Machine Reading

Oren Etzioni, Michele Banko, Michael J. Cafarella

Computer Science & Engineering
University of Washington
etzioni@cs.washington.edu

Introduction

The time is ripe for the AI community to set its sights on ***Machine Reading***---the automatic, unsupervised understanding of text. In this paper, we place the notion of "Machine Reading" in context, describe progress towards this goal by the KnowItAll research group at the University of Washington, and highlight several central research questions.

Over the last two decades or so, Natural Language Processing (NLP) has developed powerful methods for low-level syntactic and semantic text processing tasks such as parsing, semantic role labeling, and text categorization. Over the same period, the fields of machine learning and probabilistic reasoning have yielded important breakthroughs as well. It is now time to investigate how to leverage these advances to understand text.[1]

By "understanding text" I mean the formation of a coherent set of beliefs based on a textual corpus and a background theory. Because the text and the background theory may be inconsistent, it is natural to express the resultant beliefs, and the reasoning process in probabilistic terms. A key problem is that many of the beliefs of interest are only *implied* by the text in combination with a background theory. To recall Roger Schank's old example, if the text states that a person left a restaurant after a satisfactory meal, it is reasonable to infer that he is likely to have paid the bill and left a tip. Thus, inference is an integral part of text understanding.

[1] Similar observations have been made recently by Tom Mitchell (Mitchell 2005), Noah Friedland (Friedland 2005), and others. Independently, our research group has been vigorously pursuing this goal over the last three years via the KnowItAll family of unsupervised Web information extraction systems.

Related Work

Machine Reading (MR) is very different from current semantic NLP research areas such as Information Extraction (IE) or Question Answering (QA). Many NLP tasks utilize supervised learning techniques, which rely on hand-tagged training examples. For example, IE systems often utilize extraction rules learned from example extractions of each target relation. Yet MR is not limited to a small set of target relations. In fact, the relations encountered when reading arbitrary text are not known in advance! Thus, it is impractical to generate a set of hand-tagged examples of each relation of interest. In contrast with many NLP tasks, *MR is inherently unsupervised*.

Another important difference is that IE and QA focus on isolated "nuggets" obtained from text whereas MR is about forging and updating connections between beliefs. While MR will build on NLP techniques, it is a holistic process that synthesizes information gleaned from text with the machine's existing knowledge.

Textual Entailment (TE) (Dagan, Glickman, and Magnini 2005) is much closer in spirit to MR, but with some important differences. TE systems determine whether one sentence is entailed by another. This is a valuable abstraction that naturally lends itself to tasks such as paraphrasing, summarization etc. MR is more ambitious, however, in that it combines multiple TE steps to form a coherent set of beliefs based on the text. In addition, MR is focused on scaling up to arbitrary relations and doing away with hand-tagged training examples. Thus, TE is an important component of MR, but far from the whole story.

Discussion

For the foreseeable future, humans' ability to grasp the intricate nuances of text will far surpass that of machines. However, MR will have some intriguing strengths. First, MR will be fast. Today's machines already map a sentence to a "shallow" semantic representation in a few milliseconds. Second, MR will leverage statistics

computed over massive corpora. For example, Peter Turney (Turney 2002) has shown how mutual-information statistics, computed over the Web corpus, can be used to classify opinion words as positive or negative with high accuracy.

These observations suggest a loose analogy between MR and Computer Chess. The computer's approach to playing chess is very different than that of a person. Each player, human or computer, builds on their own "natural" strengths. A computer's ability to analyze the nuances of a chess position (or a sentence) is far weaker than that of a person, but the computer makes up for this weakness with its superior memory and speed. Of course, MR is an "ill-structured problem" that the computer cannot solve by mere look ahead. However, I conjecture that MR, like computer chess, will be "shallow" yet lightning fast.

Initial Steps towards Machine Reading

The field of Information extraction (IE) has taken some preliminary steps towards text understanding. IE has traditionally relied on human involvement to identify instances of a small, predefined set of relations, but modern information extraction has sought to reduce the amount of human labor necessary to extract information in a new domain or set of relations.

An important step in this direction has been the training of IE systems using hand-tagged training examples. When the examples are fed to machine learning methods, domain-specific extraction patterns can be automatically learned and used to extract facts from text. However, the development of suitable training data requires a non-trivial amount of effort and expertise.

DIPRE (Brin, 1998) and Snowball (Agichtein, 2000) further demonstrated the power of trainable information extraction systems by reducing the amount of manual labor necessary to perform relation-specific extraction. Rather than demand hand-tagged corpora, these systems require a user to specify relation-specific knowledge in the form of a small set of seed instances known to satisfy the relation of interest or a set of hand-constructed extraction patterns to begin the training process.

The KnowItAll Web IE system (Etzioni et al., 2005) took the next step in automation by learning to label its own training examples using only a small set of domain-independent extraction patterns, thus being the first published system to carry out unsupervised, domain-independent, large-scale extraction from Web pages.

When instantiated for a particular relation, these generic patterns yield relation-specific extraction rules that are then used to learn domain-specific extraction rules. The rules are applied to Web pages, identified via search-engine queries, and the resulting extractions are assigned a probability using mutual-information measures derived from search engine hit counts. For example, KnowItAll utilized generic extraction patterns like "<Class> such as <Mem>" to suggest instantiations of <Mem> as candidate members of the class. Next, KnowItAll used frequency information to identify which instantiations are most likely to be bona-fide members of the class. Thus, it was able to confidently label major cities including Seattle, Tel Aviv, and London as members of the class "Cities" (Downey, Etzioni, and Soderland 2005). Finally, KnowItAll learned a set of relation-specific extraction patterns (e.g. "headquartered in <city>") that led it to extract additional cities and so on.

