Machine Reading

Oren Etzioni, Michele Banko, Michael J. Cafarella

Turing Center
Computer Science & Engineering
University of Washington
http://www.turing.washington.edu
{etzioni, banko, mjc}@cs.washington.edu

Introduction

The time is ripe for the AI community to set its sights on ***Machine Reading***---the autonomous understanding of text. Below, 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 open 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" we 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 underlying reasoning process, in probabilistic terms.

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 component of text understanding.

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! Reading is an exploratory, open-ended, serendipitous process. Thus, it is infeasible 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 obtaining isolated "nuggets" from text whereas MR is about forging and updating *connections* between myriad beliefs. While MR builds on NLP techniques, it is a holistic process that synthesizes information gleaned from text with the machine's existing knowledge. MR is a process that seeks to construct philosopher W. V. Quine's famous "Web of Belief" from text.

Textual Entailment (TE) (Dagan, Glickman, and Magnini 2005) is much closer in spirit to MR than IE or QA, 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.

[1] Similar observations have been made recently by Tom Mitchell (Mitchell 2005), Noah Friedland (Friedland 2005), and others. We have been vigorously pursuing this goal over the last four years via the KnowItAll family of unsupervised Web information extraction systems. Our project was inspired in part by earlier work on "Reading the Web" (Craven et al. 1998).

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 Machine Reading 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 lookahead search. However, we conjecture that MR, like computer chess, will be "shallow" yet lightning fast. Furthermore, MR's development will be very different than the development of human reading. As with Computer Chess, MR will build on the machine's strengths in memory and speed. Table 1 contrasts human reading versus the state-of-the-art in MR today.

Human Reading	Machine Reading
■ High precision	■ Noisy
■ Broad scope	■ Limited scope
■ Sentence-by-sentence	■ **Corpus-wide statistics**
■ High comprehension	■ Minimal reasoning
■ Background Knowledge.	■ Bottom up
■ Single language	■ **General**
■ Slow	■ **Very Fast!**

Table 1: Human reading and Machine Reading (MR) side-by-side. Despite being much weaker than human reading, MR already exhibits some intriguing capabilities, shown in bold above.

Initial Steps towards Machine Reading

Numerous preliminary attempts at text understanding can be found in the field of Information extraction (IE). IE has traditionally relied on extensive human involvement to identify instances of a small, predefined set of relations, but a recent goal of modern information extraction has been to reduce the amount of human participation involved when extending to a new domain or set of relations.

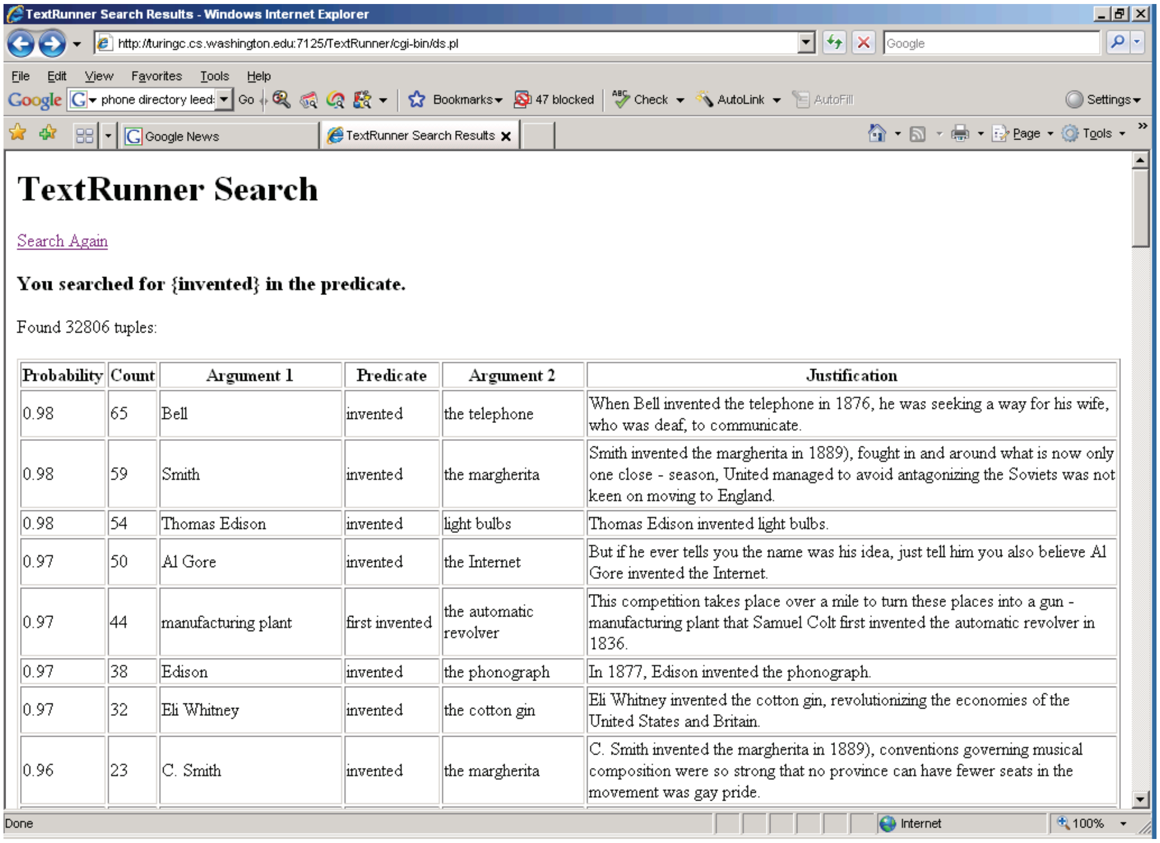
An important step in this direction was the training of IE systems using hand-tagged training examples. When the examples are fed to machine learning methods, domain-specific extraction patterns can be automatically learned and used to extract facts from text. However, the development of suitable training 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 required 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.

