

GUL.LE.VER @ GhigliottinAI: A Glove based Artificial Player to Solve the Language Game “La Ghigliottina”

Nazareno De Francesco

Turin, Italy

nazarenodef francesco@gmail.com

Abstract

The paper describes GUL.LE.VER, GUiLlottine gLoVe resOlVER, a Glove based system developed to solve the game “La Ghigliottina” which participated in the Evalita 2020 (Basile et al., 2020) task Ghigliottin-AI. The system described positioned #2, with 0.26 of Precision and 0.46 R@10, more than one guillotine is solved every four games, achieving results comparable to human players. The system proved to solve a different kind of guillotines compared to the first classified system ‘Il Mago della ghigliottina’ (Sangati et al., 2018). An approach based on these two kinds of systems may result in a boost in this field of research.

1 Introduction

“La Ghigliottina” is a language game in which the gamer has to guess the word that is most correlated with other five words, named clues. An example is the guillotine “Certificate, Son, Tragedy, Star, Venus”, the solution, in this case, is “Birth”. The game structure is simple, but some complex steps are required in order to solve a guillotine. The gamer’s background knowledge has to be rich enough to cover a large variety of fields, such as common culture, proverbs, etc. Additionally, the gamer’s reasoning has to be fast enough to give the solution in less than a minute. In this article, an artificial player for The Guillotine has been built: GUL.LE.VER, the GUiLlottine gLoVe resOlVER. It’s mostly based on the Glove (Pennington et al., 2014) vector representation of the words present in a large collected dataset, containing the

Italian Wiktionary, Wikiquote, Wikipedia (only titles), the Italian Collocations Dictionary (Tiberi, 2018), and resources scraped on the web containing Italian polirematism, proverbs and songs titles. The Glove algorithm was chosen for its intrinsic power in capturing the co-occurrence correlation between two words that are not synonyms, due to the co-occurrence matrix that the algorithm builds before the training. Other similar algorithms, such as Word2Vec, do not have this characteristic. The solution for the guillotine is searched in the vector space near the clues, obtaining a list of solution candidates. This list is descending reordered using a hybrid function composed by two parts: one part is based on the Pointwise Mutual Information (Sangati et al., 2018), the other one is based on the weighted sum of the cosine similarity between the solution candidate and the clues, in which the weight is the normalized Inverse Document Frequency of the single clue in the corpus (solutions that are correlated with the rarest clues are more important than others).

2 Related work

In order to find the solution for a particular game, a player needs to know the rules that regulate the game and, based on the game type, he also needs to possess a background knowledge that helps him in finding the solution. We can distinguish two types of games based on these two requirements: closed-world games and open-world games. Closed-world games provide the player with all the knowledge necessary for playing the game (like chess), otherwise open-world games can not be solved without additional knowledge. A particular type of open-world games is represented by language games in which word meanings play a central role (like crosswords) (Littman et al., 2000). The challenge in this type of games is found in the intrinsic ambiguity of natural language, in which a word with different meanings

Copyright ©2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

may be connected with a word or with another based on its interpretation, which is heavily dependent on the context. The needs of reasoning skills and a background knowledge to solve this type of games is the main reason for which they have attracted the attention of researchers. In this field a language game like Who Wants to be a Millionaire?, in which the player must have a wide background knowledge in order to answer a series of multiple-choice questions, has been shown to be solved mining the web, with the same performance of a human player (Lam et al., 2003). Extract common sense human knowledge from Wikipedia articles is another proposed solution that is able to challenge a human player (Molino et al., 2013). In the same category of open-world language games is set “La Ghigliottina”, an Italian quiz show in which five words are submitted to the player as clues and he has to find the unique word that is correlated with all the clues. In order to find this hidden associations between clues and solution, a human player must possess a wide background knowledge and he has to be able to perform a complex task of reasoning on it in order of finding correlations between different word meanings in different contexts. In literature, a proposed solution to this problem is OT-THO (On the Tip of my THOught) (Semeraro et al., 2009; Semeraro et al., 2012) which achieved performance similar to humans using a network representation of the background knowledge and a spreading algorithm to find the solution. “Il mago della Ghigliottina” (Sangati et al., 2018), based on a co-occurrence matrix obtained from a corpus of patterns mined on web scraped resources and the Pointwise Mutual Information as measure of word correlation, achieved super-human performance. In order to explore a new way to solve this game, GUL.LE.VER is built using similar web scraped resources, Glove algorithm for word representation and a custom word correlation measure based on cosine similarity and inverse document frequency (idf).

