A Virtual Player for "Who Wants to Be a Millionaire?" based on Question Answering

Piero Molino, Pierpaolo Basile, Ciro Santoro, Pasquale Lops,
Marco de Gemmis, and Giovanni Semeraro

Dept. of Computer Science, University of Bari Aldo Moro
Via E. Orabona, 4 - 70125 Bari, Italy
firstname.lastname@uniba.it,
c.santoro16@studenti.uniba.it

Abstract. This work presents a virtual player for the quiz game "Who Wants to Be a Millionaire?". The virtual player demands linguistic and common sense knowledge and adopts state-of-the-art Natural Language Processing and Question Answering technologies to answer the questions. Wikipedia articles and DBpedia triples are used as knowledge sources and the answers are ranked according to several lexical, syntactic and semantic criteria. Preliminary experiments carried out on the Italian version of the boardgame proves that the virtual player is able to challenge human players.

1 Introduction

Today artificial systems can compete with the best human players in a growing number of games, like chess, checkers, othello and go. These are called *closed world* games since they have a finite number of possible choices and can be solved in a formal way, even though they are hard to play due to the exponential dimension of the search space.

Recently the interest of the researchers shifted to less structured games, like sports, videogames, crosswords, where the states of the game and the actions that the player can take cannot be easily enumerated, making the search through the space of possible solutions impossible.

In particular, language games require a wide linguistic and common sense knowledge and the understanding of the meaning of natural language words. "Who Wants to Be a Millionaire" (WWBM) is a perfect example of a language game. It is a quiz where the player answers questions posed in natural language by selecting the correct answer out of four possible answers. For example, a possible question is: *"Who directed Blade Runner?"* and the four possible answers are *A) Harrison Ford B) Ridley Scott C) Philip Dick D) James Cameron.*

Even though in this game the number of possible answers is limited, the choice is dependent on the player's knowledge, her understanding of the questions and her ability to balance the confidence in the answers and the risk taken in answering.

M. Baldoni et al. (Eds.): AI*IA 2013, LNAI 8249, pp. 205–216, 2013.

This work describes a virtual player for the WWBM game, which leverages Question Answering (QA) techniques and both Wikipedia and DBpedia datasources to incorporate common sense human knowledge for playing the game.

The WWBM game allows three "lifelines" which provide some form of assistance to the player. In the original game the lifelines are: *50/50* (which randomly removes two wrong answers), *Ask the Audience* (where the audience answers the question and the percentage of people that choice each possible answer is provided to the player), *Phone-a-Friend* (where the player can phone a friend to ask the question having a specific time constraint - 30 or 60 seconds). If the answer given by the player is correct, she earns a certain amount of money and continues to play with questions of increasing difficulty, until she reaches the last question - the 15th - or she decides to stop the game by taking the money earned. If the player gives the wrong answer, she loses everything if she is answering one of the first five questions; she earns 3,000 Euros if she is answering questions from six to ten, 20,000 Euros for questions from eleven to fifteen. The amount of money earned and the lifelines vary from country to country.

In this first attempt to solve the game, we do not manage "lifelines" or the possibility to retire from the game. Our main goal is to evaluate the ability of the virtual player to correctly answer the questions of the game.

The rest of the paper is organized as follows: in section 2 we describe how our system is built. The evaluation of the system is described in section 3, while in section 4 related work are reported. Conclusions and future work close the paper.

2 The Architecture of the Virtual Player

We built a virtual player for WWBM with a layered architecture consisting of three main modules:

1. *Game Manager*: this module allows to manage the game and its specific rules.
2. *Question Answering*: this module queries Wikipedia and DBpedia datasources and retrieves the most relevant passages of text useful to select the right answer. A detailed description is provided in Section 2.1.
3. *Answer Selection*: this module adopts different criteria to assign a confidence value to each of the four possible answers for a specific question. A detailed description is provided in Section 2.2.

