Exploiting Multiword Expressions to solve “La Ghigliottina”

Federico Sangati
University L’Orientale
Naples, Italy
fsangati@unior.it

Antonio Pascucci
University L’Orientale
Naples, Italy
apascucci@unior.it

Johanna Monti
University L’Orientale
Naples, Italy
jmonti@unior.it

Abstract

English. The paper describes UNIOR4NLP a system developed to solve “La Ghigliottina” game which took part in the NLP4FUN task of the Evalita 2018 evaluation campaign. The system is the best performing one in the competition and achieves better results than human players.

Italiano. *Il contributo descrive il sistema* UNIOR4NLP, *sviluppato per risolvere il gioco “La Ghigliottina”, che ha partecipato alla sfida NLP4FUN della campagna di valutazione Evalita 2018. Il sistema risulta il migliore della competizione e ha prestazioni più elevate rispetto agli umani.*

1 Introduction

In this paper we describe UNIOR4NLP, a system which took part in the NLP4FUN task of the Evalita 2018 evaluation campaign (Basile et al., 2018). The goal of this task is to design a solver for “La Ghigliottina”, the final game of the popular Italian TV quiz show “L’Eredità”. The game involves a single player, who is given a set of five words (clues), each one linked with an unknown sixth word that represents the solution to the game. For example, given the set of clues [*fighting*, *gun*, *roof*, *eater*, *set*] the solution is *fire*, because: *the roof is on fire* is a title of a famous song, while *fire fighting*, *fire a gun*, *fire-eater*, and *set something on fire* are fixed word constructions.

UNIOR4NLP relies on the assumption that Multiword Expressions (MWEs) play an important role in solving the game: given a set of clues, the system outputs the solution word which forms the strongest connections with all of the clues.

The paper is organized as follows: in Section 2 we present related work. In Section 3 we describe the different steps we took in order to prepare and tune the UNIOR4NLP system. In Section 4 we describe our system and its functioning, while results are presented in Section 5, where we also focus on error analysis concerning both the data-set of the NLP4FUN task and our system. Finally, conclusions and future work are presented in Section 6.

2 Related work

From the very beginning of Artificial Intelligence (AI) games represented an interesting playground to test the results of research in this field (Yannakakis and Togelius, 2018). NLP plays an essential role in solving language related games and recent examples, such as the IBM Watson system in Jeopardy!$^{\text{TM}}$ (Ferrucci et al., 2013), have proven that its use can result in groundbreaking technology. An interesting test-bed for this type of approach is represented by language games, such as the *Wheel of Fortune*, *Who Wants to be a Millionaire?* and “*La Ghigliottina*”.

The game “La Ghigliottina” is particularly challenging because its solution is based on modelling how words are connected to each other. A first artificial player of the game, OTTHO (Semeraro et al., 2009; Basile et al., 2016) exploits i) resources from the web such as Wikipedia to build a lexicon and a knowledge repository and ii) a knowledge base modeling represented by an association matrix which stores the degree of correlation between any two terms in the lexicon. Word correlations are detected by connecting i) lemmas to the terms in its dictionary definition, pair of words occurring in a proverb, movie or song title, and iii) pair of similar words by exploiting Vector Space Models (Salton et al., 1975).

In our approach, we make use of similar resources but we only rely on a very limited set of syntactic constructions (patterns) to correlate words and build our association matrix.

3 Solving the Ghigliottina game

Building an automatic solver for the Ghigliottina game requires a number of preliminary steps: i) the analysis of real game instances, ii) the analysis of patterns that could help the system in solving the game, iii) the collection of the linguistic resources necessary to tune the system for the task.

3.1 Analysis of real game instances

We have analyzed a sample of 100 game instances that we personally collected from the last five editions of the TV show. We found out that in most cases each clue word is connected to the solution because they form a Multiword Expression (MWE). We have used this key observation in designing our system. We started working on our system before the announcement of the NLP4FUN task. Since our system is not supervised, the extra data-set is not adding any advantage to our system. After the official data-set was released, we found out that a good number of game instances was confirming our initial finding. However we also observed a number of unusual cases which will discuss in more depth in Section 5.2.

