Estimating Similarity among Collaboration Contributions

Kenneth Murray, John Lowrance, Doug Appelt, Andres Rodriguez
SRI International
Menlo Park, California, USA
{murray lowrance appelt rodriguez} @ ai.sri.com

ABSTRACT

The need for collaboration arises in many activities required for effective problem solving and decision making. We are developing Angler, a web-services tool that supports collaboration among participants on some focus topic. Angler overcomes some common barriers to collaboration by enabling asynchronous and distributed collaboration. Angler supports a collaboration methodology that exploits opportunities afforded by multiple participants each making contributions to the collaboration. One challenge that arises in helping participants manage their contributions and their review of others' contributions is determining when one contribution is very similar to another contribution. Two very similar contributions may suggest either a need to merge them or to further elaborate one or both of them. Indexes over the participant contributions are used to assess similarity across contributions and address this challenge. The indexes may comprise lexical or ontological information; the former indexes require fewer resources to deploy but the later appear to support better similarity estimates.

Categories and Subject Descriptors
I.2.1 Applications and Expert Systems
I.2.7 Natural Language Processing
I.7.5 Document Capture
I.7.2 Document Preparation
H.4.3 Communications Applications

General Terms
Algorithms, Experimentation.

Keywords
Knowledge management, information retrieval.

INTRODUCTION

Collaboration arises in many activities required for effective problem solving and decision making. We are exploring new approaches to facilitating effective collaboration. Angler is a web-services tool for supporting collective reasoning on some focus topic [1, 2].

K-CAP'05, October 2–5, 2005, Banff, Alberta, Canada.

Angler facilitates overcoming some of the common barriers to effective collaboration. It does so by allowing participants to be geographically distributed, while making asynchronous contributions. Angler supports a methodology for collective reasoning designed to facilitate cognitive expansion by using divergent (such as brainstorming) and convergent (such as clustering or ranking) thinking techniques.

Angler is organized around the idea of virtual (possibly hierarchical) workshops. Workshops are organized by a facilitator that brings a group of people together to accomplish a knowledge task. Simple workshops usually start in a brainstorming phase and then go through other phases, such as clustering the ideas from brainstorming, and then ranking the clusters. The facilitator for the workshop will enforce and manage the timelines. As more and more workshops are captured and stored in the Knowledge Base, a Corporate Memory is formed.

In the brainstorming phase, participants are asked to contribute "thoughts" (or ideas) to a focal topic. Participants submit thoughts as brief statements on some aspect of the overall topic; each such thought is authored as a small textual document, available for review by other participants. The thoughts of other participants are incrementally disclosed to each participant, so that he or she interleaves independent thinking with stimulation from others' ideas. Angler provides a simple interface for participants to author such thoughts, review and respond to the thoughts of others, and organize the growing set of shared contributions in meaningful ways.

After sufficient input and review, the facilitator moves the workshop into the clustering phase. The participants each organize all the contributed thoughts into clusters. These clusters suggest candidate ways to coalesce thoughts into coherent themes. The participants benefit by considering the emerging themes. This process promotes a rich interchange of ideas and perspectives, allowing each participant to express his or her own opinions pertaining to the themes, while learning from the opinions of others. Angler uses agglomerative clustering techniques [3] to calculate a candidate set of consensus clusters and determine how well aligned is the group vision. Finally, there is a consensus and ranking phase, where participants come to understand their differing views and vote on the final names for the consensus clusters.

