added by hand during the process of reviewing system results. To calculate Rt, we need one additional parameter: **#added**, which indicates the number of property-value pairs the user added to the candidate concept. Rt is then defined as:

Rt = #userandsystem / (#user + #added)

To conclude our COFFEE-CROP example, the user added one additional property-value pair not found in the corpus to the candidate ontological frame. Thus, Rt for this word sense was 0.800.

Finally, we can calculate the standard fmeasure score for both recall values:

Fb = 2 x ((P x R) / (P + R))
Ft = 2 x ((P x Rt) / (P + Rt))

Rt has an upper bound of Rb, and similarly Ft has an upper bound of Fb.

Table 2 indicates the precision, basic recall, total recall, and both fmeasures (basic and total) for each of the four word senses learned in this experiment. We interpret these results in the next section.

Table 2. The results of property-value learning.

	P	Rb	Rt	Fb	Ft
COFFEE-FOODSTUFF	0.294	0.625	0.555	0.400	0.384
COFFEE-BEVERAGE	0.449	0.750	0.500	0.562	0.473
COFFEE-CROP	0.333	1.00	0.800	0.500	0.470
VACCINE	1.00	0.250	0.250	0.400	0.400

Interpretation of Results and Future Work

We did not expect the precision and recall results of the reported experiment to be as low as they were: after all, we had intended to optimize learning results by hand selecting texts that contained useful property values, creating gold-standard TMRs, deciding how many senses each word would have, and selecting an ontological position for each candidate concept, thus optimizing its inherited inventory of property-value pairs. Given all of these prerequisites, the work was intended to focus narrowly on optimizing the automatic scoring function for candidate knowledge elements. We knew from the outset that the absolute scores would be of little interest since the experimental setup (like so many others) had little relation to any actual or envisioned real-world task. However, the uniformly low quality of results led us to contemplate a larger – and, we would suggest – ultimately more important set of issues than originally anticipated. In fact, we have come to believe that the main contribution of this paper is precisely the analysis of the issues and problems inherent in this and other similar experiments. We begin with experiment-specific lessons learned, then broaden the discussion to the basic scientific and methodological issues facing practitioners of lifelong learning by reading.

Modifying the Experimental Setup. During evaluation, we detected three aspects of experimental design that could improve the results of this or a similar experiment – again, focusing specifically on optimizing the scoring function.

1. The ontological descriptions for the concepts that served as parents for our new concepts were simply retrieved from the standing ontology and not manually rechecked before the experiment; as it turned out, they were actually of suboptimal quality – essentially, underspecified due to lack of acquirer time/attention. This led to scoring problems for all heuristics that compared the parent's value of a property with one attested in the learning corpus.

2. Since we chose to supply the learner with gold-standard TMRs as input, and since it is expensive to create gold-standard TMRs – even when supported by aspects of automatic analysis and a sophisticated acquisition environment – we agreed to learn from a limited corpus. The corpus proved to be too small to provide sufficient evidence to optimize the scoring function. One option for a future experiment would be to create gold-standard TMRs for relevant *excerpts* of sentences rather than full sentences. We estimate that this might increase fourfold the amount of data that could be produced given a set amount of acquirer time.

3. We hypothesize that scoring might be improved by merging several of the scorer functions related to the ontological nature of candidate property values. There are two reasons for this. First, psychological studies have shown that people cannot manipulate large numbers of variables in decision-making, and that small numbers of well-selected ones tend to work better (Kahneman 2011), at least in routine cases. Since we are creating our scoring function using human introspection, constraining the number of property values should be beneficial. The second reason to merge several scorers is that some of the current scorers were not truly independent. They inadvertently overlapped with respect to some phenomena, imposing disproportionate bonuses or penalties. For example, if a candidate property value was of type OBJECT or EVENT, the *baseline scorer* penalized it for being high on the ontological tree by giving it a starting score of .35, then the *ontological depth scorer* penalized it again for the same reason; similarly, both the *higher specification scorer* and the *selectional constraint scorer* evaluated whether or not a property value corroborated or conflicted with the initially recorded ontological property values – a calculation that could readily be merged into one scoring system. In sum, we hypothesize that it would be useful to merge several scorers relating to the ontological nature of candidate property values and their relationship to their respective recorded property values existing before learning occurred.

4. We could easily expand the coverage of some of the scorers to include additional relevant phenomena. For example, the *wrong domain for case-role scorer* could be expanded to cover non-case-role relations, such that the domain for any relation was tested. Similarly, the *higher specification* and *selectional constraint scorers* (or a merged variant of them, as suggested above), could be applied to scalar properties as well as relations: e.g., if a parent is defined for [TEMPERATURE: 30 <> 90] and the corpus evidence suggests that the target concept's TEMPERATURE is 40 <> 80, then the latter is a higher specification of the former.

Expanding This Experiment By Automatically Deriving Prerequisites. The most conspicuous aspect of the selected experimental setup was the extent to which we permitted the prerequisites for learning to be provided manually. This goes against our group's overall research and development methodology, which strongly prefers – even dictates – that we take responsibility for *all* aspects of text processing ourselves: preprocessing, syntactic analysis, semantic analysis, compilation of knowledge resources and development of user tools (XXXX). The reason for making prerequisite-oriented compromises in this experiment was the desire to evaluate, as cleanly as possible, one module of our overall system: the scoring mechanism. We expect that the next iteration of this experiment will have improved that sufficiently so that we can begin to iteratively replace manual efforts with automatic ones.