A MWE can be defined as a sequence of words that presents some characteristic behaviour (at the lexical, syntactic, semantic, pragmatic or statistical level) and whose interpretation crosses the boundaries between words (Sag et al., 2002). MWEs have to be considered as lexical items which convey a single meaning different from the meanings of the constituents of the MWE, such as in the idiomatic expression *kick the bucket* where the simple addition of the meanings of *kick* and *bucket* does not convey the meaning of *to die*.

We have different classes of MWEs, such as idioms (*break a leg*), verb particle constructions (*to call off*), light verbs constructions (*to provoke a reaction*). For a detailed overview of MWEs in NLP applications we refer the reader to Constant et al. (2017). For the purpose of the current task we considered only those MWEs characterized by fixed syntactic patterns described in the following section.

3.2 Pattern Analysis

A first analysis of the tuples from the sample mentioned above revealed that words in the clues are typically nouns, verbs, or adjectives, while the ones in the solutions are typically nouns or adjectives (never verbs). A more detailed investigation resulted in the definition of six patterns that identify valid MWEs connecting clue and solution pairs. We list them below with some examples from our data-set (solution words are underlined):

A B: *diario segreto* (‘diary secret’ → secret diary), *brutta caduta* (‘ugly fall’ → bad fall), *permesso premio* (‘permit price’ → good behaviour license), *dare gas* (‘give gas’ → accelerate).

A det B: *dare il permesso* (‘give the permit’ → authorize).

A prep B: *colpo di coda* (‘flick of tail’ → last ditch effort).

A conj B: *stima e affetto* (esteem and affection).

A prepart B or **A prep det B**: *virtù dei forti*, part of the famous Italian proverb *La calma è la virtù dei forti* (patience is the virtue of the strong).

A+B: compounds such as *radio* + *attività* = *radioattività* (radio + activity = radioactivity).

3.3 Linguistic Resources

On the basis of the linguistic analysis described above, we collected the linguistic resources which we deemed necessary for the task. To this end we used the following freely available corpora:

Paisà: 225 M words corpus automatically annotated (Lyding et al., 2014).

itWaC: 1.5 B words corpus automatically annotated (Baroni et al., 2009)

Wiki-IT-Titles: Wikipedia-IT titles downloaded via WikiExtractor (Attardi, 2016).

Proverbs: 1955 proverbs from Wikiquote (2016) and 371 from an online collection (Dige, 2016).

In addition, we have constructed the following lexical resources:

DeMauro-Ext: words extracted from “Il Nuovo vocabolario di base della lingua italiana”(De Mauro, 2016b), extended with morphological variations obtained by changing last vowel of the word and checking if the resulting word has frequency $\geq$ 1000 in Paisà.

DeMauro-MWEs: MWEs extracted from the “De Mauro online dictionary” (De Mauro, 2016a) composed of 30,633 entries.

4 System description

In order to build our system, we started processing the selected corpora via standard tokenization (only single word tokens) and removal of punctuation marks and non-word patterns. We next constructed two lexical sets: C_{LEX} to cover the *clue* words, and S_{LEX} to cover the *solution* words. S_{LEX} (composed of 7,942 nouns and adjectives in DeMauro-Ext) is smaller than C_{LEX} (composed of 19,414 words from the full DeMauro-Ext and DeMauro-MWEs) because solution words are almost always nouns or adjectives as described in Section 3.2.