Table 1: Example Angler "thought" Contributions

ID	Catch Phrase	Author	Description
1	Amount of fissile material	G	How much fissile material is in Country-X? In what form is it stored?
2	C2 apparatus	D	The nature of the nuclear command and control structure.
3	configuration	B	configuration of arsenal - assembled, disassembled, state of deployment
4	control over nukes	C	Does the new regime have effective control over Country-X's nuclear arsenal and facilities?
5	domestic audience	B	domestic public opinion
6	International Support	E	Level of US and international support for maintaining Country-X's nuclear security
7	new regime nuke policy	C	Will the new regime alter Country-X's existing nuclear policies concerning deterrence, etc.?
8	Nuclear Command and Control	G	Specific structures and systems; safeguards, command authority, designation
9	nuclear labs and military	B	domestic nuclear and military corporate interests
10	Personnel Reliability	F	Extent of radical Religion-Z leanings of nuclear C2 personnel, and/or level of corruption.
11	Physical security of nuclear arsenal	G	Permissive Action Links, Hardened sites etc.
12	quality and reliability of personnel	B	nature of Religion-Z among nucledar personnel
13	Regional Developments	A	Country-X, Country-Y and stability
14	Resource allocation	D	The inclination of the new leadership to allocate resources to nuclear safety and security measures (or alternatively, force expansion/enhancement).
15	Successor regime	A	Nature of the successor regime and its willingness to maintain strong positive control
16	Technical Reliability	F	Functionality of nuclear safeguard devices.
17	Threat level	E	Who wants Country-X's nukes, either to use or to sabotage?
18	Threat perceptions	D	The nature of the new leadership's threat perceptions and how it impacts nuclear decision making.
19	US contingency plans	C	Does the US have effective contingency plans to seize control of Country-X's nuclear arsenal to prevent it from falling into hostile hands?
20	US inputs	B	US inputs into safety and security of Country-X's arsenal
21	US intervention	A	US willingness to preempt should instability occur
22	Vulnerability to preemption	F	Extent of device core separation from firing technology given events surrounding succession.
23	Willingness to divert funds	E	State of Country-X's economy and willingness of govt to divert funds/budgetary allocation towards nuclear security

Angler has been used within several workshops (e.g., for scenario planning [4]) and has been shown to facilitate collaboration among participants in these workshops. Table 1 presents (slightly obfuscated versions of) some thoughts contributed during a workshop considering the consequences of a hypothetical political regime change in a country (Country-X) that has nuclear weapons. These are the 23 thoughts related to nuclear security issues. A total of 140 thoughts were contributed by the seven participants during this phase of the workshop. Each thought includes a pithy summarizing "catch phrase" as well as a concise description of the issue to be considered. Angler users face several challenges in managing thoughts during collaboration; four are:

- Similarity: detecting similar thoughts (i.e., candidates either for merging or for further elaboration).
- Coverage: appraising the topics considered and those not yet considered by an evolving set of thoughts.
- Collaboration: detecting further collaboration topics and opportunities among the participants based on their contributions.
- Semantic search: helping participants find those contributions relevant to some search topics.

In [1] we reviewed these challenges and presented preliminary work towards addressing them with a semantic index. In this paper we focus on the first challenge, presenting further development and evaluation of technology to support estimating similarity. Specifically, we have implemented term-weighting to better leverage the various indexing data used to assess similarity among Angler thoughts. Furthermore, we have gathered subjective assessments from humans to evaluate the performance of the automated assessments. The next section motivates the need to support Angler users with automated assessments of the similarity among thoughts. The following sections describe our approach to computing similarity and an initial evaluation of the approach.

Motivation

One challenge that arises in collaborative contexts such as Angler involves helping participants manage the often large number of thoughts that are contributed. A rich exchange of ideas among even a relatively small group of collaborators can quickly produce a volume of thoughts that may overwhelm some participants and impair their continuing participation, obscuring "the forest for the trees."

An important observation about such sets of thoughts is that they are not independent from each other, and semantic relations can be defined among them and used to organize them. Sometimes a thought contributed by one participant may be redundant with a thought authored by another participant (e.g., from Table 1, thought 8 nearly subsumes thought 2). Sometimes two thoughts share a significant amount of semantic content, and although one is not subsumed by the other, the two are good candidates for merging (e.g., from Table 1 thoughts 2, 4, and 8 or thoughts 19 and 21).

Establishing semantic similarity among thought contributions has two benefits. First, it helps the participants in reviewing the contributions of others and grasping how some contributions converge with or diverge from other contributions. Second, it helps the participants in authoring new thoughts by identifying which existing thoughts are most similar, so an author can consider either merging a new thought with a similar existing thought or can elaborate the new thought to clarify how it is distinct from similar existing thoughts.