The most important shift to automation involves the automatic generation of TMRs – a process that has been at the center of our group's work for over 20 years. Clearly, automatically generated TMRs will not soon be of perfect quality due to the complexity of the enterprise and the amount of static knowledge required to support it (one of the central motivations for machine learning of ontology!). However, we are working toward endowing the system with the ability to *self-evaluate its results* such that it can select high-confidence portions of TMRs as input to learning. Once we shift to automatically generated TMRs, the corpus need not longer be constrained in size or specially selected.

Clearly, improving any aspect of text processing should improve overall learning results, but we can prioritize development efforts based on the needs of the learner and our theory of scoring candidate knowledge, manifest through the inventory of scorers – which represent the inventory of features we choose to target. For example, one of our scorers – the *specific-instance penalty scorer* -- requires as input the determination of whether a given bit of knowledge in text applies to a class (*Lions roar*) or an instance (*This lion barks*). Whereas for this experiment the generic/specific-instance distinction was made using light, text-based heuristics (e.g., "this X" is an instance), in actuality, the generic status of an object or event should be explicitly recorded in TMR, having been determined using a battery of heuristics that goes far beyond the presence or absence of a given determiner. The difficulty in determining generic/specific status can readily be seen in the following dialog: - *Dogs eat cat food. - No they don't, they eat dog food! - Ugh ugh, my dog will only eat cat food!*

Some Big Issues. Here, to our minds, is where the discussion becomes really important and relevant to the field as a whole. Our experiment, like most, was designed to shield us from excessive complications as we whittled away at one corner of a very large problem. However, the complications so doggedly asserted themselves during evaluation that ignoring them would be untenable. Below we present brief discussions of complex issues that we believe must be addressed head-on by anyone pursuing automatic learning of deep ontology (i.e., ontology that includes properties and values) by reading.

1. *Ontological hierarchy & inheritance.* Most ontologies are organized as subsumption hierarchies, with concepts inheriting property-value pairs from their parents unless locally overridden: e.g., BLUEBIRD's COLOR is blue, whereas the value of color of its parent, BIRD, is a set of different colors. Ontological inheritance causes many practical acquisition problems, be it carried out by a person or a system. If acquisition can be carried out in an exclusively top-down fashion – where perfecting the description of a child is undertaken only after its parent is deemed to be described sufficiently precisely – then most problems of inheritance can be avoided. Realistically, however, strictly top-down acquisition is impossible to pursue. As a result, every time a modification to a property value is considered, the question arises of whether this modification should be carried out at locally or, instead, applied to the parent (or the grandparent...) and subsequently inherited in the normal way. For example, if a text says that bluebirds are blue, rather than editing the BLUEBIRD frame, the acquirer or system should see if the parent is, by chance, a class of ALL-BIRDS-THAT-ARE-BLUE. This is a contrived example, but it makes the point: acquisition should always take into account ontology as a whole and not be reduced to acquisition of information about a concept in isolation. This consideration strongly influences control issues in the acquisition process.

2. *What is a gold-standard ontology frame?* Any answer to this question would depend upon the demands of an application. In principle, our BLUEBIRD frame could contain all of the information in a specialist's tome about bluebirds, including all of the scripts (complex events) that a bluebird participates in, its properties at all of its life stages, etc. If we cannot ever say that a concept description is finished, then how can we evaluate a learning experiment with respect to a gold standard – what was considered "total recall" in the evaluation reported above?

3. *Generalizing over attested knowledge.* In the experiment reported here, we did not attempt to merge attested property values into larger classes: e.g., if the learner had evidence that coffee was [THEME-OF: EXPORT] and [THEME-OF: TRADE], it did not merge EXPORT and TRADE into their common parent, COMMERCE-EVENT; doing so would have also implied that coffee was the THEME-OF IMPORT, PRICE-FREEZE, SUBSIDIZE and a number of other events, which might or might not be true. Clearly, judicious merging of specific concepts into a common subtree is useful and necessary, but the set of relevant merging heuristics remains to be developed and tested.

4. *Task-oriented evaluation.* Isolated, non-real-world experiments are useful, at most, for comparisons with similar experiments but say little about the potential real-world contribution of a theory, approach or system. Our long-term program envisages starting to use our learning by reading system in the near future to support system-aided manual acquisition of static knowledge resources; then, over time graduating first to human-aided automatic acquisition and, finally, to fully automatic acquisition. This means that, in the near term, we expect the system essentially to reduce the time and effort needed for manual acquisition by proposing to acquirers knowledge (extracted from a corpus) that is already recorded in the human-understandable ontological metalanguage. Using the interface similar to that illustrated in Table 1, users will approve of, edit or reject knowledge gathered from a corpus which, we hypothesize, will take much less time than manually reading the corpus, determining how to record the knowledge using the formal metalanguage of the ontology, and actually recording it. This aspect of evaluation, which measures time saved, is much more cumbersome than the evaluation provided above, but will reflect real-world utility far better than any evaluation of machine learning in isolation, since we will anytime soon not expose our relatively high-quality ontology to unvetted machine learning results.

The most difficult research problems do not lend themselves to the kind of regular, satisfying evaluations achievable for more constrained problems, with the definition of "useful evaluation metric" presenting a quandary in itself (XXXX). Still, the problem of automatic acquisition of rich knowledge remains the single most important problem in the field, which justifies ongoing attempts at solving it.

References

Allen, James, William de Beaumont, Nate Blaylock, George Ferguson, Jansen Orfan and Mary Swift. 2011. Acquiring commonsense knowledge for a cognitive agent. Proceedings of AAAI Fall 2011 Symposium on Advances in Cognitive Systems.

Biemann, Chris. 2005. Ontology learning from text: A survey of methods. LDV-Forum 2005 – Band 20(2): 75-93.

XXXX

XXXX

Kahneman, Daniel. 2011. Thinking, Fast and Slow. Farrar, Straus and Giroux.

XXXX

XXXX

XXXX

XXXX