Zang LJ, Cao C, Cao YN *et al.* A survey of commonsense knowledge acquisition. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(4): 689-719 July 2013. DOI 10.1007/s11390-013-1369-6

A Survey of Commonsense Knowledge Acquisition

Liang-Jun Zang[1,2] (臧良俊), Cong Cao[1,2] (曹　聪), Ya-Nan Cao[3] (曹亚男), Yu-Ming Wu[1] (吴昱明)
and Cun-Gen CAO[1] (曹存根), *Member*, *CCF*

[1] *Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences Beijing 100190, China*

[2] *University of the Chinese Academy of Sciences, Beijing 100049, China*

[3] *The Second Lab, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*

E-mail: {ljzang.ict, ccao.ict}@gmail.com; caoyanan@iie.ac.cn; {wuyuming, cgcao}@ict.ac.cn

Received June 15, 2012; revised April 28, 2013.

Abstract Collecting massive commonsense knowledge (CSK) for commonsense reasoning has been a long time standing challenge within artificial intelligence research. Numerous methods and systems for acquiring CSK have been developed to overcome the knowledge acquisition bottleneck. Although some specific commonsense reasoning tasks have been presented to allow researchers to measure and compare the performance of their CSK systems, we compare them at a higher level from the following aspects: CSK acquisition task (what CSK is acquired from where), technique used (how can CSK be acquired), and CSK evaluation methods (how to evaluate the acquired CSK). In this survey, we first present a categorization of CSK acquisition systems and the great challenges in the field. Then, we review and compare the CSK acquisition systems in detail. Finally, we conclude the current progress in this field and explore some promising future research issues.

Keywords commonsense knowledge, knowledge acquisition, knowledge representation and reasoning

1 Introduction

Artificial intelligence (AI) is the science and engineering of making intelligent machines. Its ultimate goal is to make machine reach human-level intelligence. In other words, AI researchers aim to develop machines that can solve problems and achieve goals in the world as well as humans. McCarthy, the father of AI, stated that common sense was farthest from human-level intelligence among the branches of AI①.

In the 164th page of Minsky's book *The Emotion Machine*[1], he defined common sense as follows: "*Common sense includes not only what we call commonsense knowledge – the kind of facts and concepts that most of us know – but also the commonsense reasoning skills which people use for applying their knowledge.*" Many tasks in our daily life, such as reading a simple story, require only this simple knowledge possessed by a ten-year-old child rather than precise scientific knowledge. Scientific knowledge gradually separated itself from commonsense knowledge as people sought more precise descriptions of their world.

Formalizing and collecting commonsense knowledge (CSK) are two main research issues in the field of common sense. Formalizing the CSK for even simple reasoning problems is very difficult②. Commonsense researchers therefore often study toy problems such as planning in the blocks world domain, whereas recently they began to study more realistic problems. Minsky and his colleagues designed and implemented a program equipped with a small-scale knowledge base to answer some simple questions about a simple story. Unfortunately, the program fails to know what to do when confronting other new stories. After working for a couple of years on this problem, they concluded that "*you'd have to know a couple million things before you could make a machine do some commonsense thinking.*"③. Therefore, commonsense knowledge acquisition is another fundamental and important problem in this domain.

Survey
This work is supported by the National Natural Science Foundation of China under Grant Nos. 91224006, 61173063, 61035004, 61203284, and 309737163, and the National Social Science Foundation of China under Grant No. 10AYY003.

①http://www-formal.stanford.edu/jmc/whatisai/node2.html, Apr. 2013.

②http://www-formal.stanford.edu/leora/commonsense/, Apr. 2013.

③Grice said the words and Minsky quoted it in a conversation. http://www.technologyreview.com/computing/17164/page1/, Apr. 2013. More information about Grice can be found in http://plato.stanford.edu/entries/grice/, Apr. 2013.

1.1 What Is CSK

Consider first the following things everybody knows: *plants cannot speak*, *fish live in water*, *people have to open their mouth to drink*. People usually refer to this kind of knowledge as CSK. Although there has not yet been any generally accepted formal definition of CSK, many related literatures mentioned CSK's characteristics as follows:

• *Share*. CSK is possessed and shared by (possibly a group of) people.

• *Fundamentality*. People understand CSK so well that they take CSK for granted.

• *Implicitness*. Usually people do not talk or write CSK explicitly since others also know it.

• *Large-Scale*. CSK has a tremendously large scale in both amount and diversity.

• *Open-Domain*. CSK covers all aspects of our daily life rather than a specific domain.

• *Default*. CSK are default assumptions about typical cases in everyday life, so most of them are defeasible rather than definitely correct.

Based on these characteristics, we would like to define CSK as: *a tremendous amount and variety of knowledge of default assumptions about the world, which is shared by* (*possibly a group of*) *people and seems so fundamental and obvious that it usually does not explicitly appear in people's communications*.

1.2 CSK Representation and Categorization

To make computers understand what knowledge means, CSK should be coded in formal language, such as the first-order logic, rather than natural language, such as English. In knowledge representation, individuals should be distinguished from concepts and relations. Individuals denote things in the world, such as person *Ming Yao*, organization *Rocket Team*, etc. Concepts denote collections of individuals, such as *NBA Player*. Relations denote relations about individuals, such as (*Ming Yao, is employed by, Rocket Team*). CSK can be roughly divided into three types:

• *Factual Knowledge*. Factual knowledge describes facts about individuals, represented as statements about individuals. For example, (*Ming Yao, be instance of, NBA player*); (*Ming Yao, be employed by, rocket team*). In other words, factual knowledge describes individuals using concepts and relations. Facts can be represented as statements of form $C(a)$ or $R(a,b)$. That is, individual a is an instance of concept C (conceptual fact); individuals a and b have relation R (relational fact).

• *Ontological Knowledge*. Ontological knowledge describes terms (i.e., concepts and properties) in some domain, represented as statements about concepts and properties. For example, (*NBA player, be subclass of, sportsman*); (*NBA player, be employed by, NBA team*). Ontological knowledge describes relations about concepts and relations. Taxonomic knowledge, including a concept hierarchy and/or a relation hierarchy, is the backbone of an ontology. A concept hierarchy contains statements of form $C_1 \sqsubseteq C_2$ (called concept C_1 is subclass of C_2), which means that each instance of C_1 is also an instance of C_2. A relation hierarchy contains statements of form $R_1 \sqsubseteq R_2$ (called relation R_1 is subrelation of R_2), which means that relation R_1 holds between x and y implies relation R_2 also holds.

• *Rules*. Rules are the hardest to acquire. For example, *any NBA player is employed by no more than one NBA team at any time*. It can be represented as statement:

$$\begin{aligned}\forall t \forall x (\mathit{NBAPlayer}(x,t) &\rightarrow \text{Unemployed}(x,t) \vee \\ &\exists y(\text{Team}(y,t) \wedge \text{employedBy}(x,y,t) \wedge \\ &\quad \forall z(\text{employedBy}(x,z,t) \rightarrow y = z))).\end{aligned}$$

In addition, CSK entered by non-experts or extracted from text is usually represented in natural language (e.g., English). Textual fragments are ambiguous and language-dependent; whereas individuals, concepts, relations are ontological terms and abstractions of human thought, which are assigned with formal semantics and language-independent. It is far from a trivial work to convert knowledge in natural languages into formalized knowledge.

1.3 Why Do Computers Need CSK

First, versatile expert systems need commonsense knowledge[2-3]. Usually expert systems only possess the knowledge required for solving particular problems, which makes them break down when confronting other unexpected situations. Commonsense knowledge can help an expert system determine whether it can handle the task at hand. In addition, it can serve as something to which expert systems refer when reacting to new situations.

Second, interactive systems need commonsense knowledge. To interact with human users better, interactive systems (e.g., recommender systems) should be able to understand and predict users' intentions, plans, preferences, affect, context, and so on. But to do so, computers must have access to commonsense knowledge of all above aspects of human beings. At the MIT's Media Laboratory, the researchers have designed and implemented a number of intelligent interface agents[4]. These interface agents can watch the user's interactions and operate the interface as the user would.

Third, large-scale commonsense knowledge bases have been successfully applied to many tasks relating to natural language processing (NLP), for example, ambiguity solution[5-7], sentiment analysis (e.g., AffectiveSpace[8] and Sentic Computing[9]), question answering[10], story understanding and generation[11-12], and information retrieval[13-14].

Finally, commonsense knowledge has many potential commercial applications. Many things that people do in everyday life require only commonsense knowledge rather than expert knowledge. Nilsson[15] described an example of household robot, which could prepare and serve meals, do the laundry, keep the house neat, and so on. A practical example of such robots is the Honda indoor robot, which is equipped with commonsense knowledge from OMICS[16]. The robot could guess the users' desires based on current beliefs and commands, and it also generates a room's topological map with spatial labeling.

To summary, CSK has important influence on many sub-fields of AI; giving computers the capacity for common sense will make AI closer to human-level intelligence.

1.4 Brief History

In this part, we shall give an overview of some important historical developments in the field of commonsense knowledge acquisition.

Advice Taker, to our knowledge, is the first program with commonsense knowledge[17]. It is able to automatically derive a wide range of consequences from what it is told and what it knows. The objective of designing it is to develop a program which can learn from experience as human do. Since then, there emerged numerous work of representing and reasoning over commonsense knowledge. Because formalizing commonsense knowledge is a very complicated undertaking, researches often focus on small toy problems like planning in the blocks world. Compared with commonsense knowledge representation and reasoning, there was far less work on building large-scale real-world commonsense knowledge bases.

Earlier work on gathering CSK mainly relies on the codification of human contributors. This method was employed by several famous projects, such as Cyc④ [18], Open Mind Common Sense (OMCS)⑤ [19], and HowNet⑥ [20]. Lenat[18] believed handcrafting an appreciable fraction of the required knowledge (e.g., more than a million) would enable further knowledge acquisition through natural language understanding (NLU) and machine learning (ML). Therefore, a team of knowledge specialists was employed to codify commonsense assertions into the Cyc knowledge base. To gather CSK more efficiently, OMCS distributed this labor-intensive task across general public on the Web. In other words, OMCS offered a collaborative tool that supports large-scale collative effort on building commonsense database. Through the OMCS website, volunteer contributors can enter CSK in natural language or even evaluate CSK entered by others. OMCS data has been transferred to two kinds of machine readable representations: ConceptNet[21] and AnalogySpace[22]. HowNet is the largest Chinese-English bilingual commonsense knowledge base[20].

Due to the rapid development of machine learning and natural language processing techniques, some researchers tried to automate the process of acquisition. Schubert[23] believed that "there is a largely untapped source of general knowledge in texts, lying at a level beneath the explicit assertional content"; he extracted from the Penn Treebank corpus⑦ a considerable number of general possibilistic propositions, such as "a house may have windows". Torisowa discovered commonsensical inference rules of events from Japanese newspapers[24-25]. Also, some researchers relied on automatic reasoning techniques (e.g., analogy) to discover new knowledge from existing knowledge bases, like Learner[26] and AnalogySpace[22]. It is widely agreed that human participant is indispensible during CSK acquisition and evaluation. In order to retain volunteer contributors and motivate them to share their knowledge, a number of intelligent user interfaces or games have been designed and developed since 2005. In Cyc Corp., a couple of semi-interactive tools were developed to enable lightly trained volunteers to enter or confirm knowledge in natural language[27]. Luis von Ahn[28] demonstrated that playing game was an effective way to collect commonsense knowledge. Other game-based work include [29-32].

After 2005, there emerged a number of automatically constructed large-scale knowledge bases, which contain millions or even billions items of knowledge. Such knowledge bases usually employ information extraction techniques to extract knowledge from the Web (e.g. Wikipedia articles or general web pages). Notable endeavors in academic community include: Open Information Extraction[33-36], Learning by Reading[37-38],

④http://www.cyc.com/, Apr. 2013.

⑤http://openmind.media.mit.edu/, Apr. 2013.

⑥http://www.keenage.com, Apr. 2013.

⑦http://www.cis.upenn.edu/t̃reebank, Apr. 2013.

DBpedia⑧[39-40], and YAGO⑨[41-42]. The most famous commercial knowledge base may be the Google's Knowledge Graph⑩. Open knowledge bases include freebase.com⑪, Evi.com⑫, wolframalpha.com⑬, etc. In this phrase, a number of large-scale Chinese knowledge bases have also emerged, including SJTU-Zhishi.me[43], Tsinghua-ChineseKB[44], CASIA-KB[45], and the first commercial knowledge base "知立方"⑭ supporting Chinese search engines.

Today, commonsense knowledge is still partially understood, and there are few researchers working in this area. Studying common sense is a huge undertaking which needs a long term of great effort.

1.5 Related Proceedings, Journals, and Resources

Today, researches on common sense have appeared in many AI conferences including, most notably, IJCAI (International Joint Conferences on Artificial Intelligence), AAAI (AAAI Conference on Artificial Intelligence), WWW (International World Wide Web Conference), ISWC (International Semantic Web Conference), SIGIR (International ACM SIGIR Conference on Research and Development in Information Retrieval), ACL (The Association for Computer Linguistics), SIGKDD (International Conference on Knowledge Discovery and Data Mining), SIGMOD (International ACM SIGMOD Conference on Management of Data), and IUI (International Conference on Intelligent User Interfaces). Moreover, there are some symposiums focusing on common sense. International Symposium on Logical Formalizations of Commonsense Reasoning (COMMONSENSE)⑮ is a symposium that focuses on formalizing commonsense reasoning with logic. Commonsense Knowledge Symposium (CSK)⑯ pays attention to issues of CSK acquisition and reasoning as well as applications based on CSK. Related researches have also appeared in famous journals such as Communications of the ACM. There are also several open resources of CSK, including OpenCyc⑰, ResearchCyc⑱, OMCS's semantic network representation ConceptNet⑲ and its matrix-based representation AnalogySpace⑳, DBPedia⑧, Freebase⑪, YAGO and YAGO2⑨.

In this survey, we investigate the work in the field of commonsense knowledge acquisition (CSKA). We first classify CSK acquisition methods and systems into four paradigms according to knowledge sources and employed techniques (Section 2). Secondly, we present the challenges in the field and review how previous work overcomes these challenges (Section 3). From Section 4 to Section 17, we describe various CSK acquisition systems in more detail. Different systems are compared from different dimensions (in Section 18). Finally, we conclude the paper and suggest future directions (in Section 19).

2 Categorization

Building a real-world commonsense knowledge base (CSKB) is a very tremendous project, which cannot be accomplished by employing only a few specific methods. We could categorize CSKA systems under four subsettings, labor acquisition, interaction acquisition, mining acquisition, and reasoning acquisition, based on knowledge source and technique used. This is summarized in Table 1.

Table 1. Different Subsettings of CSKA

Subsetting	Source	Technique
Labor	Human mind	Knowledge engineering
Interaction	Human mind	Computer human interation
Mining	Text	Natural language processing
Reasoning	Knowledge base	Reasoning

2.1 Labor Acquisition

In this subsetting, human minds are viewed as knowledge sources. A team of trained knowledge engineers or untrained volunteers are asked to codify or enter knowledge by hands. Knowledge engineers are usually trained for codifying CSK using a certain formal language, so what they enter can be directly understood by computers. Volunteer contributors could

⑧http://wiki.dbpedia.org/About, Apr. 2013.
⑨http://www.mpi-inf.mpg.de/yago-naga/yago, Apr. 2013.
⑩http://googleblog.blogspot.co.uk2012/05/introducing-knowledge-graph-things-not.html, Apr. 2013.
⑪http://www.freebase.com, Apr. 2013.
⑫http://www.evi.com/, Apr. 2013.
⑬http://www.wolframalpha.com, Apr. 2013.
⑭"知立方" is developed by sogou.com: http://www.sogou.com, Apr. 2013.
⑮http://www-formal.stanford.edu/leora/commonsense, Apr. 2013.
⑯http://csk.media.mit.edu, Apr. 2013.
⑰http://www.cyc.com/platform/opencyc, Apr. 2013.
⑱http://www.cyc.com/platform/researchcyc, Apr. 2013.
⑲http://conceptnet5.media.mit.edu, Apr. 2013.
⑳http://csc.media.mit.edu/analogyspace, Apr. 2013.

enter CSK in natural language through some friendly input interfaces. This paradigm follows the methods of knowledge engineering (KE).

2.2 Interaction Acquisition

In this subsetting, knowledge sources are also human minds, while interaction acquisition emphasizes interaction with human contributors. Interactive property motivates their contribution and makes the entry process more enjoyable and productive. The approaches in this subsetting usually adopt computer human interaction (CHI) techniques. There are mainly two forms of interaction: interactive user interfaces (IUI) and games. An interactive user interface is capable of giving some kind of feedback to its users when they contribute knowledge. Consequently, the users are encouraged by the feeling that "the computer is learning or understanding their entries"[46]. In the game-based subsetting, the activity of entering commonsensical facts is transformed into a more enjoyable process of playing game. Users play the game to be entertained, and CSK is collected simultaneously as a side effect.