TECHNICAL APPROACH

We are investigating the use of an ontology [5, 6] for establishing semantic properties of Angler thoughts. We can then use these properties to reason in simple but useful ways about the semantic content of the thoughts to address the four collaboration challenges described above. For example, a substantial number of the semantic concepts of thought 2 are also included in thought 8, suggesting a high degree of semantic overlap. Our approach includes a term extraction (natural language) module and an ontology (knowledge base) module.

Term Extraction

The term extraction is performed by the information extraction system, TextPro. TextPro was developed as a successor to the FASTUS [7] system, and like the earlier work, it is based on applying multiple finite-state grammars to the natural language text. Each finite-state grammar adds additional cumulative information about the text under analysis. The information is expressed by placing "annotations" on the text, which are essentially sets of attributes and values associated with spans of text. Each grammar is a transducer that accepts certain specified types of annotations as input, and contributes new annotations (or modifies existing annotations) as its output.

Annotations express the extracted information about the text comprising the catch phrases and descriptions of the Angler thoughts. For example, some annotations indicate words together with their lexical properties. Other annotations indicate which spans of texts are proper names, and attributes are added to classify proper names according to the types of entities to which they refer; currently, these types include persons, organizations, locations, and facilities. Still other annotations mark basic syntactic structure, including noun groups and verb groups. Base noun groups consist of nominal heads together with determiners and prenominal modifiers. Verb groups consist of main verbs together with auxiliaries and adverbs.

Although the architecture of the TextPro system is essentially rule-based, it employs various statistical models for various purposes, so it is really a hybrid system. Currently TextPro employs a rule-based name recognizer [8] to mark proper names, and then a maximum-entropy part of speech tagger is used to assign tags to the remaining words, assuming that names (possibly containing many unknown words) have been recognized. Another maximum entropy model, trained from the Penn Treebank, is employed for prepositional phrase attachment decisions. For example, a finite state rule is written to match a sequence of phrase chunks like [verb-group] [noun-group] [preposition] [noun-group] and the statistical model uses features based on the phrase heads, and the lexical semantics of those heads to determine whether the prepositional phrase is more likely to attach to the verb-group or the noun-group.

As TextPro analyzes a text, it keeps track of the entities that have been mentioned, and their classification as to type. Mentions of entities are judged whether they corefer, and are grouped into coreference equivalence classes. The coreference classification proceeds by heuristic methods when noun groups headed by proper names or common nouns are recognized. When pronouns are involved, a maximum entropy classifier based on morphosyntactic features derived from the sentences containing the potential antecedents is employed to identify the most likely antecedent [9].

For assessing similarity among Angler thoughts, the most important contribution made by TextPro is the identification of phrase heads. These words correspond to the onto-

logical concepts that best characterize what the text is about. TextPro is also capable of making hypotheses about the basic predicate-argument structure of the sentence, indicating "who did what to whom". This information can potentially be used for even more fine-grained characterization of the conceptual content of a text.

Ontology Module

The data extracted by TextPro are then mapped into concepts defined within the ontology; our ontology is implemented as a network of semantically related concepts. We then index each thought with the set of concepts from the ontology that correspond to words in the parse of the text for the thought. This index of concepts can be viewed as a thought-specific vector in the semantic space defined by the ontology.

Each concept in the ontology is related to *ancestor* concepts using formally defined relationships that denote properties such as "is a type of," "is an instance of," "is a part of," and "in region." Figure 1 presents a segment of the ontology that illustrates several relations used to compute ancestors. The ancestors of an Angler thought are the unions of the ancestors of the concepts of the thought.

Our ontology was created manually to cover the content of the example 23 Angler thoughts and relevant background knowledge, and it was engineered to be extendable to cover the content of new thoughts arising in new collaborations involving new domains. The ontology content has been significantly influenced by the Component Library [10], by Cyc [11], and by the Suggested Upper Merged Ontology [12]. KetL, the formal language used to represent the ontology, has been influenced by CycL [13] and KIF [14].

