HIGGINS: Knowledge Acquisition Meets the Crowds

Sarath Kumar Kondreddi
Max Planck Institute for Informatics
Saarbrücken, Germany
skondred@mpi-inf.mpg.de

Peter Triantafillou
University of Patras
Rio-Patra, Greece
peter@ceid.upatras.gr

Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

We present HIGGINS, a system for *Knowledge Acquisition (KA)*, placing emphasis on its architecture. The distinguishing characteristic and novelty of HIGGINS lies in its blending of two engines: an automated *Information Extraction (IE)* engine, aided by *semantic resources* and *statistics*, and a game-based *Human Computing (HC)* engine. We focus on KA from web pages and text sources and, in particular, on deriving relationships between entities. As a running application we utilize movie narratives, from which we wish to derive relationships among movie characters.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems**]: Human information processing

Keywords

Knowledge acquisition; Information Extraction; Human Computing

1. RATIONALE

KA critically relies on IE technology, combining methods from pattern matching, computational linguistics, and statistical learning. *Open IE* methods [1] can derive a wide diversity of relational facts between entities (instances of binary relationships, ER triples) based on detecting and analyzing noun phrases (for entities) and verb-centric phrases (for relations), such as: "Vesper" "finally falls for" "Bond". Hence, automated IE plays a key role in HIGGINS. However, there are fundamental limitations of IE technology. IE methods can yield noisy or nonsensical ER triples such a "Vesper" "certainly has" "Bond" (from the sentence "... certainly has ... sized up"). This occurs because automated IE generally faces obstacles: input sentences with complex structures, use of pronouns and other anaphoras, and ambiguous wording. The following snippet (from `imdb.com`) is a daunting example: "He quickly grabs Vesper and they kiss in the stairway entrance to cover themselves." Crowdsourcing, a.k.a *Human Computing (HC)*, has been successfully employed to help with tasks where fully automated solutions are deemed inadequate [2, 3]. Our thesis is that HC is a natural alternative to overcome the limitations of automated IE. It can tap human intelligence and knowledge to assess candidate facts and to correct errors. Further, humans can readily recognize "surreal" relationships, (e.g., occurring in dreams, being imaginary, contained in intentions, etc.), such as "Vesper Lynd" "*pretends to love*" "James Bond", and note that they do not correspond to true facts. Human *intelligence* can help to resolve pronouns in complex sentences or to identify erroneous paths in the dependency-parsing of natural language. Human *knowledge* on special topics such as movies, books, or medicine can add new facts, that may be entirely missing from the text, or help derive ER triples, e.g., about movie or book characters, that are virtually impossible to extract automatically, as they may not be explicitly mentioned. Despite this great potential, to our knowledge there have been no previous attempts to employ HC and crowdsourcing platforms for the difficult KA task of extracting ER triples.

For applications with human experts, (e.g., movies with movie aficionados, or book lovers, or experts in diseases and medicine, etc.), one would expect that HC can be nicely cast into game form, thus enticing more users to contribute on the KA task. However, despite the inherent promises of HC for KA, humans alone cannot carry this burden. First, the number of real experts is typically limited. Second, these experts are not so likely to participate in online games. Hence, inevitably, HC output will contain a wide range from high-quality to highly noisy and incorrect facts. One may think that these HC errors could be compensated by large-scale crowdsourcing, with redundant Human Intelligence Tasks (HITS) and statistic reasoning over many contributors. However, there is still the issue of the total cost: each HIT may cost a few cents only, but paying for hundreds of thousands or millions of HITs quickly becomes prohibitive.

2. STRATEGY & ARCHITECTURE

HIGGINS is a novel architecture that *blends* HC inputs with machine-centric IE and thus overcomes the limitations of either a purely HC-based or purely IE-based approach to advanced KA. The key idea is to use automatic IE to generate questions and candidate answers (for multiple-choice questions), for a KA game. Our expectation is that this can improve the quality of user contributions and reduce the overall cost of crowdsourcing [1].