Secondly, we built a co-occurrence matrix M_c which stores the counts $c_{i,j}$ for every pair of words $w_i \in S_{LEX}$ and $w_j \in C_{LEX}$ such that w_i co-occurs with w_j in the resources according to patterns described in Section 3.2. Co-occurrence patterns were extracted from Paisà and itWaC with weight $w = 1$, from DeMauro-MWE with $w = 200$, from Proverbs with $w = 100$, and from Wiki-IT-Titles with $w = 50$. The weight were chosen manually taking into account the likelihood that a pattern in a given corpus represented a valid MWE. Compound patterns (A+B) were extracted from C_{LEX}: for every word w in C_{LEX} if $w = ab$, a and b are both in C_{LEX}, and a and b have at least 4 characters, the count for the pair (a, b) is incremented by 1 in the co-occurrence matrix.

Thirdly, for every pair of words w_i and w_j in M_c, we populate the association-score matrix M_{pmi} via the Pointwise Mutual Information measure:

$$M_{pmi}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i) \cdot p(w_j)} \qquad (1)$$

where

$$p(w_i) = \sum_{w_j \in C_{LEX}} M_c(w_i, w_j) \qquad (2)$$

$$p(w_j) = \sum_{w_i \in S_{LEX}} M_c(w_i, w_j) \qquad (3)$$

$$p(w_i, w_j) = \frac{M_c(w_i, w_j)}{\sum_{\substack{x \in S_{LEX} \\ y \in C_{LEX}}} M_c(x, y)} \qquad (4)$$

Finally, for a given game instance with the 5 clue words $G = (w_{c1}, w_{c2}, w_{c3}, w_{c4}, w_{c5})$, we choose the solution word $\widehat{w_s} \in S_{LEX}$ such that:

$$\widehat{w_s} = \max_{w_s \in S_{LEX}} \sum_{w_c \in G} M_{pmi}(w_s, w_c) \qquad (5)$$

that is, we choose the word in S_{LEX} which maximizes the score obtained by summing the pmi between each clue word and the candidate word. If two words are never seen co-occurring together in a pattern in the training corpora, we assign to them the lowest pmi value in M_{pmi}.

The system has been implemented in Python and the code is open source.[1] After the matrix has been loaded into memory the response time on an average laptop is around 1-2 seconds.

5 Results

According to Basile et al. (2018), Unior4NLP is the best performing system in the Evalita NLP4FUN task. Table 1 provides the detailed results, including split results on TV and Board Game (BG) subsets. The system achieved a very high performance: in more than half of the games (64/105) it is able to guess the correct word.

In the attempt to compare the performance of our AI system with that of a top player we analyzed the games played by Andrea Saccone, who has been the biggest champion of the Ghigliottina game so far: he was champion for 13 days (3-15 March 2018), and he managed to find the correct solution three times.[2] In comparison, Unior4NLP was able to win the same game instances 9 times.

SET	SIZE	MRR	R@100
TEST ALL	105	**0,64**	**0.82**
TEST TV	66	0.67	0.88
TEST BG	39	0.60	0.72
DEV ALL	315	0.56	0.80
DEV TV	204	0.61	0.85
DEV BG	111	0.48	0.71

Table 1: Results on the TEST and DEV set. Evaluations are the MRR (Mean Reciprocal Rank) and R@100 (recall at 100).

[1] https://gitlab.com/kercos/ghigliottina

[2] The players who reach the "Ghigliottina" game (the champion) continue to participate in the subsequent episodes even if they do not guess the solution word.

The plot in Figure 1 shows the distributions of the scores for the correct and missed solutions of our system on the full set of games in the development and test set (420 in total). This allows us to set a number of confidence values for our system: if the system returns a solution of a game with a score $S \geq 10$ we can be reasonably certain ($68/69 = 99\%$) that the system has guessed the correct solution, if $5 \leq S < 10$ we are above chance level ($50/70 = 71\%$), if $0 \leq S < 5$ we are at chance level ($56/112 = 50\%$) and if S is below 0, we are below chance level ($48/169 = 28\%$).