Our approach is similar in several respects to [15] except that rather than using a corporate thesaurus to provide semantic relations over terms used to index documents we exploit an ontology with simple but useful (e.g., transitive) relations. The ontology relations (e.g., in-region) go beyond the basic thesaurus relations (e.g., broader-term), but of course the ontology was manually engineered and not pre-existing or automatically extended as is the thesaurus in [15]. Unlike [6], our documents are small and an overly high-dimension space is not a concern.

Representing Thoughts

To support the simple ways in which our applications will reason over them, the representation of the Angler thoughts are annotated with terms that represent their contents. Figure 2 presents a part of the annotated representation of an Angler thought. The original text is augmented by six types of representation terms:

- TextPro tokens
- TextPro lexemes
- KetL concepts
- KetL ancestors

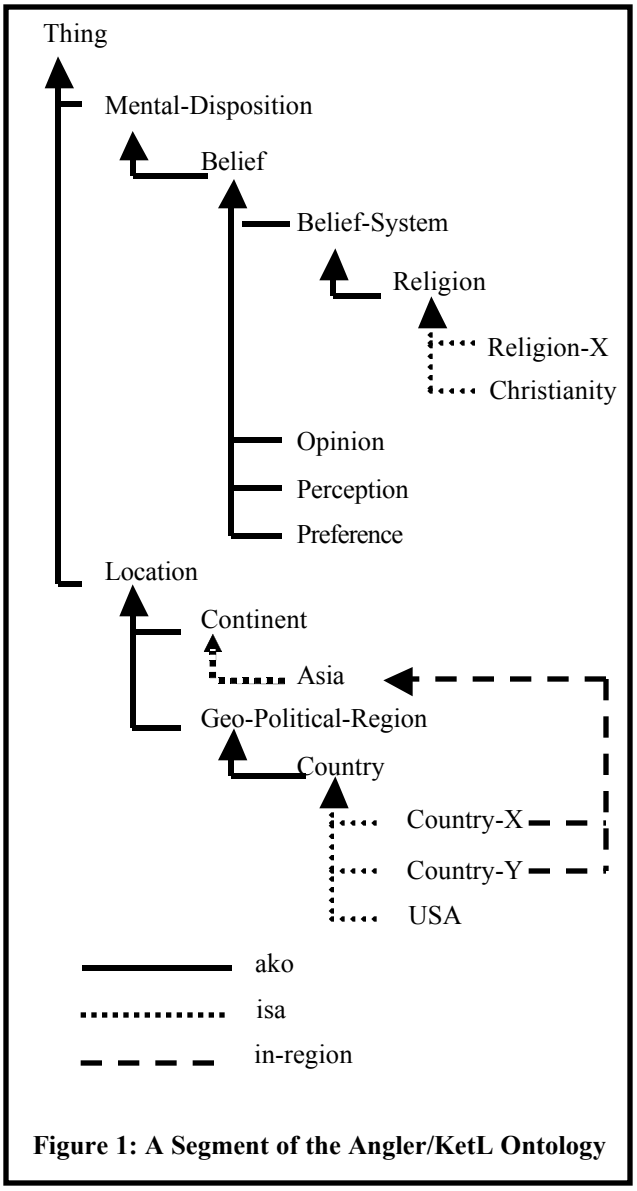

Figure 1: A Segment of the Angler/KetL Ontology

- WordNet concepts (synsets [16])
- WordNet ancestors

Each type of representation term is produced by one or more transformations of the thought textual content (the catch phrase text and the description text).

Tokens are the substrings of the original Angler thought text that TextPro deems to be treated as words. In extracting tokens from the thought text, TextPro handles issues such as punctuation, hyphenation, and apostrophes. The tokens are algorithmically extracted from text.

