Capability Modeling of Knowledge-Based Agents for Commonsense Knowledge Integration*

Yen-Ling Kuo and Jane Yung-jen Hsu

Department of Computer Science and Information Engineering
National Taiwan University
yjhsu@csie.ntu.edu.tw

Abstract. Robust intelligent systems require commonsense knowledge. While significant progress has been made in building large commonsense knowledge bases, they are intrinsically incomplete. It is difficult to combine multiple knowledge bases due to their different choices of representation and inference mechanisms, thereby limiting users to one knowledge base and its reasonable methods for any specific task. This paper presents a multi-agent framework for commonsense knowledge integration, and proposes an approach to capability modeling of knowledge bases without a common ontology. The proposed capability model provides a general description of large heterogeneous knowledge bases, such that contents accessible by the knowledge-based agents may be matched up against specific requests. The concept correlation matrix of a knowledge base is transformed into a k-dimensional vector space using low-rank approximation for dimensionality reduction. Experiments are performed with the matchmaking mechanism for commonsense knowledge integration framework using the capability models of ConceptNet, WordNet, and Wikipedia. In the user study, the matchmaking results are compared with the ranked lists produced by online users to show that over 85% of them are accurate and have positive correlation with the user-produced ranked lists.

Keywords: multi-agent system, common sense, commonsense knowledge integration, capability model, agent description.

1 Introduction

Commonsense knowledge is an essential element for building intelligent systems. It enables computers to infer new facts or to perform actions with commonsense about the world so that applications can interact with humans intelligently. Also, it helps break the software brittleness bottleneck by taking place whenever the domain-specific knowledge fails.

For thirty years, many projects [5,14] have been devoted to the collection of common sense knowledge. While significant progress has been made in building large commonsense knowledge bases (KBs), e.g. Cyc [5] and ConceptNet [3], such KBs are still

* This research was supported by a grant from the National Science Council of Taiwan, NSC 99-2221-E-002-139-MY3.

D. Kinny et al. (Eds.): PRIMA 2011, LNAI 7047, pp. 299–310, 2011.

intrinsically incomplete or even inconsistent. Therefore, it may be necessary for an intelligent system to reason with multiple commonsense KBs at the same time in order to meet the specific goals of an applications. Example 1 demonstrates a sample scenario from an application developer's viewpoint.

Example 1. Goal-oriented search engine [6]
The goal-oriented search engine is a search engine interface that uses commonsense reasoning to turn search goals specified by the user in some natural language description into effective query terms. When processing an input like "my golden retriever has a cough", it should identify that the user's search goal is to find a veterinarian/remedy for his/her dog. Unfortunately, systems developed using ConceptNet alone will be unable to find the answers, as it does not contain any knowledge about golden retrievers. Alternatively, an intelligent system can first consult the lexicon database WordNet to find a generalization of the concept, e.g. "golden retriever is a kind of dog". It is then possible to find "veterinarian" by reasoning from "dog" and "cough" in the ConceptNet semantic network.

In summary, enabling applications to reason across multiple knowledge bases improves their goal-achieving behaviors. In the dynamic world today, it is especially important that applications should be equipped with up-to-date knowledge to interact with their users. Multi-agent systems (MAS) provide a powerful paradigm to facilitate application building when multiple heterogeneous knowledge representations and reasoning are required [8]. In this paper, we present a multi-agent framework for commonsense knowledge integration, which is especially effective in supporting the reasoning with knowledge on different commonsense domains from heterogeneous KBs. It can reflect upon and improve its behavior when KBs are expanded.

One major challenge in such a multi-agent system is to describe the capability of each knowledge-based agent, given that multiple representations are used, and without common ontology. This paper proposes a distributed capability model which is built by transforming the concepts contained in the KBs into a k-dimensional space using low-rank approximation. Requests from applications are evaluated based on vector similarity to decide which KB to match up with.

This paper starts with a brief overview of commonsense knowledge representations, collection, and reasoning methods. A specific multi-agent system for common sense knowledge integration is then proposed, along with the introduction of our capability model and capability evaluation procedure. We then present our experimental setup of the capability models in ConceptNet, WordNet, and Wikipedia. The matching results are evaluated by comparing them with matchmakings made by online users to verify their correctness and relevance.

