Robospierre, an Artificial Intelligence to Solve “La Ghigliottina”

Nicola Cirillo
University of Salerno
Salerno, Italy
n.cirillo9@studenti.unisa.it

Chiara Pericolo
University of Salerno
Salerno, Italy
c.pericolo@studenti.unisa.it

Pasquale Tufano
University of Salerno
Salerno, Italy
p.tufano@studenti.unisa.it

Abstract

This paper describes Robospierre a system developed to solve the language game “La Ghigliottina” (the guillotine). To find the solution of a game instance, it relies on MWEs automatically extracted through a lexicalized association rules algorithm; on a list of proverbs; and on some lists of titles.

1 Introduction

“La Ghigliottina” is the final game of “L’Eredità”, an Italian quiz show. In this game, the player should find a word linked to a set of five clue words. For example, if these words are *table*, *works*, *watch*, *Premier League* and *police*, the player should give as solution the word *calendar*. The link between a clue and the solution is usually the fact that both these words are part of an MWE (Multi-Word Expression) e.g. *table* and *calendar* are linked because they are part of the MWE *table calendar*. However, there can be also other kind of links. For example, the two words can be both part of a proverb (e.g. *bird* and *world* in the proverb “early bird catches the world”), of a film title (e.g. *river* and *return* in “River of No Return”) or they can be linked semantically (e.g. *Suarez* and *bite* because of the Suarez’s bite to Chiellini during the 2014 World Cup). The task of solving this game was presented as the NLP4FUN task of Evalita 2018 (Basile et al., 2018).

To build our system, first, we collected and analyzed a corpus of 296 game instances: 146 from the tv show and 150 from the board game. Second, we built an association matrix launching a lexicalized association rules algorithm, developed by us, on Paisà (Lyding et al., 2014). Then, we collected from the web a list of titles of books, films, plays and songs; and a list of proverbs. Finally, we tested the system on the game instances collected and we compared it with other artificial players of “La Ghigliottina”, especially UNIOR4NLP (Sangati, Pascucci and Monti, 2018), that obtained the best performance on this task at Evalita 2018 (Basile et al., 2018).

2 Related Works

In the field of AI (Artificial Intelligence), games have ever provided challenging tasks that encouraged researchers to develop better and better systems (Yannakakis and Togelius, 2018). In regard to language games, worth citing is the IBM Watson system designed to play Jeopardy!™ (Ferrucci et al., 2013). However, only recently, the task of solving “La Ghigliottina” has attracted the attention of researchers. Besides a first attempt in 2009 (Semeraro et al., 2009), the research on this topic began in 2018 when this task was proposed at the Evalita evaluation campaign (Basile et al., 2018).

2.1 Game Analysis

Sangati, Pascucci and Monti (2018) showed that “the words in the clues are typically nouns, verbs or adjectives, while the ones in the solutions are typically nouns or adjectives (never verbs)”. They also stated that “in most cases each clue word is connected with the solution because they form an MWE”. However, MWEs are not the only possible associations, some game instances require difficult inferences in order to be solved. (Basile et al., 2018).

2.2 Artificial Players

The first artificial player of “La Ghigliottina” is OTTHO (Semeraro et al., 2009; Basile et al., 2016) which employs an association matrix that uses a spreading activation model on a knowledge repository to compute the degree of correlation between two terms (the repository was built using web sources like Wikipedia). During Evalita 2018

(Basile et al., 2018) two artificial players were presented: UNIOR4NLP (Sangati, Pascucci and Monti, 2018) and the system developed by Squadrone (2018). The first is based on MWEs. It employs an association-score matrix that was populated computing the PMI (Pointwise Mutual Information) measure for each pair of words. In computing this measure, only co-occurrences in specific patterns (that represents MWEs) were considered. The second system is based on an algorithm that works in two steps. First, the system extracts a set of possible solutions from a knowledge base using the five clue words. Then, the algorithm verifies the existence of proverbs, aphorisms, and titles in which the possible solutions and the clues co-occur.