2.3 Reasoning Acquisition

In this subsetting, potential CSK can be automatically inferred from pre-existing knowledge bases. Notable reasoning techniques include analogical reasoning, inductive reasoning, and plausible reasoning. Analogical reasoning can discover new properties of a concept from its similar concepts. Inductive reasoning technique (e.g., inductive logic programming) or plausible reasoning can be used to inductively produce rules from basic facts.

2.4 Mining Acquisition

In this subsetting, CSK can be extracted from text corpus automatically. Since the operational object is text, it is unavoidable to use natural language processing (NLP) techniques. Researchers need to design and develop mining systems. The mining systems can take as input either a domain-specific corpora such as newspapers, or a domain-open corpora such as the Web. Web-oriented systems can potentially collect more CSK due to the tremendous scale of the Web, but they are also more difficult to design due to the heterogeneity of the Web. The mining systems can also make use of existing knowledge if they have. The existing knowledge could be used to generate queries for finding commonsensical linguistic patterns or to check whether the elicited knowledge is consistent with existing knowledge.

Based on their employed techniques, mining systems can be further divided into four types: rule-based information extraction systems (RIE), open information extraction systems (OIE), machine reading systems (MR), and knowledge integration systems (KI).

3 Challenges

In this section, we describe challenges in CSK acquisition and review how these challenges are tackled in the four subsettings. Moreover, we also summarize advantages and disadvantages of methods in each subsetting. In addition to the absence of formal definition, we believe five significant challenges exist throughout the process of knowledge acquisition.

3.1 Explicitization

Human beings understand CSK so well that they take CSK for granted and ignore its existence. So, the problem is how to explicitize such "implicit" CSK in human minds. In other words, how can we help human contributors think of what they often take for granted? In engineering setting, human contributors are responsible to recall such implicit CSK, probably by lengthy calm consideration. However, when human contributors fail to recall any relevant CSK even after a long-time consideration, how to decrease the difficulty level of thinking of a piece of "implicit" CSK? An important insight, as in recommender systems[47], is: people find articulating CSK difficult, but they are good at recognizing it when they see it or some information of it. In interactive subsetting, knowledge contributors are prompted by feedback from other people or computers through some forms of interaction, say, a game. In reasoning subsetting, computers present automatically inferred CSK to human reviewers, and the reviewers simply confirm or eliminate these derived knowledge. Usually, the prompts provided by computers rely on a formally represented knowledge base and some effective reasoning techniques.

CSK can be viewed as default assumptions about typical cases of everyday life, and many naturally occurring linguistic patterns in natural language indicate such default assumptions explicitly or implicitly. For example, Google returns 26 500 results using query "houses have doors", and the statement is also implicitly indicated by the sentence "he entered the house through its open door". It was demonstrated that different literary styles or sentences are different in their difficulty grades for knowledge acquisition[48-49]. In order to overcome the implicitness, mining-based approaches make use of the redundancy of large-scale corpus to find an abundant supply of easy-to-extract commonsensical sentences. They often adopt a generate-and-test architecture: finding "suitable" commonsensical expressions scattered over the corpora (e.g., the Web), performing knowledge extraction tasks, and validate the elicited candidates using heuristics. For example, in validation phase, heuristic functions are designed to measure the degree to which an extracted statement is commonsensical. These heuristic functions are often based on corpora statistics, including word co-occurrence[24-25] and pointwise mutual information[50-51]. Finally, the cost of higher automation of mining acquisition is its low precision, i.e., the elicited statements often include many noises.

3.2 Automation

Collecting CSK is a tremendously huge project, so it is important to increase the degree of automation and decrease manual labor. The first step in reducing manual labor is moving from trained engineers to non-trained volunteers. Engineers are trained to codify each item of CSK using a certain formal language (e.g., [18, 20]); in contrast, volunteers could enter CSK in natural language (e.g., [19]), and then the knowledge is automatically transformed into formal representation (e.g., [21]). Manual labor is further reduced by transforming the activity of entering knowledge to more enjoyable interactive process like a game, or by designing knowledge elicitation systems. In games[28,30-31], players never realize that they are contributing knowledge, and their inputs are transformed into knowledge automatically. In a knowledge elicitation system, what the researchers often do is designing linguistic patterns, heuristic rules and functions. When a considerable number of CSK have been collected, they could be used to increase the degree of automation further. Specifically, reasoning approaches could infer new knowledge automatically[22,26-27]; mining approaches could automatically learn more commonsensical linguistic patterns using bootstrapping method[51-52]. Interactive approaches could increase the throughput of new knowledge[30]. Moreover, since CSK has numerous types and we cannot specify the collection targets in advance, further work might be inspired by open information extraction (OIE)[33-34]. Usually, human involvement is unavoidable because the automatically derived knowledge has to be validated by human reviewers.

3.3 Diversity

Commonsense knowledge, on the whole, is domain-independent and type-independent. It is scattered over various domains in the world and covers a large range of kinds of knowledge. The diversity challenge means the difficulty of scaling from some specific domains to general domains, or from some specific kinds of knowledge to a large variety of knowledge.

For knowledge acquisition from people, contributors are inclined to contribute the knowledge of the domain with which they are familiar. For example, a student would like to enter knowledge regarding his/her school life. To achieve the diversity of CSK, more people from various fields should be attracted to do this job. The first step in increasing the amount of contributors is moving from a team of trained knowledge engineers[18,20] to a much larger number of untrained volunteer contributors[19]. Later, the volunteer contributors are increasingly substituted by a large amount of attracted contributors. The attracted contributors are either absorbed by enjoyable entry interfaces with feedback, or contribute knowledge unconsciously during playing games.

To improve the coverage of a CSKB, knowledge extraction systems first move from small domain-specific corpora (e.g., Penn Treebank) to much larger domain-independent Web corpora. Larger corpora can potentially supply a wide variety of knowledge. Furthermore, knowledge extraction systems move from target-specific systems (e.g., a specific relation like *IsA*) to target-independent systems (e.g., automatic discovery of all potential relations). Traditional IE systems often take particular acquisition targets (e.g., *IsA* and *PartOf* relation) as input. However, commonsense knowledge has a huge number of types and it is impossible to specify all of them in advance. So Open Information Extraction (OIE) paradigm[33-34] was proposed to automatically discover all potential targets of interest rather than require them to be specified in advance.

Some reasoning-based acquisition methods, such as analogical inference and inductive logic programming, can operate over knowledge of various types and domains. So they are both independent of CSK's types and domains.

3.4 Efficiency

The efficiency challenge means the difficulty of increasing acquisition rate. Higher efficiency means higher speed of gathering CSK. It is also related to the large-scale characteristics of CSK while it focuses on the large amount of CSK.

For human knowledge sources, the key point of increasing efficiency is to gather more contributors and retain them for a longer time. More CSK would be collected if more contributors are gathered during a shorter period of time. Again, knowledge contributors moved from a team of trained knowledge engineers, to a much larger number of untrained volunteer contributors, to numerous attracted contributors. von Ahn predicted that a popular game for knowledge collection could acquire millions of facts in only a few weeks[28].

For knowledge acquisition from text, since an extraction system can be run at a very high speed, the bottleneck lies in the supply of easy-to-extract corpora. Larger commonsensical corpora can potentially supply larger collection of knowledge. So, knowledge extraction systems moved from small domain-specific corpora (e.g., Penn Treebank) to much larger domain-independent Web corpora. Unfortunately, some earlier Web extraction systems (e.g., the KownItAll system for named entity extraction[52]) require a large number of search engine queries and web page downloads to extract instances for a certain concept of a given ontology (e.g., "Paris is an instance of City"). Hence, their extraction rate is limited by the number of queries allowed by search engines. Even if the use of large-scale web archives became a common practice (hence issuing no query to a search engine), traditional extraction systems need acquisition targets specified in advance and they have to be run once for one acquisition target. For a collection of acquisition targets, such systems have to be run again and again. Open information extraction systems do not need any target as input, and importantly, they could collect all possible knowledge of interest for just one pass over the corpora.

When a certain number of knowledge has been collected and formalized, reasoning techniques[22,26-27] could be used to produce numerous potential commonsensical statements in a very efficient way.

3.5 Evaluation

The evaluation of a system for knowledge collection is often made from the following aspects: the correctness/precision, coverage/completeness, and efficiency of knowledge acquisition, together with the usefulness of the collected knowledge.

Efficiency and correctness are relatively easy to assess. Efficiency can be assessed with the amount of the collected knowledge divided by the time spent on collection. Correctness can be assessed in several ways. The most popular method is relying on a team of human subjects to rate. Another important method is to verify knowledge against large corpus. Based on corpus statistics, the likelihood of the correctness of a piece of knowledge is computed automatically. In addition, an existing knowledge base can be used to test the correctness of new knowledge by consistency checking.

There is little work on the evaluation of coverage/comprehensiveness of a collection of CSK. An important reason may be that there is no gold-standard knowledge base to serve as a standard for measuring coverage. Since coverage is hard to measure directly, some alternative methods have been proposed by researchers. First, human subjects are asked to judge whether the knowledge w.r.t. a concept is fairly comprehensive[21]. Second, machine learning techniques such as nonlinear regression are used to estimate the amount of new knowledge about a domain produced per day[32]. Third, in some Web information extraction system[53], recall is defined with respect to the set of extraction rules that the system uses as well as the sentences with which the system has actually dealt, rather than a hypothetical (but unknown) number of all possible correct extractions from the entire Web.

The collected knowledge can be evaluated against the performance of an application system in concrete tasks. This helps us measure the degree to which CSK improves the performance of the application system. For example, Cyc knowledge base has been used to QA (question and answer) systems[37-38,54]. ConceptNet and AnalogySpace have been applied to word sense disambiguation[6-7] and sentiment analysis[8-9,55].

Moreover, in SemEval-2012 Task 7, Gordon[56] presented a simple challenge for commonsense causal reasoning about everyday events. Given an English sentence describing a state of the world, the competitive systems must choose from two alternatives the one that is more likely the cause or result for the premise.

To summary, the challenge of explicitization corresponds to CSK's implicitness characteristic. The challenges of automation, diversity, and efficiency are closely associated with CSK's large-scale characteristic. They together determine the scalability of an acquisition system. The automation describes whether or not the system is labor intensive in contributing knowledge or creating training data. The diversity describes whether the system is domain-independent or target-independent. The efficiency describes whether the system is time-expensive to accomplish the acquisition task.

4 Cyc

The Cyc project was started by Lenat in 1984 and is now developed by Cycorp, Inc. Its goal is to codify millions of pieces of commonsense knowledge in machine readable form and enable machine to perform human-like reasoning on that knowledge. Cyc contains a Cyc knowledge base (Cyc KB) as well as a collection of Cyc inference engines. The knowledge in Cyc KB is coded in formal language CycL and grouped into thousands of micro-theories. So far, the Cyc KB contains nearly 500 000 terms (including about 17 000 types of relations), and about 7 000 000 assertions relating these terms㉑. The Cyc inference engine can perform general logic deduction as well as other AI well-known inference

㉑The data is retrieved in Nov. 2012 from Cyc's website: http://www.cyc.com/kb, Apr. 2013.

mechanisms over Cyc KB, including inheritance, automatic classification, and so on.

Earlier stage of Cyc mainly depended on manual knowledge entry (Subsections 4.1, 4.2, 4.3). Knowledge authoring tools were provided to help ontologists (e.g., Pretége[57]), subject matter experts[58-59], and volunteers enter and vote various types of knowledge[27]. Cyc also exploited textual resources in one of two ways: either facts or rules were extracted, converted to CycL, and incorporated directly into the knowledge base (Subsections 4.4 and 4.5); or external information sources were integrated as extensions of the knowledge base, not as part of its contents (Subsection 4.6). We finally describe various Cyc-based application systems (Subsection 4.7).

4.1 Coding by Trained Ontologists

Cyc's early stage (from 1984 to around 2000) followed the line of knowledge engineering, manually codifying facts about the world and implementing efficient inference mechanisms on that knowledge. Millions of facts and rules have been formally codified by ontologists skilled in CycL. Lenat believed that such assertions are unlikely to be published in text, even those designed for children, and described the relationship between Cyc and machine learning as chicken-and-egg[18]. A solution to this paradox is handcrafting an appreciable part of human commonsense knowledge before automatic acquisition through machine learning. During building the Cyc KB, the developers learned the following technical lessons[18]. First, each assertion should be considered true in only certain contexts. Thus all assertions were organized into thousands of micro-theories whose assertions share the same set of assumptions. Second, each assertion is assumed true by default instead of using a numeric certainty factors. Third, frame-and-slot language is not expressive enough; the developers therefore designed and used CycL, whose syntax derives from predicate calculus and Lisp programming language.

4.2 Entering by Domain Experts

To alleviate the labor-intensive knowledge entry process, Cyc provides knowledge authoring tools to subject matter experts (SMEs) to extend specific domain knowledge. Such tools include KRAKEN[58] and a dialogue system based on user interaction agenda (UIA)[59]. The number of facts, entered by an SME per hour using natural language, increased from approximately 25 (October 2000) to 35 (summer of 2001)㉒.

4.3 Entering by Volunteers

Cyc's Ground Facts, like "Cafes sell coffee", are relatively straightforward to obtain and represent. So, ground facts can be obtained from untrained volunteers through more convenient entry interfaces such as Factivore and Predicate Populator[27]. Factivore is a template-based knowledge entry tool, through which users can fill the blanks in natural language; the entered facts are then automatically translated into valid formal representations by Cyc Natural Language System. Predicate Populator allows its users to select among plausible choices, and the users find it is more convenient than filling in blanks. All knowledge entered through these interfaces can be validated by several ways. First, anything incompliant with existing knowledge can be automatically filtered. Second, multiple reviewers can be asked to vote on the ground facts. Third, automatic validation methods could verify ground facts in other corpus.

In addition, volunteers can contribute knowledge via The FACTory㉓, which is a fun game aiming to help improve Cyc's thinking.

4.4 Extracting Facts from Web

Matuszek[37] presented a method of populating Cyc knowledge base with Cyc preexisting knowledge and the Web accessed via Google.

The acquisition process could be presented as follows. First, generate interesting and productive CycL queries. A query is formed by combining a binary predicate p and a most frequent value v of the type constraint on each argument of the predicate, denoted by $p(v, ?var)$ or $p(?var, v)$, where the other argument is represented by a variable. Second, translate the queries into English search strings and pass them on to the Google API. Third, translate the relevant component of search result into one or more GAFs (Ground Atom Formular) in CycL via Cyc's natural language parsing module. Fourth, verify the GAFs via KB consistency checking, Google retrieved information, and human review. Finally, knowledge that pass through all verification is asserted into knowledge base.

4.5 Generating Rule Automatically

Creating correct rules in a formal representation is a challenging task for Cyc's knowledge engineers. Eliciting rules through interfaces from untrained users is slow and fairly ineffective. Witbrock[27] believed the difficulty was due to "the mismatch between the requirements of formal reasoning and the way humans

㉒http://cyc.com/cyc/cycrandd/areasofrandd_dir/kfd, Apr. 2013.
㉓http://game.cyc.com/, Apr. 2013.

conceptualize their own reasoning processes". In order to generate rules automatically, they applied FOIL, an inductive logic programming system[60], to the ground facts of the Cyc KB. The rule reviewers found that, among the automatically derived rules, only 7.5% were correct and 35% required only minor editing. In addition, rule evaluation could be performed at an average rate of 20 rules per hour for one reviewer.

4.6 Integration with External Data

Cyc has the ability of integrating itself with external databases, or semi-structured text (e.g., Wikipedia), or even free text.

4.6.1 SKSI

SKSI techniques[61] can be used to integrate information ranging over websites, databases and background knowledge to answer integrative queries. When knowledge sources are wrapped with wrappers, they can communicate with Cyc. These wrappers provide semantic information about the knowledge sources being mapped. Users can pose queries to Cyc that combines results from multiple external knowledge sources with the Cyc knowledge base. For example, answering "Is it raining somewhere in New England?" requires three resources. The website forecasting weather knows the weather about any city but does not know what cities are in New England, and another geographic database has information about every city in every US state, and Cyc KB knows which states constitute New England. Once Cyc integrates the other two sources, the question can be answered by posing queries to the Cyc system.

The SKSI technology uses a subset of the CycL knowledge representation language to build a conceptual model of a structured knowledge source. The conceptual model of a source's data content can be used to produce code modules. The code modules are responsible for translating appropriate fragments of CycL queries into SQL queries, executing the queries on remote SQL query engine, retrieve the results, and translate them back into CycL.

The SKSI architecture contains three interrelated layers of knowledge about the source's structure and data content. The *access knowledge layer* contains the knowledge for identifying, connecting, and submitting requests to the source on the network. The *physical schema layer* contains the knowledge that preliminarily describes how data is organized in the source. The *logical schema layer* interprets the semantics of the data content in terms of the Cyc ontology.