2 Commonsense Knowledge Bases

Due to their different design decisions, the current commonsense KBs are heterogeneous and inconsistent in representations, quantity, quality, and means of access. Therefore, a system to integrate different commonsense knowledge bases are needed for the benefit of application building. We will review these elements and illustrate their heterogeneity in this section.

2.1 Knowledge Representation

When building applications, developers may choose commonsense KBs with different knowledge representations to serve their specific requirements. The two most prominent representations for common sense are formal logical framework and semantic network, used by Cyc [5] and ConceptNet [3] respectively.

The formal logical framework is appropriate for representing precise and unambiguous facts, which facilitates the automation of commonsense reasoning. On the other hand, the semantic network is more flexible in incorporating new knowledge and contextual reasoning. It represents all sentences in the corpus as a directed graph. The nodes of this graph are *concepts*, and its labeled edges are *relations* between two concepts. For example,

- `UsedFor(a, b)`, e.g. [Spoon] is used for [eating].
- `IsA(a, b)`, e.g. [Dog] is an [animal].

2.2 Commonsense Knowledge Collection

Codifying millions of pieces of human knowledge into machine usable forms has proved to be time-consuming and expensive. While techniques for mining knowledge from corpus or web pages have been developed [12,2], it is difficult for computers to discover the commonsense knowledge underlying a text. Therefore, sources of commonsense knowledge are still majorly reliant on experts or the general public.

Expert-Developed Knowledge Bases. A team of knowledge engineers encode common sense into the KBs. This approach ensures the highest quality of data. However, it is expensive, time-consuming, and difficult to scale up.

WordNet [7] is a highly structured database of words, which are carefully crafted by expert linguists. Synonyms are grouped into *synsets* and are conneted with each other by relations. It has been successfully used in a variety of applications to measure the proximity of words.

Started in 1984, the Cyc project [5] carefully crafted knowledge into CycL, a rigorous logic-based language to ensure its correctness. Now, the OpenCyc 2.0 ontology contains hundreds of thousands of terms with millions of assertions relating the terms to each other.

Collaboratively-Built Knowledge Bases. The success of crowd-sourcing approaches led many research groups to start to use websites or games to appeal to online users for contribution. However, the knowledge collected from these sources is highly dependent on the performance of users, which also makes the KBs incomplete and inconsistent.

The Open Mind Common Sense (OMCS) project at MIT [15] has collected over a million sentences in multiple language. The English and Portuguese corpora were collected from over 15,000 contributors at the OMCS website[1] within the span of 10 years. With innovations in community-based social games, the up-to-date knowledge in

[1] http://openmind.media.mit.edu/

the Chinese ConceptNet was successfully collected and verified via question-answering between players [4].

Wikipedia[2] is one of the world's largest KBs of both encyclopaedic knowledge and commonsense knowledge. The knowledge is stored in documents connected with page links. It also provides a taxonomy by its categories, where articles can be assigned to one or more categories. The unstructured documents are thus put into a network of categories.

2.3 Commonsense Reasoning

It is straighforward to equip a variety of applications with common sense by querying the KBs using APIs. For example, one may ask if a specific assertion is present in the corpus. Furthermore, KBs with different representations may call for different reasoning methods. The semantic network is suitable for finding related and similar concepts. Measures of similarity/relatedness quantify how much two concepts are alike/related. Both relatedness and similarity measures are developed for WordNet [10], ConceptNet [17], and Wikipedia [18] so that it is possible to reason in large and noisy semantic networks. The logic framework, on the other hand, uses deduction and theorem prover to reason new facts. Heuristics are often applied to logic-based reasoning for better efficiency. OpenCyc[3] also released its planner for reasoning out actions and events with its rules and assertions.

3 Commonsense Knowledge Integration

In response to the emergence of heterogeneous commonsense knowledge sources and the different reasoning methods using them, a system for commonsense knowledge integration should utilize the reasoning abilities of KBs while maitaining their own autonomies.