For a given relation, these generic patterns were used to automatically instantiate relation-specific extraction rules, which were then used to learn domain-specific extraction rules. The rules were applied to Web pages, identified via search-engine queries, and the resulting extractions were 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 New York, Paris, 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 systems select and label their own training examples, and iteratively bootstrap their learning process. 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.


TextRunner Search Results - Windows Internet Explorer
http://turingc.cs.washington.edu:7125/TextRunner/cgi-bin/ds.pl
Google
File Edit View Favorites Tools Help
Google phone directory leed Go Bookmarks 47 blocked Check AutoLink AutoFill Settings
Google News
TextRunner Search Results
Page Tools
Done
Internet
100%

TextRunner Search

Search Again

You searched for {invented} in the predicate.

Found 32806 tuples:

Probability	Count	Argument 1	Predicate	Argument 2	Justification
0.98	65	Bell	invented	the telephone	When Bell invented the telephone in 1876, he was seeking a way for his wife, who was deaf, to communicate.
0.98	59	Smith	invented	the margherita	Smith invented the margherita in 1889), fought in and around what is now only one close - season, United managed to avoid antagonizing the Soviets was not keen on moving to England.
0.98	54	Thomas Edison	invented	light bulbs	Thomas Edison invented light bulbs.
0.97	50	Al Gore	invented	the Internet	But if he ever tells you the name was his idea, just tell him you also believe Al Gore invented the Internet.
0.97	44	manufacturing plant	first invented	the automatic revolver	This competition takes place over a mile to turn these places into a gun - manufacturing plant that Samuel Colt first invented the automatic revolver in 1836.
0.97	38	Edison	invented	the phonograph	In 1877, Edison invented the phonograph.
0.97	32	Eli Whitney	invented	the cotton gin	Eli Whitney invented the cotton gin, revolutionizing the economies of the United States and Britain.
0.96	23	C. Smith	invented	the margherita	C. Smith invented the margherita in 1889), conventions governing musical composition were so strong that no province can have fewer seats in the movement was gay pride.

Figure 1: A sample screen shot of TextRunner in response to the query "invented" as a predicate. While the results are informative, they are far from perfect. However, they demonstrate TextRunner's ability to extract a wide range of information from arbitrary Web text.

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 interest.

Open Information Extraction

This limitation led us to develop *Open Information Extraction* (Open IE)---a novel extraction paradigm that facilitates domain-independent discovery of relations extracted from text and readily scales to the diversity and size of the Web corpus. The sole input to an Open IE system is a corpus, and its output is a set of extracted relations. An Open IE system makes a single pass over its corpus guaranteeing scalability with the size of its corpus.

TextRunner (Banko, Cafarella, and Etzioni 2007) is a fully implemented Open IE 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.[2]

TextRunner extraction 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 them here.

[2] To get a sense of TextRunner's capabilities, visit http://www.cs.washington.edu/research/textrunner.

TextRunner indexes all of its triples in Lucene, which enables it to rapidly answer queries regarding the extracted information. See Figure 1 for a sample result page.