KnowItAll is *self supervised*---instead of utilizing hand-tagged training data, the system selects and labels its own training examples, and iteratively bootstraps its learning process. In general, self-supervised systems are a species of unsupervised systems because they require *no* hand-tagged training examples whatsoever. However, unlike classical unsupervised systems (e.g., clustering) self-supervised systems *do* utilize labeled examples and *do* form classifiers whose accuracy can be measured using standard metrics. Instead of relying on hand-tagged data, self-supervised systems autonomously "roll their own" labeled examples.

While self-supervised, KnowItAll is *relation-specific*---it requires a laborious bootstrapping process for each relation of interest, and the set of relations of interest has to be named by the human user in advance. This is a significant obstacle to MR because during reading one often encounters unanticipated concepts and relations of great interest.

TextRunner

This limitation led us to develop TextRunner (Cafarella, Banko, Etzioni 2006), a system that seamlessly extracts information from each sentence it encounters. Instead of requiring relations to be specified in its input, TextRunner *learns* the relations, classes, and entities from the text in its corpus in a self-supervised fashion.

TextRunner's extractor module reads in sentences and rapidly extracts one or more textual triples that aim to capture (some of) the relationships in each sentence. For example, given the sentence "Berkeley hired Robert Oppenheimer to create a new school of theoretical physics", the extractor forms the triple (Berkeley, hired, Robert Oppenheimer). The triple consists of three strings where the first and third are meant to denote entities and the intermediate string is meant to denote the relationship between them. There are many subtleties to doing this

kind of extraction with good recall and precision, but we will not discuss there here.

TextRunner collects all of its triples into an *extraction graph*---a textual approximation to an entity-relationship graph, which is automatically extracted from Web pages. The graph is an intermediate representation that is more informative than a mere page-hyperlink graph but far easier to construct than a semantic network. The graph collects relationship information about particular entities (e.g., Oppenheimer) as edges emanating from a single node.

The extraction graph suffers from numerous problems including inconsistencies, synonymy, polysymy, entity duplication, and more. Each of these problems represents an area of active research for our group. Despite its limitations, the graph is efficiently indexed in Lucene, which enables TextRunner to rapidly answer queries regarding the extracted information (e.g., what are relationships between Oppenheimer and Berkeley?).

TextRunner operates at very large scale. In a recent run, it processed ninety million Web pages yielding over 1,000,000,000 extractions with an estimated precision of close to 70%. Clearly, TextRunner is an early embodiment of the idea that MR will be fast but shallow.

While TextRunner is a state-of-the-art IE system, its ability to read is very primitive. Its value is in showing that NLP techniques can be harnessed to begin to understand text in a domain-independent and unsupervised manner. We are now working on composing TextRunner extractions into coherent probabilistic theories, and on forming generalizations based on extracted assertions.

Conclusion

We have argued that the time is ripe for Machine Reading to join Machine Learning and Machine Translation as a full-fledged subfield of AI. We have described several initial steps in this direction, but numerous open problems remain.

One open problem worth highlighting is *recursive learning*---how can an MR system leverage the information it has read to date to enhance its understanding of the next sentences it encounters? Humans become exponentially more proficient at a task as they practice it---can we develop MR systems that exhibit some of that amazing learning capability?

In conclusion, Machine Reading is an ambitious undertaking but the pieces of the puzzle are at hand, and the payoff could lead to a solution to AI's infamous knowledge acquisition bottleneck.

Acknowledgements

This research was supported in part by NSF grant IIS-0312988, DARPA contract NBCHD030010, ONR grant N00014-05-1-0185 as well as a gift from Google. Thanks go to Dan Weld for helpful comments on a previous draft, and to all members of the KnowItAll Research Group for their insights and contributions. This work was performed at the University of Washington's Turing Center.

References

Agichtein, E., and Gravano, L., 2000. Snowball: Extracting Relations from Large Plain-Text Collections. Proceedings of the Fifth ACM International Conference on Digital Libraries.

Brin S., 1998. Extracting Patterns and Relations from the World Wide Web. WebDB Workshop.

Cafarella M., Banko M., and Etzioni O., 2006. Relational Web Search. University of Washington Technical Report, UW-CSE-2006-04-02.

Dagan, I., Glickman, O., and Magnini, B. 2005. The PASCAL Recognising Textual Entailment Challenge. In Proceedings of the first PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, U.K.: Pattern Analysis, Statistical Modelling and Computational Learning, Inc.

Downey, D., Etzioni, O., and Soderland, S. 2005. A Probabilistic Model of Redundancy in Information Extraction. In Proceedings of the 19th International Joint Conference on Artificial Intelligence, 1034-1041. Edinburgh, Scotland: International Joint Conference on Artificial Intelligence, Inc.

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

Friedland, N. 2005. Personal Communication.

Mitchell, T. 2005. Reading the Web: A Breakthrough Goal for AI. Celebrating Twenty-Five Years of AAAI: Notes from the AAAI-05 and IAAI-05 Conferences. *AI Magazine* 26(3):12-16.

Turney, P. D. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 417-424. Philadelphia, Penn.: Association for Computational Linguistics, Inc.