3.1 Multi-agent Framework

A multi-agent system is fitting for this open and dynamic environment to achieve the interoperation of commonsense KBs. We devised a common sense integration framework (see figure 1) to provide integrated reasoning service for application to use.

Instead of integrating knowledge sources into a single ontology, the key idea of this framework is to treat knowledge as resources that different reasoning methods can access. The integration of knowledge is achieved via matchmaking and composition of different reasoning methods. Following are the detailed descriptions of the agents in figure 1.

- **KB agent:** A KB agent is responsible for a commonsense KB. It monitors knowledge in the KB and is equipped with behaviors that make the KB complete. If a matchmaker asks for a KB agent's capability in handling a request, the KB agent will answer the query based on its belief (i.e. the capability model) of the KB.

[2] http://www.wikipedia.org
[3] http://www.opencyc.org/

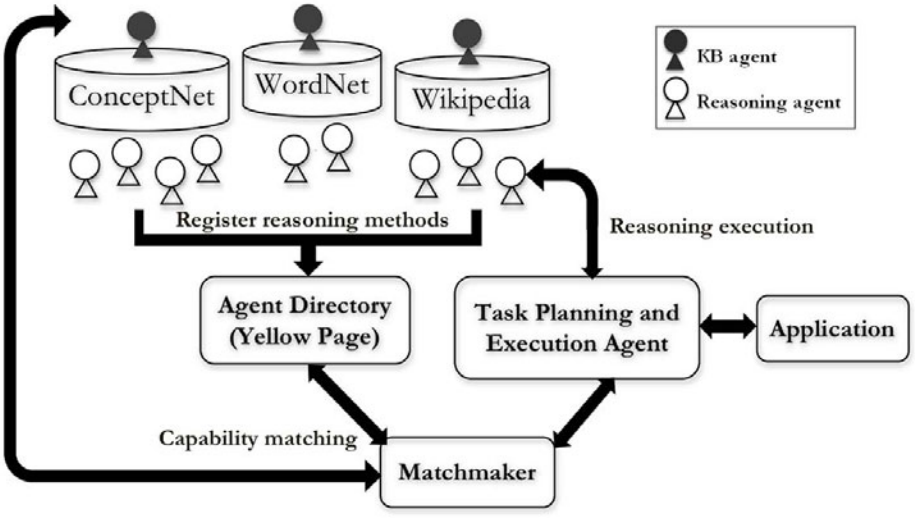

Fig. 1. The multi-agent framework for commonsense knowledge integration

- **Reasoning agent:** There are multiple reasoning agents for a KB. Each agent performs an atomic reasoning task of the KB. Once the KB agent updates the KB, the quality of reasoning results is also improved.
- **Agent directory:** An agent directory records the types of reasoning agents in the system and the KBs they can access.
- **Matchmaker:** A matchmaker forms sample queries to KB agents to check which knowledge bases are able to handle its request. Then, it returns a list of candidate reasoning agents which are sorted by their capability to the task planning agent.
- **User agent:** A user agent represents a specific application. It states the needs of an application to the task planning agent and communicates results with users.
- **Task planning and execution agent:** Based on the requirements of a user agent, a task planning and execution agent decides whether to form a composite task or to send the request directly to the matched reasoning agents.

3.2 Challenges

There are three challenges in realizing such a framework. The major challenge is the capability matching of large and heterogeneous KBs. Unlike service matchmaking in service-oriented computing, there is no common ontology for applications to identify the capability of commonsense KBs. The second one is to maintain good reasoning quality for reasoning agents. It requires the complete KBs. The third challenge is the composition of reasoning methods. The dependency of reasoning agents should be specified to facilitate composite reasoning.

The second and third challenges have been studied in knowledge acquisition and MAS community respectively and are not in the scope of our discussion. In this paper, we focus on the first challenge, the capability model of KB agents. Instead of querying

every KB at the same time (which causes a very high overhead), we use a vector space to model the belief of KB agents.

4 Related Work

Many approaches to integrating knowledge and describing capability of agents has been proposed in MAS. The differences between these systems and commonsense knowledge integration are discussed in this section.

