ABSTRACT

Automatic information extraction techniques for knowledge acquisition are known to produce noise, incomplete or incorrect facts from textual sources. Human computing offers a natural alternative to expand and complement the output of automated information extraction methods, thereby enabling us to build high-quality knowledge bases. However, relying solely on human inputs for extraction can be prohibitively expensive in practice. We demonstrate human computing games for knowledge acquisition that employ human computing to overcome the limitations in automated fact acquisition methods. We provide a combined approach that tightly integrates automated extraction techniques with human computing for effective gathering of facts. The methods we provide gather facts in the form of relationships between entities. The games we demonstrate are specifically designed to capture hard-to-extract relations between entities in narrative text— a task that automated systems find challenging.

Categories and Subject Descriptors

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

Keywords

Knowledge acquisition; Information Extraction; Human Computing

1. INTRODUCTION

Automated knowledge acquisition (KA) systems help capture the vast reserve of knowledge on the Web, enabling the construction of large-scale Knowledge Bases (KBs). The acquired knowledge in KBs such as www.freebase.com, www.dbpedia.org, www.yago-knowledge.org etc. contain millions of entities and relationships between them. Most of the existing KBs represent their knowledge in the form of subject-predicate-object (SPO) triples, based on the ubiquitous Resource Description Model (RDF) model. The stored SPO triples can be processed algorithmically to reason about the entities and their relations, thereby enabling the systems to answer complex questions posed to them.

A large number of automated KA systems exist and they employ a host of information extraction (IE) techniques ranging from pattern matching, linguistic analysis to statistical learning, to find entities and their relationships from textual sources. Recently proposed methods under the OpenIE paradigm use light-weight shallow processing techniques to efficiently gather facts from large corpora. The rationale is to trade-off precision for high recall through efficient processing. They primarily rely on the structural properties of the textual sentences to extract facts, most commonly by capturing a relation phrase occurring between two consecutive noun phrases in the sentence. Depending on the complexity of the sentences, such methods can produce noise, incomplete or incoherent facts. Sentences involving multiple entities, possible references to them, ambiguous wording and complex structure are common-place in descriptive text (news, blogs, biographies, narratives of movies, book stories etc.). Although current methods identify entities with reasonable precision, the quality of extraction suffers when uninformative or spurious relations are captured, or if the relations between the entities are identified incorrectly. For example an input sentence “After severing his limbs, Obi-Wan leaves Vader for dead” may yield the incorrect/uninformative SPO triple “Obi-Wan” leaves “Vader”. Input instances such as “Obi-Wan allows himself to be killed during a lightsaber rematch” can be even more challenging.

The paradigm of Human Computing (HC) [3, 4] has been advocated for a variety of tasks for which automatic methods fall short. We believe HC is a natural alternative to overcome the limitations of the automated methods for relational-centric KA. The reasons are two-fold. Firstly, humans are better able to comprehend the context, resolve anaphoras and relate the entities when presented with natural language text. Secondly, the background knowledge of humans gives the capability not only to assess existing factual knowledge but also add hard-to-extract facts to the KB. Despite these inherent advantages, no previous efforts have been attempted to employ HC to extract facts, particularly to find relationships between entities. Our work focuses on building HC games where players answer factoid questions on the relations between entities and in the process generate fact triples for the system.

One might think of acquiring facts solely from HC gameplay inputs, say by asking the player to type in the relation or choosing from a large pool of relations. Such tasks would
be tedious and are not enjoyable as games. To overcome this, the system needs to present the player with questions on important entities and only a handful of judiciously ranked candidate relations between them to choose from. Our approach makes concrete contributions towards ranking candidate relations between entities that will form basis for the games.

We have developed HIGGINS [5], a framework for human intelligence games for KA. In this paper, we demonstrate the potential of HIGGINS with the three-fold aim of a) its effectiveness in achieving high accuracy when extracting relational facts about entities in narrative text, b) exemplifying the inherent challenges and c) generating enjoyable and useful games for extraction from difficult text.

