Literary Natural Language Generation with Psychological Traits

Luis-Gil Moreno-Jiménez, Juan-Manuel Torres-Moreno, Roseli Wedemann

Literary Natural Language Generation with Psychological Traits*

Luis Moreno-Jiménez[1,4], Juan-Manuel Torres-Moreno[1,3][0000−0002−4392−1825], and Roseli Wedemann[2][0000−0001−7532−3881]

[1] LIA/Avignon Université, France
luis-gil.moreno-jimenez@alumni.univ-avignon.fr
juan-manuel.torres@univ-avignon.fr
[2] Universidade do Estado do Rio de Janeiro, Brazil
roseli@ime.uerj.br
[3] Polytechnique Montréal, 69121 (Québec), Canada
[4] Universidad Tecnológica de la Selva, Mexico

Abstract. The area of Computational Creativity has received much attention in recent years. In this paper, within this framework, we propose a model for the generation of literary sentences in Spanish, which is based on statistical algorithms, shallow parsing and the automatic detection of personality features of characters of well known literary texts. We present encouraging results of the analysis of sentences generated by our methods obtained with human inspection.

Keywords: Natural language processing · Generation of literary texts · Psychological traits.

1 Introduction

Automatic Text Generation (ATG) is a task that has been widely studied by researchers in the area of Natural Language Processing (NLP) [13, 20–23]. Results from several investigations have presented very encouraging results that have allowed the establishment of progressively more challenging objectives in this field. Currently, the scope of automatic generation of text is being expanded and there is much recent work that aims at generating text related to a specific domain [8, 10, 18]. In this research scenario, algorithms have been developed, oriented to such diverse purposes as creating chat-bots for customer service in commercial applications, automatic generation of summaries for academic support, or even developing generators of literary poetry, short stories, novels, plays and essays [3, 17, 25].

The creation of a literary text is particularly interesting and challenging, when compared to other types of ATG, as these texts are not universally and perpetually perceived. Each individual has a personal conception of a literary work and perceives it in a different way. Furthermore, this perception can also

* Supported partially by Conacyt (Mexico) and Université d'Avignon (France).

vary depending on the reader's mood. It can thus can be assumed that literary perception is subjective and from this perspective, it is difficult to ensure that text generated by an algorithm will be perceived as literature. To reduce this possible ambiguity regarding literary perception, we consider that literature is regarded as a text that employs a vocabulary which may be largely different from that used in common language and that it employs various writing styles, such as rhymes, anaphora, etc. and figures of speech, in order to obtain an artistic, complex, possibly elegant and emotional text [11]. This understanding gives us a guide for the development of our model whereby literary sentences can be generated.

We present here a model for the generation of literary text (GLT) which is guided by psychological characteristics of the personality of characters in literature. The model is based on the assumption that these psychological traits completely determine a person's speech. We thus generate literary sentences based upon a situation or context, and also on different psychological traits. For this, we have been inspired by various works which we discuss in Section 2. It is thus possible to perform an analysis of the personality of an individual through their writing, considering parts of speech such as verbs, adjectives, conjunctions, and the spinning of words or concepts, as well as other characteristics.

In Section 2, we present a review of the main literature that has addressed topics related to ATG, with focus on those that proposed methods and algorithms integrated into this work. We describe the corpus used to train our models in Section 3. Section 4 describes in detail the methodology followed for the development of our model. In Section 5, we show some experiments, as well as the results of human evaluations of the generated sentences. Finally, in Section 6 we present our conclusions.

2 Related Work

The task of ATG has been widely addressed by the research community in recent years. In [21], Szymanski and Ciota presented a model based on Markov sequences for the stochastic generation of text, where the intention is to generate grammatically correct text, although without considering a context or meaning. Shridaha et al. [20] present an algorithm to automatically generate descriptive summary comments for Java methods. Given the signature and body of a method, their generator identifies the content for the summary and generates natural language text that summarizes the method's overall actions.

In recent years, work has been developed for GLT, with an artistic, literary approach, as discussed in the Introduction. The works of Riedl and Young [17] and Clark, Ji and Smith [3] propose stochastic models, based on contextual analyzes, for the generation of fictional narratives (stories). Zhang and Lapata [25] use neural networks and Oliveira [12, 13] uses the technique of *canned text* for generating poems. Some research has achieved the difficult task of generating large texts, overcoming the barrier of phrases and paragraphs, such as the MEXICA project [15]. MEXICA employs a computational model based on

the *engagement-reflection*, creativity paradigm that generates narratives about pre-Columbian Aztec mythology [18].