Knowledge Integration Systems. MAS has been proposed to integrate and reuse information in web environment for many years. Most of them opt to combine loosely coupled sources into integrated wholes, such as KRAFT [11] and KSNet [16]. InfoSleuth[9] is another system for integrating heterogeneous information sources. An application-specific ontology is used as a basic ontology to locate different queries. However, these approaches are not feasible for commonsense knowledge integration, because there is no common ontology and exists conflicts in different commonsense KBs such that it may results in errors if we combine them into a single one.

Capability Descriptions in Matchmaking. One of the most important issues in matchmaking is to describe agents' capability. Several languages are defined to represent agent capabilities, e.g. LARKS (Language for Advertisement and Request for Knowledge Sharing) [19], WSDL (Web Service Description Language)[4], etc. With the representations of capabilities, attributes can be matched using constraint satisfaction or semantic similarity [13]. However, this process usually requires ontologies to facilitate generalizing service needs and calculating semantic similarities. Also, the description languages cannot reflect the differences resulted from the knowledge the agents can access. For example, even if the agents in WordNet and ConceptNet provide the same reasoning method, e.g. finding related concepts, they may return different results because of different knowledge coverage.

5 Capability Model

The major challenge in finding a reasoning agents to handle a query is the variety of domains contained in the KBs. Since the capability of a reasoning agent is reflect on the coverage of knowledge base, each KB agent should be equipped with a compact model of its KB, and be able to inform the capable reasoning agents to provide the required knowledge quickly. In this section, we introduce the proposed capability model for KB agents and an algorithm to evaluate requests.

5.1 Representation of Capability Model

In most KBs, we can argue that they can be determined by or associated with a small set of eigenconcepts. Therefore, we can use the eigenconcepts as the capability model of KB agents.

[4] http://www.w3.org/TR/wsdl20/

Correlation Matrix. Consider a commonsense knowledge base $\mathcal{K}$, a sentence is always represented as a triple, $(c_i, \textit{relation}, c_j)$, where c_i and c_j are concepts. Each concept can form a vector of related concepts $\boldsymbol{v}$, where the j_{th} component of the vector, v_j, is the number of triples containing c_i and c_j in $\mathcal{K}$. Thus, we can construct an $n \times n$ *correlation matrix*, where n is the number of concepts in $\mathcal{K}$. The *correlation matrix* reflects the capabilities of a KB agent by describing the relatedness of concepts in the KB it can access.

Knowledge Represented by Eigenconcepts. In order to evaluate the query quickly, we need to reduce the dimension of a correlation matrix without losing its capability. We re-formulate the high-dimensional concept correlation matrix into a k-dimensional eigenspace, where $k \ll n$. A concept in $\mathcal{K}$ is then represented as a vector in a k-dimensional space spanned by *eigenconcepts*. It is the responsibility of a KB agent to identify the dimensions that are useful in summarizing the KB while truncating dimensions that are less relevant.

This process is also referred to as "dimensionality reduction." *Low-rank approximation* is an approach to achieving dimensionality reduction. Given $m \times n$ matrix A, we aim to approximate it by a matrix of rank k, which is much smaller than m and n. In this paper, we use singular value decomposition (SVD) to achieve low-rank approximation.

5.2 Notations

In what follows, let A be the correlation matrix constructed from a commonsense KB. We use $U\Sigma U^T$ to denote the SVD of A since A is a symmetric matrix. The diagonal entries of Σ are singular values of A and denoted as σ_i. Similarly, we use $A_k = U_k \Sigma_k U_k^T$ to denote the best rank k approximation to A and use $A^{(c)}$ to refer to the column of concept c in A. The projection of the concept c in A onto the first k column of U is denoted as a_c. Finally, we denote the 2-norm of a vector a_c by $\|a_c\|$ and the 2-norm of a matrix A by $\|A\|_2$.

5.3 Capability Modeling: Choosing the Best k

In order to create a capability model for describing a KB, we apply SVD on the correlation matrix: $A \approx A_k = U_k \Sigma_k U_k^T$, where

- U_k: a $n \times k$ matrix that relates concepts to eigenconcepts
- Σ_k: a $k \times k$ diagonal matrix of singular values σ_i that assigns weights to each eigenconcept.