In that paper, the conceptual model of a source was illustrated with a National Weather Service example. The authors claimed "the real value of the work is in providing a proof of concept of integrating existing mapping and Web extraction technology with the semantic information available in large ontologies".

4.6.2 Integration with Wikipedia

Medelyan and Legg[62] designed a step-by-step heuristic mapping procedure, which mapps Cyc terms to Wikipedia articles that describe corresponding concepts represented by the terms. To overcome terminology differences, it uses rich synonymy relations in both resources. To deal with sense ambiguity, it analyzes semantic similarity of possible mappings to context categories in the neighboring Cyc ontology. If several Cyc terms are mapped to the same article, two consecutive tests based on CSK in Cyc KB are used to further correct such mappings. On 9 333 manual alignments by one person, the method achieves an F-measure of 90%. On 100 alignments by six human subjects, the average agreement of the method with the subject is close to their agreement with each other. The mapping covers 62.8% of Cyc categories relating to commonsense knowledge.

4.7 Cyc-Based Applications

4.7.1 Cyc-Based Question Answering

MySentient Answers[10] is a commercial prototype system for question-answering that integrates the deductive QA module implemented by Cyc and the IR-based QA module developed by CNLP㉔. MySentient generates better results than either approach could yield individually. Cyc KB improves the system's overall ability from three aspects: firstly, expanding key concepts of a question to improve NLP-based passage retrieval; secondly, generating question types for passage retrieval; thirdly, translating the results of deductive QA to natural language strings that explain answers to an user.

The Cyc Analytic Environment (CAE㉕) provides a multi-domain semantic platform to provide analysts with answers to complex questions. The CAE allows analysts to pose questions in English as well as other interfaces appropriate to their domain. The CAE interprets the analysts' questions, identifies the information sources required to answer it, and integrates domain and general knowledge with open-source and enterprise data to provide answers in English or via other modalities appropriate to the domain and the informa-

㉔http://www.cnlp.org/, Apr. 2013.

㉕http://www.cyc.com/enterprise-solutions/solutions, Apr. 2013.

tion to be conveyed (e.g., maps, time-lines, or charts). The analyst may drill-down into the answer to view the CAE's rationale, including all the supporting rules and data. CAE applications in several domains have been developed or are under development, including: medical records analysis, intelligence/counter-terrorism analysis, financial analysis.

Cycorp and IBM have worked together in order to build a QA system QUIRK㉖ with desired abilities of: 1) not only answering a question but also providing a justification for the answer itself; 2) integrating heterogeneous data sources ranging over free text, databases, and knowledge bases; 3) answering a question by combining results from a collection of data sources. QUIRK will employ a blackboard architecture, where a combination of agents will jointly contribute to the task of answering questions. Cyc's inference engine and IBM's GuruQA Information Retrieval engines are both agents interacting with the blackboard.

4.7.2 *Other Applications*

Cyc has been used in word sense disambiguation[5], semantic web[63-64], integration of heterogeneous data sources[61], intelligent search, etc. You can find many real-world business solutions at the website of Cycorp.

5 ThoughtTreasure

Eric Muller[65] started the development of ThoughtTreasure in 1994, established the company Signiform to pursue commercial applications of ThoughtTreasure in 1997, and closed the company and stopped further development of ThoughtTreasure in 2000㉗.

ThoughtTreasure contains a commonsense knowledge base and an architecture for natural language understanding. The commonsense knowledge base contains declaratively and procedurally represented commonsense knowledge. It include 35 023 English words/phrases, 21 529 French words/phrases, 51 305 commonsense assertions, and 27 093 concepts㉘. Some sample items are: "Soda is a drink", "At the end of a phone call, one says goodbye and hangs up". ThoughtTreasure uses multiple representations, including logic, finite automata, grids, and scripts. The architecture of ThoughtTreasure for natural language understanding includes text agency, syntactic component, semantic component, generator, planning agency and understanding agency.

ThoughtTreasure could understand simple stories by means of simulation. The key idea of its approach is simulation. That is, to make a computer understand stories, it needs to build a computer that can construct simulations (i.e., models) of the states and events described in a story. A simulation is a sequence of states, each of which stores a snapshot of the mental world of each story character and physical world. To achieve this, different agents in ThoughtTreasure are employed to work on different parts of the simulation. Given a story, the understanding process is to maintain a simulation of the story. Afterwards, simple questions about the story can be answered.

An application of ThoughtTreasure is a smart calendar[66]. The smart calendar can extract information from an entry, perform commonsense reasoning, and help fill in missing information or point out potential problems. Eric Muller gave the following example: for a date "lunch with Lin at frank's steakhouse", the calendar would warn "You are taking Lin who is a vegetarian to a steak house" if it know "Lin is a vegetarian".

6 HowNet

HowNet[20] is an online extra-linguistic knowledge system for meaning computation in human language technology, uncovering relationships between concepts or attributes of concepts. Dong leads the research and development of HowNet. Its knowledge dictionary has more than 160 000 records, which are codified with Knowledge Database Mark-up Language (KDML). For example, the concept "Doctor|医生" defined below can be paraphrased as "a human being, who has the attribute of occupation; he gives medical treatment to; he belongs to the domain of medicine"[20].

$$\begin{aligned} DEF = \{ & human\text{–人} : \\ & HostOf = \{Occupation\text{–职位}\}, \\ & domain = \{medical|\text{医}\}, \\ & \{doctor\text{–医治} : agent = \{\sim\}\}\ \}. \end{aligned}$$

HowNet has several characteristics as follows. First, all concepts are denoted by words and expressions in both Chinese and English. Second, all concepts are defined on the top of sememes, the smallest units of meaning. In our example, "human|人", "Occupation|职位", "medical|医", "doctor|医治" are all sememes. All sememes have been classified into four subclasses, including entity, event, attribute, and attribute-value; they are also organized into taxonomies respectively. To generate a comprehensive collection of sememes, en-

㉖http://www.itl.nist.gov/iaui/894.02/projects/aquaint/proceedings/kickoff/ProgramSummary/Cycorp_QUIRK_summary.doc, Apr. 2013.

㉗Erik Mueller moved to IBM Research, where he was a member of the team that developed Watson (computer).

㉘http://web.media.mit.edu/~lieber/Teaching/Common-Sense-Course-02/ThoughtTreasure.ppt, Apr. 2013.

gineers first listed all senses of 4 000 frequently-used Chinese characters and deleted all the duplications to form an initial set of around 1 500 sememes; then, using the sememes, the engineers did trial tagging over 50 000 Chinese words and expressions as well as their equivalent expressions in English. After three-year experiment and modification, they accomplished this labor-intensive and time-consuming task. Third, a number of meaning computation devices have been developed to test HowNet, including concept relevance calculator, concept similarity measure, and so on.

7 Common Sense Computing Initiative

The Common Sense Computing Initiative (CSCI) group at the MIT Media Lab㉙ focuses its research on common sense computation, specifically including "learning and inferring common sense knowledge, creating grounded applications, and understanding how people talk about their beliefs and opinions"㉚. Famous projects include Open Mind Common Sense (OMCS), OMCS's semantic network representation ConceptNet and matrixed-base representation AnalogySpace. The various applications of collected CSK can be found at the website of CSCI㉛.

7.1 OMCS

The project Open Mind Common Sense (OMCS) (http://openmind.media.mit.edu) aims to collect commonsense knowledge via its website from ordinary people over the Web. OMCS supports collecting knowledge in multiple languages, including English (1 040 067 statements), Chinese (356 277), Portuguese (233 514), Korean (14 955), Japanese (14 546), Dutch (5 066), etc.㉜ They defined CSK as "all those aspects of the world that we all understand so well we take them for granted". So far, OMCS has built the second largest commonsense knowledge base after the Cyc project, containing about 1 000 000 statements in English and plenty of statements in other languages.

7.1.1 Traditional Entry Interfaces

Original OMCS was designed to collect knowledge in only English. Singh *et al.*[19] designed 25 kinds of activities for collecting CSK from volunteers. Each activity had its own interface to facilitate users to enter one kind of CSK. For example, given a short story "Bob had a cold and Bob went to the doctor", a user might enter knowledge like "Bob was feeling sick", "The doctor made Bob feel better", etc. All users were encouraged to input sentences that even a child could understand.

Singh[46] evaluated the collected assertions of OMCS and corrected some of its deficiencies. Seven human judges were asked to rate 3 009 items (i.e., commonsense assertions) selected from OMCS corpus. After removing the items that were judged as garbage (about 12.3%), the remaining items were rated based on four attributes: generality, truth, neutrality, and sense. Average ratings for the four attributes showed that, the items might range from specific ones to general ones, a large part of them (75%) were roughly correct, most of them (82%) were relatively unbiased, and most of them (85%) made sense. Another experiment showed that most items (84%) were known by people at grade or high school level. In addition, an important inspiration from OMCS is that entry interface should be easy to use because "participants would leave if they encounter difficulties".

In a latter version, OMCS allows untrained volunteers to select a concept he/she is interested in and fill the templates associated with that concept. For example, given the template "__can be used to__", one could fill in "a pen" and "write", or more complex phrases such as "take the dog for a walk" and "get exercise". Also it allows volunteers to construct new templates when necessary and to evaluate knowledge of each other. Compared with Cyc, OMCS has greater efficiency of collecting commonsense statements, while its statements contain more noises and need to be formalized.

7.1.2 Integration with External Resources

More recently, OMCS incorporated Verbosity㉝ as a novel way for people to contribute to OMCS. Also, the data in Verbosity was filtered and adapted for use in OMCS[67]. Experimental results show that the filtered data of Verbosity has a comparable quality to OMCS's existing data.

7.1.3 Open Mind Sister Projects

There are several sister projects to OMCS. Some projects use similar collecting methods to accumulate CSK in different languages, including OMCS in Portuguese[68], OMCS in Dutch[69], OMCS in Brasil㉞, and GlobalMind[70] (including Korean, Japanese, and

㉙http://csc.media.mit.edu, Apr. 2013.
㉚http://csc.media.mit.edu/node/5, Apr. 2013.
㉛http://csc.media.mit.edu/node/7, Apr. 2013.
㉜The data was retrieved in Nov. 2012.
㉝Verbosity is a game with purpose for collecting CSK. We will introduce it later.
㉞http://www.sensocomum.ufscar.br:8080/omcs, Apr. 2013.

Chinese). Some projects aim to collect knowledge in a certain domain or knowledge of certain kind. Open Mind Indoor Common Sense (OMICS)[16] restricts the domain to indoor home and office environments. The knowledge collected by OMICS was applied to indoor mobile robots. Common Consensus[29] is a Web-based game for collecting and validating knowledge about human's goals. Open Mind Common Sentics[71] is an emotion-sensitive intelligent platform for collecting affective commonsense knowledge through label sequential rules, crowd sourcing, and GWAP (Game with A Purpose[72]) techniques; it also provide an NLP tool for extracting cognitive and affective information relevant to short texts.

7.2 Open Mind Commons

Open Mind Commons[73] is a novel interactive interface that can supply interesting feedback to and dialogue with its users, which confirms to users that the computer is understanding and learning from the knowledge they enter. For a topic of interest, the system makes analogical inferences based on the knowledge it already has on the topic. Then, the system generates some relevant questions and asks the user to confirm them. These questions help the system fill in gaps in its knowledge and make its knowledge more connected. For example, the system asks "A bicycle would be found on the street. Is this common sense?". Moreover, justification for the statement is also given: "A bicycle is similar to a car" and "I have been told that a car would be found on the street". Users can click either "Yes" button to confirm or "No" button to reject the potential inference. If a user rejects a potential inference, the computer will ask the user to change it to make it true. Moreover, the interface allows its users to refine knowledge entered by other users and see ratings of their contributions by other users. Users can also see new inference results made on the basis of their new contributions.

7.3 20 Questions

The game 20 Questions[30] (20Q) aims to make computer figure out what kind of thing is being discussed with its players. More specifically, the computer produces and asks natural language questions based on the knowledge in OMCS. According to what a player answers, the computer attempts to guess the object in question. The design objectives of this game have two aspects: first, motivating volunteer contributions; second, increasing the throughput of new knowledge when interacting with a user. In order to produce reasonable questions, statistical classification methods are used to discover the most informative characteristics in the OMCS knowledge base. A hierarchy of concept clusters, together with the features that define the clusters, is created using a beta-binomial mixture model. The model determines which features will be the most informative in distinguishing clusters, and these features are used to produce questions. After a user answers all the questions, the object to be guessed will be projected into a cluster, and the most salient characteristics of that object will be learned. Experimental results showed that users liked the game and that the game increased the throughput of new knowledge and the speed of knowledge acquisition.

7.4 ConceptNet

ConceptNet is the semantic network representation of the knowledge collected from OMCS projects as well as other external resources. Since 2002, ConceptNet has experienced several times of revision.

7.4.1 *ConceptNet-2*

ConceptNet-2[21] contains 1.6 million assertions over 30 000 concepts. The nodes of it denote concepts, such as *kitchen table* and *eat breakfast*, and the edges connecting two nodes denote semantic relations between concepts, such as *UsedFor*. Two nodes and one edge constitute an assertion, for example, *UsedFor*(*kitchen table, eat breakfast*). There were 1.25 million *k-line*[35] assertions indicating generic semantic relations such as *ConceptuallyRelatedTo*, and other 400 000 assertions covering 20 relation types (e.g., *EffectOf*).

ConceptNet-2 was automatically constructed by three stages: extraction, normalization, and relaxation. First, binary predicates were extracted from the semi-structured OMCS using regular expressions with syntactic and semantic constraints. Each argument of predicate is composed of combination of four syntactic components: verbs, noun phrases, prepositional phrases, and adjectival phrases. For example, from the semi-structured sentence *the effect of* [*falling off a bike*] *is* [*you get hurt*], binary predicate *EffectOf* (*falling off a bike, get hurt*) was extracted. Second, all concepts in binary predicates were normalized. For instance, phrases were stripped of determiners and modals, and words were stripped of tense and number. The concept *falling off a bike* was normalized as *fall off bike*. Third, heuristic relaxations were performed over the normalized assertions to obtain additional more generalized knowledge. For example, *IsA* (*apple, red round object*) and *IsA* (*apple, red fruit*) imply *PropertyOf* (*apple, red*).

[35] A term introduced by [74].

In evaluation, five human judges selected 100 concepts (10 common and 90 different) and assessed the knowledge associated with them. They found that the coverage of assertions about a concept is moderate and widely varied, and the noisiness is low and relatively unvaried. They also discovered the missing rate of concepts is more than tolerable because one out of each ten concepts desired by the judges is unavailable.

ConceptNet-2 tool-kit supports various contextual commonsense reasoning tasks. It implements three node-level functionalities, including contextual neighbours, analogy, and projection. It also implements four document-level functions, including topic-gisting, disambiguation and classification, novel-concept identification, and affect sensing. Compared with WordNet[75] and Cyc, Liu and Singh [21] claimed that WordNet was optimized for lexical categorization and word similarity, Cyc is good at formalized logic inference, but ConceptNet-2 does well at making practical context-based reasoning over text.

7.4.2 ConceptNet-3

ConceptNet-3[76] was constructed following a four-layer software design pattern named CSAMOA (Common Sense Application Model of Architecture[77]). The new architecture and the embedded NLP tools enable ConceptNet-3 to readily extract knowledge from different forms of natural language input and convert its edges (i.e., binary predicate) back to natural language.

In ConceptNet-3, each assertion is accompanied with a score of reliability and a parameter of polarity. A high score of reliability means there are multiple statements of OMCS mapping to the predicate, and this score can be increased or decreased by one point by one reviewer. The polarity parameter has a value of either "1" or "−1", where the negative value indicates a negative statement. An example of negative statement is *people do not want to be hurt*.

Havasi *et al.* conducted an experiment for investigating how often the assertions of ConceptNet-3 aligned with those of WordNet and Brandeis Semantic Ontology[78]. They did not compare ConceptNet-3 with the Cyc KB because the structure of Cyc was not easy to be aligned with that of ConceptNet. Test data contain three relationship types including *IsA*, *PartOf*, and *UsedFor*. Before comparison, all concepts of these relationships (or arguments of predicates) were normalized to a single word. The major operation of comparison is to check whether an equivalent relationship also holds between two corresponding concepts in WordNet or BSO. The comparative experiment drew the following conclusion: ConceptNet-3 has some overlap with the two expert created resources (from 20% to 45%), but many useful statements of ConceptNet-3 do not appear in the other two resources, e.g., *a son is part of a family*.

7.4.3 ConceptNet-4

ConceptNet-4 could represent all knowledge from the family of OMCS projects in different languages, including English OMCS, OMCS no Brasil, OMCS in Dutch, and GlobalMind (in Korean, Japanese, and Chinese). It also incorporates knowledge collected from online games. Moreover, ConceptNet-4 provides a web API for accessing and querying its data.