Personality analysis is a complicated task that can be studied from different perspectives (see [24, 19] and references therein). Recent research has investigated the relation between the characteristics of literary text and personality, which can be understood as the complex of the behavioural, temperamental, emotional and mental attributes that characterise a unique individual [6, 7]. In [7], a type of personality is detected from the analysis of a text, using an artificial neural network (ANN), which classifies text into one of five personality classes considered by the authors (Extroversion, Neuroticism, Agreeableness, Conscientiousness, Openness). Some characteristics that are considered by the authors are writing styles and relationships between pairs of words.

3 Corpus

We have built two corpora in Spanish consisting of the main works of Johann Wolfgang von Goethe and Edgar Allan Poe, called **cGoethe** and **cPoe**, respectively. These corpora are analyzed and used to extract information about the vocabulary used by these authors. We later chose an important work for each author where the emotions and feelings, *i.e. psychological traits*, of main characters are easily perceived by readers. For Johann Wolfgang von Goethe, we have selected the novel *The Sorrows of Young Werther* [5] and for Edgar Allan Poe, we select the story *The Cask of Amontillado* [16]. The two corpora generated from these literary works were used to extract the sentences, that were later used as a basis for the generation of new sentences, as we describe in section 4.1. We also use the corpora 5KL described in [11] for the final phase of the sentence generation procedure described in 4.2.

To build the **cGoethe** and **cPoe** corpora, we processed each constituent literary work (originally found in heterogeneous formats), creating a single document for each corpus, encoded in *utf8*. The processing of each constituent work consisted of automatically segmenting the phrases into regular expressions, using a program developed in PERL 5.0, to obtain one sentence per line in each corpus, **cGoethe** and **cPoe**.

From the segmented phrases in **cGoethe** and **cPoe**, we selected only those that belong to the works *The Sorrows of Young Werther* (set 1) and *The Cask of Amontillado* (set 2). From sets 1 and 2, we manually extracted phrases that we considered to be very *literary*, to form two new corpora, **cWerther** and **cCask**, respectively. In this step, we chose phrases with complex vocabulary, a directly expressed message and some literary styles like rhymes, anaphoras, etc. Table 1 shows basic statistical information of the **cGoethe** and **cPoe** corpora. Table 2, shows similar information for **cWerther** and **cCask**. Table 3 shows statistical information of the 5KL corpus. The 5KL corpus contains approximately 4 000 literary works, from various authors and different genres, and is extremely useful, as it consists of an enormous vocabulary, which forms a highly representative set to train our Word2vec based model, described in Section 4.

Table 1: Corpora formed by main works of each author. K represents one thousand and M represents one million.

	Sentences	**Tokens**	**Characters**
cPoe	5 787	70 K	430 K
Average for sentence	–	12	74
cGoethe	19 519	340 K	2 M
Average for sentence	–	17	103

Table 2: Corpora of literary phrases of one selected work.

	Sentences	**Tokens**	**Characters**
cCask	141	1 856	11 469
Average for sentence	–	13	81
cWerther	134	1 635	9 321
Average for sentence	–	12	69

4 Model

We now describe our proposed model for the generation of literary phrases, which is an extension of Model 3 presented in [11]. The model consists of two phases described as follows. In the **first phase**, the Partially Empty Grammatical Structures (PGS), each composed by elements constituted either by parts of speech (POS) tags or function words[5], are generated. The PGS's are constructed through a morphosyntactic analysis made with FreeLing[6] of the literary phrase corpora, **cWerther** and **cCask**, described in the Section 3. The POS tags of the generated PGS's are replaced by real words during the second phase.

FreeLing [14] is a commonly used tool for morphosyntactic analysis, which receives as input a string of text and returns a POS tag as output, for each word in the string. The POS tag indicates the type of word (verb, noun, adjective, adverb, etc.), and also information about inflections, i.e., gender, conjugation and number. For example, for the word "Investigador" FreeLing generates the POS tag `[NCMS000]`. The first letter indicates a **N**oun, the second a **C**ommon noun, **M** stands for **M**ale gender and the fourth gives number information (**S**ingular).

[5] Function words (or functors) are words that have little or ambiguous meaning and express grammatical relationships among other words within a sentence, or specify the attitude or mood of the speaker, such as prepositions, pronouns, auxiliary verbs, or conjunctions.

[6] FreeLing can be downloaded from: http://nlp.lsi.upc.edu/freeling

Table 3: Corpus 5KL, composed of 4 839 literary works.