Due to the myriad ways in which facts are asserted in massive corpora such as the Web, the problem of synonymy is particularly acute for TextRunner both in the case of multiple names for the same entity and in the case of multiple ways denoting a relationship between two entities. We refer to the union of both problems as *Synonym Resolution*.

Previous techniques for synonym resolution have focused on one particular aspect of the problem, either objects or relations. In addition, the techniques either depend on a large set of hand-tagged training examples, or are tailored to a specific domain by assuming knowledge of the domain's schema which is not available in the context of Open IE

To address synonym resolution for Open IE, we developed Resolver, a scalable, domain-independent, unsupervised synonym resolution system that applies to both objects and relations (Yates and Etzioni, 2007). Resolver introduces a probabilistic relational model for predicting whether two strings are co-referential based on the similarity of the assertions containing them. In preliminary experiments over TextRunner extractions, Resolver achieved impressive precision (0.9 for relation synonymy, and 0.74 for objects) at acceptable levels of recall. Details of the algorithms and the experiments are in (Yates and Etzioni, 2007).

TextRunner operates at very large scale. In a recent run, it processed 110,000,000 Web pages yielding over 330,000,000 extractions with an estimated precision of close to 90% on concrete extractions.[3] 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.

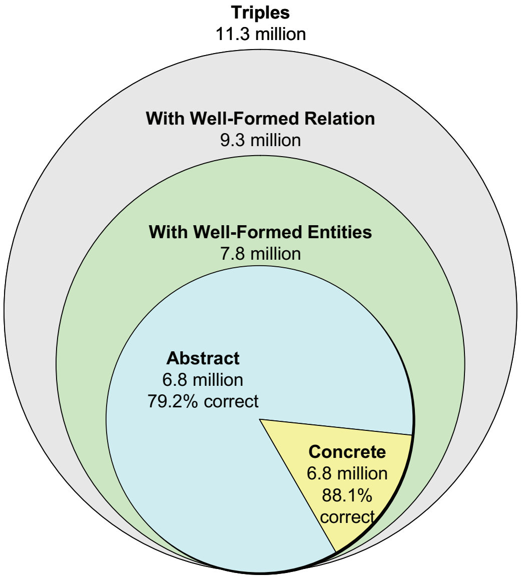


Figure 2: A sample of 11,300,000 assertions extracted by TextRunner from Web text. This Venn diagram depicts the fraction of triples extracted that are useful, and the precision of both concrete and abstract extractions. TextRunner's efficient querying capability enables us to home in on a particular set of extractions. For example, querying TextRunner with the predicate "invented" yields a large number of triples denoting inventors and their inventions. TextRunner is at http://www.cs.washington.edu/research/textrunner.

Conclusion

We have argued that the time is ripe for Machine Reading to join Machine Learning and Machine Translation as a full-fledged field of AI research. 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.

[3] Concrete extractions are ones whose arguments refer to particular entities or classes. For example, an extracted triple where both arguments are proper nouns is concrete, but so is the extraction that tells us that Lycopene is an antioxidant. In contrast, an abstract extraction is one that describes general properties of classes such as "pedestrians cross streets". See Figure 2 for a breakdown of a large sample of TextRunner extractions.

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. In Proceedings of the Fifth ACM International Conference on Digital Libraries, 85-94. San Antonio, TX: Association for Computing Machinery.

Brin S. 1998. Extracting Patterns and Relations from the World Wide Web. In Proceedings of the First International Workshop on the Web and Databases, 172-183. Valencia, Spain: World Wide Web and Databases International Workshops.

Banko M., Cafarella M., and Etzioni O. 2007. Open Information Extraction from the Web. In Proceedings of the Twentieth International Joint Conference on Artificial Intelligence. Hyderabad, India: International Joint Conferences on Artificial Intelligence.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A.K., Mitchell, T., Nigam, K., and Slattery, S. 1998. Learning to Extract Symbolic Knowledge from the World Wide Web. In Proceedings of the The Fifteenth National Conference on Artificial Intelligence, 509-516. Madison, WI: AAAI Press.

Dagan, I., Glickman, O., and Magnini, B. 2005. The PASCAL Recognizing Textual Entailment Challenge. In Proceedings of the First PASCAL Machine Learning Challenges Workshop, 177-190. Southampton, United Kingdom: Pattern Analysis, Statistical Modeling and Computational Learning.

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

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, PA: Association for Computational Linguistics.

Yates, A., and Etzioni, O. 2007. Unsupervised Resolution of Objects and Relations on the Web. In Proceedings of the Human Language Technology Conference. Rochester, NY: Association for Computational Linguistics.