For each question of the game, the *Game Manager* delegates to the *Question Answering* module the selection of the most relevant passages of text, which might contain the correct answer. The *Question Answering* module returns the passages with the highest scores, together with the title of the articles they are extracted from. Those results are processed by the the *Answer Selection* module which computes the confidence of each possible answer using different heuristics. Finally, the *Game Manager* selects and provides the best answer.

2.1 Question Answering Framework

We exploit QuestionCube [9, 10], a multilingual QA framework created using NLP and IR techniques, in order to obtain relevant passages of text. The overall architecture of the framework is shown in Figure 1.

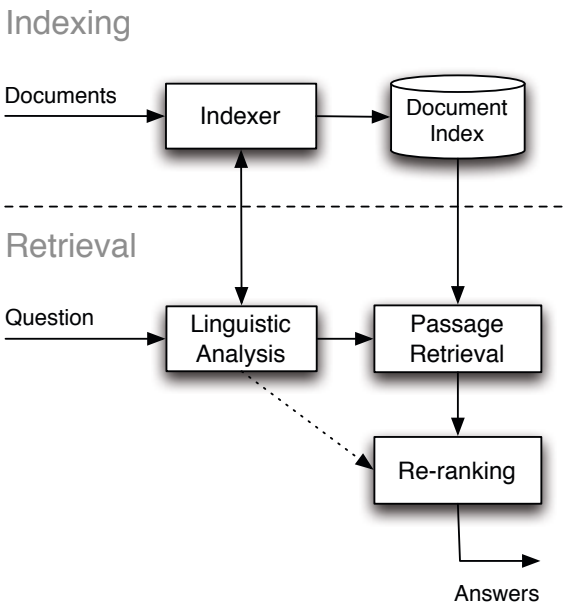

Fig. 1. QuestionCube architecture

The linguistic analysis is carried out by a full-featured NLP pipeline. It includes stopword removal, stemming, part-of-speech tagging, lemmatization, shallow parsing and word sense disambiguation for both English and Italian. Each NLP component adopts state-of-the-art algorithms for the specific task.

The passage retrieval step is carried out by Lucene 4[1], a standard off-the-shelf retrieval framework that allows TF-IDF, Language Modeling and BM25 [13] weighting. Inside the QA process this component is useful to filter passages not related to the question.

The question re-ranking component is designed as a pipeline of different scoring criteria that exploits the different linguistic features obtained from the NLP process (terms, stems, lemmas, chunks, senses). Those criteria include:

– the overlap of specific linguistic features between the question and the answer (or the title of the original document containing the answer)

[1] Available at `http://lucene.apache.org/`

- a density and frequency measure of the question linguistic features inside the answer (or the title of the original document containing the answer). The exact overlapping subsequence, the minimal overlapping span [11] and the number of linguistic features of the question terms in a single sentence of the answer can be considered
- similarity measures based on TF-IDF, Language Modelling and BM25 weighting schemes
- measures based on static properties of the answer and the documents it comes from, such as the length, the number of in-links and out-links (if available), the centrality in the network of documents measured as degree, PageRank or HITS scores
- measures based on distributional semantic models, such as Latent Semantic Analysis [3], Random Indexing [7] and Latent Semantic Analysis over Random Indexing [14]. Further details are available in [10].

We derive a global re-ranking function combining the different scores using the CombSum function [16]. CombSum can be replaced by Machine Learning to Rank algorithms, such as Logistic Regression (reported to be very effective in IBM's DeepQA / Watson [1]), RankNet [2], RankBoost [6] and LambdaMART [17] if enough training data is available. More details on the framework and a description of the main scorers are reported in [9, 10].

In order to build the virtual player for the WWBM game, Wikipedia and DBpedia are used as datasources. To this purpose, we used Wikiedi[2], a specific application built using the QuestionCube framework, which exploits the unstructured data coming from the Wikipedia articles to provide answers to the questions of the WWBM game. Wikiedi also integrates DBpedia as a source for the structured data that can be found in the infoboxes and templates of Wikipedia articles. Search on DBpedia relies on a custom triple searcher built to maximize recall of correct answers. This choice makes the application robust enough to manage both factoid and non-factoid questions. Factoid questions are those whose answers are short excerpts of text, usually named entities, dates or quantities. Non-factoid questions are those whose answers have the form of a passage of text.