	Sentences	**Tokens**	**Characters**
5KL	9 M	149 M	893 M
Average for sentence	2.4 K	37.3 K	223 K

The last 3 characters give information about semantics, named entities, etc. We will use only the first 4 symbols of the POS tags.

In the **second phase**, the POS tags in the PGS are replaced by a corresponding vocabulary, using a semantic approach algorithm based on an ANN model (Word2vec[7]). The 5KL corpus, described in Section 3, was used for the Word2vec training, as well as the following parameters: only words with more than 5 occurrences in the 5KL corpus were considered, the context window has a dimension of 10, the dimensions of the vector representations were tested within a range of 50 to 100, being 60 the dimension with the best results. The Word2vec model used in this work is the *continuous skip-gram model* (Skip-gram), which receives a word (*Query*) as input and, as output, returns a set of words (*embeddings*) semantically related to the *Query*. The detailed process for the generation of sentences is described in what follows.

4.1 Phase I: PGS Generation

For the generation of each PGS, we use methods guided by fixed morphosyntactic structures, called *Template-based Generation* or *Canned Text*. In [10], it is argued that the use of these techniques saves time in syntactic analysis and allows one to concentrate directly on the vocabulary. The canned text technique has also been used in several works, with specific purposes, such as in [4, 8], where the authors developed models for the generation of simple dialogues and phrases.

We use canned text to generate text based on the corpora of templates given by **cWerther** and **cCask**, described in Section 3. These corpora contain flexible grammatical structures that can be manipulated to create new phrases. The templates in a corpus can be selected randomly or through heuristics, according to a predefined objective. The process starts with the random selection of an original phrase $f_o \in$ corpus of length $N = |f_o|$. A template PGS is built from the words of a sentence f_o, where verbs (v), nouns (n) or adjectives (a) are replaced by their respective POS tags. Other words, particularly the function words, are retained. f_o is analyzed with FreeLing to identify the content words (verbs, nouns, adjectives), so that words of f_o with "values" in $\{v, n, a\}$ are replaced by their respective POS tags. These content words provide most of the information in any text, regardless of their length or genre [2]. Our hypothesis is that by changing only these content words, we simulate the generation of phrases by homo-syntax: different semantics, same structure. The output of this process is a PGS with function words that give grammatical support and POS tags that will be replaced later, in order to change the meaning of the sentence. The general architecture of this model is illustrated in Figure 1, where full boxes represent functional words and empty boxes represent POS tags to be replaced.

4.2 Phase II: Semantic Analysis and Substitution of POS Tags

In this phase, the POS tags of the PGS generated in Phase I are replaced. Tags corresponding to nouns are replaced with a vocabulary close to the context

[7] Word2vec is a group of related ANN models, used to produce word embeddings [1].

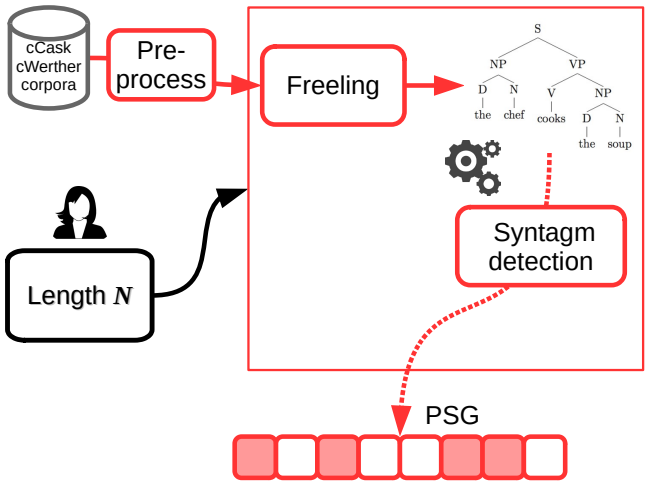


Fig. 1: *Canned text* model for generating a PGS.

defined by the user (the query), while verbs and adjectives are replaced with a vocabulary closer (similar in meaning) to the original terms of the sentence. The idea is to preserve the style and emotional-psychological content, that the author intended to associate with the characters that he is portraying in his work.

The 5KL corpus is pre-processed to standardize text formatting, eliminating characters that are not important for semantic analysis such as punctuation and numbers. This stage prepares the Word2vec training data which uses a vector representation of the 5KL corpus. We use Gensim[8], a Python implementation of Word2vec. A query Q, provided by the user, is given as input to this algorithm, and its output is a set of words (*embeddings*) associated with a context defined by Q. In other words, Word2vec receives a term Q and returns a lexicon $L(Q) = (Q_1, Q_2, ..., Q_m)$, which represents a set of $m = 10$ words semantically close to Q. We chose this value of m because we found that if we increase the number of words obtained by Word2vec, they start to lose their relation to Q. Formally, we represnt a mapping by Word2vec as $Q \rightarrow L(Q)$.

