Preliminary Experiments on an Improved Artificial Player for a Word Association Game

Alberto Coffrini[1], Stefania Monica[2], and Federico Bergenti[1]

[1] Dipartimento di Scienze Matematiche, Fisiche e Informatiche
Università degli Studi di Parma, 43124 Parma, Italy
alberto.coffrini@studenti.unipr.it, federico.bergenti@unipr.it
[2] Dipartimento di Scienze e Metodi dell'Ingegneria
Università degli Studi di Modena e Reggio Emilia, 42122 Reggio Emilia, Italy
stefania.monica@unimore.it

Abstract. This paper presents recent developments of a software system that acts as an artificial player for a popular word association game. The game was proposed for the Evaluation Campaign of Natural Language Processing and Speech Tools for Italian in 2020, and it attracted the interest of various researchers. Several aspects of the recent developments of the artificial player are discussed, from the collection of the texts used to acquire sufficient linguistic knowledge, to the improvements of the algorithm employed to play the game. Preliminary, but encouraging, experimental results are also discussed in comparison with other artificial players for the same game.

Keywords: Word association games · Lexical semantics · Natural language processing · Artificial intelligence

1 Introduction

Natural Language Processing (*NLP*) is a broad research field that studies the interactions between computers and human languages in the attempt to make computers speak and understand human languages (e.g., [13]). By its nature, NLP is an interdisciplinary field located at the intersection of linguistics, computer science, and artificial intelligence.

The history of NLP includes a long list of particular problems that were addressed and effectively solved, but many other problems are still open and challenging. Among the traditional problems of NLP, it is worth mentioning automatic translation (e.g., [8]), which is the problem of generating a fluent text in a target language preserving the meaning of the original text written in a source language. A second traditional problem of NLP is text classification (e.g., [11]), which is the task of categorizing texts on the basis of their contents. Finally, a third traditional problem of NLP is information retrieval (e.g., [12]), which is the problem of automatically obtaining relevant information from texts.

Besides these traditional problems, the recent advent of new technologies promoted the interest in new application contexts for NLP. For example, the increasing pervasiveness of personal assistants such as Microsoft Cortana, Apple Siri, Amazon Alexa, and Google Home, renewed the interest in tasks related to automatic speech recognition (e.g., [10]). In addition, the diffusion of chatbots accelerated the research on question answering (e.g., [1]). Finally, the massive use of social networking services contributed to spread the interest in tasks related to sentiment analysis (e.g., [2]).

A plethora of approaches have been experimented over the years to effectively solve NLP problems. For example, logic programming has been playing a crucial role in NLP since the very first studies on computational linguistics (e.g., [7]). Logic programming is based on facts and rules, which is a feature shared with the ordinary approach to describe the surface grammars of human languages. This shared feature makes the use of logic programming particularly well suited to accomplish NLP tasks. Note that inductive logic programming (e.g., [14]) and probabilistic logic programming (e.g., [15, 16]) have also been successfully applied to accomplish NLP tasks. In addition to logic-based methods and techniques, statistical methods have been extensively used in the context of NLP (e.g., [5]). Such methods are typically based on decision trees and hidden Markov models. More recently, several approaches based on neural networks (e.g., [4]) and deep learning (e.g. [18]) have been successfully applied solve NLP problems.

The analysis of the specific application context is crucial to design and implement effective NLP systems. Actually, the common approach to design NLP systems is based on the identification of the relevant NLP problems to be solved. Such problems are then addressed using specific methods that are often designed for the purpose. It is common opinion among researchers interested in NLP that the use of methods specifically designed to target the problems at hand is the only viable approach to accomplish complex NLP tasks.

The NLP problem discussed in this paper was proposed for the *Evaluation Campaign of Natural Language Processing and Speech Tools for Italian* (*EVALITA*) in 2020, and it is called *Ghigliottin-AI* [3]. The challenge is to build a software system that can play a word association game called *La Ghigliottina* (Italian for *The Guillotine*), which is the closing game of a popular Italian television show. The rules of the game are simple, and they can be summarized as follows. Given five words in Italian, the player needs to guess a sixth word that must be related to each one of the five words. Various relationships among words are acceptable. For instance, two words can be related because they are synonyms, antonyms, or because they form a compound word. Similarly, two words can be related because they are included in a proverb or a movie title.

