Overview of the EVALITA 2018 Solving language games (NLP4FUN) Task

Pierpaolo Basile and **Marco de Gemmis** and **Lucia Siciliani** and **Giovanni Semeraro**
Department of Computer Science
Via E. Orabona, 4 - 70125 Bari
University of Bari Aldo Moro
{firstname.lastname}@uniba.it

Abstract

English. This paper describes the first edition of the "Solving language games" (NLP4FUN) task at the EVALITA 2018 campaign. The task consists in designing an artificial player for *"The Guillotine" (La Ghigliottina, in Italian)*, a challenging language game which demands knowledge covering a broad range of topics. The game consists in finding a word which is semantically correlated with a set of 5 words called clues. Artificial players for that game can take advantage from the availability of open repositories on the web, such as Wikipedia, that provide the system with the cultural and linguistic background needed to find the solution.

Italiano. *Questo lavoro descrive la prima edizione del task "Solving language games" (NLP4FUN) task, proposto durante la campagna di valutazione EVALITA 2018. Il task consiste nella realizzazione di un giocatore artificiale per "La Gigliottina", un gioco linguistico molto sfidante, la cui soluzione richiede conoscenze in svariati campi. Il gioco consiste nel trovare una parola il cui significato è correlato a quello di un insieme di 5 parole, chiamate indizi. Un giocatore artificiale per questo task potrebbe sfruttare diverse sorgenti di conoscenza disponibili online, come Wikipedia, che forniscano al sistema le conoscenze linguistiche e culturali necessarie per arrivare alla soluzione.*

1 Motivation

Language games draw their challenge and excitement from the richness and ambiguity of natural language, and therefore have attracted the attention of researchers in the fields of Artificial Intelligence and Natural Language Processing. For instance, IBM Watson is a system which successfully challenged human champions of Jeopardy!, a game in which contestants are presented with clues in the form of answers, and must phrase their responses in the form of a question (Ferrucci et al., 2010; Molino et al., 2015). Another popular language game is solving crossword puzzles. The first experience reported in the literature is Proverb (Littman et al., 2002), that exploits large libraries of clues and solutions to past crossword puzzles. WebCrow is the first solver for Italian crosswords (Ernandes et al., 2008).

The proposed task consists in designing a solver for *"The Guillotine" (La Ghigliottina, in Italian)* game. It is inspired by the final game of an Italian TV show called "L'eredità". The game, broadcast by Italian National TV, involves a single player, who is given a set of five words - the 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 has a hidden association with the solution. Once the clues are given, the player has one minute to find the solution. For example, given the five clues: *sin*, *Newton*, *doctor*, *New York*, *bad*, the solution is *apple*, because: the apple is the symbol of original sin in Christian theology; Newton discovered the gravity by means of an apple; *"an apple a day keeps the doctor away"* is a famous proverb; New York city is also called *"the big apple"*; and *"one bad apple can spoil the whole bunch"* is a popular phrase which figuratively means that the person doing wrong can have a negative influence on those around him. "La Ghigliottina" is a challenging language game which demands knowledge covering a broad range of topics. Artificial players for that game can take advantage from the availability of open repositories on the web, such

as Wikipedia, that provide the system with the cultural and linguistic background needed to understand clues (Basile et al., 2016; Semeraro et al., 2009; Semeraro et al., 2012).

The task is part of EVALITA 2018, the periodic evaluation campaign of Natural Language Processing (NLP) and speech tools for the Italian language (Caselli et al., 2018).

The paper is organized as follows: Section 2 reports details about the task, the dataset and the evaluation protocol, while Section 3 describes the systems participating in the task, and Section 4 shows results.

2 Task Description: Dataset, Evaluation Protocol and Measures

An instance of the game consists of a set of 5 clue words and 1 word given as the official solution for that instance. We provided:

- a training set for the system development, containing 315 instances of the game;
- a test set for the evaluation, containing 105 instances of the game.

In order to measure the performance of the participants on games having different levels of difficulty, we provided instances taken both from the TV game and from the official board game. In the training set, 204 instances (64.8%) came from the TV game, 111 (35.2%) from the board game. In the test set, 66 instances (62.9%) were collected from the TV game, 39 (37.1%) from the board game. In order to discourage participants from cheating (e.g. finding the solution manually), in the test set we included 300 fake games automatically created by us. Obviously, fake games were not taken into account in the evaluation.