3 Our Approach

Our approach is quite similar to the approach of Sangati, Pascucci and Monti (2018) since it also relies on MWEs and makes use of an association matrix to find the solution of the game. However, there are some differences between our approach and theirs.

First, we used MWEs only to find links between two words in Italian corpora while UNIOR4NLP used them also to find associations in other resources like titles and proverbs (Sangati, Pascucci and Monti, 2018). We decided that, in a title and in a proverb, a simple co-occurrence is a valid link. In fact, there are game instances in which a clue is linked to the solution because both appear in the same title or proverb, even if they do not form an MWE. For example, in a game instance, the clue *occasione* (opportunity) is linked to the solution *ladro* (thief) because both appear in the famous Italian proverb “l’occasione fa l’uomo ladro” (opportunity makes a thief) even if they do not form any MWE.

In regard to the links extracted from Italian corpora, we used association rules (Agrawal and Srikant, 1994) instead of PMI. We decided to use this measure because, in MWEs, there is a head and the rest of the expression depends on it. For example, in the MWE *pesca con la mosca* (fly fishing), the word sequence *con la mosca* (with the fly) rarely appear without the noun *pesca* (fishing | peach). However, the noun *pesca* will appear a lot of times without being followed by the word sequence *con la mosca*. The PMI between the terms *pesca* and *mosca* will be low because the noun *pesca* has a relatively high frequency. Conversely, with association rules, this same link will be considered much stronger.

Another difference is that we produced a rule for every MWE and then the link between two words is defined as the score of the rule that has the highest score among all the rules in which one word appear in the consequent and the other in the antecedent (see Subsection 4.1). On the other hand, Sangati, Pascucci and Monti (2018) computed a single PMI value between two words considering all the MWEs in which these words occur. If the two systems compute the link between the words *dare* (to give) and *mano* (hand) and, in the corpus, these two words occur in the MWEs *dare una mano* (give a hand | to help) and *dare la mano* (hold hands). UNIOR4NLP will consider both these MWEs in computing the PMI between *dare* (to give) and *mano* (hand) while our system will generate two different rules: (*una mano* → *dare*) and (*la mano* → *dare*), then it will assign at the link between *dare* and *mano* the highest score between the scores of the two rules. This means that probably UNIOR4NLP will give at this link a higher score than our system.

The last difference is that Sangati, Pascucci and Monti (2018) prioritized the strength of the links over their number while we did the opposite. In fact, they considered all the words linked to each other with at least a minimum score. In this way, it is impossible to determine the number of clues to which a word is linked because every word is always linked with all the five clues. Conversely, in our system, a word is usually linked with only a subset of words. Given a game instance, our system tends to answer with a word that is linked to as many clues as possible.

4 System Description

Robospierre is composed of a scoring system and 7 linguistic resources: an association matrix, a list of proverbs, 5 lists of titles and a list of compound words. This system takes in input a set of five clues that represents a game instance. For each clue, it extracts from the resources all the words that are linked to that clue. Then, a score value is assigned to each word (it represents the strength of that link). The words extracted in this way form the set of candidate solutions. This set is then processed by the scorer that ranks each candidate solution according to the strength of the links between it and the five clues. Finally, the answer produced by the system is the candidate solution that has the highest rank.

4.1 Association Matrix

The association matrix is an *S-C* matrix where S is the set of candidate solutions and C is the set of possible clues. To list the possible clues, we took the words whose lemma occurs in Paisà (Lyding et al., 2014) at least 10 times. Then we performed the POS tagging on these lemmas with Nooj (Silberztein, 2018) using as lexical resources _Sdic_it.nod, Dnum.nom, tronche.nod, toponimi.nod, ElisioniContrazioni.nod and as syntactic resources DNUM.nog (Vietri, 2014). From the list obtained, we extracted only nouns, adjectives, verbs, and prepositions and then we inflected them (with Nooj). On the other hand, the set of candidate solutions is a subset of the set of possible clues containing only nouns and adjectives.