The software system discussed in this paper is an artificial player for this game. A description of the initial design and implementation of the player was presented in [6], and this paper focuses on recent developments of the player. In particular, this paper outlines an enhanced algorithm for the player, and it shows preliminary experimental results that document the improved performance of the player with respect to the performance discussed in [6].

This paper is organized as follows. Section 2 discusses the collection and the processing of the texts used to acquire the needed linguistic knowledge. Section 3 outlines the main characteristics of the algorithm used by the artificial player. Section 4 examines the metrics used by the artificial player and outlines relevant improvements. Section 5 shows preliminary experimental results based on real instances of the game. Section 6 concludes the paper and outlines possible future research directions.

2 Collection and Processing of Texts

Various steps were involved in the design and implementation of the artificial player. The first step concerned the collection of a relevant amount of texts from various Web sources. Then, collected texts were properly processed to remove punctuation marks and words that are not used in the game, such as articles and prepositions. The obtained cleaned text were then further processed to identify pairs of related words by simply grouping words that are close to each other. Each pair of related words was then associated with the number of its occurrences in the cleaned texts. The collection of the pairs of related words is particularly relevant for the construction of the artificial player because the game is based on finding relationships among words. Therefore, the obtained pairs of related words were stored and used as the knowledge base of the artificial player, as discussed in Section 3.

2.1 Collection of Texts

The words needed to play the considered word association game are from the Italian lexicon, and therefore all collected texts are written in Italian. The collected texts are taken from diverse Web sources, and they concern various topics. The diversification of the collected texts ensures that a large amount of diverse pairs of related words can be retrieved and used to play the game. Note that some of the collected texts were authored in Italian, while the others are professional translations from other languages, such as English, French, and Spanish.

The set of collected texts includes thirteen books that were downloaded free of charge from e-book platforms. The set of books includes some books from Italian authors, such as *Pinocchio* and *Zeno's Conscience*, and some books professionally translated from other languages, such as *Alice's Adventures in Wonderland*, *The Little Prince*, and *Don Quixote*.

The set of collected texts also includes a large assortment of Italian texts called *Corpus Paisà* (www.corpusitaliano.it), which is a free corpus of approximately 1.5 GB of text files. Originally, Corpus Paisà was created with the aim of providing authentic and freely available texts to learn Italian. Today, Corpus Paisà is mostly used as a resource for research activities related to the Italian language, and it is commonly considered as a valuable resource to acquire linguistic knowledge for the Italian language.

Besides the books and Corpus Paisà, some texts from the Italian edition of Wikipedia were also downloaded. These texts include the titles of all the Italian articles of Wikipedia (corresponding to nearly 100 MB of text) and 150 full articles written in Italian. The chosen articles concern various topics, such as cooking, science, and music.

Finally, a long list of proverbs, compound words, and idiomatic phrases was also collected from various Web sources. This list is particularly useful to successfully play the game because proverbs, compound words, and idiomatic phrases are often used in the game. As a matter of fact, preliminary empirical observations of game instances aired on television confirm that at least one of the given five words is often related to the correct sixth word through proverbs, compound words, or idiomatic phrases.

2.2 Processing of Texts

The collected texts are processed to remove punctuation marks and words that are not used in the game. Then, the obtained cleaned texts are used to create a set of pairs of related words to form the knowledge base of the artificial player.

First, all punctuation marks are replaced with an uncommon symbol, namely \$, which is used to break sentences. Then, the words that are not useful to play the game are removed. For example, articles and prepositions can be removed from the collected texts without affecting the performance of the artificial player. Actually, articles and prepositions are so common in Italian that the rules of the game prohibit to use them. Finally, conjugated verbs are removed from the collected texts because they are also prohibited by the rules of the game.