Any knowledge resource can be used to build an artificial player, except further instances of the game. For each instance of the game, a ranked list of maximum 100 tentative solutions must be provided.

2.1 Data Format

Both development and test set were provided in XML format:

```
<games>
  <game>
        <id>3fc953bd...</id>
        <clue>uomo</clue>
        <clue>cane</clue>
        <clue>musica</clue>
        <clue>casa</clue>
        <clue>pietra</clue>
        <solution>chiesa</solution>
        <type>TV</type>
  </game>
  ...
</games>
```

The XML file consists of a root element *games* which contains several *game* elements. Each game has five *clue* elements and one *solution*. Moreover, the element *type* specifies the type of the game: *TV* or *boardgame*.

The ranked list of solutions must be provided in a single plain text file, according to the following format:

```
id solution score rank time
```

Values were separated by a whitespace character; time taken by the system to compute the list was also reported in milliseconds. An example of a ranked list of solutions is reported below:

```
3fc953bd-... porta 0.978 1 3459
3fc953bd-... chiesa 0.932 2 3251
3fc953bd-... santo 0.897 3 4321
...
3fc953bd-... carta 0.321 100 2343
...
```

2.2 Evaluation

As evaluation measure, we adopt a weighted version of Mean Reciprocal Rank (MRR). Since time is a critical factor in this game, the Reciprocal Rank is weighted by a function which lowers the score based on the time taken by the computation. In fact, in the TV game, the player has only one minute to provide the solution. Taking into account these factors, the evaluation measure was:

$$\frac{1}{|G|}\sum_{g\in G}\frac{1}{r_g}max(\frac{1}{t_g},\frac{1}{10}) \quad (1)$$

where G is the set of games and r_g is the rank of the solution, while t_g denotes the minutes taken by the system to give the tentative solutions. Systems that took more than 10 minutes are equally penalized.

The evaluation was performed only on the 105 test games, for which we knew the correct solution (results provided for fake games were excluded).

We provided a separate ranking for TV and boardgame, but the final ranking was computed on the the whole test set.

3 Systems

Twelve teams registered in the task, but only two of them actually submitted the results for the evaluation. A short description of each system follows:

UNIOR4FUN - The system described in (Sangati et al., 2018) is based on the idea that clue words and corresponding solution are often part of a multiword expression. Therefore, the system exploits six linguistic patterns[1] that identify valid multiword expressions connecting clue and solution pairs. The core of the proposed solution is a set of freely available corpora and lexical resources built by the authors, which are used to find potential solutions by computing mutual information.

System by Luca Squadrone - In (Squadrone, 2018), the author proposed an algorithm based on two steps. In the first one, for each clue of a game, a list of relevant keywords is retrieved from linguistic corpora, so that each clue is associated with keywords representing the concepts having a relation with that clue. Then, words at the intersection of the retrieved sets are considered as candidate solutions. In the second step, another knowledge source made of proverbs, book and movie titles, word definitions, is exploited to count co-occurrences of clues and candidate solutions.

4 Results

Table 1: System results.

System	MRR	MRR (std)	Solved
UNIOR4NLP	0.6428	0.6428	81.90%
Squadrone	0.0134	0.0350	25.71%

Results of the evaluation in terms of MRR are reported in Table 1. The best performance is obtained by the *UNIOR4NLP* team. They reached a remarkable performance: MRR is very high, thus showing that the system is able to place the solution in the first positions of the ranking. We report, also, the standard MRR ($MRR(std)$) computed without taking into account the time. We notice that for UNIOR4NLP the value is equal to MRR: the system is able to provide the solution always in the first minute, while the Squadrone system takes more time for solving games.

Table 2 reports the results by game type (66 instances from the TV game and 39 instances from the boardgame). UNIOR4NLP shows similar results for both the game types, while the system proposed by Squadrone performs better on board games.

One possible explanation for this difference is that board games are meant just for fun; they are designed for the average player, whereas those taken from the TV game are more difficult to solve because they are intended to challenge the contestants of the show who try to win a money prize. Therefore, TV games generally have very specific clues and require more extensive knowledge about world facts and particular topics to find the solution than the average player has. As a consequence, the UNIOR4NLP solution based on specific multiword expressions extracted from several knowledge sources shows a more balanced performance than the other system.