Lexemes are the standardized versions of words extracted as tokens. In converting tokens to lexemes, TextPro handdles issues such as case, tense, plurality, and proper nouns. Lexemes are entries in a manually engineered corpus of canonical word stems.

```
Angler-Force-Thought-2
  isa : ( Angler-Force-Thought )
  author : ( Author-D )
  catch-phrase-text : ( "C2 apparatus" )
  description-text : ( "The nature of the nuclear command and control structure." )
  textpro-term : ( "nuclear" "c2" "structure" "of" "apparatus" "nature" "the" "command and control" )
  textpro-token : ( "The" "nuclear" "C2" "structure" "of" "apparatus" "nature" "the" "command and control" )
  ketl-word : ( English-Word-nuclear English-Word-c2 English-Word-structure English-Word-of English-Word-apparatus
              English-Word-nature English-Word-the English-Word-command and control )
  ketl-concept : ( Nuclear-Technology Structure Device State-Description Command-and-Control )
  ketl-ancestor : ( Control Command Structure Technology Resource Device Cognitive-Resource Mechanism
              Interaction Nuclear-Technology Command-and-Control Artifact Knowledge Description Agent Agent-Role
               State-Description Authority )
  wordnet-concept: ( Wn-nuclear-adjective-576833 Wn-structure-noun-3431817 Wn-apparatus-noun-2200827
              Wn-nature-noun-3716446 Wn-control-noun-4045365 Wn-command-noun-5359574 )
  wordnet-ancestor : ( Wn-act-noun-17487 Wn-speech_act-noun-5354574 Wn-command-noun-5359574
              Wn-control-noun-4045365 Wn-abstraction-noun-13018 Wn-quality-noun-3714294 Wn-object-noun-9457
              Wn-power-noun-4041746 Wn-apparatus-noun-2200827 Wn-equipment-noun-2644452
              Wn-nuclear-adjective-576833 Wn-structure-noun-3431817 Wn-entity-noun-1740 Wn-attribute-noun-18604
              Wn-nature-noun-3716446 Wn-artifact-noun-11937 Wn-instrumentality-noun-2859872 )
```

Figure 2: Representing an Angler Thought

Concepts are terms defined in the Angler ontology. The ontology is expressed using the Knowledge Engineering Toolkit Language (KetL). The concepts of an Angler thought include those concepts associated with the lexemes extracted by TextPro from the textual content of the thought. The lexemes extracted from the thought texts are mapped into concepts by using a translation table that is constructed as part of the knowledge engineering effort that produces the ontology. The mapping occurs via the "English-Word" concepts (see Figure 2). For example, English-Word-nuclear explicitly relates the word "nuclear" with the ontology concept Nuclear-Technology; the term and the relations are created by asserting:
(denotes (English-Word "nuclear") Nuclear-Technology).

The ancestors of a concept include those terms in the ontology that are in some way implied by the concept. For example, Weapon is an ancestor of Pistol, and Africa is an ancestor of Egypt.

The WordNet concepts are simply terms created in the ontology that correspond to relevant WordNet synsets. The WordNet concepts of an Angler thought are precisely those whose synsets include any lexeme extracted from the textual content of the thought. (Note, for presentation in Figure 2, only the first synset for each lexeme is included.)

The WordNet ancestors include terms in the ontology corresponding to all synsets accessible going up the hierarchies along the transitive relations in WordNet (e.g., hypernym, entailment, etc.).

Table 2 summarizes the number of terms required to represent the 23 thoughts for each of the different representation types. The data in Table 2 show that in the representation of these 23 thoughts there is a natural compaction that occurs during the transformations that produce the first three of the six types of representation terms: on average, the 12 tokens appearing in a thought is reduced to 8.5 concepts. However, on average, about four additional ancestors are added to the representation of a thought for each concept. In the presence of a very large corpus of thoughts, the ancestors may be efficiently computed when needed rather than stored for each thought as presented in Figure 2.

Table 2: Representation Use Summary

	min	max	Total	Avg
Token	4	22	277	12.0
Lexeme	4	22	267	11.6
KetL-Concept	5	14	196	8.5
KetL-Ancestor	21	76	1003	43.6
Wn-Concept	13	118	997	43.3
Wn-Ancestor	44	271	2753	119.7

Effectively, each KetL concept referenced by any thought defines a folder that includes all the thoughts referencing that term; the folder is created as a consequence of the representation via the slot-inverse of ketl-concept. Similarly, each KetL ancestor defines a folder containing thoughts that reference concepts deemed relevant to that ancestor term. This structure defines a semantic index for the Angler thought contributions.

Having created this semantic index, we can leverage it to reason about the Angler thought contributions in simple but useful ways that address the four technical challenges identified above. The following sections describe in detail how the semantic index is used to assess similarity among Angler thoughts.