7.4.4 ConceptNet-5

ConceptNet-5 (http://conceptnet5.media.mit.edu) is the latest version released in May 2012. Its development was led by Rob Speer and advised by Catherine Havasi. It aims to "grow freely and absorb knowledge from many sources, with contributions from many different projects"[79].

ConceptNet-5 has the capability of "blending" (a term introduced in [80]) many different knowledge sources. In other words, it collects sources of facts rather than facts. In addition to the content of ConceptNet-4, it also includes:

- knowledge from the English Wikipedia with two extraction tools: DBPedia extracting knowledge from the infoboxes that appear on articles, and ReVerb extracting relational knowledge from the actual text of each article;
- knowledge from the English Wiktionary;
- knowledge from WordNet 3.0[75,81];
- knowledge collected from games, including English word game Verbosity[28], and nadya.jp in Japanese, and "pet game" in Chinese[31].

Moreover, ConceptNet-5 keeps growing as its developers find new knowledge sources and the ways to integrate their knowledge. By April 2012, ConceptNet-5 contains 12.5 million edges, representing about 8.7 million assertions connecting 3.9 million concepts. Its most represented language is English.

ConceptNet-5 is conceptually represented as a hypergraph. Nodes denote concepts (i.e., words or phrases) and edges represent relations between concepts. A relation can be either an interlingual relation, such as *IsA* or *UsedFor*, or an automatically-extracted relations that are specific to a language, such as "is known for" or "is on" in English. An assertion might be represented by either an edge, when it is learned from some knowledge source, or a large bundle of edges, when it is learned in many different ways.

ConceptNet-5 separates data from the interface for accessing that data. The data is a flat list of edges, available in JSON or as tab-separated values. The flat file is very convenient to do statistics, merge, and convert format, especially as the input for many machine learning tools. However, a flat file is not particularly efficient for querying, so the developers use Apache Solr and MongoDB to build indexes on top of the data. Users can efficiently search the data for edges with many kinds of queries, such as all lemmas (normalized words) in edges and prefixes of any URI allows. ConceptNet-5 can be freely downloaded, redistributed or reused under licenses.

The developers asked people to evaluate a random sample of the edges of ConceptNet-5 through a website. People could classify the statement as "Generally true", "Somewhat true", "I don't know", "Unhelpful or vague", "Generally false", and "This is garbled nonsense". During two days, 81 responses (all English speakers) were returned, including the evaluation of a total of 1 888 statements, or 1 193 if "I don't know" answers were ignored. According to the sources where the edges come, the results were grouped as follows. Existing data of ConceptNet-4, Wiktionary (translation), DBPedia and Wikipedia performed very well, where the rate of "Generally true" or "Somewhat true" is about 80% out of definite responses (i.e., the responses discarding "I don't know"). The performances of WordNet data, Wiktionary (English-only), and Verbosity were barely satisfactory, where the responses of "Generally true" or "Somewhat true" falled in between 60% and 70% out of all definite responses. They explained that the low precision of WordNet data was probably due to that WordNet edges inherently generate assertions that sound too unnatural. Not surprisingly, the ReVerb data performed poorly since extracting knowledge from free text is the hardest task. The few negative-score edges (previous ConceptNet contributors rated them as "false") were mostly rated as "false" as expected.

7.5 AnalogySpace

AnalogySpace[22] is an analogy reasoning technique that can generate the analogical closure of a knowledge base through dimensionality reduction.

The ideological predecessor of AnalogySpace is Leaner[26], which can discover potential knowledge from OMCS knowledge base by "cumulative analogy" via two steps as follows. In the similarity step, the nearest neighbors of a given object are identified based on the number of features they share. In the analogy step, it is hypothesized that the given object has a certain feature if its similar neighbors have the feature. For example, the object *newspaper* has similar objects *magazine*, *book*, and *map*. Learners would guess *newspaper* also contains information if *magazine*, *book*, and *map* all contain information.

When the dimension of concepts or features is very high, the similarity notion employed by Leaner will become too brittle to work well. The similarity is less powerful to capture the similar concepts that share features that are themselves similar but not identical. AnalogySpace corrects this deficiency by employing another generalized similarity, adding more resistance to noise and more power to the process of analogy. Based on the assumption that similarity is a linear operation over vectors, they run truncated Singular Value Decomposition (SVD) on the concept/feature matrix $\boldsymbol{A}$ and get an approximate matrix $\boldsymbol{A}_k$. As a result, all concepts (or features) in the space of features (or concepts) are projected into a reduced-dimensional space of eigenvectors. The dimensions of this new space represent the most salient aspects of the knowledge base. If a concept and a feature do not connect in the matrix $\boldsymbol{A}$ but their corresponding entry in the matrix $\boldsymbol{A}_k$ has a high value, then the concept and the feature will probably comprise a true commonsense statement.

To evaluate the inference using AnalogySpace, 40 college students were asked to rate the truth of assertions derived by AnalogySpace. Experimental results showed that these assertions were often agreed by human judges, more than 70% of which were rated as "Generally true" or "Sometimes/Somewhat true".

Two tools are provided for users to work with AnalogySpace. First, Divisi[36] is an open source software package. It is a library for reasoning by analogy and association over knowledge bases that can be represented as semantic networks. Second, Luminoso[37] is a visualizer for AnalogySpace, which visualizes the semantics that AnalogySpace brings to any set of text documents. Users could interactively analyze and understand their natural language data, discovering differences in the meaning of those documents (e.g., the opinions, perspectives, and topics they express).

7.6 ConceptMiner

Eslick[51] studied the issue of automatically extracting CSK from the Web and developed a system called ConceptMiner. He chose three specific relationships as test samples: *DesireOf*, *EffectOf*, and *CapableOf*. The system ConceptMiner employs extraction patterns and makes use of the knowledge in ConceptNet. It comprises three main parts: a pattern miner, an instance miner, and a number of filters.

[36] http://csc.media.mit.edu/divisi, Apr. 2013.
[37] http://csc.media.mit.edu/analogyspace/luminoso, Apr. 2013.

The pattern miner is responsible for discovering linguistic patterns expressing some specific relationship from the Web. First, some relation instances of ConceptNet (e.g., *DesireOf* (dog, attention)) are used to derive queries (e.g., "dog * bark"). Second, search the Web using the queries for textual contexts. One of top results is "My/PRP dog/NN loves/VBZ attention/NN ./." (after POS tagging). Finally, generalize over the specific contexts to yield more general patterns. For instance, the above context is generalized into pattern of the form: "$\langle X\rangle/NN\ \ loves/VBZ\ \ \langle Y\rangle/NN$".

The instance miner is responsible for finding more relation instances with the derived patterns. It first derives search queries from patterns, then searches the Web, and finally extracts potential relation instances.

A sequence of filters are used to strip bad instances in the post-processing stage. Some useful filters include: concept filter remains only concepts which are present in ConceptNet; PMI filter uses pointwise mutual information to cut off the instances that have too high or too low values; inferential distance strips negative ones using the inferential distance between two concepts in ConceptNet. The experimental result confirms that the Web contains a lot of relational CSK which is similar to that of ConceptNet.

8 Game with A Purpose

Luis von Ahn first proposed the paradigm *Games with A Purpose* (GWAP)[72,82]: games are played by people and produce useful computation as a side-effect. He designed a game named *Verbosity* to collect CSK from players[28]. Inspired by von Ahn's work, Hsu's research group⑱ developed *Rapport*, *Virtual Pet*, and *GOKC* to collect CSK from people on the Web.

8.1 Verbosity

Verbosity[28] is an game with purpose of collecting CSK, which is designed for two players — a "narrator" and a "guesser". Given a word, the narrator can offer some hints about it in order to get the guesser to guess the word. Meanwhile, the guesser must guess the secret word from the clues. In Verbosity, the clues take the form of sentence templates with blanks, and the narrator can fill in the blanks with any string without containing the secret word. For example, given the word *laptop*, the narrator might prompt the guesser: "it has a keyboard". In addition, Verbosity allows a single player to play the game with a "bot" partner which can be created using the collected data.

In their evaluation experiments, each player played for more than 20 minutes on average, and some played for over 3 hours. These numbers show that Verbosity is fun to play. Also, the design of Verbosity ensures that the collected facts have a high accuracy. 85% of the 200 randomly selected facts were rated as true by all six raters, and among the remaining 15% facts, many of them were debatable.

8.2 Community-Based Games: Rapport Game and Virtual Pet Game

Kuo and her colleagues[31] explored how games for CSK collection could take the advantage of rich interactions in an online community. In order to make players retain/return for sustained contribution, they investigated modes of interaction among players and built user model according to their participation goals. Two games based on social community platforms were designed and implemented, and they both collect CSK in a question-answering fashion. The Rapport Game help users to make friends with strangers or enhance social connection with their friends by asking and answering questions with matching answers. In Virtual Pet Game, players teach their pets simple facts in order to raise the intelligence of their virtual pets.

Kuo *et al.* analyzed quantitatively the collected data over the first six months since the launch of games. Virtual Pets Game outperformed OMCS[19] in collection speed and quantity. That is, less contributors used less time in collecting more CSK. The Rapport Game did not perform as expected, which suggests interface design may strongly influence the collection result. Different from Verbosity[28], community-based games benefit from their friend-invitation mechanism, which brings the most active players' friends into the Rapport game and in turn increases speed and quantity of knowledge collection.

8.3 Goal-Oriented Knowledge Collection

In order to fill the gap between an existing CSKB (i.e., Chinese ConceptNet) and the ideal complete CSKB, Kuo and Hsu[32] presented an approach, Goal-Oriented Knowledge Collection (GOKC), to populating knowledge within a target domain (e.g., sport field). The approach issues questions to its players and collects their answers as commonsense knowledge. Unlike other games such as Verbosity[28] and Virtual Pet Game[31], which have shown their efficiency in collecting large amount of knowledge from online users, this new approach shows its effectiveness in collecting new knowledge.

⑱The Semantic Group at Intelligent Agents Lab led by Hsu at National Taiwan University, https://sites.google.com/site/iagent-commonsense/home, Apr. 2013.

They got two findings after doing experiments on collecting knowledge from three domains: electric appliance, food, and sport. First, the amount of new knowledge with regards to a specific domain decreases over time. Second, the rate of decay depends on the size of domain knowledge. They utilized a heuristic function $NewInfo_{k-day}$ to estimate, for a given question, the rate of new knowledge produced at the k-th day. If its value w.r.t. a question is lower than a threshold (empirically decided), another new question will be put forward to players actively. New questions are inferred using a relation network, where two relations $Relation1$ and $Relation2$ are linked if $Relation1(x, y)$ and $Relation2(y, z)$ are true for every possible concept x, y, and z. For example, if we know *AtLocation* is linked with *HasSubevent* and we get an answer "classroom" to the question "You are likely to find___in a school", we can ask players a new question "One thing you will do when you in classroom is___".

The approach successfully drew 10 572 answers for the "food" domain. All the collected answers were verified by online voting. Experimental results show that 92.07% of them are good and 95.89% of them are new, which is a significant improvement over the original Virtual Pet game.

9 KNEXT

Schubert[23] believes that "*there is a largely untapped source of general knowledge in texts, lying at a level beneath the explicit assertional content*". He led the development of KNEXT system, which extracts CSK of the form *general possibilistic propositions* from textual corpus Penn Treebank. There, *general* means it is not predetermined specific kind of facts such as part-whole or causality, and *possibilistic* means the propositions are possible in the world. For example, given the sentence "he entered the house through its open door", they can infer that "it is possible for a male to enter a house", "houses probably have doors", "doors can be open", etc.

Rather than specialized extraction patterns for specific relationships (e.g., *is-a* and *part-of*), Schubert employed general phrase structure coupled with compositional interpretive rules to derive general possibilistic propositions from the Penn Treebank. The process is described in detail as follows. Given a parse tree in Penn Treebank, the system uses general phrase structure patterns to match the tree in bottom-up fashion. For each successfully matched sub-tree, the system first abstracts the interpretations of each essential constituent of it. Abstraction operations include stripping modifiers and inessential conjuncts and generalizing individual terms to types. Such abstraction operations often yield general presumptions about the world underlying the assertions. For example, "a small office at the end of a long dark corridor" could be abstracted to "an office". Then, compositional interpretive rules are utilized to combine all abstracted interpretations and finally derive a general possibilistic proposition.

Schubert and Tong[48] evaluated the general possibilistic propositions extracted from the Brown corpus in the Penn Treebank. Five human judges (including two authors) were asked to rate the extracted propositions using six levels: reasonable general claim, reasonable but specific or obscure, vacuous, false, something missing, and hard to judge. A human judge was allowed to assign one and only one level to one proposition. Firstly, Schubert and Tong investigated the dependence of extracted propositions on literary style. Experimental result showed that literary style did make a difference to the quality of extracted propositions. Secondly, Schubert and Tong assessed the overall quality of extracted propositions. Nearly 60 percent of propositions were marked as "reasonable general claims" by at least one human judge, suggesting that their method might be of some use in deriving world knowledge. They claimed that their work was complementary to Cyc rather than an alternative.

10 Target-Specified Acquisition

In this section we focus on work for collecting CSK of specific relations, such as entailment or paraphrasing, temporal relation (e.g., probably-follow or as-before), contradiction, causality, and so on.

10.1 Textual Entailment

Paraphrases are pairs of natural language expressions (e.g., phrases or sentences) that convey almost the same information. Textual entailments are pairs of natural language expressions such that a human who believes the first element of a pair would most probably conclude that the other element is also true. Paraphrasing and textual entailment methods/systems recognize, generate, or extract pairs of natural language expressions that are paraphrasing and textual entailment. A paraphrase can be seen as a bidirectional textual entailment, so the methods from the two areas are often similar. So far a large number of work for paraphrase or textual entailment have been published. More details could be found in the survey paper [83]. Here we are concerned about extraction methods for paraphrases (possibly templates) or textual entailment (templates).

In [94], the authors classify extraction methods into three groups. 1) Extraction methods based on the distributional hypothesis. This kind of methods are based on the assumption that linguistic expressions (or templates) occurring in similar contexts (or with similar slot values) tend to have similar meanings, and hence regard such linguistic expressions (or templates) as paraphrases. 2) Extraction methods that use bootstrapping. This kind of methods start with seed linguistic expression templates or seed values of slots of templates. Then, the iteration process repeats the two steps: using slot values of templates to obtain templates, and using the new templates to obtain more slot values. The iteration process terminates when no new templates or no new values of slots of templates can be obtained from the corpus, or when it reaches a maximum number of iterations. 3) Extraction methods based on alignment. This kind of methods usually use two comparable corpus or a parallel corpora where sentences have to be aligned or have been aligned.

Ideally one could evaluate extraction systems by measuring both precision (i.e., the percentage of the extracted pairs that are correct) and recall (i.e., the percentage of all paraphrase or entailment pairs that have been extracted). Alternatively, one may count the total number of extracted pairs at different precision levels. Human judges are often employed to rate the extractions. In addition, extraction methods could also be indirectly evaluated against various application systems. For example, one may measure how much the extracted pairs improve the performance of information extraction, query expansion, summarization, phrase alignment in monolingual parallel corpora, etc.

10.2 Extracting Specified Event Relations from Japanese Text

Torisawa and Hashimoto at National Institute of Information and Communications Technology (NICT, Kyoto, Japan) designed and implemented systems for extracting specified event relations from Japanese text.

Torisawa *et al.*[24−25] used unsupervised methods to elicit commonsensical inference rules between events from coordinated sentences in Japanese newspapers. He extracted two kinds of inference rules with temporal constraints: *probably-follow* rule[24] and *as-before* rule[25]. An inference rule between events e_1 and e_2 holds if and only if there is an implication relation and a particular temporal order *probably-follow* or *as-before* between them. For example, an *as-before* rule is "if someone enforces a law, usually someone enacts the law at the same time as or before the enforcing of the law".

The two kinds of inference rules are extracted using similar procedures. First, a rule candidate is formed by extracting two event expressions such as "(enforce, law)" and "(enact, law)", where the two expressions come from coordinated phrases or sentences and the two verbs share the noun as their arguments. Second, each rule candidate is rated using a heuristic scoring function, which is based on co-occurrence between verbs and co-occurrence between verb and noun. Finally, the rules whose scores are greater than certain threshold are viewed as reliable rules, whereas the others are thrown away.

Hashimoto *et al.*[84] introduced the concept of *excitation* (a semantic property of predicate), and applied it to the automatic acquisition of *contradiction* and *causality* relations between events in Japanese text. For example, "destroy cancer" and "develop cancer" have contradiction relation, and "increase in crime" and "heighten anxiety" have causality relation.

The concept of excitation could be used to characterize predicates with semantic orientations, namely "excitatory", "inhibitory", and "neutral". Excitatory templates imply that the main effect of a predicate on its argument is activated or enhanced (e.g., "cause X", "preserve X"), while inhibitory templates imply that the effect is deactivated or suppressed (e.g., "ruin X", "prevent X"); neutral examples include "related to X" and "close to X". A bootstrapping approach was also proposed to acquire excitation templates based on some language-independent constraints on narrative structures of text.