Answer re-ranking adopts most of the scorers available in the framework, including those based on the distributional semantics. This enables an approximate matching between question and answers that reduces the impact of question ambiguity.

Table 1 reports the list of five passages returned by Wikiedi for the question "Who directed Blade Runner?". Each passage contains the title of the article it is contained in, and the score computed by Wikiedi.

2.2 Answer Selection

In order to assign a confidence score to each of the four possible answers for a specific question of the WWBM game, we adopt different criteria based on the

[2] Available at `www.wikiedi.it`

Table 1. The list of five passages returned by Wikiedi for the question "Who directed Blade Runner?"

Article Title	Passage Text	Score
Ridley Scott	Sir Ridley Scott (born 30 November 1937) is an English film director and producer. Following his commercial breakthrough with Alien (1979), his best-known works are the sci-fi classic Blade Runner (1982) and the best picture Oscar-winner Gladiator (2000).	5.32
Blade Runner	Blade Runner is a 1982 American dystopian science fiction action film directed by Ridley Scott and starring Harrison Ford, Rutger Hauer, and Sean Young. The screenplay, written by Hampton Fancher and David Peoples, is loosely based on the novel Do Androids Dream of Electric Sheep? by Philip K. Dick.	5.1
Blade Runner	Director Ridley Scott and the film's producers "spent months" meeting and discussing the role with Dustin Hoffman, who eventually departed over differences in vision. Harrison Ford was ultimately chosen for several reasons.	5
Blade Runner	The screenplay by Hampton Fancher was optioned in 1977. Producer Michael Deeley became interested in Fancher's draft and convinced director Ridley Scott to film it.	4.9
Blade Runner	Interest in adapting Philip K. Dick's novel Do Androids Dream of Electric Sheep? developed shortly after its 1968 publication. Director Martin Scorsese was interested in filming the novel, but never optioned it.	1.2

analysis of the passages returned by the QA module. Each individual criterion returns a confidence in the answers, obtained by dividing the score of each answer by the sum of the scores of the four possible answers. Follows a description of each criterion, explained by taking into account the example in Table 1:

- ***Title Levenshtein*:** this criterion computes the Levenshtein distance (metric for measuring the difference between two sequences of characters) between the candidate answer and the title of the answer returned by Wikiedi. As the Levenshtein distance is a distance measure rather than a similarity measure, we compute $\frac{max(len(a),len(t))-lev(a,t)}{max(len(a),len(t))}$, where $len(a)$ is the length of the candidate answer, $len(t)$ is the length of the title of the Wikipedia page containing the answer provided by Wikiedi, and $lev(a,t)$ is the Levenshtein distance between the candidate answer and the title of the page. This allows to have scores in the $[0,1]$ interval. In our example, the answer B) Ridley Scott occurs in the title of the page containing the passage, so it gets the maximum score of 1 since all the characters are the same, while all the other answers get $\frac{12-11}{12} = 0.083$. The final confidence is 0.8 for answer B) and 0.066 for the others.