Handling narrative text brings its own set of challenges. Our demonstration will feature gathering relations between character roles from movie plots and book stories. The character role entities in narratives occur in various surface forms (e.g., “Jack Sparrow” as “Jack” or “Captain of the Black Pearl” or “Pirate Lord”) and the problem gets even harder for relational phrases (e.g., “begins to like” is similar to “has romantic feelings for” or “falls for”) bringing in higher degree of sparseness. We build extensive entity and relational phrase lexicons and use specifically designed statistical language models for phrase-relatedness and phrase-ranking to overcome these issues. As a by-product of playing the HIGGINS games, new facts are acquired and assessed. In this paper we demonstrate different games based on the HIGGINS strategy to extract facts from these textual narratives. The demonstration will prove the enjoyability of games while simultaneously producing high quality facts for the system.

2. HIGGINS OVERVIEW

The architecture of HIGGINS is designed to enable the use of automated IE methods to generate questions and possible answers that can be crowdsourced to obtain high quality facts.

IE Engine. The IE engine is designed to aggressively capture entity-centric and relation-centric phrases from the arbitrarily complex natural language text. It employs a suite of existing NLP techniques such as regex-based noun phrase detection, pronoun resolution and dependency parsing for this purpose. The IE engine gathers potential facts candidates, given a corpus of movie, book or news narratives. The entities and their aliases in the text are identified using regex matching on parts-of-speech. To resolve pronouns and anaphoras, the IE engine uses Stanford co-reference tool with additional gender-based heuristics. For every pair of interesting entities occurring within the same context (for e.g., a sentence), the shortest path in the dependency parse tree is extracted, which typically captures the relation between them [1]. Errors encountered in any of the above steps will be handled in the HC phase.

To overcome the sparseness issue for relational phrases, the IE engine builds an extensive mapping of closely-related phrases. This is achieved by tapping into available semantic resources like WordNet, ConceptNet [6] and phrase dictionaries from Reverb [7] and Patty [8]. To take care of the very special phrases that are seen in the cases of human relationships in books or movies, patterns between co-occurring entities in the ClueWeb09 corpus are mined and added to the dictionary of phrases.

HC Engine. The first stage generated fact hypotheses i.e., SPO triples along with their context C (the correspond-ing movie or book) which are fed into the HC engine. Game questions are constructed from these CSPO quads and presented to the game player to validate the fact hypotheses. Alternatively, these questions can be presented with one of the S, P, O, C as empty slots to be filled in. In this case, the HC engine generates possible alternatives for the slots and presents them as multiple choice. We focus particularly on the hard case where the relation P is to be determined. The alternatives for P are generated by ranking the candidate phrases in the dictionary built by the IE engine. We use statistical language models for this purpose. The question and the answer candidates are presented to the players and their responses are collected and aggregated by the system.

Games are perceived differently by the system and by the players. The system understands the notion of CSPO quads and how they can be built from an underlying dictionaries of entities and relational phrases. Players, on the other hand, view these items in a simpler and intuitive form, either as natural language text with gaps to be filled or as form fields. The players receive points upon providing a response to the question. They obtain bonuses only when their responses to the questions are in agreement with other players in the system, which makes the games fun to play.

3. KNOWLEDGE ACQUISITION GAMES

The fact hypotheses generated as CSPO quads by the HIGGINS IE engine form the basis for different game scenarios at the HC stage. Each HIGGINS game involves a sequence of interactions between the system and one or more players. The system designs questions by withholding one of the C, S, P or O values and presenting alternatives to the player. The player chooses the appropriate alternative, providing a completed quad to the system. Alternatively a fully bound CSPO quad can be presented to the player to be validated (in this paper we omit this case due to space reasons).

3.1 Game Scenarios

The following list describes different possible game scenarios by leaving out different variables, and their benefits to the system:

- **Variable S/O:** In this case the system leaves out S or O to be determined by the player. This game scenario is helpful for the anaphora resolution task. Candidates for the unknown anaphora are generated by HIGGINS and presented as multiple-choice to the player.

- **Variable P:** By leaving P unbounded, the player is tasked to determine the appropriate relation in which the two entities participate within the context. This game scenario can be applied by the system for the relation extraction task. Here’s an example question: “In the movie *Shawshank Redemption*, how are Andy Dufrense and Ellis Boyd Red related?”. HIGGINS generates possible relevant relation phrase candidates for P by processing the dependency path between S and O and ranking the phrases in its dictionary.