The use of excitation in extracting contradiction pairs and causality pairs can be explained using two assumptions. The assumption about contradiction is: contradiction pairs are often similar in distribution but have a sharp contrast in excitation value. The assumption about causality is: if a pair of events have a strong excitation tendency and they are connected by an AND/THUS-type connective in a sentence, then the pair probably has causal relation. Their methods extracted one million contradiction pairs with over 70% precision, and 500 000 causality pairs with about 70% precision from a 600 million page Web corpus. Interestingly, by combining the extracted causality pairs and contradiction pairs, they generated one million more plausible causality pairs. Such causality pairs cannot be acquired in any single sentence in their corpus with reasonable precision.

10.3 NKI

In China, the group of Large Scale Knowledge Processing (at Institute of Computing Technology, Chinese Academy of Sciences) directed by Cao has done a lot of work on commonsense knowledge acquisition[49,85-89]. Zhu *et al.*[49] crawled and extracted web pages in Chinese for commonsensical sentences from which CSK is relatively easier to acquire. Tian *et al.*[85] introduced a framework of acquiring and analyzing psychological commonsense concepts. Peng *et al.*[86] proposed a method for mining commonsensically associated events w.r.t. a given core event, and discovering the mapping between the participants of the core event and those of the associated events. Cao *et al.* extracted specific kinds of Chinese CSK from the Web, including commonsensical properties of concepts[87], such as "snow is white", comparative commonsense knowledge[88], such as "a man is generally stronger than a woman", and causal relation between events[89], such as "criticism leads to upset".

11 Extract Structured Data from Wikipedia

Wikipedia, one of the largest knowledge sources of mankind, is maintained by thousands of contributors and keeping growing. Hence, it would be a fantastic knowledge source for CSK acquisition.

11.1 YAGO/YAGO2

Suchanek *et al.*㊴ [41] built the YAGO ontology from WordNet (which provides large amount of entities) and Wikipedia (which provides a clean taxonomy). Entities include *individuals* (e.g., Albert Einstein or string "Albert Einstein"), *classes* (e.g., scientist), *relations* (e.g., taxonomic relation *subclassOf* and non-taxonomic relation *bornInYear*), and *fact identifiers* (each is mapped to exactly one fact). The YAGO ontology contains more than 1 million entities and 5 million *facts* (i.e. triples of form "(*entity*1, *relation*, *entity*2)"). Each fact is assigned with a confidence value (between 0 and 1), and new facts from new sources can be added to YAGO (i.e., YAGO is extendable).

YAGO adopts a carefully designed combination of rule-based and heuristic methods to extract the TYPE relation (e.g., (Albert Einstein, type, Physicist)), the subclass relation (e.g., (Physicist, subclassOf, Scientist)), the means relation (e.g., ("urban center", means, city)), other relations (e.g., (Albert Einstein, bornInYear, 1879)), and meta-relations (e.g., a meta-relation describes Albert Einstein using its URL). Empirical evaluation of fact correctness showed that the accuracy of YAGO was about 95%. The sizes of entities and facts in YAGO ontology (millions) were much larger than those in KnowItAll, SUMO, WordNet, and OpenCyc ontologies (hundreds of thousands).

YAGO2 is an extension of YAGO, where entities and facts are placed in both time and space dimensions. YAGO2 contains 9.8 million entities and 447 million facts, which were extracted automatically from Wikipedia, GeoNames, and WordNet. YAGO2 also employs rule-based extraction method. In contrast to YAGO's hardwired rules in its source code, YAGO2 stores extraction rules in text files, which allows easy extension without changing source code. Human judgement for sampled facts showed the facts in YAGO2 remained an accuracy of 95%.

11.2 Freebase

Freebase⑪ [90] is a scalable graph database used to structure general human knowledge. The data in Freebase can be collaboratively created, structured, and maintained by people and softwares. Freebase provides an AJAX/Web based user interface for humans and an HTTP/JSON based API for softwares. Metaweb Query Language (MQL) is Freebase's data query and manipulation language. Freebase currently contains more than 1.8 billion facts and 39 million topics㊵ .

The users can edit the properties of an existing topic (e.g., "Richard Feynman"). Freebase provides automatic suggestion to help the user enter new knowledge. For example, Freebase presents a candidate list of the siblings of "Richard Feynman". Furthermore, the user can edit the schema of Freebase database. Freebase provides the *schema editor* for schema creation and evolution. For example, the users can add a "Derived Drug" property to "MedicinalPlant" schema, and specify the value of the property is expected to be an instance of "Drug".

11.3 DBPedia

DBpedia[91] is a crowd-sourced community effort to extract structured information from Wikipedia. DBpedia is freely available on the Web. The motivation of building DBpedia is to make the tremendous information in Wikipedia easier to be used in new interesting ways; in turn it can inspire new mechanisms for improving the encyclopedia itself.

DBpedia extracts RDF triples from Wikipedia articles. The DBpedia community now adopts a flexible and extensible extraction framework㊶ , which contains 4 core components. Firstly, the Source package provides an abstraction over a source of Media Wiki pages. Secondly, the WikiParser module transforms an Media Wiki page source into an abstract syntax tree (AST). Thirdly, the Extractor module uses an extractor to map from a page node to a graph of statements about it. Finally, the Destination module provides an abstraction over a destination of RDF statements.

The DBpedia knowledge base has been interlinked with various other data sources (e.g., Freebase and

㊴Max Planck Institute for Informatics (MPII), Germany, Apr. 2013.
㊵This data is retrieved in June, 2013 from Freebase's website.
㊶http://wiki.dbpedia.org/Documentation?v=k2l, Apr. 2013.

YAGO) on the Web according to the Linked Data principles. Linked Data is a method to publish data on the Web and to interlink data between different data sources. Using RDF links "(URI1, owl:sameas, URI2)", it is possible to connect the DBpedia entities (e.g., Spain) with additional information from other data sources (e.g., Freebase and OpenCyc). People or softwares can follow these links to retrieve additional information about Spain.

The DBpedia knowledge base has been used to many applications. For example, it supports more complicated queries on Wikipedia, location-based information services, annotation of Web content, etc. You can find more detailed information about the applications of DBpedia in its website.

12 KnowItAll

To answer the questions "How can a computer accumulate a massive body of knowledge? What will Web search engines look like in ten years?"[42], Etzioni has been leading the KnowItAll research group at the University of Washington to design and develop a variety of Web information extraction systems.

The KnowItAll system[43][53] extracts large collections of facts (e.g., names of cities) from the Web in a domain-independent, unsupervised, autonomous, scalable manner. The paradigm Open Information Extraction (Open IE)[34] has been proposed to extract a large number of relations from arbitrary text on the Web at once, without specifying the targets to be extracted. Notable Open IE systems include the first generation system TextRunner[33] and the second generation systems ReVerb[35] and R2A2[36].

Open IE extractions have been applied to tasks such as learning selectional preferences[92], acquiring common sense knowledge[93], and recognizing entailment[94-95]. In addition, Open IE extractions have been mapped onto pre-existing ontologies[96].

12.1 Web Named Entity Extraction

The KnowItAll system[52-53] aims to extract large collections of facts from the Web in an autonomous, domain-independent, and scalable manner. The input of it includes an extensible ontology containing class names and relation names, as well as a small set of general domain-independent extraction templates like "NP1 such as NPList2". In the preliminary experiments, KnowItAll ran for four days on a single machine and extracted over 50 000 facts regarding *cities*, *states*, *countries*, *actors*, and *films*.

For each class and relation in the ontology, KnowItAll automatically generates extraction rules and search queries from the set of generic templates. For example, an extraction rule is generated by instantiating the generic template "NP1 such as NPList2" with the class name *city*, which can be used to find city names from sentences like "We provide tours to cities such as Paris, Nice, and Monte Carlo"[52]. Such simple sentences are downloaded via a number of search engines using queries, such as "cities such as".

In order to estimate the correctness of extracted facts and discriminate a class's instances from non-instances, KnowItAll uses features of "web-scale" statistics regarding extracted instances and discriminator phrases. Discriminator phrases, like "actors such as X", are generated from the class name, such as "actor", or the keyword phrase from an extraction rule, such as "such as", where X can be replaced by the extracted instances. Based on search engine hit counts, i.e., the number of results returned by search engines in response to a query, the assessor calculates pointwise mutual information between candidate extractions and their discriminator phrases, and derives a set of features. For example, if the PMI between "Bratt Pitt" and "actors such as Bratt Pitt" is high, this gives evidence that "Bratt Pitt" is indeed a valid instance of the class *Actor*. These features are then combined by a Naive Bayes classifier and used to assess the likelihood of extracted facts are correct. KnowItAll automatically selects k most informative ones of the generated discriminators in a bootstrapping manner.

KnowItAll suggests a challenge: how to improve its recall and extraction rate while retaining its precision. In [53], three distinct methods are implemented to address this challenge. First, *pattern learning* automatically learns domain-specific extraction rules used to find additional extractions. Second, *subclass extraction* automatically identifies subclasses of a class of interest and extracts instances of them. Third, *list extraction* is specially designed for finding lists of class instances. These methods are based on or bootstrap from KnowItAll's domain-independent methods, and thus they also do not require hand-tagged training examples. Importantly, they improve KnowItAll with a 4-fold to 8-fold increase in recall and meanwhile they retain similar precision.

12.2 TextRunner

TextRunner[33] is the first web-scale Open Information Extraction (OIE) system, which is capable of extracting all potential relationships in one time by

[42] http://www.cs.washington.edu/people/faculty/etzioni/research/, Apr. 2013.
[43] The system has the same name as the research group.

scanning corpus once. It takes only a corpus as input, without relation specificity or hand-tagged samples, and outputs a set of relational tuples consisting of two strings denoting two entities and a string denoting their relationship. It now has elicited 1 000 000 000 (one billion) distinct relational extractions for the Web㊹. TextRunner consists of three key modules:

• The *self-supervised learner* takes as input a small corpus sample without hand-tagged data, and its output is a classifier that labels candidate extractions as "trustworthy" or not. First, the learner uses an unlexicalized parser[97] to automatically identify a set of extractions, and label them as trustworthy or untrustworthy using a set of heuristic constraints. Then, these extractions are mapped into feature vectors and used as training samples to a Naive Bayes classifier. Finally, the trained classifier will be called by the extractor module to label its extracted candidate extractions.

• The *single-pass extractor* is capable of extracting tuples for all potential relations by making a single pass over the entire corpus. It first identifies basic noun phrases as entities using a noun phrase chunker. It then identifies normalized relations by eliciting the text between the noun phrases and heuristically discarding non-essential modifiers (e.g., some adverbs modifying verbs). Finally, the extracted tuples are labeled by the classifier produced by the learner, and the "trustworthy" extractions are remained and scored.

• The *redundancy-based assessor* evaluates each retained extraction using a probabilistic model of redundancy[98]. Given a tuple $t = (e_i, r_{i,j}, e_j)$ extracted from k different sentences, the model can precisely and efficiently estimate the probability that the relation $r_{i,j}$ holds between e_i and e_j.

Comparative experiments with KnowItAll showed that the two systems found almost identical amount of correct extractions, but TextRunner's average error rate was 33% lower than KnowItAll's. In addition, TextRunner is much more efficient than KnowItAll. KnowItAll takes relation names as input and has to scan web corpus again and again, while TextRunner does it in only a single pass of web corpus. You can find more details regarding the relationship between traditional IE and Open IE in [99].

12.3 ReVerb

The Open IE system ReVerb㊺[35] employs a specified model of relations for extraction instead of a model learned from training data. The model implements two simple but effective constraints on binary relationship expressions, which can be used to eliminate two types of frequent errors, incoherent extractions and uninformative extractions, in the results of prior Open IE systems (TextRunner[33] and WOE[100]).

Fader *et al.*[35] illustrated the two types of frequent errors, analyzed the reasons of producing such errors, and proposed solutions. Incoherent extractions are the relation phrases that have no meaningful interpretation. For examples, given the sentence "The guide contains dead links and omits sites", previous Open IE systems return "contains omits" as a relation expression. The reason for this type of error is that the extractor makes a sequence of decisions about whether to contain each word between two noun phrases. Uninformative extractions are the relation expressions that omit critical information. For example, given the sentence "Faust made a deal with the devil", previous Open IE systems return "(Faust, made, a deal)" rather than "(Faust, made a deal with, the devil)". This type of error is caused by improper handling of the relation expressions expressed by Light Verb Constructions (LVCs), where the noun phrase "a deal" also contributes to the meaning of the predicate. To solve such errors, a syntactic constraint on relational expressions was introduced: every multi-word relation phrase must be a contiguous sequence of words that begins with a verb and ends with a preposition, avoiding unmeaningful relation phrases and allowing relation phrases to include nouns.

However, the syntactic constraint leads to overly-specific relation phrases, for example, "is offering only modest greenhouse gas reduction targets at". To address this problem, Fader *et al.* introduced the lexical constraint: "a binary relation phrase ought to appear with at least a minimal number of distinct argument pairs in a large corpus".

ReVerb takes as input a POS-tagged and NP-chunked sentence and returns a set of triples of form (Arg1, Rel, Arg2). It comprises two components:

• *Relation extractor* searches, for each verb in a sentence, the longest sequence of words that satisfies the syntactic constraint and the lexical constraint. The syntactic constraint is implemented using regular expression, and the lexical constraint utilizes a large dictionary of relation phrases to record distinct arguments. The constraints cover nearly 85% of the relation phrases. on a sample of Web sentences.

• *Argument extractor* finds the nearest noun phrases to the left/right of each identified relation phrase, where the noun phrase is not a relative pronoun, nor WHO-adverb, nor existential "there".

㊹http://videolectures.net/ijcai2011_etzioni_webscale/, Apr. 2013

㊺ReVerb and the data used in their experiments have been released to the research community, available at http://reverb.cs.washington.edu/, Apr. 2013.

Fader *et al.* used a logistic regression classifier to assign a confidence score to each extraction. All features used by the model are independent with specific relation and can be efficiently computed. The training data set contains 1 000 sentences with their extractions, which are manually labeled as either "correct" or "incorrect". Trading recall for precision can be achieved by tuning a confidence threshold.

On a test set of 500 sentences sampled from the Web, Fader *et al.* compared ReVerb with other five systems: ReVerb$^{\neg lex}$ (ReVerb without lexical constraint), TextRunner (trained with the Penn Treebank), TextRunner-R (trained with the ReVerb extractions), WOE^{pos} and WOE^{parse}. Each Open IE system returned confidence scores for its extractions and, for a given confidence threshold, the precision and recall of the output were computed. A precision-recall curve was then drawn via varying the confidence threshold, and the area under the curve (AUC) was computed.

Experimental results showed that the AUC of ReVerb was 30% higher than that of WOE^{parse} and more than two times of that of WOE^{pos} or TextRunner. ReVerb achieved an AUC 23% higher than REVERB$^{\neg lex}$. Also, ReVerb extractions provided a useful training data for TextRunner-R, of which the AUC is 71% higher than that of TextRunner-R. In particular, more than 30% of ReVerb extractions could be extracted with a precision 0.8 or higher, while the other systems returned none at that precision.

12.4 R2A2

TextRunner and ReVerb assume that arguments are simple noun phrases (NPs), thus failing to identify complicated structures of arguments, e.g., NPs with prepositional attachments, lists of NPs, independent clauses, and NPs with relative clauses. Experimental results showed that 65% of ReVerb's errors had correct relation phrases but wrong arguments. To reduce such errors, Christensen *et al.*[36] proposed an argument learning component ArgLearner, along with ReVerb to constitute the final system R2A2.

ArgLearner uses statistical classifiers to detect the left bound and the right bound of each argument. Since Arg2 (i.e., the second argument of a relation) almost always follows the relation phrase, there is no need to built a separate classifier for the left bound of Arg2. Features come from two sources. Standard features include those describing the NP itself and the context around the NP, as well as those describing the whole sentence (e.g., sentence length, POS-tags, capitalization, and punctuation). Other features came from manual analysis about argument bounds. For example, they created regular expression indicators to detect whether Arg2 was followed by an independent clause or verb phrase. For the challenge of training data, they used a set of post-processing heuristics to convert the training data for semantic role labeling (SRL) into the form suitable for Open IE training.

Experiments for ArgLearner[36] showed that the accuracy of identification of Arg1 (i.e., the first argument of a relation) right bound was 96%, and 92% for Arg1 left bound and 73% for Arg2 right bound. Compared with ReVerb, R2A2 substantially increased both precision and recall on a dataset containing both Web and newswire sentences.

13 Probase

Probase㊻ [101], which is developed by Microsoft Research, contains a probabilistic taxonomy of concepts and enable computers to conceptualize like human. Conceptualization includes both instantiating concepts to their typical instances (e.g., from *largest company* to *China Mobile*) and abstracting one or multiple instances to the likely concepts they belong to (e.g., from *China, India, Brazil* to *emerging market*).