2.1 Linguistic Resources

Based on the previous related works, the linguistic resources involved in this project are:

- The italian Wikipedia, only titles, downloaded via WikiExtractor (Attardi, 2012).
- The italian Wiktionary, downloaded via WikiExtractor.

- The italian Wikiquote, downloaded via WikiExtractor.
- The “Dizionario delle Collocazioni” (Tiberi, 2018) containing 200.000 combinations of words in Italian.
- A collection of 369 italian proverbs (Dige, 2016)
- A collection of more than 3700 common sayings, scraped on different websites .
- A collection of more than 6000 italian polirematics, scraped on different websites. 678 italian song titles (Paldo, 2013).

These corpora was preprocessed, using tokenization (single words only) and punctuation removing, obtaining a unique corpus to feed the Glove algorithm.

3 System description

The system can be described in 6 steps:

1. **Glove training:** the corpus is used to train a Glove model that represents the words in corpus in a compact vector space. The best parameters used to train the algorithm are empirically obtained: Vector_size 600, Vocab_min_count 200, Window 10, Iteration 50, Xmax 10, Alpha 0.75, Eta 0.05. They proved to be the best parameters for the Evalita training dataset.
 - (a) The Vocab_min_count setted to 200 corresponds to a vocabulary of 28873 unique words represented.
2. **Setting search space, ‘looking into neighbors’:** starting from the clues, a list of $5 \times M$ solution candidates is built finding the M most similar words to each clue in order of cosine similarity. The result search space is significantly smaller than the entire vocabulary. This solution gives faster and more accurate results than the exhaustive search on the vocabulary.
3. **Filtering candidates:** the solution candidates list is filtered by:
 - (a) removing all words except Nouns and Adjective (verbs and conjunctions are never solutions for the game).

- (b) removing Adjectives too, if one of the clues is already an adjective.
- (c) removing words that are present in a custom blacklist and not present in a custom whitelist. The blacklist contains lists of non-ambiguous proper nouns, cities names, foreign words, etc.

4. **Reordering, the cosine based score function:** the filtered list is reordered in descending order based on the following formula:

$$(a) \quad F(t) = \frac{\alpha}{\beta} \times \frac{(\sum_{i=1}^n (\cos(s, c_i) \times nIDF(c_i))) \div N}{1 + \sigma}$$

- (b) The first part of the formula are two arbitrary weights that can be manually set up in order to give more importance to the weighted mean of the cosines or the standard deviation.
- (c) The second part of the formula has: as a numerator, the weighted mean of the cosines between the solution candidate and clues. The weight is the normalized Inverse Document Frequency of the clue in the corpus. This gives a boost to the solutions that are correlated to the most rare clues, starting with the assumption that a rare clue has less possible meaningful combination in the corpus, so a candidate solution highly correlated with that may be corresponding to the solution of the game. As a denominator, there is the standard deviation of the cosines (not weighted). This is intended to give a boost to the solutions that are correlated with all the clues in a balanced way, avoiding such solutions that are very highly correlated to a clue but not to the others.
- (d) A cosine threshold can be set in order to discard cosines that are lower than that, penalizing those that are too low. In this case, the cosines lower than zero are penalized automatically to -1 (the lower bound of the cosine similarity function), avoiding solutions that have opposite meaning compared to the clues.