After the elimination of punctuation marks and prohibited words, cleaned texts are further processed to obtain the set of pairs of related words used by the artificial player. A *word pair* is a couple of two subsequent words in the same sentence. The identification of word pairs is particularly relevant because the game is based on finding relationships among words. Actually, every instance of the game implicitly involves five word pairs because each one of the given five words must be related to the sixth word. This is the reason why all cleaned texts are parsed to extract word pairs. Note that, while parsing cleaned texts, the number of times that each word pair appears in cleaned texts is stored together with the word pair.

The adopted nomenclature to refer to word pairs and related metrics is as follows. Given a word pair, the first word of the pair is called *token* and the second word is called *related token*. Therefore, a generic word pair is a couple

$$\langle token, related\ token \rangle. \tag{1}$$

Each word pair is associated with its *occurrence*, which is the number of times that the pair is found in cleaned texts. For each (direct) word pair, its *inverse word pair* is formed by exchanging the token with the related token. Note that the occurrence of the inverse word pair is set equal to the occurrence of the direct word pair.

The reason for considering inverse word pairs is as follows. As outlined in Section 3, the artificial player considers the given five words as tokens, and it searches for the sixth word among the corresponding related tokens. Since every word can be either included in the set of five words or it can be the sixth word, inverse pairs are needed to ensure that words can be equally considered as tokens and as related tokens.

The following example is shown to explain the cleaning process and the nomenclature of tokens, related tokens, and occurrences. Let us assume that the sequence of words *acquistare un computer* (Italian for *buy a computer*) is found in the collected texts. The word *un* (Italian for *a*) is an article and, as explained earlier in this section, it is removed during the cleaning process. After removing the article, the two words *acquistare* and *computer* are close to each other in the cleaned text and, therefore, the word pair

$$\langle acquistare, computer \rangle \tag{2}$$

is created. In this case, *acquistare* is considered as the token of the pair and *computer* is considered as the related token of the pair. Let us assume that the word pair (2) is found eight times in all cleaned texts, then the occurrence of the word pair (2) is set equal to eight.

As discussed in Section 3, the word pair (2) ensures that if *acquistare* is one of the given five words, then *computer* is considered as a candidate for the sixth word. However, if *computer* is one of the given five words, then the artificial player should consider *acquistare* as a candidate for the sixth word. Since the two words should be considered as related regardless of which one is the token and which one is the related token, the inverse word pair of (2) is also created. The occurrence of the inverse pair is set equal to eight because it equals the occurrence of the direct pair. The inverse pair allows finding *acquistare* as a candidate for the sixth word when *computer* is one of the given five words.

The word pairs obtained by the collected texts are stored together with their respective occurrences, and they are used by the artificial player to play the game, as discussed in the following section. Currently, the set of word pairs used by the artificial player comprises more than 34,000 tokens, and every token is related with a number of related tokens between 100 and 1,000.

3 The Algorithm of the Artificial Player

This section outlines the algorithm used by the artificial player to play the game. The input of the algorithm is a set of five words, and the computed output is a sixth word that is related with each one of the given five words. An enhanced variation of the algorithm is presented in Section 4.

The algorithm starts by searching each one of the given five words as a token of a word pair in the set of word pairs obtained using the collected texts. Then, assuming that all the given five words are actually found as tokens of word pairs, the algorithm considers the set of the five tokens

$$T = \{t_i\}_{i=1}^{5}. \tag{3}$$

For each token t_i, with $1 \leq i \leq 5$, the set R_i of its related tokens is also considered. All the related tokens of the five tokens are treated as valid candidates for the sixth word, and therefore, the algorithm searches the sixth word in the set R obtained as the union of the five sets R_i, with $1 \leq i \leq 5$. In other words, the sixth word is searched in

$$R = \bigcup_{i=1}^{5} R_i. \tag{4}$$

Note that if some of the given five words are not found as tokens of the available word pairs, then R is computed as the union of the sets of related tokens of the words that were actually found as tokens of word pairs.

Let us denote a generic element of the set R as r_j. Assuming that the word pair $\langle t_i, r_j \rangle$ is found in the set of word pairs, the occurrence $o_{i,j}$ of the pair is immediately available from the collected texts as discussed in the previous section. On the contrary, if the pair $\langle t_i, r_j \rangle$ is not found in the set of word pairs, the occurrence of the pair is conventionally set to zero.