HIGGINS' strategy for KA of ER triples has two phases:

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

[1]Work supported by the 7th Framework IST programme of the European Union, through the Longitudinal Analytics of Web Archive data (LAWA) project, contract no. 258105

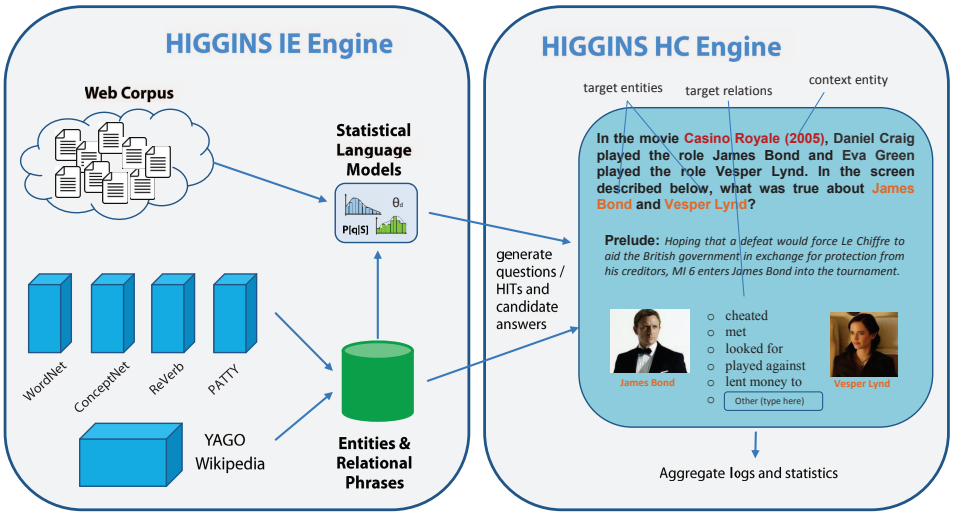

Figure 1: Overview of the Higgins Architecture

1. IE phase: We employ automated IE on Web corpora, in order to derive candidates for ER triples, with an open set of potential relations. We use a suite of techniques from computational linguistics, including dependency parsing (with the Stanford Parser) and pronoun resolution (with our own customized method). The resulting triples are usually of mixed quality, necessitating the second stage.
2. HC phase: The sets of candidates from the IE stage and their underlying patterns are then used to generate HITs in game form. Abstractly, each HIT presents the user with a *knowledge quad* of the form $(c, e1, r, e2)$ where $e1$ and $e2$ are entities, r is a relation, and c is a cue or textual context. One or more of the components $e1$, r, and $e2$ can be empty slots (variables) to be filled by the user; we may present a multiple-choice list to the user to pick the missing value. The quads are presented in the form of questions, with candidate answers and additional free-text fields for entering further values. We focus on the case where the relationship r is left to be filled, and both entities and the context are given.

HIGGINS' design principles are:

1. The IE engine is tuned to work very aggressively (aiming for high recall) capturing as many relational patterns as possible, and we expand this set by specifically designed *statistical (translation) language models (LM's)*.
2. We use statistics and heuristics to generate interesting questions about *important* entities and *salient but not obvious* relationships. Candidate answers for multiple-choice input are judiciously *ranked*, using corpus-collected statistics. An additional *diversification* step serves to avoid boring the user with near-duplicate choices.
3. The statistically derived relational phrases for candidate answers are complemented by phrases from *semantic resources*, specifically WordNet, ConceptNet, ReVerb, and PATTY [4, 5, 6, 7]. All this information is combined by a mixture model for the LM that generates, expands, and ranks relationships for a given context.

Figure 1 depicts the **system architecture** and main components of HIGGINS. It also shows a sample question from a game instance.

3. EXPERIMENTAL RESULTS

We have conducted experiments using Wikipedia movie plots and crowds "in the wild" (with CrowdFlower) and crowds of students in our lab. The results show that the full HIGGINS architecture can more than double recall and precision of derived ER triples (achieving recall and precision numbers over 70%) compared against a purely statistical IE engine or a purely semantics-based IE engine. Further, only a small number of game players are required in order to achieve this performance with good inter-annotator agreement – a fact that translates into reduced dollar-costs for human engagement.

4. REFERENCES

[1] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni: Open Information Extraction from the Web. IJCAI 2007.

[2] A. Doan, R. Ramakrishnan, A.Y. Halevy: Crowdsourcing systems on the World-Wide Web. Commun. ACM 54(4): 86-96, 2011.

[3] E. Law, L. von Ahn: Human Computation. Morgan & Claypool Publishers 2011.

[4] C. Fellbaum, G. Miller (Eds.): WordNet: An Electronic Lexical Database, MIT Press, 1998. `http://wordnet.princeton.edu`.

[5] R. Speer, C. Havasi: Representing General Relational Knowledge in ConceptNet 5, LREC 2012. `http://conceptnet5.media.mit.edu`.

[6] A. Fader. S. Soderland, O. Etzioni: Identifying Relations for Open Information Extraction, EMNLP 2011. `http://reverb.cs.washington.edu`

[7] N. Nakashole, G. Weikum, F. Suchanek: PATTY: A Taxonomy of Relational Patterns with Semantic Types. EMNLP-CoNLL 2012: 1135-1145. `http://www.mpi-inf.mpg.de/yago-naga/patty/`.