Our methodology for developing this application emphasizes evaluation. Several transformations occur in the representation of an Angler thought, and each results in an alternative representation. For each such transformation we want to be able to empirically assess the relative costs for performing that transformation, the relative benefits with respect to the quality of the resulting relations and properties established for the thoughts, and scalability issues, including how difficult it is to support the transformation in new domains. In the following evaluation of similarity assessments we compare the assessments supported by each of these six distinct types of representation data.

Computing Similarity

We are developing support to help participants identify when a thought (e.g., perhaps a newly authored thought) is semantically related to other thoughts. For a given thought, we use the semantic index to identify all other thoughts that share some ancestor term and so may be semantically relevant. Next we rank the relevant thoughts for semantic similarity. We have experimented with a few different ranking scores; all examples in this paper are scored with the cosine measure

$$\frac{X \bullet Y}{\| X \| \times \| Y \|}$$

where X is the vector of terms from one thought and Y is the vector of terms from another thought. This is a traditional similarity measure applied to the terms of a document [17]. We are applying this measure to each of the six types of representation data in order to compare the match quality supported by each type. We currently are weighting terms using a standard tf.idf measure [17] that considers both the term-frequency (how often a term is referenced by a document) and the inverse document frequency (in how many documents is the term referenced):

$$(1 + (\log tf)) * \left(\log \frac{n}{df} \right)$$

In considering the ancestors of a thought when computing semantic similarity, we are effectively performing ontology-based query expansion [18, 19].

EVALUATION

To evaluate the relative performance of each of the indexes we computed N-most similar thoughts for each of the 23 thoughts represented, for N values of 1 and 2, using each of the six data types. While there are at most two distinct similar thoughts for each data type, there are up to eight possible distinct similar thoughts for each thought because the indexes support different determinations of similarity from each other; thus for each thought there are between two and eight candidate similar thoughts. Next we randomized the order of these candidates and asked three human subjects to rate the similarity of each candidate; the subjective estimations were on a scale of 1 (very low similarity) to 10 (very high similarity). Then we averaged the subjective values so that for each thought and candidate pair we had an average (across the three human estimates) subjective score of their similarity.

Table 3 presents the average subjective scores for each of the six data types and each of the three humans and an average across all the humans (ten rows) and for assessments that included the best one candidate and the best two candidates (two columns). For example:

- The cell in row 4 and column 1 indicates that averaging over the 23 thoughts, the single highest-rated candidate for each thought found using the KetL-Ancestor vectors was deemed, on average by the three human evaluators, to have a similarity score of 6.1.
- The cell in row 4 and column 2 indicates that averaging over the 23 thoughts, the highest-two rated candidates using the KetL-Ancestor vectors was deemed by the average of the three human evaluators to have a similarity score of 5.6.
- The cell in row 7 and column 1 indicates that averaging over the 23 thoughts, the single highest-rated candidate for each thought by evaluator Human-1 was deemed, on average by the other two human evaluators, to have a similarity score of 6.5.

Table 3: Average Relevance Scores

	Best 1	Best 2
Token	5.4	4.7
Lexeme	5.4	5.0
KetL-Concept	5.6	5.1
KetL-Ancestor	6.1	5.6
Wn-Concept	4.7	4.7
Wn-Ancestor	4.7	5.0
Human-1	6.5	5.7
Human-2	6.8	6.4
Human-3	6.3	5.5
Human avg.	6.5	5.9

The data in Table 3 indicate that the ontology supports assessments of similarity that are close to quality of human assessments. Specifically, for N=1 (column 1), the KetL ancestor scores improve upon the lexeme scores but are not statistically different from the human average at confidence level $p<.2$. However, the human assessments do not reflect strong agreement, indicating how open and subjective is the task of the assessing semantic similarity among thoughts.

Table 4 presents the precision and recall scores for the same six data types and three humans (nine rows) and for assessments that included the same best one candidate and best two candidates (two columns) as the average human assessments. Note that precision (the ratio of retrieved candidates considered relevant) and recall (the ratio of relevant candidates retrieved) are the same in these tests since

in both cases the ratio denominator equals the batch size (i.e., the number of candidates retrieved equals the number of targets considered relevant), 1 for the first column and 2 for the second column.