Wu *et al.*[101] claimed Probase is unique in two aspects. Firstly, Probase has a much larger concept space (2.7 million concepts) than other knowledge bases such as ResearchCyc (about 120 thousand). Secondly, Probase measures the plausibility and typicality of knowledge using probabilities, which serve as the priors and likelihoods for probabilistic reasoning in Probase.

Probase extracts *IsA* relation iteratively using a fixed set of extraction patterns from 1.68 billion web pages. In each iteration, Probase extracts new *IsA* relation pairs and uses them to increase the precision and recall of extraction in the next iteration. *IsA* relations are extracted by three steps. Firstly, a list of candidate super-concepts (denoted as X) and a list of candidate sub-concepts (denoted as Y) are generated using the extraction patterns. Secondly, the ratio of likelihood between any two candidate super-concepts is computed, and only one candidate super-concept (e.g., x) having the largest likelihood is selected as a super-concept. Thirdly, on the basis of the observation that a candidate sub-concept is more likely to be valid if it is closer to the super-concept, they find the farthest candidate sub-concept y_k such that the likelihood $p(y_k|x)$ is above a threshold. All candidate sub-concept between x and y_k are regarded as the valid sub-concepts of the super-concept x.

Probase adopts a probabilistic framework where both plausibility of an *IsA* relation and typicality of an instance of a concept are computed using joint proba-

㊻http://research.microsoft.com/en-us/projects/probase, Apr. 2013.

bility and conditional probability respectively. An *IsA* relation (x, y) is judged to be false if all evidences (i.e., sentences) $s_1, \ldots, s_n$ supporting (x, y) are false. Formally, under the independence assumption, the plausibility of (x, y) is represented as

$$\mathcal{P}(x, y) = 1 - p(\bigwedge_{1}^{n} \overline{s_i}) = 1 - \Pi_1^n(1 - p_i),$$

where the confidence p_i of an evidence s_i is generated by a Naive Bayes model with a training set from WordNet. The typicality is the probability of an instance given a concept, which is represented as

$$\mathcal{T}(i|x) = \frac{n(x, i) \cdot \mathcal{P}(x, i)}{\Sigma_{i' \in I_x} n(x, i) \cdot \mathcal{P}(x, i)},$$

where $n(x, i)$ is the number of evidences supporting (x, i), $\mathcal{P}(x, i)$ is the confidence of (x, i), and I_x is the set of individuals of concept x.

Probase was compared with four other taxonomies, namely WordNet[75], WikiTaxonomy[102], YAGO[41], and Freebase[90]. The coverage of each taxonomy was examined against web search queries from Bing's query log in a two-year period. A query is covered by a taxonomy if the query includes at least one concept or instance in the taxonomy. Experimental results showed Probase is more comprehensive (81% w.r.t. top 50 million queries) than four other taxonomies. The average precision of *IsA* relations in Probase was estimated to be 92.8%, which outperforms other taxonomies rather than YAGO's 95%㊼ . In addition, Probase has been applied to semantic web search[103], short text understanding[104], etc.

14 Qualitative Reasoning

The Qualitative Reasoning Group (QRG)㊽ at Northwestern University, led by Forbus, conducts research on qualitative representation and reasoning for capturing both everyday and expert reasoning about quantities, space, time, and causality. A research project relevant to CSK acquisition is *learning by reading*㊾[38] to develop systems capable of populating their knowledge via understanding text and diagrams. In addition, we investigate the reasoning technique of learning plausible inference patterns over ground facts[54], which can improve the scalability of knowledge base construction. This section does not include other work that is relevant to common sense but is not directly used to collect CSK.

14.1 Learning Reader

Learning Reader[38] is a system that learns by reading simple texts. The system could extract knowledge from what it has read (via its Reader component), improve its understanding of what it has read (via its Ruminator component), and evaluate what it has learnt by answering questions (via its QA system). All components operate over a large knowledge base extracted from ResearchCyc.

Learning Reader produces formal representations (in the form of assertions of CycResearch KB) for input text snippets, identifying pre-existing knowledge and creating new ones when none can be found. It adopts Direct Memory Access Parsing (DMAP) model for natural language understanding. DMAP understands a text snippet as a sequence of references to concepts and incrementally matches those references against phrasal patterns. Matched phrasal patterns generate additional higher-order conceptual references. DMAP used 30 000 phrasal patterns, where only 50 were hand-generated and the other were automatically translated from linguistic knowledge in the ResearchCyc KB.

The QA system uses a set of parameterized question templates to derive questions, and generates a set of formal queries for each question template to answer the questions. In order to ensure the tractability of reasoning, the QA system restricts the set of axioms used for reasoning and restricts axioms to Horn clauses.

The Ruminator imitates the human capability of reflecting upon what they have read by generating interesting questions to consider. There are three kinds of questions, including standard Journalist's questions (who, what, when, where, why, how), questions by analogy with a prior case, and questions by analogy with a generalization for a topic. The Ruminator could perform inference in two subsettings: purely deductive inference and promiscuous conjecture acceptance (PCA) inference. PCA includes additional inferences by analogy with prior cases.

The experimental corpus consists of 62 simple stories (containing 956 sentences) about the Middle East in English. The set of questions asked was generated by filling the parameterized question templates in the QA system with the entities appearing in the knowledge base resulting from reading the entire corpus. These question were asked in four different conditions: without reading, only reading, reading plus rumination (purely deductively), and reading plus rumination (allowing PCA). In these four subsettings, the percent of the questions that can be answered increased from 10%

㊼YAGO uses a clear data source Wikipedia rather than web pages.
㊽http://www.qrg.northwestern.edu/, Apr. 2013.
㊾http://www.qrg.northwestern.edu/projects/LearningReader/lr_index.html, Apr. 2013.

(without reading) to 37% (only reading), to 50% (after deductive rumination), and to 60% (after rumination with PCA). The improvement demonstrated that the system did learn via reading and rumination did promote learning more from the texts that it had read. However, the correctness of answers dropped from 100% to nearly 90% in turn. Especially after rumination with PCA, out of the 91 new questions that the system could answer, 47 questions were answered incorrectly.

14.2 Plausible Inference Patterns

Sharma and Forbus[54] studied how to learn and use plausible inference patterns (PIPs) to do plausible inference on ground facts of ResearchCyc KB, which could significantly improve the performance of QA systems and the scalability of knowledge base construction. Informally, a PIP is an inference chain containing predicate types between two entities. An PIP example is

$$\begin{aligned}&\mathit{FamilyRelationSlot}(?x, ?y)\\ &\wedge\ \mathit{FamilyRelationSlot}(?y, ?z)\\ &\rightarrow \mathit{PersonalAssociationPredicate}(?x, ?z),\end{aligned}$$

where *FamilyRelationSlot* and *PersonalAssociationPredicate* denote predicate types and $?x, ?y, ?z$ are variables representing entities. This pattern means that two predicates of type *FamilyRelationSlot* can plausibly combine to infer assertions involving *PersonalAssociationPredicate*. The major source of knowledge base contents came from ResearchCyc, where both concepts and predicates are arranged in hierarchy.

The algorithm FPEQ can find all plausible explanations for a given fully ground query (i.e., a predicate and two entities). First, it constructs a graph around the entities mentioned in the query. Then, it incrementally searches the facts in the graph under the guidance of a set of PIPs. Finally, it returns all explanations (i.e., PIPs configured with specific predicates and entities) which plausibly entail the query.

Moreover, Sharma *et al.* showed that PIPs and their quality could be learned by reinforcement learning. Users provided feedback about the correctness of the final answer, +1 for correct ones and −1 for incorrect ones respectively. Initially, PIPs were obtained by replacing predicates in axioms with their predicate types. An iterative algorithm was employed to find the correct level of generalization (of predicate types) according to its estimation of reward.

The experimental results showed that PIP-based QA system generated more answers than QA system based on traditional deductive reasoning. The experiments were done on five kinds of question templates. A template example is "Where did <Event> occur?", where <Event> indicates the kind of thing for which the question makes sense. Each question template was expanded to a collection of queries by randomly selecting facts from the KB. The baseline QA system used a simple backchainer and included all axioms for these predicates and their subgoals through depth 3. The FPEQ algorithm obtained a much higher recall over the baseline (120% improvement) while remained a high average accuracy (94%).

15 NELL

NELL[50] (Never-Ending Language Learner) is a computer system that learns knowledge and improves its learning forever. Each day it performs two tasks: 1) extract facts from web text to populate a growing knowledge base; 2) attempt to improve its reading competence (extracting more facts more precisely) over time.

In NELL's prototype implementation[105], two kinds of knowledge[51] are collected: 1) knowledge about which noun phrases refer to which specified semantic categories, such as cities, companies, and sports teams, 2) knowledge about which pairs of noun phrases satisfy which specified semantic relations, such as "hasOfficesIn(organization, location)".

Four subsystems have been implemented to learn extraction/inference models and use the models to extract/infer new knowledge. 1) Coupled Pattern Learner (CPL)[106] learns a free-text extractor that extracts candidate instances of categories or relations from sentences on the Web. The extractor consists of a set of contextual patterns like "mayor of X" and "X plays for Y". CPL assigns an estimated probability to each candidate instance using the heuristical formula $1 - 0.5^c$, where c is the number of contextual patterns that extract the candidate. 2) Coupled SEAL (CSEAL)[106] learns a semi-structured extractor which extracts novel instances of categories or relations from lists and tables on the Internet. It takes existing instances from each category or relation as positive examples and mutual exclusion relationships as negative examples. CSEAL assigns an estimated probability to each candidate instance using the same method as CPL, except that c is the number of unfiltered wrappers that extract the candidate instance. 3) Coupled Morphological Classifier (CMC) learns a set of binary L2-regularized logistic regression models, each of which is for a specified category or relation. Beliefs from the KB are used as training instances, and mutual exclusion relationships are used to identify negative instances. Also, CMC is used to exam-

[50] rtw.ml.cmu.edu/rtw/, Apr. 2013.

[51] These linguistic knowledge (about English) is important part of commonsense knowledge.

ine candidate instances proposed by other components. 4) Rule Learner (RL) learns a set of probabilistic Horn clauses, which are used to infer new relation instances from other already-known relation instances.

Carlson *et al.* adopted a bootstrap learning method, and its preliminary implementation operates the following process iteratively. Firstly, with the current KB, each subsystem component runs until completion. Then, a knowledge integrator decides which novel candidates can be promoted to the KB. The larger number of beliefs provided by the growing KB retains each subsystem to perform better in the next iteration.

Given an initial seed ontology defining 123 categories and 55 relations, NELL ran for 67 days and acquired 242 453 new facts (95% category facts and 5% relation facts) with average estimated precision of 74%. To evaluate its performance, all iterations during execution are divided into three stages: iterations 1∼22, iterations 23∼44, and iterations 45∼66. During these periods, the numbers of new facts are very similar, falling in between 76 000 and 89 000; the estimated precision drops, from 90% to 71% to 57% (74% average). The data show that NELL retains fairly high precision for many iterations of learning with a consistent collecting rate.

16 LarKC

LarKC㉜, the Large Knowledge Collider, is a platform that supports "massive, distributed, incomplete" semantic reasoning over billions of data[107-108]. Practical examples include various context-sensitive and personalized mobile services in the telecom sector, which require processing of billions of RDF triples or more in near-realtime. Traditional reasoning methods or systems in the Semantic Web community are strictly based on logic and do not scale up to the challenge of processing datasets of such size. Therefore, a novel reasoning infrastructure is required to adapt to the requirements required by different use cases.

LarKC is a pluggable algorithmic framework implemented on a distributed computational platform, which makes reasoning at Web scale possible by trading quality for computational cost via embracing incompleteness and unsoundness. The plug-in architecture enables LarKC to integrate logical reasoning with diverse techniques and heuristics from diverse areas, e.g., databases or machine learning. They are equipped into LarKC as plug-ins and combined in complex workflows to achieve a task. The distributed and parallel computing enables LarKC to achieve massive inference by distributing problems across heterogeneous computing resources.

Researchers can design and experiment with plug-ins. The plug-ins are grouped into categories based on their functionality: 1) *identifier* — restraining the scope of all available data to only those that possibly contribute to answering a user's query; 2) *transformer* — transforming data from one representation to another; 3) *selecter* — selecting which parts of the identified data are required for reasoning; 4) *reasoner* — performing different kinds of reasoning (deductive, inductive, etc.) on the identified/selected/transformed data; 5) *decider* — supervising the construction, management, control and execution of the plug-ins within the workflows.

LarKC has been successfully applied to: 1) data integration in the pharmaceutical and biotechnology domain, and 2) urban computing.

17 Others

Pasca *et al.*[109] studied the problem of acquiring thousands of open-domain classes of instances, along with relevant sets of open-domain class attributes. Vanderwende[50] used handcrafted lexical-syntactic patterns to identify sentences implying default assumptions. For example, a sentence with an adverbial clause of time.

18 Comparisons

While many researchers have developed various commonsense knowledge acquisition systems, to our knowledge, there is no work on comparing and evaluating them in a unified framework. The reason may be that the acquisition tasks, which are defined by knowledge source and knowledge to be acquired, are different, and the techniques used are also different. The motivation of such comparison is to give some lights on developing more effective CSK acquisition approaches.

18.1 Overall Comparison

Labor acquisition is the most direct method of collecting CSK, which can produce (well-formalized) CSK with relatively high accuracy. Importantly, the collected knowledge can serve as the foundation for other automatic systems. However, labor acquisition leaves most of work to human contributors. Human contributors are responsible for explicitizing the hidden CSK in their mind. They have to address the problem of large scale by long-term tedious effort. A hand-coded knowledge base usually suffers from "knowledge gaps": it possibly returns nothing in response to a requested concept from a user[21].

Interactive acquisition is capable of decreasing cognitive burden of human contributors and making them

㉜http://www.larkc.eu/, Apr. 2013.

contribute knowledge in an easy and enjoyable way (sometime unconsciously). As a result, this paradigm retains volunteer contributors for a longer time and collects CSK more efficiently (compared with labor acquisition). Some intelligent interactive interfaces can guide its users to enter "what it does not know"[30], which increases the throughput of new knowledge and improves the coverage of knowledge on the whole.

Reasoning acquisition aims to find and fill the gaps in an existing knowledge base, improving its comprehensiveness. Furthermore, reasoning acquisition can produce CSK in a very efficient way since this process is fully automatic. Unfortunately, the performance of reasoning acquisition heavily depends on the amount of existing knowledge, and this knowledge must be well formalized so that it could be understood by computers. Another deficiency of reasoning acquisition is its low precision. The automatically derived knowledge has to be validated by human or via other statistic methods before being put into knowledge base.

A huge advantage of mining acquisition is the existence of ample textual resources. The scale and redundancy of corpus make mining acquisition a suitable framework for CSK acquisition. Large-scale and domain-independent corpus, such as billions of web pages on the Web, make it possible to collect a huge and diverse collection of CSK. Data redundancy in corpus can be effectively used to estimate the correctness of commonsensical statements[98,110]. In addition, a mining system can extract CSK from text automatically and efficiently. On the other hand, mining acquisition also has some limitations. First, there is numerous obvious knowledge that no one has bothered to record it, whereas most mining systems focus on extracting CSK from text that is relatively easy to process. Second, web corpus contains various noises and incorrect knowledge. For example, Google search engine returns much more results (about 10 700 000) for "dog can fly" than "dog cannot fly" (about 27 600). Third, it is not a trivial work to formalize the knowledge in natural language.

18.2 Item by Item Comparison

In this subsection, we compare CSK acquisition systems on the basis of several important features characterizing the systems. All comparative dimensions could be divided into three groups: task, solution, and evaluation. Table 2 summarizes the comparison results.

The first column (Affiliation) denotes the affiliations that develops CSK acquisition systems. The second column (System) denotes CSK acquisition system names. The third column (Section) denotes the numbers of the sections/subsections describing the CSK acquisition systems. The fourth column (Class) denotes the categories of acquisition systems according to Section 2. The classification include labor (L), interactive user interface (IUI), game (G), reasoning (R), rule-based information extraction (RIE), bootstrapping information extraction (BIE), open information extraction (OIE), machine reading (MR), and knowledge integration (KI).

The task dimension describes what can be done by a system. Specifically, this dimension describes the input and output of a system, as well as the circumstance to which it is applicable.

• *Knowledge Source* (*Src.*): From the table, we can see that various knowledge sources have been used in CSK collection. Human contributors include knowledge engineers (KE), domain experts (DE), volunteer contributors (VC), and game players (GP). Data sources include tagged corpus such as Penn Treebank (PTB), raw corpus without any tagging such as newspapers, semi-structured data such as wikipedia (wiki) and web pages (Web), and structured database (DB).

• *Knowledge Base* (*KB*): In some cases, pre-existing knowledge bases are used to assist knowledge acquisition. If the system does not require a KB, then "-", otherwise the KB(s) is given.