5. **Solution certainty:** if the difference between the first and the second score result is more than a Solution certainty threshold, the first

candidate is proposed as a solution for the game. If not, the candidate list is reordered again using the Pointwise Mutual Information (pmi), calculated on the corpus proposed, as the third multiplied part of the formula $F(t)$. This helps in the situation in which the real solution is between the first three/four results before the final reordering.

6. **Solution proposed:** the first candidate of the reorder list is proposed as a solution for the game.

4 System implementation

The system is entirely implemented in Python 3.7. The principal libraries used are:

- gensim (Řehůřek et al., 2011)
- spacy-stanza (Peng et al., 2020)
- nltk (Loper et al., 2002)
- numba (Lam et al., 2015)
- numpy

The Glove algorithm (Pennington et al., 2014) is the C implementation provided by Stanford and the model obtained is loaded through gensim. A Flask python server was setup to respond to the evaluation requests via API.

5 Results

The table 1 shows the results obtained by GUL.LE.VER on the Evalita-GhigliottinAI Dev dataset and Test dataset.

Set	Size	Precision	R@5	R@10	R@100
Dev	300 pt	0,32	0,44	0,51	0,69
Test	350 pt	0,27	0,38	0,46	0,62
Dev*	300 pt	0,32	0,37	0,44	0,68
Test*	350 pt	0,28	0,40	0,48	0,65

Table 1: Results on the TEST and DEV set. Evaluations are the Precision (number of correct solutions / the number of guillotines) and R@5, R@10, R@100 (recall at 5, 10, 100).

The 5% difference in the Precision between the Dev set and the Test set is in part due to a blacklist overfitted on the dev set. 10 solutions are found to be erroneously in the blacklist. Putting them in the whitelist gives a more balanced result, slightly higher for the Test dataset and a little lower for the Dev dataset, as shown by the Dev* and Test*

rows. The system seems biased by solutions that are very frequent in corpus: it responded ‘uno’ 23 times and none of them were the correct solution. Another example: it responded ‘senza’ 9 times, only one time guessing the correct solution. An important point to underline is that almost half of the solutions are found in the first 10 proposed results, with approximately 40% of them in the first 5, with 57% and 56% in the first 20 for Test and Dev set respectively (not reported in Table 1). This seems very promising for future upgrading, finding a better way to clean the candidates list and/or fine tuning the reorder function.

The last point of analysis is a brief comparison between GUL.LE.VER and ‘Il Mago della Ghigliottina’. Selecting only the resolved guillotines from the Test Set and submitting them to the Telegram version of ‘Il Mago della Ghigliottina’, 18 guillotines were not resolved by Sangati et al., 2018 system. These guillotines (in table 2) represent 4.8% of the total test guillotines and can be resolved only by the proposed solution.

Clue1	Clue2	Clue3	Clue4	Clue5	Gullever	Mago
fazzoletto	alto	allungare	braccio	osso	collo	naso
studio	vestire	notte	povero	montalbano	giovane	panni
paradiso	bordo	sud	nino	casa	benvenuti	angolo
vecchio	cavallo	end	soda	conquista	west	polenta
mettere	moto	collo	baffi	brutta	piega	giro
mamma	scena	scuola	re	crudo	nudo	gonna
volo	dare	mezzi	ente	intervento	assistenza	pronto
idee	bocca	isola	sottomarino	spock	vulcano	porto
finestra	vestire	volto	chiara	chiaro	scuro	luna
pari	pace	sosta	motivo	famiglia	senza	apparente
cura	pietre	alto	azzurro	occhi	sole	cielo
acqua	onda	capo	sempre	essere	verde	andata
città	tv	vita	oggi	gioco	ragazzi	frenetico
bandiera	coltelli	caponi	marx	italia	fratelli	regno
dare	camera	consiglio	misura	stato	sicurezza	deciso
regola	parole	alberi	perfetto	fa	tre	quadrato
leggero	barba	togliere	viso	inganno	trucco	velo

Table 2: Guillotine resolved by GUL.LE.VER and not resolved by Il Mago della Ghigliottina.