The best rank k is chosen in algorithm 1 so that the resulted space is the best approximation to describe the correlation of concepts in a KB.

The confidence of choosing k by $\sigma_k - \sigma_{k+1} \leq \theta$ is motivated by the Eckart-Young theorem.

Theorem 1. *Eckart-Young theorem [1]*
Let $A = U\Sigma V^T = U\mathit{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)V^T$. *For any* k *with* $0 \leq k \leq r$,

$$\|A - A_k\|_2 = \sigma_{k+1} \tag{1}$$

Algorithm 1. Capability Model ($\mathcal{K}$)

Require: A commonsense knowledge base $\mathcal{K}$
Ensure: A set of eigenvectors that span the capability model U_k and the projection A_k of concepts in $\mathcal{K}$
1: Build a correlation matrix A from triples in $\mathcal{K}$
2: Apply SVD on the correlation matrix: $A = U\Sigma U^T$
3: Choose the largest k such that $\sigma_k - \sigma_{k+1} \leq \theta$
where θ is a small constant
4: Represent concepts in the k-dimensional eigenspace:
$A_k = U_k^T A$
5: **return** U_k and A_k

Theorem 1 implies that $\sigma_k - \sigma_{k+1}$ is the key factor of incorporating the k_{th} column of U into the compact representation of capability model. If we set θ small enough, we are returned with the representative eigenconcepts so that the error between real and modeled capability of KB is within σ_{k+1}.

5.4 Capability Evaluation for Matchmaking

With the capability model, the KB agent is now able to answer whether it can handle a request from applications. Here, we introduce two kinds of concept vectors.

- **Knowledge-based concept vector**, a_c: The vector of concept c is $a_c = U_k^T A^{(c)}$. It represents the supply of a KB.
- **Application-based concept vector**, v: For any application that would like to incorporate commonsense knowledge, we can find a corpus to describe the application, e.g. using google snippets to describe the application that requires up-to-date news. A concept can be represented by a vector v constructed from the co-occurred concepts in the corpus. This vector states the need of an application.

We can also project v onto the capability model, i.e. $v_k = U_k^T v$. The capability of KB to handle the request is correspondent to the similarity between a_c and v_k. The algorithm to evaluate the capability of a KB is illustrated in algorithm 2. The evaluation score returned by algorithm 2 is defined by the cosine similarity of a_c and v_k.

Algorithm 2. Evaluate Capability (U_k, A_k, c)

Require: A capability model U_k, a projection A_k of concepts in $\mathcal{K}$, and a testing concept c
Ensure: An evaluation score sim that indicates the capability of $\mathcal{K}$ to handle a request
1: Construct term-frequency vector v of concept c
from the corpus of application, e.g. google snippets
2: Project v onto the capability model: $v_k = U_k^T v$
3: Get vector of c from A_k: $a_c =$ column of c in A_k
4: Calculate similarity of a_c and v_k: $sim = \frac{a_c \cdot v_k}{\|a_c\| \|v_k\|}$
5: **return** sim

Once the matchmaker gets the request from the task planning agent, algorithm 3 is used for matching suitable KBs. The matchmaker asks every KB agent to get the evaluation score of each KB respectively, and then sort the KBs based on their evaluation scores in descending order. Finally, the KBs with evaluation scores > 0.5 are returned to the task planning agent. The task planning agent will send requests to the reasoning agents of the matched KBs to get answers.

Algorithm 3. Matchmaking $(\mathcal{K}, c)$

Require: A set of available KBs $\mathcal{K}$ and a query concept c
Ensure: A sorted list of matched KBs L
1: Construct application-based vector v
2: **for** K_i in $\mathcal{K}$ **do**
3: sim_i = Evaluate Capability (U_{ik}, A_{ik}, c, v)
4: **end for**
5: **for** K_i in $\mathcal{K}$ **do**
6: **if** $sim_i > 0.5$ **then**
7: Add KB K_i into L
8: **end if**
9: **end for**
10: Sort L in descending order based on the evaluation scores of KBs
11: **return** L

6 Evaluation

In order to evaluate the proposed capability model, we incorporated the model into the matchmaking in the commonsense knowledge integration framework. The matchmaking results produced by our approach are compared with the matches made by online users.