To populate the matrix, we developed a lexicalized association rules algorithm based on Apriori (Agrawal and Srikant, 1994). In our algorithm, a rule is an implication $A \rightarrow B$ where A and B are sequences of words. To generate the possible rules, our algorithm uses a function written by us: *genMWE*. This function takes five arguments: *D*, *antecedent*, *consequent*, *position* and *lemmatize*. *D* is a text; *antecedent* and *consequent* are sequences of POS tags that represent respectively the possible antecedents and the possible consequents of the rules. The argument *position* tells the function where it must search for the consequent in relation to the position of the antecedent. It can take the values *forward*, *backward* and *both*. The value *forward* means that the consequent directly follows the antecedent in the text, the value *backward* means that the consequent directly precedes the antecedent and the value *both* means that the consequent can either follow or precede the antecedent. The argument *lemmatize* can take a Boolean value. If it takes *true*, the antecedents of all the rules will be lemmatized. For example, if we run the function on a text with parameters *antecedent = PREP N*, *consequent = N*, *position = backward* and *lemmatize = false*; it will generate rules such as (*di credito → carta*) (credit card), (*di credito → carte*) (credit cards), (*da guardia → cane*) (watchdog), etc. Table 1 shows the parameters used in our experiment. While the algorithm is generating the candidate rules, it counts the occurrences of every rule ($ws_j \rightarrow ws_i$) and the occurrences of the word sequences ws_j that match the pattern of POS tags given as consequent. Finally, the algorithm computes, for every rule, the confidence (1), the lift (2) and a score value (3) used to solve the game instances.

Rules	Position	Lemmatize	Example
N → N	both	False	lupo → cane
A → N	both	False	intenzioni → buone
PREP N → N	backward	False	di vista → punto
PREP DET N → N	backward	False	con la mosca → pesca
CONG N → N	backward	False	e gatti → cani
N → PREP	backward	False	permesso → con
N → V	backward	True	via → andare
DET N → V	backward	True	la spugna → gettare
PREP N → V	backward	True	con mano → toccare
PREP DET N → V	backward	True	per i fondelli → prendere

Table 1: Parameters given to the genMWE function

$$conf_r = \frac{Count(ws_i, ws_j)}{Count(ws_j)} \tag{1}$$

$$lift_r = \frac{conf_r}{P(ws_i)} \tag{2}$$

$$score_r = Count(ws_i, ws_j)\, conf_r \times 100 \tag{3}$$

We pruned the rules that disrespect one or more of the following constraints:

- Count(ws_i, ws_j) > 1
- $conf_r > 0.001$
- $lift_r > 1$
- $score_r > 2$

Once generated the rules, the score of a link in the association matrix between a pair of words w_i, w_j is defined in the following equation (4).

$$score_{w_i, w_j} = \max_{r \in R_1 \subseteq R}(score_r) \tag{4}$$

Where R_1 is a subset of R containing all the rules in which the word sequence ws_i includes the word w_i or the word w_j and the word sequence ws_j includes the other word of the pair. If there are no rules with this feature, the two words w_i, w_j are not linked to each other.

To populate the association matrix, we ran this algorithm on the Paisà corpus (Lyding et al., 2014).

4.2 Lists

To handle the links where the two words are part of a proverb or of a title, we collected from the web the following lists:

- Proverbs: A list of 2048 Italian proverbs collected from Wikiquote.[1]

- Films: A list of 13098 film titles collected from Film.it.[2]

- Books: A list of 1633 book titles collected from Cultura&Svago.[3]

- Songs: A list of 984 Italian song titles collected from various web sources.[4]

- Plays: A list of 739 play titles collected from Wikipedia.[5]

We consider linked two words that appear in the same element of one of these lists. We assigned at these links a fixed score value (see Subsection 5.1).

4.3 Compound Words

The link between a clue and the solution can be also the fact that both the words appear in a compound word. For example, the words *police* and *man* are linked because they appear in the compound word *policeman*. However, there are game instances where the two words appear concatenated in a word that is not a compound. For example, *franco* (frank) and *forte* (strong) can be linked because of the word *Francoforte* (Frankfurt) although this word is not a compound.