- **Variable C:** An interesting game scenario can be generated when the context C is unbound. A set of SPO triples for the same C (movie or book) are presented to the player. The player, in a series of interactions with the system or another player, tries to determine C. The key idea here is to consistently anonymize S and O when presented to the partner. In a two-player setting, one
player constructs SPO triples with anonymized S and O and a randomly-paired partner tries to guess C. The constructed SPO triples from the game are used by the system to form new facts.

Using movie and book narratives, we demonstrate two games, MovIE Wizard and Book Wizard, that are based on the scenarios where P or S/O are unbound. We also present a two-player game, MovIE Gurus, that demonstrates the scenario where C is unbound.

### 3.2 IMPLEMENTATION

We create our movie data set from the Wikipedia lists of movies. For book stories we use the articles from Sparknotes (http://www.sparknotes.com/sparknotes).

**Question generation.** From each article we extract the story plot, the character descriptions and the available metadata. The sentences from the article which describe the individual character roles are mined for noun phrases using regular expressions over parts-of-speech tags. This provides the system with character roles and their aliases (e.g.: “Don Vito”, “Vito Andolini”, “Don of the Corleone family”, “native Sicilian”, etc.). Next we process the plot text to detect existence of relations among the character roles in the following manner. The plot text is tokenized into sentences and the role-name aliases are resolved. For each sentence, the shortest paths between pair of roles in the dependency parse tree are obtained. Each non-empty shortest path containing a verb or common noun indicates a potential relation between the role pair.

**Candidate answer generation:** The relation candidates for two entities are generated by ranking the relational phrases in the Higgins dictionary, taking into account the shortest paths in the dependency parse trees generated in the above step. We achieve this by using a statistical translation model that combines both the relation-phrase relatedness scores calculated offline and the relatedness of the phrases to the dependency paths. The model assigns a relevance score to each relation in the dictionary, based on which they are ranked. To eliminate near duplicates relations, an additional diversification step is performed to remove all phrases that overlap significantly with a higher-ranked relation. We finally randomize the top-5 ranked relations and present them to the game player. In case of the anaphora resolution task, we use the top-ranked relation and rank the entities based on proximity to the anaphora.

### 4. DEMONSTRATION

The demonstration shows the three games: MovIE Wizard, Book Wizard and MovIE Gurus (http://www.mpi-inf.mpg.de/yago-naga/higgins).

1. **MovIE Wizard and Book Wizard games.** The MovIE Wizard game aims to establish relations between movie character roles by posing factoid questions to the player based on its storyline (Book Wizard is a functionally identical game on book stories). The player starts the game by picking a movie of his choice from the list of movies provided. The game server provides a question from the selected movie which consists of three parts. The first part states the question on two prominent characters in the movie. The second part contains hidden cues which are sentences about the character-pair in the plot. The third part contains five relation phrases presented in multiple-choice, an option for no relation between the pair, and an optional input-field. The task of the player is to choose an appropriate relation that holds between the two characters. If there exists a relation and none among the multiple-choice holds, the player types in a relational phrase. Otherwise, the option for no relation has to be chosen. The player may choose to use the hidden cues to arrive at an answer.

   **Scoring scheme.** The player score has two components: a) fixed points, called base points, for responding with an answer to the question and b) bonus points, awarded to the player based on the fraction of players who responded with the same answer. If the player chooses to utilize the hidden cue, the base points for the question are halved. Upon skipping the question no points are added to the score. The game server presents only unanswered questions to the player. The overall score of the player is determined by aggregating the base and bonus points. The position of the player among all players, ranked on overall score, is shown at the end of a game session. The objective of the player is to maximize the score and achieve top ranking among all the players.

2. **MovIE Gurus game.** MovIE Gurus is a two-player interactive fact-generation game with multiple rounds. In each round, the player is assigned to a partner, randomly picked from all players of the game. Upon pairing, the player is presented with a movie title and its storyline. The player is also provided with the list of characters in the movie and anonymous names corresponding to each of them. The player chooses any pair of characters from the list and types in a relational phrase. This phrase along with the anonymized character names is shown on the screen of the partner to help guess the movie title. The player repeats the process until the partner correctly guesses the movie title, in which case the round is won. The round is lost when either of the players chooses to give up or upon stipulated time out. Both the players earn points when the round is won. From a successful round, the system benefits by harnessing the fact triples that are constructed by the player-typed relational phrases and the respective pairs of movie characters.

### 5. REFERENCES