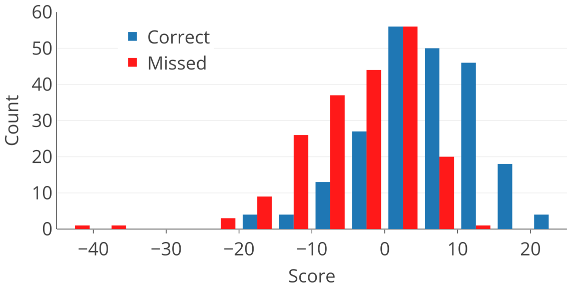


Figure 1: Distribution of the correct and missed solutions with respect to their score.

5.1 Data-set analysis

In the development data-set we found several cases which fall outside the patterns we observed in out data-set. For instance, we noticed the presence of digits in some clues or solution words (1973, 33), game instances with a clue being also the solution (‘sostanza’, ‘fuori’), and words being spelled in different ways (‘tenère’, ‘tenere’).

Moreover, we also observed a number of ‘clue - solution’ pairs which are very difficult to relate. We list below some examples with some possible explanation:

- g - orecchio: (g - ear) the letter ‘g’ has the shape of a ear.
- classe 1973 - 33: (class 1973 - 33) this game instance was from 2006, and that year people born in 1973 were 33 years old.
- ...—... - titanic: the clue being the S.O.S. beacon in morse code.

One possible reason for these inconsistent cases is that Board Game edition use slightly different criteria to correlate words,[3] and that those from the TV set date back to the very first editions of the TV game (when correlation criteria where probably not yet well defined).

5.2 System error analysis

In this section we analyze some types of errors that our system makes, and we provide some suggestion for possible improvement.

Word similarity Although quite rare, few of the clue-solution links can be explained by the similarity relation. An example is the clue-solution pair *sincero-franco* (sincere-frank). Those are not easily captured by patterns of the types described in Section 3.2, but could be included by means of automatic detection of word similarity via Vector Space Models (Salton et al., 1975) as done in Basile et al. (2016).

Missing words As explained in Section 3.2, we restricted the set of words in the solution set. This choice, while helping the system to restrict the search space, leads to some coverage issues. For instance, *pennello* (brush) is one of the solutions of the games in the test data not present in our solution set. In the future we would like to experiment increasing the size of the solution set while avoiding performance and memory problems.

Wrong PoS Our system analyzes words in their surface form, so it cannot distinguish cases where the same word-form can have multiple Part of Speech (PoS) (with different meaning). To avoid this problem we could envision a system which takes PoS and word-sense disambiguation into consideration.

Multiword clues Although the great majority of the clues are constituted by a single word, there are a few exceptions (typically names of saints). The current system considers only single-word tokens, so if a game has a 2-word clue, it is regarded as two separate clues (their contribution is then average to obtain the final score). The system could be optimized by using a tokenizer which keeps specific types of bigrams connected.

Association metrics As described in Section 4, we compute the association score between any pair of words in the matrix via the Pointwise Mutual Information measure (pmi). There is still a big number of alternative measures (Pecina, 2010) that might lead to higher performance.

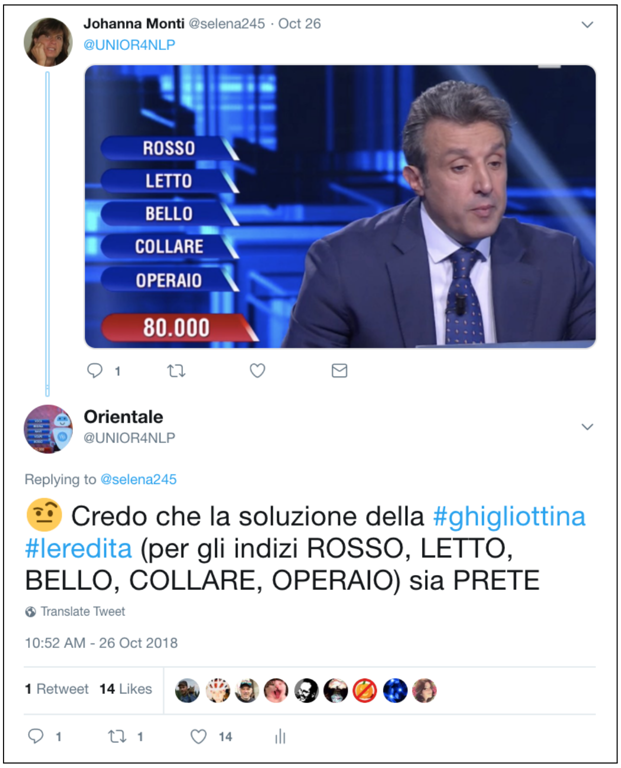
[3] This is supported by results in Table 1, where Board Game results are lower than those from the TV set.