[1] Wikiquote. Proverbi italiani. https://it.wikiquote.org/wiki/proverbi_italiani

[2] Film.it, Film A-Z. https://www.film.it/film/film-a-z/

[3] Cultura&Svago, Mille titoli letteratura mondiale. https://www.culturaesvago.com/mille-titoli-letteratura-mondiale/

[4] Il blog di Alessandro Paldo, Le 1000 canzoni italiane più belle di sempre. http://alessandro-paldo.blogspot.com/2013/10/1-10-1.html?m=1
Panorama, Le 100 canzoni italiane più belle del ventunesimo secolo (fino ad ora...). https://www.panorama.it/musica/le-100-canzoni-italiane-piu-belle-del-ventunesimo-secolo/
Le Canzoni d'Amore, Canzoni d'amore Italiane: una lista di brani tra i più belli di sempre. http://www.lecanzonidamore.it/canzoni-d-amore-italiane/classifiche-italiane/250-canzoni-d-amore-italiane-una-lista-di-brani-tra-i-piu-belli-di-sempre.html

[5] Wikipedia, Elenco di opere teatrali. https://it.wikipedia.org/wiki/Progetto:Teatro/Elenco_di_opere_teatrali

To handle these links, we consider linked two words that appear compounded in a noun listed in the set of possible clues used in the association matrix (see Subsection 4.1). We assigned at this links a fixed score value (see Subsection 5.1).

4.4 Scoring System

Given five clues (a game instance), our system uses the resources presented above to rank the possible solutions and give an answer. This occurs in six steps:

1. For every clue $c \in C$, it generates a set of candidate solutions S finding all the words linked to c in the matrix, in the lists, and in the compound words.

2. It generates, for every candidate solution $s \in S$ a set of scores $V_{s,c}$ that contains a score for every resource in which the clue c and the candidate solution s are linked (5).

$$V_{s,c} = \{score_{s,c,1}, score_{s,c,2}, \dots, score_{s,c,n}\} \quad (5)$$

3. From the set of scores of every candidate solution, the system keeps only the highest (6).

$$v_{s,c} = \max_{i=1,n}(score_{s,c,i}) \quad (6)$$

4. Then, it standardizes every score in an interval (between 0 and 100) and adds to the value obtained a bonus of 100 that represents the existence of a link between that candidate solution and the clue (7)(8)(9).

$$max = \max_{s \in S}(v_{s,c}) \quad (7)$$

$$min = \min_{s \in S}(v_{s,c}) \quad (8)$$

$$std_{s,c} = \left(\frac{v_{s,c} - min}{max - min} \times 100\right) + 100 \quad (9)$$

5. Once completed the steps 1-4 for all the clues in the game instance, the system sums all the scores of that candidate solution to produce its final score f_s (10).

$$f_s = \sum_{c \in C} std_{s,c} \quad (10)$$

6. The answer given by the system is the candidate solution that obtains the highest final score value (11).

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}}(f_s) \quad (11)$$

5 System Evaluation

To evaluate the artificial players of “La Ghigliottina” Basile et al. (2018) made use of the MRR (Mean Reciprocal Rank) measure weighted by a function that lower the score according to the time taken by the system to provide the answer (12).

$$MRR = \frac{1}{|G|} \sum_{g \in G} \frac{1}{r_g} \max\left(\frac{1}{t_g}, \frac{1}{10}\right) \quad (12)$$

In this equation, G is the set of game instances, r_g is the rank that the solution of the game g has in the set of answers produced by the system, and t_g is the time (in minutes) that the system takes to provide the set of answers (Basile et al., 2018).

The first 100 answers that the system provides are considered in computing the MRR and a game instance is considered solved when the solution is among these 100 answers. According to this evaluation, UNIOR4NLP (Sangati, Pascucci and Monti, 2018) obtained an MRR of 0.6428 and solved the 81.90% of the game instances while Squadrone (2018) obtained an MRR of 0.0134 and solved the 25.71% of the game instances.