The conventional extension of the occurrence of a word pair is used to define the *frequency* f_j of the generic related token r_j as

$$f_j = \sum_{i=1}^{5} o_{i,j}, \tag{5}$$

which is the sum of the occurrences $\{o_{i,j}\}_{i=1}^{5}$ of all word pairs that include the considered related token.

Note that, according to the definition of frequency, the higher are the occurrences of word pairs, the higher is the frequency. This means that if the word pair $\langle t_i, r_j \rangle$ for a generic related token r_j is found frequently in the cleaned texts, then the value of the frequency f_j is expected to be high. Since the sixth word is expected to appear often as a related token of the given five words, then it can be concluded that the sixth word is also expected to have a high frequency.

Besides the frequency, each word in the set of related tokens R is also associated with a second metrics. The *match* of a generic related token r_j, denoted as m_j, is defined as the number of tokens for which r_j is a related token in the available word pairs. In other words, the match of a generic related token r_j is equal to the number of word pairs $\langle t_i, r_j \rangle$ in the set of word pairs that have different t_i. Possible values for the match of a generic related token r_j are integer numbers from 1 to 5. In particular, if r_j is related to only one of the given five words, then m_j is equal to 1. On the contrary, if r_j is related to all the given five words, then m_j is equal to 5.

The values of frequency and match are evaluated for each related token r_j in R, and they are used altogether to find the best candidate for the sixth word. First, the set of candidates for the sixth word is restricted to the related tokens with the largest match. This guarantees that the sixth word is related to as many words as possible. Then, the sixth word is chosen in the restricted set as the one with the highest frequency. If two ore more related tokens share the same frequency, then the sixth word is randomly chosen among them.

In order to clarify the ideas behind the algorithm used by the artificial player, and to exemplify the computation of the values of frequency and match, let us consider the five tokens in $T = \{t_i\}_{i=1}^{5}$ and the following simple set of word pairs that include nine related tokens

$$\begin{array}{l} \langle t_1, r_1\rangle,\ \langle t_1, r_2\rangle,\ \langle t_1, r_4\rangle \\ \langle t_2, r_1\rangle,\ \langle t_2, r_3\rangle,\ \langle t_2, r_4\rangle,\ \langle t_2, r_7\rangle \\ \langle t_3, r_1\rangle,\ \langle t_3, r_3\rangle,\ \langle t_3, r_4\rangle \\ \langle t_4, r_1\rangle,\ \langle t_4, r_4\rangle,\ \langle t_4, r_5\rangle,\ \langle t_4, r_8\rangle,\ \langle t_4, r_9\rangle \\ \langle t_5, r_1\rangle,\ \langle t_5, r_4\rangle,\ \langle t_5, r_6\rangle \end{array} \tag{6}$$

For each pair, consider the following values of their occurrences

$$\begin{array}{l} o_{1,1} = 3,\quad o_{1,2} = 4,\ o_{1,4} = 4 \\ o_{2,1} = 2,\quad o_{2,3} = 3,\ o_{2,4} = 2\quad o_{2,7} = 8 \\ o_{3,1} = 7,\quad o_{3,3} = 5,\ o_{3,4} = 2 \\ o_{4,1} = 4,\quad o_{4,4} = 2,\ o_{4,5} = 7,\ o_{4,8} = 6,\ o_{4,9} = 6 \\ o_{5,1} = 10,\ o_{5,4} = 1,\ o_{5,6} = 9 \end{array} \tag{7}$$

Let us now consider some of the nine related tokens, and let us evaluate the values of their frequencies and matches. Consider, for example, the related token r_1. Since r_1 is a related token of all the five tokens, its match is $m_1 = 5$. The frequency f_1 of the related token r_1 is

$$f_1 = \sum_{i=1}^{5} o_{i,1} = 26. \tag{8}$$

Let us now consider the related token r_3, which is related only to t_2 and t_3. In this case, the match is $m_3 = 2$, and the frequency is

$$f_3 = \sum_{i=1}^{5} o_{i,3} = 8. \tag{9}$$

Note that the value of f_3 is obtained recalling that $o_{1,3}$, $o_{4,3}$, and $o_{5,3}$ are conventionally set to 0 because r_3 is not related to any of the tokens t_1, t_4, and t_5. Finally, let us consider the related token r_6. Since r_6 is a related token only for the token t_5, its match is $m_6 = 1$ and its frequency is simply $f_6 = o_{5,6} = 9$.