6.1 Experimental Setup

The commonsense knowledge integration framework is implemented using JADE (Java Agent DEvelopment Framework). ConceptNet, WordNet, and Wikipedia are chosen to be our experimental KBs. The number of concepts in each KB are shown in table 1. Despite the large number of concepts in these KBs, they are quite inconsistent. For example, the overlap of ConceptNet and WordNet is only 4.79%.

Table 1. Statistics of knowledge bases

Knowledge base	Number of concepts
ConceptNet	274,477
WordNet	128,391
Wikipedia	3,440,143

Three reasoning agents are designed for each KB.

- **Topic agent:** Find the topics of a given concept, e.g. "computer" and "computer science" are topics of "cpu".
- **Related agent:** Find related concepts of a given concept, e.g. "teacher", "student", "blackboard" are related concepts of "school".
- **Similarity agent:** Calculate the similarity score of two concepts, e.g. the similarity of "cow" and "horse" is 0.889 in ConceptNet.

In this experiment, snippets from google's search results are selected as our corpus to create the application-based concept vector v in the matchmaking. Related concepts of a specific query term are returned by "related agent" of the matched KB.

Collecting Matchmaking Lists from Online Users. In order to evaluate the matchmaking results, we incorporated the proposed capability model into the system. The matchmaking results produced by our approach are compared with the matches made by online users. In order to evaluate the matchmaking results, we collected a user-produeced matching list as the ground truth. The list was collected from workers on Amazon Mechanical Turk[5], the largest crowd-sourcing market in the world. First, we uniformly sampled 100 concepts from each KB as input queries from applications. About half of these concepts were found in at least two KBs. Every worker in our task was given a concept and three web pages containing related concepts of that concept. The web pages are generated from the three KBs. What the worker required to do was to browse the web pages and rate the relatedness of the web page and the given concept. Every concept was rated by 3 workers to increase its validity. In this process, we created a ranking of the KBs for each concept according to their relatedness with the query concept. The KB ranked first was considered as the matched KB for finding related concepts.

Building a Correlation Matrix. Since the correlations of concepts are in different forms for different KBs, we had to use a different method to create the triples required by KB agents. Triples in ConceptNet are its assertions; triples in WordNet are its words and the relations between words; triples in Wikipedia are the pages and their links with other pages. The θ in algorithm 1 is set to 0.1 for every KB agent.

6.2 Experimental Result

In this experiment, our matchmaking mechanism used the evaluation scores of the query concept in each KB to create a ranking of KBs for each query concept. Accuracy and rank correlation are two measures we used for evaluating matchmaking correctness and relevance against the user-produced ranking.

Correctness. Intuitively, the correctness of a matchmaking mechanism is whether it can find user's desired result. If the result ranked first in the matchmaking is also the

[5] https://www.mturk.com/

KB ranked first by online users, we marked it as a "match". The accuracy is thus defined as the proportion of matches to query terms:

$$accuracy = \frac{\#\ of\ matches}{\#\ of\ query\ concepts}$$

For all sampled concepts, the accuracy is 93.32%. If we only consider the concepts that can be found in at least two knowledge bases, the accuracy is 87.67%, which is slightly lower than the former case. This drop of accuracy appears in the concepts with polysemy. For such concepts, it is likely that the application-based vector may not be aligned with the user's goal, therefore causing errors in matching KBs. It indicates that with application-based vectors corretly representing an applications' goal, the matchmaker can produce an accurate match using the proposed capability model.

Relevance. We also compare the rank correlation of the matchmaking produced by the matchmaker and online users. The relevance of our matchmaking results and ranked list produced by online users were measured by Kendall's τ rank correlation coefficient τ_B. If the two ranked lists are in perfect agreement, τ_B is 1; if they are perfect disagreement, τ_B is -1. In our experiment, the average τ_B are 0.818 and 0.695 respectively for all query concepts and query concepts in at least two knowledge bases respectively. A positive correlation was found between the two lists. This result suggests that the capability model corresponds to our understanding of the knowledge bases. With our capability model, we can represent knowledge bases well and make judgment that is relevant to human intuition.