Basile et al. (2016) evaluated OTTHO using the precision-k measure. A game is considered k-solved if the solution has rank k or higher in the set of answers provided by the system (13).

$$precision - k = \frac{k\text{-}solved\ game\ instances}{total\ game\ instances} \quad (13)$$

With k = 1, the best model of OTTHO obtained a precision of about 0.25 on tv games and about 0.30 on board games. With k = 100, it obtained a precision of about 0.50 on tv games and about 0.70 on board games (Basile et al., 2016).

In order to evaluate our system, we collected 294 game instances where the solution was provided: 146 from the tv show and 150 from the board game. Then, we submitted them to the system and computed the MRR (12) considering only the first 100 candidates solutions ranked according to their final scores (10).

To see how the different linguistic resources affect the performance, we tested different version of our system: one with only the association matrix; one with the association matrix and the compound words; and one with the matrix, the compound words and the lists of titles that represents the full system.

Finally, in order to compare our system to UNIOR4NLP (Sangati, Pascucci and Monti, 2018), we submitted the same game instances to the Telegram bot version of UNIOR4NLP and then we computed the precision-k (13) of the two systems for k = 1 (since the UNIOR4NLP bot provides only one answer).

5.1 Parameters Used in the Tests

We assigned to the links in the compound words (see Subsection 4.3) a score of 100 since these links seemed very reliable associations.

To the links in the lists of titles (see Subsection 4.2), we assigned a score of 5 because higher values seemed to worsen the performance of the system and, with lower values, the full model (matrix + compound + titles) gives the same answers of the previous one (matrix + compound).

5.2 Analysis of the Results

The result of the first test are displayed in Table 2. Our system obtained a quite good result if compared to the other systems. It was also able to provide the answer always in the first minute as UNIOR4NLP did (Basile et al., 2018). It performed better on the tv games than on the board games. Maybe because in the tv games, the links are more often based on MWEs while in the board game, there are more links based on titles, proverbs and semantic associations and our system does not treat these links as good as it treats the links based on MWEs (the links based on semantic associations are not even treated). This hypothesis is confirmed by the fact that the list of proverbs and the lists of titles worsen the performance of the system (see Table 3).

We suppose that this problem is caused by the

	All	Tv	Board game
MRR	0.4140	0.4794	0.3660
Correct Answers	72.30%	80.82%	64.00%

Table 2: Result of first test

Models	Precision-1		
	All	Tv	Board game
Matrix	0.3480	0.4014	0.2933
Matrix + compounds	0.3514	0.4178	0.3067
Matrix + compounds + titles	0.3446	0.4178	0.3000
UNIOR4NLP	0.5608	0.6643	0.4600
	Tot (296)	Tot (146)	Tot (150)

Table 3: Result of second and third tests

fact that we assigned at every link in the lists the same score. However, there are titles and proverbs that are more likely to produce reliable links and some others that are not. The more an element is known, the more the links in it must be reliable. Maybe, assigning at every element in the lists a score that represents how much that element is known, might lead to an improvement of system performance. This score might be based on the number of results retrieved when that element is searched with a search engine like Google.

The result of the third test are displayed in Table 3. As the result show, our system was not able to reach the performance of UNIOR4NLP. However, we found among the game instances 20 games to which our system answered correctly while UNIOR4NLP did not. We will analyze some of these instances that are of particular interest.

The first is the following:

CLUES: *cravatta; neve; S. Martino; pizza; altare*
ANSWER: pala

Our system gave to this game instance the correct answer *pala* (shovel | blade | altarpiece) while UNIOR4NLP gave the answer *bianca* (white). We suppose that UNIOR4NLP gave this answer because, sometimes, it overestimates the strength of a link and ignores the other links. We believe that the answer *bianca* is mainly due to the clue *neve* (snow) since UNIOR4NLP considered both the compound noun *Biancaneve* (Snow-white) and the frequent co-occurrence between the adjective *bianca* and the noun *neve* to compute the PMI between these two terms. On the other hand, our system found three weak links: between *pala* and *neve*; between *pala* and *pizza* and between *pala* and *altare* (altar). These links were sufficient to assign to this word the highest rank among the candidate answers produced.