The authors' corpora, **cGoethe** or **cPoe**, previously analyzed using FreeLing to obtain PGS's, had a POS tag associated to each word. Now, with FreeLing, each POS tag is used to create a set of words, with the same grammatical information (identical POS tags). An Association Table (AT) is generated as a result of this process. The AT consists of k entries of the type: $\text{POS}_k \rightarrow$ list of words $v_{k,i}$, with same grammatical information. Formally $\text{POS}_k \rightarrow \boldsymbol{V}_k = \{v_{k,1}, v_{k,2}, ..., v_{k,i}, ...\}$. To generate a new phrase, each tag $\text{POS}_k \in \text{PSG}$, $k = 1.2, ..., m$, is replaced by a word selected from the lexicon $\boldsymbol{V}_k$, given by AT.

To choose a word in $\boldsymbol{V}_k$ to replace POS_k, we use the following algorithm. A vector is constructed for each of the three words defined as:

[8] Available in: https://pypi.org/project/gensim/

- o: is the k_{th} word in the phrase f_o, corresponding to tag POS_k;
- Q: word defining the *query* provided by the user;
- w: candidate word that could replace POS_k, $w \in V_k$.

For each word o, Q and w, 10 closest words o_i, Q_i and w_i $i = 1, ..., 10$ are obtained with Word2vec. These 30 words are concatenated and represented by a vector $\boldsymbol{U}$ with dimension 30. The dimension was set to 30, as a compromise between lexical diversity and processing time. The vector $\boldsymbol{U}$ can be written as:

$$\boldsymbol{U} = (o_1, ..., o_{10}, Q_{11}, ..., Q_{20}, w_{21}, ..., w_{30}) = (u_1, u_2, ..., u_{30})\,. \tag{1}$$

Words o, Q and w generate three numerical vectors of 30 dimensions respectively, $\text{o} \rightarrow \boldsymbol{X} = (x_1, ..., x_{30})$, $\text{Q} \rightarrow \boldsymbol{Q} = (q_1, ..., q_{30})$, and $\text{w} \rightarrow \boldsymbol{W} = (w_1, ..., w_{30})$, where the elements x_j of $\boldsymbol{X}$ are obtained by taking the distance $x_j = dist(o, u_j) \in [0, 1]$, between o and each word $u_j \in \boldsymbol{U}$, provided by Word2vec. Obviously, the word o will be closer to the first 10 words u_j than to the remaining ones. A similar process is used to obtain the values of $\boldsymbol{Q}$ and $\boldsymbol{W}$ from Q and w, respectively. Cosine similarities are then calculated between $\boldsymbol{Q}$ and $\boldsymbol{W}$, and $\boldsymbol{X}$ and $\boldsymbol{W}$ as

$$\theta = \cos(\boldsymbol{Q}, \boldsymbol{W}) = \boldsymbol{Q} \cdot \boldsymbol{W} / |\boldsymbol{Q}||\boldsymbol{W}|\,, \quad 0 \leq \theta, \leq 1\,, \tag{2}$$

$$\beta = \cos(\boldsymbol{X}, \boldsymbol{W}) = \boldsymbol{X} \cdot \boldsymbol{W} / |\boldsymbol{X}||\boldsymbol{W}|\,, \quad 0 \leq \beta, \leq 1\,. \tag{3}$$

This process is repeated m times, once for each word $w = v_{k,i}$ in V_k, and similarity values θ_i and β_i, $i = 1, ..., m$, are obtained for each $v_{k,i}$. Once these m similarity values are obtained, the averages $\langle\theta\rangle = \sum \theta_i / m$ and $\langle\beta\rangle = \sum \beta_i / m$ are calculated. The normalized ratio $\left(\frac{\langle\theta\rangle}{\theta_i}\right)$ indicates how large the similarity θ_i is with respect to the average $\langle\theta\rangle$ that is, how close is the candidate word $w = v_{k,i}$ to the *query* Q. The ratio $\left(\frac{\beta_i}{\langle\beta\rangle}\right)$ indicates how reduced the similarity β_i is to the average $\langle\beta\rangle$, that is, how far away the candidate word w is from word o of f_o. These fractions are obtained for each pair (θ_i, β_i) and combined to calculate a score Sn_i as

$$Sn_i = \left(\frac{\langle\theta\rangle}{\theta_i}\right) \cdot \left(\frac{\beta_i}{\langle\beta\rangle}\right)\,. \tag{4}$$