6 Conclusions and future work

In this paper we described UNIOR4NLP, an artificial player of “La Ghigliottina”, a challenging game which requires linguistic knowledge to be solved. We described the preliminary steps that we made before developing our system (identifying linguistic patterns that are relevant in the game) as well as the algorithms and the methodology we adopted. The system achieved a high performance but we believe that with further tuning it can still be improved.

Future work will focus on adopting the same methodology to automatically create novel game instances: using the same association-matrix we can choose a random word (the solution) and present the list of 5 clues with high score.

In order to make our system easily testable by the scientific community and general public, we have built an interactive version which can be accessed via a Telegram bot[4] and on Twitter[5] (see Figure 2).


Johanna Monti @selena245 · Oct 26

@UNIOR4NLP

Orientale
@UNIOR4NLP

Replying to @selena245

Credo che la soluzione della #ghigliottina #leredita (per gli indizi ROSSO, LETTO, BELLO, COLLARE, OPERAIO) sia PRETE

10:52 AM - 26 Oct 2018

1 Retweet 14 Likes

Figure 2: Screenshot of @UNIOR4NLP twitter response.

Acknowledgments

This research has been partly supported by the PON Ricerca e Innovazione 2014/20 fund. Authorship contribution is as follows: Johanna Monti is author of Sections 1, 2, 3.3 and 6; Federico Sangati of Section 4 and 5, and Antonio Pascucci of Sections 3.1. and 3.2.

References

Giuseppe Attardi. 2016. Wikiextractor. http : //attardi.github.io/wikiextractor. Last accessed on the 1st October 2018.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

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.

Pierpaolo Basile, Marco de Gemmis, Lucia Siciliani, and Giovanni Semeraro. 2018. Overview of the evalita 2018 solving language games (nlp4fun) task. 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)*. CEUR.org, Turin, Italy.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke Van Der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Tullio De Mauro. 2016a. Il Nuovo De Mauro (Online). https://dizionario.internazionale.it. Last accessed on the 1st October 2018.

Tullio De Mauro. 2016b. Il Nuovo vocabolario di base della lingua italiana (pdf version). https : //www.internazionale.it/opinione/tullio-de-mauro/2016/12/23/il-nuovo-vocabolario-di-base-della-lingua-italiana. Last accessed on the 1st October 2018.

Antonio Dige. 2016. Raccolta di proverbi e detti

[4] https://t.me/Unior4NLPbot

[5] https://twitter.com/UNIOR4NLP

italiani. http://web.tiscali.it/proverbiitaliani. Downloaded on the 24th April 2018.

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

Verena Lyding, Egon Stemle, Claudia Borghetti, Marco Brunello, Sara Castagnoli, Felice Dell'Orletta, Henrik Dittmann, Alessandro Lenci, and Vito Pirrelli. 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.

Pavel Pecina. 2010. Lexical association measures and collocation extraction. *Language Resources and Evaluation*, 44(1-2):137–158.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15. Springer.

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Giovanni Semeraro, Pasquale Lops, Pierpaolo Basile, and Marco De Gemmis. 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.

Wikiquote. 2016. Proverbi italiani. https://it.wikiquote.org/wiki/Proverbi_italiani. Downloaded on the 24th April 2018.

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