- ***LCS***: this criterion computes the Longest Common Subsequence between the candidate answer and the passages of text returned by Wikiedi. In our case, answer A) Harrison Ford gets a score equal to 13, answer B) Ridley Scott gets a score equal to 12, answer C) Philip Dick gets a score equal to 11, and answer D) James Cameron gets a score equal to 0 since it does not occur in any of the passages returned by Wikiedi. The final confidence is 0.36 for the candidate answer A), 0.33 for the candidate answer B), 0.31 for the candidate answer C), and 0 for the candidate answer D).
- ***Overlap***: this criterion computes the Jaccard index between the set of terms in the candidate answer and the set of terms in the passages of text returned by Wikiedi. In our example, answers A), B) and C) get a score of 0.04651, while the anser D) gets 0. The final confidence is 0.33 for the candidate answer A), B), C), and 0 for the candidate answer D).
- ***Exact Substring***: this criterion computes the length in characters of the longest common substring between the candidate answer and the answers from Wikiedi. We normalize the score using the length of the candidate answer. In our example, answer A) Harrison Ford gets a score of $\frac{13}{13} = 1$, answer B) Ridley Scott gets a score of $\frac{12}{12} = 1$, answer C) Philip Dick gets a score of $\frac{6}{11} = 0.54$, and answer D) James Cameron gets score 0. The final confidence is 0.395 for the candidate answer A) and B), 0.21 for the candidate answer C), and 0 for the candidate answer D).
- ***Density***: this criterion computes the density of the terms in the candidate answer inside the passages of text returned by Wikiedi, using the minimal overlapping span method described in [11]. In our example, considering only the first passage returned by Wikiedi, answers A) and B) get a score of 1, answer C) gets a score of 0.66 (as the passage reports the full name Philip K. Dick, adding an extra token between the two tokens of the answer), while answer D) gets score 0. The final confidence is 0.375 for the candidate answer A) and B), 0.25 for the candidate answer C) and 0 for the candidate answer C).

Each criterion is parametrized using four parameters: 1) the number of passages returned by Wikiedi; 2) the use of the score of the passages as weights for computing the final value; 3) the level of linguistic analysis to adopt, and 4) the use of the question expansion. Question expansion means that the system asks four different questions obtained by the concatenation of the original question and the four possible candidate answers. For example, the virtual player queries Wikiedi using the following questions: "Who directed Blade Runner? Harrison Ford", "Who directed Blade Runner? Ridley Scott","Who directed Blade Runner? Philip Dick" and "Who directed Blade Runner? James Cameron".

The outcomes of the previous criteria have been also combined using *Majority Vote* and *CombSum*, in order to obtain the final confidence score for each candidate answer. When Majority Vote is used, the confidence of each candidate answer is computed as the ratio between the number of different criteria voting for that candidate answer, and the total number of criteria. When CombSum is used, the scores of the different criteria are standardized and then summed.

3 Experiment

The goal of the evaluation is twofold:

1. to assess the performance of the virtual player
2. to compare the accuracy of the virtual player with that of human players.

The first experiment aims at evaluating the effectiveness and robustness of the virtual player for different kinds of questions. The second experiment aims at comparing the performance of the system and of the human players by varying the difficulty of the questions.

The experiments have been carried out using a dataset of 262 questions obtained from the official WWBM Italian boardgame. To the best of our knowledge this is the first attempt to measure the accuracy of a virtual player for the Italian version of the game. This means that we do not have results representing a baseline for our system.

The metric adopted for the evaluation is *accuracy*, computed as the ratio between the number of correct answers (n_c) and the total number of answers (n), and *c@1*, adopted in the *2010 CLEF QA Competition* [12], computed as $c@1 = \frac{1}{N}\left(n_c + n_n \frac{n_c}{N}\right)$, where N is the number of the questions, n_c is the number of the correct answers provided by the system and n_n is the number of unanswered questions. This measure allows the system to leave the questions unanswered, but the gain in doing so depends on the accuracy, so the metric favors those systems that do not answer the questions they would have answered wrong.

3.1 Results of Experiment 1

Results of the first experiment are reported in Table 2. The first column describes the answer selection criterion and the adopted parameters. Both individual and composite criteria are reported.

It is worth to note that the composite criteria outperform the individual ones in terms of accuracy and percentage of unanswered questions.

The best combination is reported in the first row of the table and exploits CombSum of the methods using the following parameters: (1) LCS over 20 passages using lemmas and scores with stopwords removed; (2) Substring over 20 passages adopting keywords and passages score; (3) Overlap over 20 passages, using lemmas and passages score; (4) Density over the first passage, using lemmas and removing stopwords; (5) LCS criterion over 20 passages, using keywords and passages score, with stopwords removed and expanding the question with the four possible answers.