Among the nine related tokens considered in the example, r_1 and r_4 are those with the largest match. As a matter of fact, r_1 and r_4 are related tokens of each one of the five tokens $\{t_i\}_{i=1}^{5}$, so that $m_1 = m_4 = 5$. Since the frequency of r_1 is $f_1 = 26$, and the frequency of r_4 is $f_4 = 11$, the sixth word proposed by the artificial player for this example is the related token r_1.

As discussed in [6], in order to test the validity of the proposed algorithm, 100 random instances of the game were taken from the instances that actually aired on television. The artificial player was able to find the correct sixth word for 24 of the considered game instances. Even if this success rate may seem low, it is worth noting that human players often fail in finding the correct sixth word, and the

expected success rate of human players is low. Regarding other artificial players, to the best of our knowledge, only two other players have been proposed to play the considered game, namely *Il Mago della Ghigliottina* [17] and *GUiLlotine gLovE replayer* (*GUL.LE.VER.*) [9]. The success rate of the first player is 68.6%, and the success rate of the second player is 26%. Hence, the success rate of the first player is significantly higher than the success rate obtained using the algorithm outlined in this section, while the success rate of the second player is comparable with the success rate obtained using the proposed algorithm. In order to improve the performance of the proposed algorithm, further refinements are discussed in the next section, and improved results are presented in Section 5.

4 The Improved Algorithm

Various tests on the algorithm described in the previous section were performed to possibly identify relevant improvements. During such tests, it was noticed that word pairs associated with high occurrences can have a negative impact on the performance of the artificial player. In order to better understand the role of these word pairs, let us consider the following instance of the game:

- *Punto* (Italian for *point*)
- *Saggio* (Italian for *essay*)
- *Arte* (Italian for *art*)
- *Occhio* (Italian for *eye*)
- *Giudizio* (Italian for *judgment*)

The correct sixth word of this instance of the game is *critico* (Italian for *critical*). As a matter of fact, one can say in Italian: *punto critico* (Italian for *critical point*); *saggio critico* (Italian for *critical essay*); *critico d'arte* (Italian for *art critic*); *occhio critico* (Italian for *critical look*); and *giudizio critico* (Italian for *critical assessment*). However, the sixth word proposed by the artificial player using the algorithm described in the previous section is *riferimento* (Italian for *reference*). Note that, in Italian, the first word of the previous list is commonly used together with the proposed sixth word. As a matter of fact, *punto di riferimento* (Italian for *point of reference*) is a common phrase in Italian.

In order to understand the reason why the artificial player fails to find the correct sixth word, let us denote the given five words as $\{t_i\}_{i=1}^{5}$, and let us denote the word *critico* as r_1 and the word *riferimento* as r_2. The occurrences for the word pairs that include the related token r_1 (namely, the word *critico*) are:

$$o_{1,1} = 121,\ o_{2,1} = 20,\ o_{3,1} = 178,\ o_{4,1} = 31,\ o_{5,1} = 50. \tag{10}$$

Instead, the occurrences for the word pairs that include the related token r_2 (namely, the word *riferimento*) are:

$$o_{1,2} = 652,\ o_{2,2} = 3,\ o_{3,2} = 14,\ o_{4,2} = 1,\ o_{5,2} = 3. \tag{11}$$

Note that all the occurrences $\{o_{i,1}\}_{i=1}^{5}$ of the word pairs that include the related token r_1 are greater than 0. Therefore, the match m_1 is equal to 5. Similarly,

all the occurrences $\{o_{i,2}\}_{i=1}^{5}$ of the word pairs that include the related token r_2 are greater than 0. Therefore, the match m_2 is also equal to 5. Since both words have the same match, let us consider their frequencies. The frequency f_1 of the related token r_1 is

$$f_1 = \sum_{i=1}^{5} o_{i,1} = 400, \tag{12}$$

while the frequency f_2 of the related token r_2 is

$$f_2 = \sum_{i=1}^{5} o_{i,2} = 673. \tag{13}$$

Since the frequency of the related token r_2 is higher than the frequency of the related token r_1, the player chooses the wrong sixth word, which is the related token r_2 (namely, the word *riferimento*).