7 Conclusion

This paper proposed a distributed capability model for large and heterogeneous KBs. Instead of integrating KBs into a single one, the model has the following features:

- it can express multiple KBs without a common ontology.
- it can update the belief of agents in a dynamic environment with frequently updated knowledge.

Using the capability model created from concept correlation matrix, we are able to identify differences, e.g. different senses of concepts, for KBs with no common ontology. The capabilities of agents are described based on the queries they can answer. Experiments have been conducted to match KB for finding related concepts. Compared to the user-produced ranking lists, our proposed method shows a high accuracy of 87.6% and a positive rank coefficient. With the capability model of KBs, many complex reasoning tasks can then be built on top of the commonsense knowledge integration system that matches relevant reasoning agents to applications.

References

1. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika 1(3), 211–218 (1936)
2. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Methods for domain-independent information extraction from the web: an experimental comparison. In: Proceedings of AAAI 2004 (2004)

3. Havasi, C., Speer, R., Alonso, J.: ConceptNet 3: A flexible, multilingual semantic network for common sense knowledge. In: Recent Advances in Natural Language Processing, Borovets, Bulgaria (September 2007)
4. Kuo, Y.L., Lee, J.C., Chiang, K.Y., Wang, R., Shen, E., Chan, C.W., Hsu, J.Y.j.: Community-based game design: experiments on social games for commonsense data collection. In: Proceedings of the ACM SIGKDD Workshop on Human Computation (2009)
5. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11), 33–38 (1995)
6. Liu, H., Lieberman, H., Selker, T.: GOOSE: A Goal-Oriented Search Engine with Commonsense. In: De Bra, P., Brusilovsky, P., Conejo, R. (eds.) AH 2002. LNCS, vol. 2347, pp. 253–263. Springer, Heidelberg (2002)
7. Miller, G.A.: WordNet: A lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
8. Minsky, M.: The Society of Mind. Simon and Schuster (1988)
9. Nodine, M., Fowler, J., Ksiezyk, T., Perry, B., Taylor, M., Unruh, A.: Active information gathering in InfoSleuth. International Journal of Cooperative Information Systems 9(1/2), 3–28 (2000)
10. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet: Similarity - measuring the relatedness of concepts. In: Proceedings of AAAI 2004 (2004)
11. Preece, A., Hui, K., Gray, A., Bench-capon, T., Joes, D., Cui, Z.: The KRAFT architecture for knowledge fusion and transformation. In: Proceedings of the 19th SGES International Conference on Knowledge-Based Systems and Applied Artificial Intelligence (1999)
12. Schubert, L., Tong, M.: Extracting and evaluating general world knowledge from the brown corpus. In: Proceedings of the HLT-NAACL Workshop on Text Meaning (2003)
13. Singh, M.P., Huhns, M.N.: Service-Oriented Computing: Semantics, Processes, Agents. Wiley (2005)
14. Singh, P.: The public acquisition of commonsense knowledge. In: Proceedings of AAAI Spring Symposium (2002)
15. Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L.: Open Mind Common Sense: Knowledge Acquisition from the General Public. In: Meersman, R., Tari, Z. (eds.) CoopIS 2002, DOA 2002, and ODBASE 2002. LNCS, vol. 2519, pp. 1223–1237. Springer, Heidelberg (2002)
16. Smirnov, A.V., Pashkin, M., Chilov, N., Levashova, T.: Multi-Agent Architecture for Knowledge Fusion from Distributed Sources. In: Dunin-Keplicz, B., Nawarecki, E. (eds.) CEEMAS 2001. LNCS (LNAI), vol. 2296, pp. 293–302. Springer, Heidelberg (2002)
17. Speer, R., Havasi, C., Lieberman, H.: AnalogySpace: Reducing the dimensionality of common sense knowledge. In: Proceedings of AAAI 2008 (2008)
18. Strube, M., Ponzetto, S.P.: WikiRelate! computing semantic relatedness using wikipedia. In: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI 2006 (2006)
19. Sycara, K., Widoff, S., Klusch, M., Lu, J.: Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. Autonomous Agents and Multi-Agent Systems 5, 173–203 (2002)