We have carefully analyzed the questions for which all the criteria failed to provide the correct answer. We found out that some of these questions would require a different kind of knowledge sources to be answered. For example, some of them would require mathematical knowledge, and some others would require English language knowledge. By removing this small subset of questions from the dataset, *accuracy* and *c@1* of the top-ranked criterion increases to 78.43% and 79.66%, respectively.

Table 2. Evaluation results. Criteria: **MV** Majority Vote, **ES** Exact Substring, **LCS** Longest Common Subsequence, **TL** Title Levenshtein. Parameters: the first number is the number of passages, **K** keyword, **L** lemma, **S** uses the score of the passage, **SW** applies stopword removal, **QE** expands the question with the four possible answers.

Criterion	Accuracy	Unansw.	c@1
CombSum: LCS(20,L,S,SW), ES(20,K,S), Overlap(20,L,S), Density(1,L,SW), LCS(20,K,S,SW,QE)	**76.34%**	1.91%	**77.79%**
CombSum: TL(1,K,S,QE), ES(25,K,S), Overlap(25,L,Scored), LCS(25,K,S,SW,QE), LCS(25,L,S,SW,QE)	71.37%	0.38%	71.64%
MV: TL(1,K,S,QE), ES(25,K,S), Overlap(25,L,S), LCS(25,K,S,SW,QE), LCS(25,L,S,SW,QE)	71.76%	0.00%	71.75%
CombSum: TL(1,SW), LCS(25,L,S,SW,QE), LCS(25,K,QE), LCS(25,K,S,SW,QE), Overlap(25,L,S), ES(25,K,S), Overlap(2,K), Overlap(5,K), Overlap(5,K,S), ES(1,K,S), ES(20,L,S)	71.76%	1.14%	72.57%
LCS(20,L,S,SW)	64.89%	16.79%	75.78%
ES(20,L,S)	55.73%	23.66%	68.91%
Overlap(20,L,S)	58.40%	29.01%	75.33%
LCS(20,L,S,SW,QE)	72.90%	1.91%	74.29%
ES(25,K,S)	56.87%	22.14%	69.46%
Overlap(25,K,S)	59.92%	27.48%	76.38%
LCS(25,K,S,SW,QE)	41.60%	20.61%	50.17%
LCS(25,K,L,SW,QE)	71.76%	1.91%	73.12%
Overlap(5,K)	45.80%	44.28%	66.08%
ES(1,K,S)	27.48%	64.50%	45.20%
TL(1,K,SW)	3.44%	96.18%	6.73%
TL(1,K,S,QE)	27.48%	0.00%	27.48%
LCS(25,K,QE)	41.22%	20.61%	49.71%
Overlap(2,K)	36.26%	59.16%	57.71%
Overlap(5,L,S)	45.80%	44.28%	66.08%

The overall outcome of the experiment is consistent with the results presented in [8] (in term of accuracy), even though it is not possible a direct comparison since the experimental settings, the dataset and the language (English) of the questions are different.

3.2 Results of Experiment 2

We involved 60 human players selected using the availability sampling strategy: 85% of the players are graduated, and 10% got a PhD. Each user played the game at least 7 times, and we ensured that each question proposed to the human player was novel so that she never received the same question during different games.

Figure 2 reports the results of the virtual player in terms of *accuracy*, compared with the average results of the human players, for each level of the game.