Let us analyze in detail the occurrences that contribute to the frequencies to better understand the reasons for the failure. For each one of the given five words, let us compare the values of the occurrences

$$\begin{aligned} o_{1,1} &= 121 < 652 = o_{1,2} \\ o_{2,1} &= 20 > 3 = o_{2,2} \\ o_{3,1} &= 178 > 14 = o_{3,2} \\ o_{4,1} &= 31 > 1 = o_{4,2} \\ o_{5,1} &= 50 > 3 = o_{5,2} \end{aligned} \tag{14}$$

Note that token t_1 is more often paired with the related token r_2 than with the related token r_1. As a matter of fact, the occurrence of the pair $\langle t_1, r_2 \rangle$ is $o_{1,2} = 652$, while the occurrence of the pair $\langle t_1, r_1 \rangle$ is $o_{1,1} = 121$. On the contrary, the remaining four tokens are more often paired with related token r_1 than with related token r_2. Moreover, the word pairs that include these four tokens and the related token r_2 have very low occurrences, which suggests that these four tokens are rarely used together with the related token r_2. At the opposite, the word pairs that include the same four tokens and the related token r_1 have occurrences greater than 20, which suggests that these four tokens are used quite often together with the related token r_1.

These considerations hint that the correct sixth word of the considered instance of the game is the related token r_1, as it is indeed the case, since it is used often with all the given five words. However, the only high value of the occurrences of the word pairs that include the related token r_2 (namely, $o_{1,2}$) causes the frequency f_2 to be greater than the frequency f_1, which causes the player to choose the wrong sixth word.

In order to overcome the problems that caused the player to fail in this game instance, a threshold on the occurrences of word pairs is introduced to lower the impact of high frequencies. The occurrences that exceed the threshold are set equal to the threshold, so that frequencies are kept within a known range.

The threshold was set empirically in the current implementation of the artificial player. The values of the threshold between 10 and 30 were considered and, after an extensive experimental campaign, the threshold was set to 13. As a matter of fact, this value corresponds to the maximum success rate of the artificial player for the considered game instances.

In order to better understand how the threshold is used, let us reconsider the example discussed earlier in this section. Let us first reconsider the occurrences of the word pairs that include the related token r_1 after the introduction of the threshold. Since all the occurrences for the related token r_1 shown in (10) are greater than the threshold, they are all set equal to the threshold

$$o_{1,1} = o_{2,1} = o_{3,1} = o_{4,1} = o_{5,1} = 13. \tag{15}$$

Let us then reconsider the occurrences for the word pairs that contain the related token r_2 after the introduction of the threshold. From (11) it can be observed that only $o_{1,2}$ and $o_{3,2}$ are greater than the threshold and, therefore, they are set equal to the threshold. The occurrences for the word pairs that include the related token r_2 after the introduction of the threshold are

$$o_{1,2} = o_{3,2} = 13,\ o_{2,2} = 3,\ o_{4,2} = 1,\ o_{5,2} = 3. \tag{16}$$

These changes on the occurrences do not influence the values of the match of the two related tokens r_1 and r_2, and they both remain equal to 5. Instead, the new occurrences have an impact on the frequencies f_1 and f_2. The frequency f_1 of the related token r_1 evaluated after the introduction of the threshold is

$$f_1 = \sum_{i=1}^{5} o_{i,1} = 65. \tag{17}$$

The frequency f_2 of the related token r_2 evaluated after the introduction of the threshold is

$$f_2 = \sum_{i=1}^{5} o_{i,2} = 33. \tag{18}$$

Observe that, according to these new frequencies, the player proposes the correct sixth word, namely the related token r_1. As a matter of fact, the new frequency of the related token r_1 is now higher than the frequency of the related token r_2.