Table 4: Average Recall/Precision Scores

	Best 1 (%)	Best 2 (%)
Token	35	39
Lexeme	30	41
KetL-Concept	39	48
KetL-Ancestor	48	63
Wn-Concept	22	35
Wn-Ancestor	26	39
Human-1	57	65
Human-2	61	67
Human-3	52	67

Table 5 presents the average precision data for each data type when considering as relevant those candidates deemed among the top two similarity scores by any human evaluator. Thus there are as many as six targets deemed relevant for each thought for the six data types (up to four targets for the human evaluators).

Table 5: Precision Scores for any Human Pick

	Best 2 (%)
Token	74
Lexeme	83
KetL-Concept	85
KetL-Ancestor	87
Wn-Concept	78
Wn-Ancestor	83
Human-1	87
Human-2	83
Human-3	78

The evaluation data suggest that the KetL ancestor index provides estimates of similarity that are substantially more effective than are the lexical indexes (e.g., the token and lexeme). The similarity-assessment performance using the KetL ancestor index is 50% better that the lexical data within 95% of the average across the three human evaluators for the identifying the top one or two most similar thoughts (Table 4, column 2). This is very encouraging, and we expect this level of performance to be deemed useful by Angler users.

Representation Reuse

The similarity assessment described above suggests one potential benefit from maintaining a semantic, ontology-based index for Angler thoughts. Next we consider the expense of supporting this index. Specifically, we consider the representational requirements that might be expected to handle additional thoughts: given a new thought, how many new lexemes and concepts and ancestors might we expect to have to define in order to represent the new thought?

One way to consider these requirements is by identifying the incremental need for additional representation terms for the 23 example Angler thoughts. Table 6 presents the incremental needs for tokens, lexemes, concepts, and ancestors required to represent each of the thoughts.

The data in Table 6 indicates the number of new terms required for each thought should be less than the number required by the preceding thoughts because later thoughts can reuse the representation terms required by prior thoughts. Since this assessment is sensitive to the order by which thoughts are encountered, the data in Table 6 summarizes 50 runs, each run considering the 23 thoughts in a random order, and then averaging over those runs, and then presenting the averaged incremental need for each type of representation data for every other thought.

As indicated in Table 6, initially the requirement for the ontology is substantial; however, the number of new concepts and ancestors required for each new thought diminishes quickly. In particular, the burden of providing the background knowledge required to define the ancestors is initially quite high, but it rapidly converges to levels only slightly higher than the concepts (e.g., only one new ancestor for each three new concepts). This suggests that the overhead of defining the hierarchical background knowledge in the ontology for a given domain may (on a *per thought* average) trend towards a negligible level because of the substantial reuse of this knowledge across many thoughts in that domain.

Table 6: Incremental Representation Requirements

Thought	Tokens	Lexemes	Concepts	Ancestors
1	11.3	10.9	8.1	47.7
3	9.3	8.6	7.2	18
5	8.9	8.3	5.1	10.7
7	8.1	6.9	6.4	7.6
9	7.3	7	5.3	7.1
11	7.2	6.7	4.9	6.3
13	6.8	6.0	4.4	5.2
15	6.6	5.9	3.8	5.6
17	6.0	5.6	3.7	4.3
19	6.3	5.7	3.4	4.4
21	6.6	5.7	3.6	3.8
23	6.5	4.8	3.1	3.4

DISCUSSION AND FUTURE WORK

WorldNet [16] represents alternative word senses. Because the ontology was developed on an as-needed basis to cover the concepts referenced in the represented thoughts there are a minimum of alternative senses captured in the denotations of the words appearing in the represented thoughts. We anticipate that as the number of represented thoughts increases ambiguity will accrue due to alternative word denotations. Our preliminary experiments with WorldNet, where all possible word senses are considered, suggest that

as this ambiguity is introduced substantial loss in performance may occur (Tables 3 and 4). Mitigating the difficulties arising from multiple word sense is probably the single greatest challenge to fielding a practical solution for detecting similar thoughts for Angler users. Therefore we are focusing future work on addressing this challenge with a knowledge-capture tool.