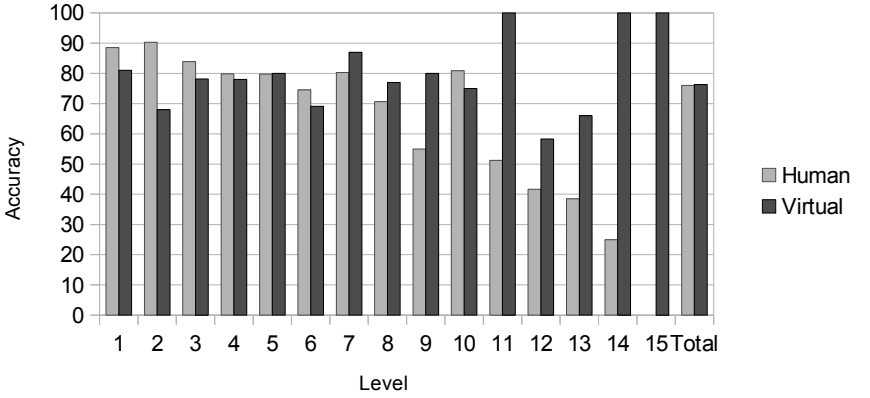

Fig. 2. Accuracy per level

The average accuracy of the virtual player is 76.33%, comparable with the average accuracy reported by humans that is 76.02%.

We performed a further analysis which takes into account the accuracy for each of the fifteen levels of the game separately. Usually lower levels correspond to easier questions, while higher levels correspond to more complex questions. This analysis unveils that human players have higher accuracy on low levels of the game and lower accuracy on higher levels, while the *virtual player* behaves in an opposite way. This means that the virtual player is able to provide the correct answer for the most difficult questions, but it also requires some abilities for providing answers to very simple questions.

This observation is in line from what was previously found in literature [8].

4 Related Work

Teaching a computer how to play games and competing against human players has always been a challenging task since the early days of computing. Games are a good test-bed for Artificial Intelligence as they allow to test the limits of the artificial agents.

Language games are a particular type of open-world games where there is the need to understand the meaning of words and a big amount of knowledge and reasoning are essential to compete at human level. Some examples of language games can be Trivial Pursuit, Punning Riddles, Humoristic Acronyms.

An interesting language game is crosswords, as it relies on linguistic knowledge and requires constraints satisfaction over the possible answers. A system for solving this game is Web-Crow [4], an artificial agent that exploits the Web as a source of information. The problem is subdivided in finding the best words that answer the definitions and in placing them inside the grid. In order to find the best words, the system queries Google with reformulations of the original

definitions and analyzes the best 200 result pages. Web-Crow achieves a 68.8% of correct words and a 79.9% of correct letters, showing the potential of the Web as a resource for language games.

Another interesting game is Guillotine, a game broadcasted by the Italian National TV company. It involves a single player, who is given a set of five words (clues), each linked in some way to a specific word that represents the unique solution of the game. Words are unrelated to each other, but each of them is strongly related to the word representing the solution. In [15], the authors propose a system for playing guillotine, called OTTHO, that implements a *knowledge infusion* process which analyzes unstructured information stored in open knowledge sources on the Web to create a memory of linguistic competencies and world facts that can be effectively exploited by the system for a deeper understanding of the information it deals with.

A virtual player for "Who Wants to Be a Millionaire?" is described in [8]. The authors exploit the great amount of knowledge in the Web in order to answer the questions. The query formulation module adopts NLP techniques in order to create different queries. The queries are then sent to Google and the number of obtained results is used to rank the answers, exploiting the redundancy of the information sources. This system reaches an accuracy of 72% showing how useful unstructured data can be for this kind of task, but still fails with questions that require common sense reasoning and access to structured information. The main difference with our work is the adoption of Wikipedia and DBpedia as a selected and reliable source of information rather than the whole Web. Moreover the adoption of a QA framework instead of a search engine allows us to get a more reliable passage filtering.

In February 2011 the IBM Watson supercomputer, adopting technology from the DeepQA [5] project, has beaten the champions of the Jeopardy! TV quiz. In Jeopardy! the classic quiz is reversed, the player is given an ambiguous or ironical piece of the answer and has to find the question for it. To accomplish it, Watson applied several different NLP, IR and ML techniques focusing on the Open-domain QA, answering questions without domain constraint. Watson analyzed 200 million content elements, both structured and unstructured, including the full text of Wikipedia. Watson shows how competitive are actual NLP and ML technologies in managing big amounts of data and exploiting it to compete with humans in a field where they have been considered unbeatable for a long time.