• *Acquisition Objective* (*Obj.*): What kind of knowledge is to be acquired: factual knowledge (Fact), or ontological knowledge (Onto.), rules, or combination of them? For example, "F/O" denotes both factual knowledge and ontological knowledge. If facts about entities are to be acquired, does the system acquire *relations* (Rel) between named entities, or *concepts* (C) or *named entities* (NE) that are instances of specific concepts[53]? From the table, we can find that 1) most automatic mining systems are designed to collect facts about entities from text; 2) human contributors like to enter ontological knowledge that describes relations about concepts and relations; 3) few system are designed to extract rules because rules are hard to acquire from both human contributors and mining systems.

• *Knowledge Representation* (*Rep.*): What language is used to represent CSK? Natural language (NL) or formal languages (FL), e.g., CycL or RDF? Most collected knowledge is codified with natural languages rather than formal languages. The reason may be that ordinary people who have no AI background use only natural language, and it is not a trivial work to transfer natural language representation into formal representation. Is the collected knowledge organized into a structure? What is the structure? For example, Cyc's assertions are organized into micro-theory (MT); OMCS data are

[53] More details about knowledge categorization can be found in Section [1.2].

Table 2. Comparison of CSK Acquisition Systems

Affiliation	System	Section	Class	Task				Technique				Evaluation			
				Src.	KB	Obj.	Rep.	Dom.	Tar.	Tech.	Part.	Cor.	Eff.	Cov.	App.
Cycorp	Ontologist	4.1	L	KE	Cyc	All	CycL	No	No	KE	Codify	C, H	Yes	No	Yes
	KRAKEN/UIA	4.2	IUI	DE	Cyc	All	NL	Yes	No	DLG	Enter	-	Yes	No	
	Fact./Pred.Pop.	4.3	IUI	VC	Cyc	All	NL	Yes	No	NLP	Enter	C, H	Yes	No	
	Fact. Extraction	4.4	MR	Web	Cyc	Fact	CycL	No	Yes	NLP, QA	-	C, S, H	Yes	No	
	Rule Generation	4.5	R	-	Cyc	Rule	CycL	No	No	ILP	-	H	No	No	
	SKSI	4.6.1	KI	DB, Web	Cyc	-	CycL/NL	No	No	SKI	-	C, S, H	Yes	No	
	Cyc+Wiki	4.6.2	KI	Wiki	Cyc	-	CycL	No	No	HR	-	H	No	No	
Sign.	ThoughtTreasure	5	L	KE	-	All	LAGS	No	No	KE	Codify	C, H	Yes	No	Yes
Keen.	HowNet	6	L	KE	-	Onto.	FL	No	Yes	KE	Codify	H	No	No	Yes
MIT	OMCS	7.1	L	VC	-	Onto.	NL	No	Yes	KE	Enter	H	Yes	No	Yes
	OMCommons	7.2	IUI	VC	OM	Onto.	NL	No	Yes	DLG, AR	Answer	H	Yes	No	
	20 Questions	7.3	G	GP	OM	Onto.	VS	No	Yes	DLG, ML	Answer	H	Yes	No	
	ConceptNet	7.4	KI	-	OM	F/O	SN	No	Yes	RIE	-	H	No	Yes	
	AnalogySpace	7.5	R	-	OM	Onto.	Matrix	No	No	AR, DR	-	H	No	No	
	ConceptMiner	7.6	RIE	Web	CN	Onto.	NL	No	Yes	BIE	-	S, H	Yes	No	
CMU	Verbosity	8.1	G	GP	-	Onto.	NL	No	Yes	GD	Answer	H	Yes	No	No
NTU	RG/VPG	8.2	G	GP	-	Onto.	NL	Yes	Yes	DLG, UM	Q/A	S, H	Yes	No	No
	GOKC	8.3	G	GP	CN	Onto.	NL	Yes	No	DLG	Q/A	S, H	Yes	Yes	No
RU	KNEXT	9	RIE	PTB	-	Onto.	FL	No	No	RIE	-	H	No	No	No
MPII	YAGO/YAGO2	11.1	RIE	Wiki	WN	F/O	RDF	No	Yes	RIE, SKI	-	H	Yes	Yes	
MT Inc.	Freebase	11.2	IUI	Wiki, VC	-	F/O	RDF	No	No	RIE, SKI	Enter	No	No	No	Yes
DBPedia	DBPedia	11.3	RIE	Wiki	-	F/O	RDF	No	No	RIE, SKI	-	S, H	Yes	Yes	Yes
UW	KnowItAll	12.1	BIE	Web	-	NE	NL	No	Yes	BIE	-	S, H	Yes	Yes	Yes
	TextRunner	12.2	OIE	Web	-	Rel	NL	No	No	OIE	-	S, H	Yes	Yes	
	ReVerb	12.3	OIE	Web	-	Rel	NL	No	No	OIE	-	S, H	Yes	Yes	
	R2A2	12.4	OIE	Web	-	Rel	NL	No	No	OIE	-	S, H	Yes	Yes	
MSR	Probase	13	BIE	-	-	C/O	NL	No	Yes	BIE, PR	-	H	Yes	Yes	
NU	ReadingLearner	14.1	MR	RC	Cyc	Fact	CycL	No	No	MR, AR	-	H	No	No	Yes
	PIP	14.2	R	-	Cyc	Rule	CycL	No	No	ILP	-	H	No	No	
CMU	NELL	15	MR	Web	-	Fact	NL	Yes	Yes	MR	-	S, H	No	No	Yes

Note: For column 2 System, some literatures do not present system names, so we name their work with knowledge source "Ontologist" and "Cyc+Wiki"; systematic infrastructure "UIA"; employed technique "SKSI"; acquisition target "Fact Extraction", "Rule Generation", "GOKC", and "PIP"; UIA: User Interaction Agenda; Fact.: Factivore; Pred.Pop.: Predicate Populator; OMCommons: Open Mind Commons; RG/VPG: Rapport Game and Virtual Pet Game. LAGS: logic, automata, grid, and script.

organized into semantic network (SN) in ConceptNet. Many other systems do not organize their collected knowledge into a whole.

The technique dimension describes what a system does to accomplish its acquisition task. Which techniques have been employed? Does the used technique depend on specific domains? Is the used technique designed for specific kind of knowledge? How do users interact with the system?

• *Domain Dependency*(*Dom.*): Is the system designed for a specific domain (Yes) or not (No)? Labor acquisition methods are usually domain-independent since they support knowledge contribution by people familiar with different domains. General reasoning acquisition techniques, such as *Analogy* and *Inductive Logic Programming*, are often domain-independent. Interaction acquisition and mining acquisition may be either domain-dependent or domain-independent. An interactive user interface or a game will be domain-specific if it is designed for a special community, e.g., social communities on the Web; it will be domain independent if it is designed for general public. A mining system will be domain-specific if it takes as input a corpora of specific domain (e.g., financial news of Penn Treebank); it will be domain-independent if designed for web corpus. Domain dependency is closely related to the characteristics of knowledge source.

• *Target Dependency* (*Tar.*): Is the system designed to collect specified types of facts or rules? For example, OMCS is target-specific since it uses knowledge templates of specific kinds of relations (e.g., *UsedFor* and *CapableOf*); while TextRunner is target-independent since it learns and extracts all potential relations from the Web. From the table, we can find that most systems are designed to collect specific kinds of knowledge. However, commonsense knowledge is of huge variety and it is impossible to specify all of them in advance. Thus, target-independent systems have advantage in collecting CSK, whereas they are much more complicated to design and implement.

• *Technique Used* (*Tech.*): This dimension describes the specific techniques used by a system. Labor acquisition usually adopts traditional knowledge engineering methods (KE). Interactive acquisition employs various techniques, including question and answer (QA), dialogue (DLG), user modeling (UM), machine learning (e.g., classification, clustering, dimension reduction (DR)). Popular reasoning techniques include analogical reasoning (AR), inductive logic programming (ILP), and probabilistic reasoning (PR). Mining acquisition uses various NLP techniques, which can be divided into Rule-Based IE (RIE), Bootstrapping IE (BIE), Open IE (OIE), and machine reading (MR). Rule-Based IE needs experts to code extraction rules. Bootstrap IE trains systems by bootstrapping from a small set of "seed" data. Open IE system's sole input is a corpus without any human input. It is also possible to use semantic knowledge integration (SKI) techniques to link different knowledge resources.

• *Participation Manner* (*Part.*): In what manner does human take part in collecting CSK? In labor acquisition, human contributors enter and codify knowledge directly. In interaction acquisition, human contributors interact with machines or other contributors by asking and answering questions. Most reasoning systems or mining systems run without human interception whenever they have been built.

Finally, the evaluation dimension describes how the collected knowledge is evaluated. Evaluation can be done from four aspects: correctness, coverage, efficiency, and usefulness.

• *Correctness* (*Cor.*): Did the researchers evaluate the correctness of the collected knowledge? If yes, how? From the table, we can see that human validation (H) is the most popular manner of evaluation. The second popular way is verifying knowledge against large corpus, i.e., statistics validation (S). In addition, consistency checking (C) is used to test new collected knowledge against existing knowledge.

• *Efficiency* (*Eff.*): Did they evaluate the efficiency of knowledge acquisition? Efficiency is relatively easier to evaluate, so it is assessed by many systems.

• *Coverage* (*Cov.*): Did they evaluate the coverage or comprehensiveness of the collected CSK? Limited work has been done on evaluating coverage due to the absence of fair assess methods. ConceptNet[21] relies on human subjects to assess coverage. GOKC[32] uses nonlinear regression to estimate the amount of new knowledge about a domain produced per day. Web IE systems usually approximate the hypothetical (but unknown) number of all correct extractions from the entire Web.

• *Application* (*App.*): Has the collected knowledge been equipped to some application system(s) in order to measure its usefulness? Only large-scale CSKBs have been put to practice, such as Cyc KB. The reason may be that real applications often need a large-scale CSKB instead of a small/toy one. Large scale is the secret of success.

19 Discussion and Conclusions

In this survey paper, we have reviewed several current trends of commonsense knowledge acquisition. CSK acquisition is classified into four different subsettings: labor acquisition, interaction acquisition, reasoning acquisition, and mining acquisition. Most earlier work focused on labor acquisition and supervised mining acquisition. In recent years, interaction acquisition and unsupervised mining acquisition have attracted more and more attention.

Although these studies offered some promising results in addressing the challenges of CSK acquisition, there are still some questions that are not solved well or need to be explored deeply.

First, more sophisticated approaches or systems should be developed to address the implicitness challenge. Most methods of explicitizing CSK from human minds rely heavily on human's introspection. In our opinion, commonsense knowledge could be thought of with ease or without lengthy calm consideration, if proper hints are given. So it is an interesting research issue that in what manners we can remind human contributors of thinking of CSK. In addition, most mining systems are based on the assumption that large-scale corpus can supply a subset of abundant easy-to-extract commonsensical sentences. More sophisticated mining systems should also be developed to discover the knowledge beneath literal content.

Second, most acquisition methods ignore the connection among CSK, collecting CSK regarding a single concept for one time. In fact, CSK is closely linked to each other. People often associate other knowledge from the knowledge they are entering. Therefore, we could plan a collection of commonsensical questions before presenting them to knowledge contributors. Thus, contributors can enter a cluster of knowledge at one time instead of entering them independently.

Third, most acquisition methods focus on simple object notations, collecting their properties and relations, while little effort has been made on events (or processes) with complex internal structures. Current researchers often view events as a whole and do not distinguish them from objects. In fact, events can involve objects as their participants, and thus events can be viewed as lying on the higher level than that of objects. An interesting research issue is regarding events, for example, which property of an event can be inherited by its participants and vice versa. For instance, the location where an event happened is just the location of its object participants during the event's process.

Finally, reasonable and fair evaluation method and evaluation workbench are unavailable. As we can see from the comparison section, most evaluations focus on the correctness of collected knowledge and the efficiency of acquisition. Little work has been done on the coverage/comprehensiveness of the collected CSK since gold-standard collection of CSK is unobtainable. In addition, there are few application evaluations by which we can measure the usefulness of the collected knowledge. As a result, a workbench providing a more fair comparison environment is a necessity, which can attract as many commonsense researchers as possible.

References

[1] Minsky M. The Emotion Machine. Simon & Schuster New York, 2006.

[2] McCarthy J. Some expert systems need common sense. *Annals of the New York Academy of Sciences*, 1984, 426:129-137.

[3] Lenat D, Prakash M, Shepherd M. CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine*, 1986, 6(4):65-85.

[4] Lieberman H, Liu H, Singh P, Barry B. Beating common sense into interactive applications. *AI Magazine*, 2004, 25(4):63-76.

[5] Curtis J, Cabral J, Baxter D. On the application of the Cyc ontology to word sense disambiguation. In *Proc. the 19th Int. Florida AI Research Society Conf.*, May 2006, pp.652-657.

[6] Dahlgren K, McDowell J. Using commonsense knowledge to disambiguate prepositional phrase modifiers. In *Proc. the 5th National Conf. Artificial Intelligence*, Aug. 1986, pp.589-593.

[7] Havasi C, Speer R, Pustejovsky J. Coarse word-sense disambiguation using common sense. In *Proc. the AAAI Fall Symposium Series*, Nov. 2010.

[8] Cambria E, Hussain A, Havasi C, Eckl C. Affectivespace: Blending common sense and affective knowledge to perform emotive reasoning. In *Proc. the 1st WOMSA at CAEPIA 2009*, Nov. 2009, pp.32-41.

[9] Cambria E, Hussain A, Havasi C, Eckl C. Sentic computing: Exploitation of common sense for the development of emotion-sensitive systems. In *Lecture Notes in Computer Science 5967*, Hutchison D, Kanade T, Kittler J *et al.* (eds.), 2010, pp.148-156.

[10] Curtis J, Matthews G, Baxter D. On the effective use of Cyc in a question answering system. In *Proc. IJCAI Workshop. Knowledge and Reasoning for Answering Questions*, Aug. 2005.

[11] Liu H, Singh P. Makebelieve: Using commonsense knowledge to generate stories. In *Proc. the 18th National Conf. Artificial Intelligence*, Jul.28-Aug.1, 2002, pp.957-958.

[12] Ong E. A commonsense knowledge base for generating children's stories. In *Proc. the AAAI Fall Symposium Series on Common Sense Knowledge*, Nov. 2010, pp.82-87.

[13] Liu H, Lieberman H, Selker T. Goose: A goal-oriented search engine with commonsense. In *Proc. the 2nd Int. Conf. Adaptive Hypermedia and Adaptive Web-Based Systems*, Jun. 2006, pp.253-263.

[14] Hsu M, Chen H. Information retrieval with commonsense knowledge. In *Proc. the 29th SIGIR 2006*, Aug. 2006, pp.651-652.

[15] Nilsson N. Artificial Intelligence: A New Synthesis. Morgan Kaufmann, 1998.

[16] Gupta R, Kochenderfer M. Common sense data acquisition for indoor mobile robots. In *Proc. the 19th National Conf. Artificial Intelligence*, Jul. 2004, pp.605-610.

[17] McCarthy J. Programs with common sense. In *Proc. the Teddington Conf. the Mechanization of Thought Processes*, Dec. 1958.

[18] Lenat D B. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 1995, 38(11): 33-38.

[19] Singh P, Lin T, Mueller E, Lim G, Perkins T, Zhu W. Open mind common sense: Knowledge acquisition from the general public. In *Proc. Conf. Cooperative Information Systems*, Oct.30-Nov.1 2002, pp.1223-1237.

[20] Dong Z, Dong Q. HowNet and the Computation of Meaning. Singapore: World Scientific Publishing Company, 2006.

[21] Liu H, Singh P. Conceptnet—A practical commonsense reasoning tool-kit. *BT Technology Journal*, 2004, 22(4): 211-226.

[22] Speer R, Havasi C, Lieberman H. AnalogySpace: Reducing the dimensionality of common sense knowledge. In *Proc. the 23rd AAAI*, Jul. 2008, pp.548-553.

[23] Schubert L. Can we derive general world knowledge from texts? In *Proc. the 2nd Int. Conf. Human Language Technology Research*, Mar. 2002, pp.94-97.

[24] Torisawa K. An unsupervised learning method for commonsensical inference rules on events. In *Proc. the 2nd CoLogNet-EIsNET Symposium*, Dec. 2003.

[25] Torisawa K. Acquiring inference rules with temporal constraints by using Japanese coordinated sentences and noun-verb co-occurrences. In *Proc. the Human Language Technology Conf. the North American Chapter of the Association of Computational Linguistics* (*HLT/NAACL*), Jun. 2006, pp.57-64.

[26] Chklovski T. Learner: A system for acquiring commonsense knowledge by analogy. In *Proc. the 2nd Int. Conf. Knowledge Capture*, Oct. 2003, pp.4-12.

[27] Witbrock M, Matuszek C, Brusseau A, Kahlert R, Fraser C, Lenat D. Knowledge begets knowledge: Steps towards assisted knowledge acquisition in Cyc. In *Proc. the AAAI Spring Symposium on Knowledge Collection from Volunteer Contributors*, Mar. 2005, pp.99-105.