However, despite the UNIOR4NLP system obtained remarkable results, very difficult games, requiring some kind of inference, are missed. For example, for the following clues: *uno, notte, la trippa, auto, palazzo*[2], the solution is *portiere* (porter). In order to solve that game, two difficult inferences are needed:

- *uno* is the number generally assigned to the role of the goolkeeper (portiere) in football teams;
- *"La Trippa"* is the surname of "Antonio La Trippa", a character of the Italian movie "Gli onorevoli", whose job is the porter (portiere) of a building.

We hope that in a further edition of this task participants will take into account these kind of games in which the simple co-occurrence of words it is not enough for solving the game. This is the most

[1] We must underline that patterns are extracted from a set of 100 games collected by authors. This is in contrast with the task guidelines; however, the games are not used for training the system.

[2] In English: one, night, "la trippa" (it was intended as a surname in this case), car, building

Table 2: System results for TV and boardgame

System	MRR (TV)	Solved (TV)	MRR (board)	Solved (board)
UNIOR4NLP	0.6528	86.36%	0.6001	71.79%
Squadrone	0.0068	25.75%	0.0245	25.64%

challenging aspect of this game. In order to compare system performance by taking into account the different levels of difficulty of the games, we plan to annotate guillottines with this information provided by human players. A deeper analysis of the results obtained by each system is provided in the corresponding technical reports (Sangati et al., 2018; Squadrone, 2018).

Finally, by looking at the statistics about the participation (12 registered teams, but only 2 of them submitted the results), we conclude that the task is attractive but perhaps it is too hard to solve. For further task editions, we plan to support the participants by providing pre-processed textual resources useful for solving the task.

5 Conclusions

Language games draw their challenge and excitement from the richness and ambiguity of natural language. This type of games are inconsistent with the closed world assumption: no fixed sets of rules are sufficient to define the game play. The proposed task consisted in building an artificial player for a challenging language game which requires from the player a strong linguistic and cultural background. The systems participating in the task were designed according to this idea: solving the game strongly depends on the background knowledge of the system. On the other hand, the results demonstrated that filling in the system with a solid background knowledge is not enough to find the solution, but strong NLP algorithms are required to discover hidden correlation among words. In fact, only the system based on specific linguistic patterns and multiword expressions was able to achieve high performance. Moreover, some games required a non-trivial inference step. For this kind of games, systems must be equipped with deeper reasoning capabilities. We hope that in further editions of the task, participants will propose solutions that deal with this issue.

References

Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. 2016. Solving a complex language game by using knowledge-based word associations discovery. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(1):13–26.

Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso. 2018. Evalita 2018: Overview of the 6th evaluation campaign of natural language processing and speech tools for italian. In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018)*, Turin, Italy. CEUR.org.

Marco Ernandes, Giovanni Angelini, and Marco Gori. 2008. A web-based agent challenges human experts on crosswords. *AI Magazine*, 29(1):77.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.

Michael L Littman, Greg A Keim, and Noam Shazeer. 2002. A probabilistic approach to solving crossword puzzles. *Artificial Intelligence*, 134(1-2):23–55.

Piero Molino, Pasquale Lops, Giovanni Semeraro, Marco de Gemmis, and Pierpaolo Basile. 2015. Playing with knowledge: A virtual player for who wants to be a millionaire? that leverages question answering techniques. *Artificial Intelligence*, 222:157–181.

Federico Sangati, Antonio Pascucci, and Johanna Monti. 2018. Exploiting Multiword Expressions to solve "La Ghigliottina". In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, Turin, Italy. CEUR.org.

Giovanni Semeraro, Pasquale Lops, Pierpaolo Basile, and Marco de Gemmis. 2009. On the Tip of My Thought: Playing the Guillotine Game. In Craig Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1543–1548. Morgan Kaufmann.

Giovanni Semeraro, Marco de Gemmis, Pasquale Lops, and Pierpaolo Basile. 2012. An artificial player for a language game. *IEEE Intelligent Systems*, 27(5):36–43.

Luca Squadrone. 2018. Computer challenges guillotine: how an artificial player can solve a complex language TV game with web data analysis. In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, Turin, Italy. CEUR.org.