Another interesting game instance is the following:

CLUES: introduzione; cowboy; fungo; 23; fare tanto
ANSWER: cappello

UNIOR4NLP gave to this game instance, the answer *proiettili* (bullets). Our system gave the correct answer *cappello* (hat). Maybe, the answer of UNIOR4NLP was due to the overestimation of the link between *proiettili* and the clue *cowboy* while it underestimated the link between this clue and the word *cappello*. We believe that this happened because *cappello* occurs in more contexts than *proiettili*. On the other hand, our system gave the correct answer *cappello* because it was strongly linked with the word sequence *da cowboy* (like cowboys) since this sequence almost always occurs in the MWE *cappello da cowboy* (cowboy hat).

The last game instances that we will analyze is the following:

CLUES: *andare; musica; occhi; mano; buona*
ANSWER: palla

To this game instance, our system answered *palla* (ball) and UNIOR4NLP answered *pallino* (cue ball | dot). We suppose that this error is caused by the MWE *andare a pallino* (right on cue) that appear in the online dictionary “Il Nuovo De Mauro” (De Mauro, 2016) which was employed by UNIOR4NLP as linguistic resource. UNIOR4NLP considered a co-occurrence in this dictionary as strong as 200 co-occurrences in the Italian corpora so this link obtained a higher PMI than that between *andare* and *palla* but, actually, the MWE *andare in palla* (be confused) is much more common than *andare a pallino*.

6 Conclusions

We described and tested Robospierre, a system developed to solve the word game “La Ghigliottina” (the guillotine). The result of the tests showed that, even if its result were below state-of-the-art, it was able to solve some game instances that the state-of-the-art system did not solved.

In the future, we plan to improve the extraction of the links in the MWEs extracting them from a bigger corpus. We also intend to assign at every element in the list of proverbs and in the lists of titles a score that represents how much that element is known.

Reference

Agrawal Rakesh and Srikant Ramakrishnan. 1994. “Fast algorithms for mining association rules.” *Proc. 20th int. conf. very large data bases, VLDB. Vol. 1215.*

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

Basile Pierpaolo, de Gemmis Marco, Siciliani Lucia, and Semeraro Giovanni. 2018. "Overview of the evalita 2018 solving language games (nlp4fun) task." In Caselli Tommaso, Novielli Nicole, Patti Viviana, and Rosso Paolo, editors*, Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, CEUR.org, Turin, Italy.

De Mauro Tullio. 2016. *Il Nuovo De Mauro (Online)*. Available at: dizionario.internazionale.it.

Ferrucci David A., Levas Anthony, Bagchi Sugato, Gondek David, and Mueller Erik T. 2013. "Watson: Beyond jeopardy!" In *Artif. Intell., 199* pages 93–105.

Lyding Verena, Stemle Egon, Borghetti Claudia, Brunello Marco, Castagnoli Sara, Dell'Orletta Felice, Dittmann Henrik, Lenci Alessandro, and Pirrelli Vito. 2014. "The PAISÀ corpus of italian web texts." In *Proceedings of the 9th Web as Corpus Workshop (WaC-9),* pages 36–43. Association for Computational Linguistics, Gothenburg, Sweden.

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

Semeraro Giovanni, Lops Pasquale, Basile Pierpaolo, and De Gemmis Marco. 2009. "On the tip of my thought: playing the guillotine game." In *Proceedings of the 21st International Jont Conference on Artifical Intelligence, IJCAI'09*, pages 1543–1548. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Silberztein Max. 2018. NooJ Manual, Available for download at: www.nooj4nlp.net

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

Vietri Simonetta. 2014. "The italian module for nooj." In *Proceedings of the First Italian Conference on Computational Linguistics*, CLiC-it 2014. Pisa University Press.

Yannakakis Georgios N. and Togelius Julian. 2018. *Artificial Intelligence and Games*. Springer.