5 Experimental Results

The current version of the artificial player uses the modified algorithm that employs a threshold on occurrences to limit the problems caused by related tokens with high frequencies, as discussed in the previous section. It is expected that the adoption of the modified algorithm can improve the success rate of the player with respect to its initial performance because a preliminary informal analysis of the game instances in which the player failed suggests that wrong sixth words were often caused by the presence of related tokens with high frequencies.

The following is an example of an instance of the game in which the current version of the player found the correct sixth word:

- *Originale* (Italian for *original*)
- *Mattino* (Italian for *morning*)
- *Segretaria* (Italian for *secretary*)
- *Curare* (Italian for *treat*)
- *Straordinaria* (Italian for *extraordinary*)

The player was able to correctly identify the correct sixth word, which is *edizione* (Italian for *edition*). As a matter of fact, one can say in Italian: *edizione originale* (Italian for *original edition*); *edizione del mattino* (Italian for *morning edition*); *segretaria di edizione* (Italian for *script girl*); *curare un'edizione* (Italian for *edit a publication*); and *edizione straordinaria* (Italian for *special edition*).

In some cases, the player cannot find the correct sixth word. Indeed, it is worth noting that, in such cases, it returns a word that is still related to the given five words. For example, consider the following instance of the game:

- *Volo* (Italian for *flight*)
- *Dare* (Italian for *give*)
- *Mezzi* (Italian for *means*)
- *Ente* (Italian for *society*)
- *Intervento* (Italian for *intervention*)

The correct sixth word is *assistenza* (Italian for *assistance*). As a matter of fact, one can say in Italian: *assistenza di volo* (Italian for *flight assistance*); *dare assistenza* (Italian for *give assistance*); *mezzi di assistenza* (Italian for *means of assistance*); *ente di assistenza* (Italian for *rescue society*); and *intervento di assistenza* (Italian for *assistance intervention*).

In this example, the player does not return the correct sixth word, and it returns the word *controllo* (Italian for *control*). This word is not the correct sixth word, but it is strictly related to all the given five words. As a matter of fact, one can say in Italian: *controllo del volo* (Italian for *flight control*); *dare il controllo* (Italian for *give the control*); *mezzi di controllo* (Italian for *means of control*); *ente di controllo* (Italian for *control unity*); and *intervento di controllo* (Italian for *control intervention*). Note that the identified relationships that link the the word *controllo* with each one of the given five words are all correct, and they are commonly used in Italian.

The current version of the player was tested using the same 100 instances of the game that were considered in [6] to compare the performance of the original algorithm, as described in Section 3, with the performance of the modified algorithm that uses the threshold. Using the modified version of the algorithm, the new success rate of the player is 47%, which ensures that for nearly half of the considered instances of the game the correct sixth word is proposed. Therefore, it can be concluded that the use of the threshold for the occurrences leads to a significant increase of the success rate, which is (almost) doubled with respect to the success rate of the previous version of the player.

In addition, note that the sixth word proposed by the current version of the artificial player is strongly related with at least four (and sometimes five) of the given five words in 28 of the 100 instances of the game, even if the sixth word is not actually correct. Therefore, in 75 of the 100 instances of the game, the player returns a sixth word that is correct (47 cases) or that is strongly related with the given five words (28 cases). This result is encouraging and further improvements are planned to increase the success rate of the player.

Finally, it is worth noting that the adoption of the modified algorithm ensures that the current version of the artificial player can outperform *GUL.LE.VER.*, which exhibits a success rate equal to 26%. On the contrary, the adoption of the modified algorithm is not sufficient to obtain a success rate better than the success rate of *Il Mago della Ghigliottina*, which equals 68.6%. Finally, note that the mentioned success rates were not obtained using a common set of game instances, and therefore their relevance to compare the players is limited.