The knowledge-capture tool will manage and coordinate simultaneous extensions to both the ontology module and the natural language module. As it extends the Angler/KetL ontology with a new concept (perhaps corresponding to a new word sense) it will also update the lexicon and the linguistic models that TextPro uses for extracting words and identifying word senses. We plan to investigate the hypothesis that attaching each new concept (each new word sense) in the ontology to example uses (e.g., relevant texts of Angler thoughts) of the words that denote the new concept will provide the data for statistical and constraint models for adequately disambiguating those multiple word senses that are represented in the evolving system. Each multiple word sense exists because some use required it and that use (that thought text) can contribute to a model of how to distinguish the sense for alternative sense of the word. Thus the hybrid knowledge-capture tool manages coordinated extensions to the ontology and to the growing relevant corpus of word uses.

ACKNOWLEDGMENTS

We are grateful for the contributions to this work by our colleagues Thomas Boyce, Janet Murdock, Jerome Thomere, and Erik Yeh.

REFERENCES

[1] Murray, K., Lowrance, J., Appelt, D., Rodriguez, A. 2005. Fostering Collaborations with a Semantic Index over Textual Contributions. In Proceedings of the AAAI Spring Symposium on AI Technologies for Homeland Security. http://www.ai.sri.com/pubs/files/1125.pdf

[2] Rodriguez, A., Boyce, T., Lowrance, J., and Yeh, E. 2005. Angler: collaboratively expanding your cognitive horizon. Submitted to the First International Conference on Intelligence Analysis Methods and Tools.

[3] King, B. 1967. Step-wise clustering procedures. Journal of the American Statistical Association, 69:86–101.

[4] Schwartz, P. 1991. *The Art of the Long View: Planning for the Future in an Uncertain World.* Currency Doubleday.

[5] Gruber T. 1993. A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220.

[6] Hotho, A., Mädche, A., and Staab, S. 2001. Ontology-based Text Clustering, IJCAI Workshop Text Learning: Beyond Supervision, 2001

[7] Hobbs, J., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., and Tyson, M. 1996. "FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text" in Schabes, Y. (ed) Finite State Devices for Natural Language Processing, MIT Press.

[8] Appelt, D., and Martin, D. 1999. Named Entity Extraction from Speech: Approach and Results Using the TextPro System. In Proceedings of the DARPA Broadcast News Workshop.

[9] Kehler, A., Appelt, D., Taylor L., and Simma, A. 2004. "Competitive Self-Trained Pronoun Interpretation," in Proceedings of HLT/NAACL, pp. 33-36.

[10] Barker, K., Porter, B., and Clark, P. 2001. A Library of Generic Concepts for Composing Knowledge Bases. In Proceedings of the First International Conference on Knowledge Capture.

[11] Cycorp: What Does Cyc Know. http://www.cyc.com/cyc/technology/whatiscyc_dir/whatdoescycknow

[12] Teknowledge. SUMO. http://ontology.teknowledge.com

[13] Cycorp. The Syntax of CycL. http://www.cyc.com/cycdoc/ref/cycl-syntax.html

[14] Genesereth, M., and Fikes, R. 1992. Knowledge Interchange Format Version 3.0 Reference. Stanford University.

[15] Clark, P., Thompson, J., Holmback, H., and Duncan, L. 2000. Exploiting a Thesaurus-Based Semantic Net for Knowledge-Based Search. In Proceedings of the Twelfth Conference on Innovative Applications of Artificial Intelligence.

[16] Miller, G. 1995. WordNet: a lexical database for English Communications of the ACM, Volume 38, Issue 11.

[17] Manning, C., and Schutze, H. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

[18] Bodner, R., and Song, F. 1996. Knowledge-based approaches to query expansion in information retrieval. In Lecture Notes in Computer Science, volume 1081.

[19] Navigli, R., and Velardi, P. 2003. An Analysis of Ontology-based Query Expansion Strategies, Workshop on Adaptive Text Extraction and Mining. In Proceedings of the 14th European Conference on Machine Learning.