[28] von Ahn L, Kedia M, Blum M. Verbosity: A game for collecting common-sense facts. In *Proc. the ACM SIGCHI Conf. Human Factors in Computing Systems*, Apr. 2006, pp.75-78.

[29] Lieberman H, Smith D, Teeters A. Common consensus: A web-based game for collecting commonsense goals. In *Proc. Int. Conf. Intelligent User Interfaces*, Jan. 2007.

[30] Speer R, Krishnamurthy J, Havasi C, Smith D, Lieberman H, Arnold K. An interface for targeted collection of common sense knowledge using a mixture model. In *Proc. the 14th Int. Conf. Intelligent User Interfaces*, Feb. 2009, pp.137-146.

[31] Kuo Y, Lee J, Chiang K, Wang R, Shen E, Chan C, Hsu J. Community-based game design: Experiments on social games for commonsense data collection. In *Proc. the ACM SIGKDD Workshop on Human Computation*, Jun. 2009, pp.15-22.

[32] Kuo Y, Hsu J. Goal-oriented knowledge collection. In *Proc. the AAAI Fall Symposium Series*, Nov. 2010.

[33] Banko M, Cafarella M, Soderland S, Broadhead M, Etzioni O. Open information extraction from the web. In *Proc. the 20th IJCAI*, Jan. 2007, pp. 2670-2676.

[34] Etzioni O, Fader A, Christensen J, Soderland S, Center M. Open information extraction: The second generation. In *Proc. the 22th Int. Joint Conf. Artificial Intelligence*, Jul. 2011, pp.3-10.

[35] Fader A, Soderland S, Etzioni O. Identifying relations for open information extraction. In *Proc. the Conf. Empirical Methods in Natural Language Processing*, Jul. 2011, pp.1535-1545.

[36] Mausam J C, Soderland S, Etzioni O. Learning arguments for open information extraction. Technical Report, University of Washington, 2011.

[37] Matuszek C, Witbrock M, Kahlert R, Cabral J, Schneider D, Shah P, Lenat D. Searching for common sense: Populating Cyc$^{\mathrm{TM}}$ from the Web. In *Proc. the National Conf. Artificial Intelligence*, Jul. 2005, pp.1430-1435.

[38] Forbus K, Riesbeck C, Birnbaum L, Livingston K, Sharma A, Ureel L. Integrating natural language, knowledge representation and reasoning, and analogical processing to learn by reading. In *Proc. the 22nd National Conf. Artificial Intelligence*, Jul. 2007, pp.1542-1547.

[39] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A nucleus for a web of open data. In *Proc. the ISWC 2007/ASWC2007*, Nov. 2007, pp.722-735.

[40] Bizer C, Lehmann J, Kobilarov G, Auer S, Becker C, Cyganiak R, Hellmann S. DBpedia: A crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2009, 7(3): 154-165.

[41] Suchanek F M, Kasneci G, Weikum G. YAGO: A core of semantic knowledge. In *Proc. the 16th Int. Conf. World Wide Web*, May 2007, pp.697-706.

[42] Hoffart J, Suchanek F M, Berberich K, Weikum G. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 2013, 194: 28-61.

[43] Niu X, Sun X, Wang H, Rong S, Qi G, Yu Y. Zhishi.me: Weaving Chinese linking open data. In *Proc. the 10th Int. Conf. the Semantic Web*, Oct. 2011, pp.205-220.

[44] Wang Z C, Wang Z G, Li J Z, Pan J Z. Knowledge extraction from Chinese wiki encyclopedias. *Journal of Zhejiang University—Science C*, 2012, 13(4): 268-280.

[45] Zeng Y. Extracting, linking and analyzing the Web of structured Chinese data. Technical Report, Institute of Automation, Chinese Academy of Sciences, 2012.

[46] Singh P. The public acquisition of commonsense knowledge. In *Proc. the AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, Mar. 2002.

[47] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 2005, 17(6): 734-749.

[48] Schubert L, Tong M. Extracting and evaluating general world knowledge from the Brown corpus. In *Proc. the HLT-NAACL 2003 Workshop on Text Meaning*, Vol.9, May 2003, pp.7-13.

[49] Zhu Y, Zang L, Wang D, Cao C. A manual experiment on commonsense knowledge acquisition from web corpora. In *Proc. the Int. Conf. Machine Learning and Cybernetics*, Vol.3, Jul. 2008, pp.1564-1569.

[50] Vanderwende L. Volunteers created the web. In *Proc. AAAI Spring Symposium on Knowledge Collection from Volunteer Cotnributors*, Mar. 2005.

[51] Eslick I. Searching for commonsense [Master Thesis]. Massachusetts Institute of Technology, 2006.

[52] Etzioni O, Cafarella M, Downey D, Kok S, Popescu A, Shaked T, Soderland S, Weld D, Yates A. Web-scale information extraction in KnowItAll: (Preliminary results). In *Proc. the 13th Int. Conf. World Wide Web*, May 2004, pp.100-110.

[53] Etzioni O, Cafarella M, Downey D, Popescu A, Shaked T, Soderland S, Weld D, Yates A. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 2005, 165(1): 91-134.

[54] Sharma A, Forbus K. Graph-based reasoning and reinforcement learning for improving Q/A performance in large knowledge-based systems. In *Proc. the AAAI Fall Symposium Series*, Nov. 2010.

[55] Liu H, Lieberman H, Selker T. A model of textual affect sensing using real-world knowledge. In *Proc. the 8th Int. Conf. Intelligent User Interfaces*, Jan. 2003, pp.125-132.

[56] Gordon A S, Kozareva Z, Roemmele M. SemEval-2012 Task7 : Choice of Plausible Alternatives : An Evaluation of Commonsense Causal Reasoning. In *Proc. the 1st Joint Conference on Lexical and Computational Semantics*, Jun. 2012.

[57] Gennari J H, Musen M A, Fergerson R W, Grosso W E, Crubézy M, Eriksson H, Noy N F, Tu S W. The evolution of Protégé: An environment for knowledge-based systems development. *Int. Journal of Human-Computer Studies*, 2003, 58(1): 89-123.

[58] Panton K, Miraglia P, Salay N, Kahlert R C, Baxter D, Reagan R. Knowledge formation and dialogue using the KRAKEN toolset. In *Proc. AAAI*, Jul. 2002, pp.900-905.

[59] Witbrock M, Baxter D, Curtis J, Schneider D, Kahlert R, Miraglia P, Wagner P, Panton K, Matthews G. An interactive dialogue system for knowledge acquisition in Cyc. In *Proc. Int. Joint Conf. Artificial Intelligence*, Aug. 2003.

[60] Quinlan J, Cameron-Jones R. Induction of logic programs: FOIL and related systems. *New Generation Computing*, 1995, 13(3): 287-312.

[61] Masters J, Matuszek C, Witbrock M. Ontology-based integration of knowledge from semi-structured web pages. Technical Report, Cycorp, 2006.

[62] Medelyan O, Legg C. Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense. In *Proc. AAAI Workshop on Wikipedia and AI*, Jul. 2008.

[63] Pierce C, Booth D, Ogbuji C, Deaton C, Blackstone E, Lenat D. SemanticDB: A semantic Web infrastructure for clinical research and quality reporting. *Current Bioinformatics*, 2012, 7(3): 267-277.

[64] Witbrock M, Panton K, Reed S, Schneider D, Aldag B, Reimers M, Bertolo S. Automated OWL annotation assisted by a large knowledge base. In *Proc. the ISWC Workshop on Knowledge Markup and Semantic Annotation*, Nov. 2004, pp.71-80.

[65] Mueller E. Natural language processing with ThoughtTreasure. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.9720, 1998.

[66] Mueller E. A calendar with common sense. In *Proc. the 5th Int. Conf. Intelligent User Interfaces*, Jan. 2000, pp.198-201.

[67] Speer R, Havasi C, Surana H. Using verbosity: Common sense data from games with a purpose. In *Proc. the 23nd Int. Florida AI Research Society Conf.*, May 2010.

[68] Anacleto J, Lieberman H, Tsutsumi M, Neris V, Carvalho A, Espinosa J, Godoi M, Zem-Mascarenhas S. Can common sense uncover cultural differences in computer applications? In *Proc. the 19th World Computer Congress*, August 2006, pp.1-10.

[69] Eckhardt N. A kid's open mind common sense [Ph.D. Thesis]. Tilburg University, 2008.

[70] Chung H. GlobalMind — Bridging the gap between different cultures and languages with commonsense computing [Master Thesis]. Massachusetts Institute of Technology, 2006.

[71] Cambria E, Xia Y, Hussain A. Affective common sense knowledge acquisition for sentiment analysis. In *Proc. the 8th Int. Conf. Language Resources and Evaluation*, May 2012, pp.3580-3595.

[72] von Ahn L. Games with a purpose. *IEEE Computer Magazine*, 2006, 39(6): 92-94.

[73] Speer R. Open mind commons: An inquisitive approach to learning common sense. In *Proc. the Workshop on Common Sense and Intelligent User Interfaces*, Jan. 2007.

[74] Minsky M. The Society Of Mind. Simon and Schuster, 1988.

[75] Fellbaum C. WordNet: An Electronic Lexical Database. The MIT press, 1998.

[76] Havasi C, Speer R, Alonso J. ConceptNet 3: A flexible, multilingual semantic network for common sense knowledge. In *Proc. the 22nd AAAI*, Sept. 2007.

[77] Alonso J. CSAMOA: A common sense application model of architecture. In *Proc. the Workshop on Common Sense and Intelligent User Interfaces*, Jan. 2007.

[78] Pustejovsky J, Havasi C, Sauri R *et al.* Towards a generative lexical resource: The brandeis semantic ontology. In *Proc. the Language Resources and Evaluation Conf.*, May 2006.

[79] Speer R, Havasi C. Representing general relational knowledge in ConceptNet 5. In *Proc. the 8th Int. Conf. Language Resources and Evaluation*, May 2012, pp.3679-3686.

[80] Havasi C, Speer R, Pustejovsky J, Lieberman H. Digital intuition: Applying common sense using dimensionality reduction. *IEEE Intelligent Systems*, 2009, 24(4): 24-35.

[81] von Assem M, von Ossenbruggen J. Wordnet 3.0 in RDF. http://semanticweb. cs. vu. nl/lod/wn30/, Sep. 2011, pp.10-24.

[82] von Ahn L, Dabbish L. Designing games with a purpose. *Communications of the ACM*, 2008, 51(8): 58-67.

[83] Androutsopoulos I, Malakasiotis P. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 2010, 38(1): 135-187.

[84] Hashimoto C, Torisawa K, De Saeger S, Oh J, Kazama J. Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the Web. In *Proc. EMNLP-CoNLL 2012*, Jul. 2012.

[85] Tian W, Cao C, H W. Representation, acquisition and analysis of psychological commonsense concepts. *Computer Science*, 2004, 31(6): 5-12. (in Chinese)

[86] Peng H, Cao C. Research on mining of associated events and discovering roles relaionship. *Computer Science*, 2010, 12. (in Chinese)

[87] Cao Y, Cao C, Zang L, Zhu Y, Wang S, Wang D. Acquiring commonsense knowledge about properties of concepts from text. In *Proc. the 5th Int. Conf. Fuzzy Systems and Knowledge Discovery*, Aug. 2008, Vol.4, pp.155-159.

[88] Cao Y, Cao C, Zang L, Wang S, Wang D. Extracting comparative commonsense from the Web. In *Proc. the IFIP International Conference on Intelligent Information Processing*, Oct. 2010, pp.154-162.

[89] Cao Y, Cao C, Zang L, Wang S. Web mining for causal relations between events. *Information: An International Interdisci Journal*, 2011, 15(1).

[90] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proc. SIGMOD2008*, 2008, pp. 1247 - 1250.

[91] Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes P N, Hellmann S, Morsey M, Kleef v P, Auer S, Bizer C. Dbpedia — A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web Journal*, 2013, Under review.

[92] Ritter A, Mausam, Etzioni O. A latent dirichlet allocation method for selectional preferences. In *Proc. the 48th Annual Meeting of the Association for Computational Linguistics*, Jul. 2010, pp.424-434.

[93] Lin T, Mausam, Etzioni O. Identifying functional relations in web text. In *Proc. the Conf. Empirical Methods in Natural Language Processing*, Oct. 2010, pp.1266-1276.

[94] Schoenmackers S, Etzioni O, Weld D S, Davis J. Learning first-order Horn clauses from Web text. In *Proc. the Conf. on Empirical Methods in Natural Language Processing*, Oct. 2010, pp.1088-1098.

[95] Berant J, Dagan I, Goldberger J. Global learning of typed entailment rules. In *Proc. the 49th HLT-ACL*, Jun. 2011, vol.1, pp. 610-619.

[96] Soderland S, Roof B, Qin B, Xu S, Mausam, Etzioni O. Adapting open information extraction to domain-specific relations. *AI Magazine*, 2010, 31(3): 93-102.

[97] Klein D, Manning C. Accurate unlexicalized parsing. In *Proc. the 41st Annual Meeting on Association for Computational Linguistics*, Jul. 2003, Vol.1, pp.423-430.

[98] Downey D, Etzioni O, Soderland S. A probabilistic model of redundancy in information extraction. In *Proc. Int. Joint Conf. Artificial Intelligence*, Jul.30-Aug.5, 2005, Vol.19.

[99] Banko M, Etzioni O. The tradeoffs between open and traditional relation extraction. In *Proc. the 46th ACL*, Jun. 2008, pp.28-36.

[100] Wu F, Weld D S. Open Information Extraction Using Wikipedia. In *Proc. the 48th ACL*, Jul. 2010, pp. 118-127.

[101] Wu W, Li H, Wang H, Zhu K Q. Probase: A probabilistic taxonomy for text understanding. In *Proc. the ACM Int. Conf. Management of Data*, Nov. 2012, pp. 481–492.

[102] Ponzetto S P, Strube M. Deriving a large scale taxonomy from Wikipedia. In *Proc. AAAI2007*, Jul. 2007, Volume 22.

[103] Wang Y, Li H, Wang H, Zhu K Q. Towards topic search on the Web. Technical Report, Microsoft Research, 2010.

[104] Song Y, Wang H, Wang Z, Li H, Chen W. Short text conceptualization using a probabilistic knowledge base. In *Proc. IJCAI2011*, Jul. 2011, pp. 2330–2336.

[105] Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka Jr E, Mitchell T. Toward an architecture for never-ending language learning. In *Proc. the 24th AAAI*, Jul. 2010, Vol.2.

[106] Carlson A, Betteridge J, Wang R *et al.* Coupled semi-supervised learning for information extraction. In *Proc. the 3rd ACM Int. Conf. Web Search and Data Mining*, Feb. 2010, pp.101-110.

[107] Fensel D, von Harmelen F, Andersson B *et al.* Towards LarKC: A platform for Web scale reasoning. In *Proc. Int. Conf. Semantic Computing*, Aug. 2008, pp.524-529.

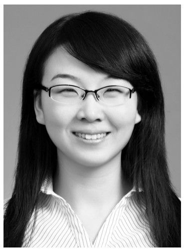
[108] Assel M, Cheptsov A, Gallizo G *et al.* Large knowledge collider: A service-oriented platform for large-scale semantic reasoning. In *Proc. the International Conference on Web Intelligence, Mining and Semantics*, May 2011.

[109] Pasca M, Van Durme B. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proc. the 46th Annual Meeting of the Association for Computational Linguistics*, Jun. 2008, pp.19-27.

[110] Turney P. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Proc. the 12th European Conf. Machine Learning*, Sept. 2001, pp. 491-502.

Liang-Jun Zang is a Ph.D. candidate in the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing. His current research interests include knowledge acquisition, representation and reasoning, ontology, mathematical logic, machine learning, and NLP.

Cong Cao is a Ph.D. candidate in the Key Laboratory of Intelligent Information Processing, ICT, CAS. His current research interests include data mining and commonsense knowledge acquisition.



Ya-Nan Cao is an assistant researcher in Institute of Information Engineering, Chinese Academy of Sciences (CAS). She received the Ph.D. degree in computer science from the Institute of Computing Technology, CAS in 2012. Her current research interest includes large-scale knowledge acquisition, information retrieval, and text mining.

Yu-Ming Wu received his Bachelor degree from Dalian Jiaotong University in 2002, his M.S. degree from Capital Normal University, Beijing, in 2008, and his Ph.D. degree in computer software and theory from ICT, CAS, in 2012. Now he is an assistant professor in ICT, CAS. His research interests include grammar learning, natural language processing, large-scale knowledge processing, and ontology learning.

Cun-Gen Cao received his M.S. and Ph.D. degrees in software from the Institute of Mathematics, Chinese Academy of Sciences in 1989 and 1993 respectively. He is a full-time professor of ICT, CAS, since 2000. He is leading a research group of large-scale knowledge engineering and knowledge-intensive applications.