5 Conclusions and Future Work

In this work we propose a virtual player for the game "Who Wants to be Millionaire?". In order to answer the questions from the quiz, our system leverages Natural Language processing and Question Answering techniques and exploits Wikipedia and DBpedia as datasources. A preliminary experiment on the Italian version of the boardgame show that the system is able to provide a correct answer for 76% of questions, and its performance in terms of accuracy is comparable with that of human players.

As future work we plan to add a decision making module able to 1) evaluate whether to provide the answer for a question or to retire from the game; 2) manage the "lifelines" provided by the rules of the game. Furthermore, we shall investigate on the improvement of the answer selection strategy by exploiting learning to rank approaches.

References

1. Agarwal, A., Raghavan, H., Subbian, K., Melville, P., Lawrence, R.D., Gondek, D., Fan, J.: Learning to rank for robust question answering. In: CIKM, pp. 833–842 (2012)
2. Burges, C.J.C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.N.: Learning to rank using gradient descent. In: ICML, pp. 89–96 (2005)
3. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)
4. Ernandes, M., Angelini, G., Gori, M.: Webcrow: A web-based system for crossword solving. In: Veloso, M.M., Kambhampati, S. (eds.) AAAI, pp. 1412–1417. AAAI Press/The MIT Press (2005)
5. Ferrucci, D.A., Brown, E.W., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J.M., Schlaefer, N., Welty, C.A.: Building Watson: An Overview of the DeepQA Project. AI Magazine 31(3), 59–79 (2010)
6. Freund, Y., Iyer, R.D., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research 4, 933–969 (2003)
7. Kanerva, P.: Sparse Distributed Memory. MIT Press (1988)
8. Lam, S.K., Pennock, D.M., Cosley, D., Lawrence, S.: 1 billion pages = 1 million dollars? mining the web to play "who wants to be a millionaire?". In: Meek, C., Kjærulff, U. (eds.) UAI, pp. 337–345. Morgan Kaufmann (2003)
9. Molino, P., Basile, P.: Questioncube: a framework for question answering. In: Amati, G., Carpineto, C., Semeraro, G. (eds.) Proceedings of the 3rd Italian Information Retrieval (IIR) Workshop, Bari, Italy, January 26-27. CEUR Workshop Proceedings, vol. 835, pp. 167–178. CEUR-WS.org (2012)
10. Molino, P., Basile, P., Caputo, A., Lops, P., Semeraro, G.: Exploiting distributional semantic models in question answering. In: Sixth IEEE International Conference on Semantic Computing, ICSC 2012, Palermo, Italy, September 19-21, pp. 146–153. IEEE Computer Society (2012)
11. Monz, C.: Minimal span weighting retrieval for question answering. In: Gaizauskas, R., Greenwood, M., Hepple, M. (eds.) Proceedings of the SIGIR Workshop on Information Retrieval for Question Answering, pp. 23–30 (2004)
12. Penas, A., Forner, P., Rodrigo, A., Sutcliffe, R.F.E., Forascu, C., Mota, C.: Overview of ResPubliQA 2010: Question Answering Evaluation over European Legislation. In: Braschler, M., Harman, D., Pianta, E. (eds.) Working Notes of ResPubliQA 2010 Lab at CLEF 2010 (2010)
13. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. 3, 333–389 (2009)
14. Sellberg, L., Jönsson, A.: Using random indexing to improve singular value decomposition for latent semantic analysis. In: LREC (2008)

15. Semeraro, G., de Gemmis, M., Lops, P., Basile, P.: An artificial player for a language game. IEEE Intelligent Systems 27(5), 36–43 (2012)
16. Shaw, J.A., Fox, E.A.: Combination of multiple searches. In: The Second Text REtrieval Conference (TREC-2), pp. 243–252 (1994)
17. Wu, Q., Burges, C.J.C., Svore, K.M., Gao, J.: Adapting boosting for information retrieval measures. Inf. Retr. 13(3), 254–270 (2010)