6 Conclusion and future work

In this paper is described GUL.LE.VER, an artificial player to solve the game “La Ghigliottina”, based on the Glove word vector algorithm, whose power is its co-occurrence matrix reduction. An hybrid pmi approach is proposed as fallback in case of uncertainty. The system achieved good performance in the Evalita2020 task, with results comparable to humans. A comparison made with the solutions proposed by the best system, the Sangati et al., 2018 ‘Il Mago della Ghigliottina’, suggests that the proposed approach is capable of solving different kinds of guillotines compared to

the first one. As future work, a even more hybrid solution between these two kinds of approaches should be implemented, hoping it will be result in a boost in this field of research.

References

- Paola Tiberii, *Dizionario delle collocazioni: le combinazioni delle parole in italiano*, Zanichelli, 2018.
- M. L. Littman, Review: *Computer language games*, in Proc. Comput. Games, 2nd Int. Conf., Rev. Papers, T. A. Marsland and I. Frank, Eds., 2000, vol. 2063, pp. 396404, ser. LNCS, Springer.
- S. K. Lam, D. M. Pennock, D. Cosley, and S. Lawrence, *Mining the web to play Who wants to be a millionaire?*, in Proc. 19th Conf. Uncert. Artif. Intell., C. Meek and U. Kjrulff, Eds.2003, pp. 337345.
- P. Molino, P. Basile, C. Santoro, P. Lops, M. de Gemmis, and G. Semeraro, M.Baldoni, C. Baroglio, G. Boella, and R. Micalizio, Eds., *A virtual player for who wants to be a millionaire? based on question answering*, in Proc. AI*IA 2013: Adv. Artif. Intell.13th Int. Conf. Italian Assoc. Artif. Intell., 2013, vol. 8249, pp. 205216, ser. Lecture Notes in Comput. Sci..
- G. Semeraro, M. de Gemmis, P. Lops, and P. Basile, *An artificial player for a language game*, IEEE Intell. Syst., vol. 27, no. 5, pp.3643, Sep.Oct. 2012.
- G. Semeraro, P. Lops, P. Basile, and M. de Gemmis, *On the tip of my thought: Playing the Guillotine game*, in Proc. 21st Int. Joint Conf. Artif. Intell., 2009, pp. 15431548, Morgan Kaufmann.
- Basile, Pierpaolo and de Gemmis, Marco and Lops, Pasquale and Semeraro, Giovanni *Solving a complex language game by using knowledge-based word associations discovery*. IEEE Transactions on Computational Intelligence and AI in Games 8.1 (2014): 13-26.
- Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro. 2020. Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online. CEUR.org.
- Sangati Federico, Antonio Pascucci, and Johanna Monti. *Exploiting Multiword Expressions to solve “La Ghigliottina”*. Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018). Vol. 2263. Accademia University Press, 2018.
- Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. *Glove: Global vectors for word representation*. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

- Attardi, Giuseppe. *WikiExtractor*. (2012).
- Antonio Dige. 2016. *Raccolta di proverbi e detti italiani*. <http://web.tiscali.it/proverbiitaliani>.
- Paldo, Alessandro. 2013. <http://alessandro-paldo.blogspot.com/2013/10/1-10-1.html>
- Lam, Siu Kwan, Antoine Pitrou, and Stanley Seibert. *Numba: A llvm-based python jit compiler*. Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. 2015.
- Řehůřek, Radim, and Petr Sojka. *Gensim—statistical semantics in python*. Retrieved from genism.org (2011).
- Loper, Edward, and Steven Bird. *NLTK: the natural language toolkit*. arXiv preprint [cs/0205028](https://arxiv.org/abs/cs/0205028) (2002).
- Qi, Peng and Zhang, Yuhao and Zhang, Yuhui and Bolton, Jason and Manning, Christopher D *Stanza: A python natural language processing toolkit for many human languages*. arXiv preprint [arXiv:2003.07082](https://arxiv.org/abs/2003.07082) (2020).