6 Conclusion

This paper discussed the design of an artificial player for a specific word association game. The first step in the construction of the artificial player involved the collection of sufficient texts to acquire needed linguistic knowledge. The collected texts were processed to extract word pairs, their occurrences, and two other metrics called frequency and match. The collected word pairs and their metrics form the knowledge base used by the player. Note that a suitable threshold on the values of the occurrences was defined to improve the success rate of the player. The player was tested on 100 instances of the game, and its success rate was 47%. Future developments of the player include the extension of the collected texts to include new word pairs. In addition, the use of additional metrics is planned to increase the success rate of the player.

References

1. Abacha, A.B., Zweigenbaum, P.: MEANS: A medical question-answering system combining NLP techniques and Semantic Web technologies. Information Processing & Management **51**(5), 570–594 (2015)
2. Ahmed, K., Tazi, N., Hossny, A.H.: Sentiment analysis over social networks: An overview. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics. pp. 2174–2179 (2015)
3. Basile, P., Lovetere, M., Monti, J., Pascucci, A., Sangati, F., Siciliani, L.: Ghigliottin-AI @ EVALITA 2020: Evaluating artificial players for the language game "La Ghigliottina". In: Basile, V., Croce, D., Maro, M.D., Passaro, L.C. (eds.) Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020). CEUR Workshop Proceedings, vol. 2765. RWTH Aachen (2020)
4. Belinkov, Y., Glass, J.: Analysis methods in neural language processing: A survey. Transactions of the Association for Computational Linguistics **7**, 49–72 (2019)

5. Chater, N., Manning, C.D.: Probabilistic models of language processing and acquisition. Trends in Cognitive Sciences **10**(7), 335–344 (2006)
6. Coffrini, A., Monica, S., Bergenti, F.: On the design of an artificial player for a popular word game. In: Italian Conference on Computational Logic (CILC 2021). CEUR Workshop Proceedings, vol. 3002, pp. 122–132. RWTH Aachen (2021)
7. Dahl, V.: Natural language processing and logic programming. The Journal of Logic Programming **19-20**, 681–714 (1994)
8. Dorr, B.J., Jordan, P.W., Benoit, J.W.: A survey of current paradigms in machine translation. Advances in Computers **49**, 1–68 (1999)
9. de Francesco, N.: GUL.LE.VER @ GhigliottinAI: A glove based artificial player to solve the language game "La Ghigliottina". In: Basile, V., Croce, D., Maro, M.D., Passaro, L.C. (eds.) Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020). CEUR Workshop Proceedings, vol. 2765. RWTH Aachen (2020)
10. Këpuska, V., Bohouta, G.: Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). In: IEEE 8$^{\text{th}}$ Annual Computing and Communication Workshop and Conference (CCWC). pp. 99–103 (2018)
11. Kowsari, K., Jafari, M., Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., D. Brown, D.: Text classification algorithms: A survey. Information **10**(4) (2019)
12. Lewis, D.D., Jones, K.: Natural language processing for information retrieval. Communications of the ACM **39**(1), 92–101 (1996)
13. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
14. Mooney, R.J.: Inductive logic programming for natural language processing. In: Muggleton, S. (ed.) Inductive Logic Programming. pp. 1–22. Springer (1997)
15. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R., Cota, G.: Probabilistic logic programming on the Web. Software Practice and Experience **46**(10), 1381–1396 (2016)
16. Riguzzi, F., Lamma, E., Alberti, M., Bellodi, E., Zese, R., Cota, G.: Probabilistic logic programming for natural language processing. In: Chesani, F., Mello, P., Milano, M. (eds.) AI*IA Workshop on Deep Understanding and Reasoning: A Challenge for Next-generation Intelligent Agents. CEUR Workshop Proceedings, vol. 1802, pp. 30–37. RWTH Aachen (2016)
17. Sangati, F., Pascucci, A., Monti, J.: "Il Mago della Ghigliottina" @ GhigliottinAI: When linguistics meets artificial intelligence. In: Basile, V., Croce, D., Maro, M.D., Passaro, L.C. (eds.) Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020). CEUR Workshop Proceedings, vol. 2765. RWTH Aachen (2020)
18. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine **13**(3), 55–75 (2018)