The higher the value of Sn_i, the better the candidate word $w = v_{k,i}$ complies to the goal of approaching Q and moving away from the semantics of the original sentence. This goal aims to obtain the candidate $v_{k,i}$ closer to Q although still considering the context of the sentence. We use the candidate with maximum value of Sn_i to replace the nouns. However, to replace verbs and adjectives, we choose the candidate with maximum $S\text{va}_i$, given by

$$S\text{va}_i = \left(\frac{\theta_i}{\langle\theta\rangle}\right) \cdot \left(\frac{\langle\beta\rangle}{\beta_i}\right)\,, \tag{5}$$

so as to find the candidate word $v_{k,i}$ closer to the meaning of the original phrase.

Finally we sort the list of values of Sn_i (nouns) or $S\text{va}_i$ (verbs and adjectives) in decreasing order and choose, at random, from the first three elements,

the candidate word $v_{k,i}$ that will replace the POS_k tag. The result is a newly generated phrase $f(Q, N)$ that does not exist in the corpora, but maintains the psychological mood (emotional content) of f_o. The model is shown in Figure 2.

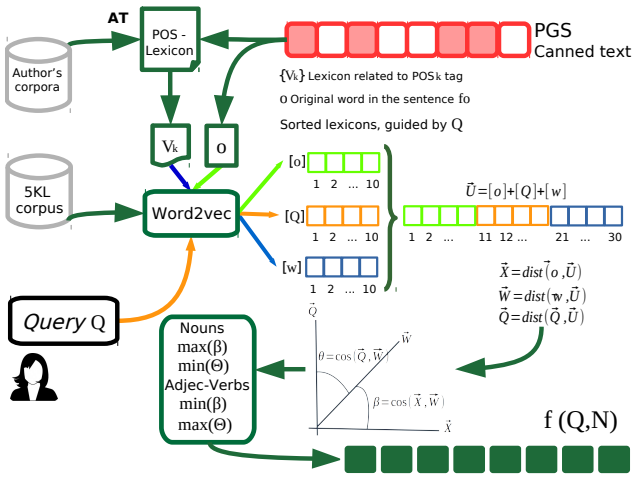


Fig. 2: Model for semantic approximation based on geometrical interpretation.

5 Evaluation and Results

A manual evaluation protocol has been designed to measure some characteristics of the phrases generated by our model. For baseline comparison, we used the model with the best results observed in experiments described in [11]. Although the baseline evaluation was done with a different scale of parameter values than in the current case, the comparison we present here helps us to understand the human evaluators' perception of the generated sentences.

Sentences have been generated using the corpora 5KL, **cGoethe**, **cPoe**, **cWerther** and **cCask**, as explained in Section 4. The queries employed for generating the sentences in Spanish are $Q \in \{$`ODIO, AMOR, SOL, LUNA`$\}$ (in English {`HATE, LOVE, SUN, MOON`}). We show some examples of sentences generated in Spanish in our experiments, manually translated to English.

Sentences generated using cCask, cPoe and 5KL

1. f(`LOVE,12`) = But I do not think anyone has ever promised against good will.
2. f(`HATE,11`) = A beautiful affection and an unbearable admiration took hold of me.
3. f(`SUN,9`) = So much does this darkness say, my noble horizon!
4. f(`MOON,9`) = My light is unhappy, and I wish for you.

Sentences generated using cWerther, cGoethe and 5KL

1. f(`LOVE,11`) = Keeping my desire, I decided to try the feeling of pleasure.
2. f(`HATE,7`) = I set about breaking down my distrust.
3. f(`SUN,17`) = He shouted, and the moon fell away with a sun that I did not try to believe.
4. f(`MOON,12`) = Three colors of the main shadow were still bred in this moon.

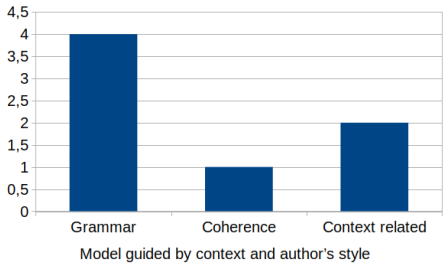
The experiment we performed consisted of generating 20 sentences for each author's corpora and the 5KL corpus. For each author, 5 phrases were generated with each of the four queries $Q \in \{$`LOVE, HATE, SUN,MOON`$\}$. Five people were asked to read carefully and evaluate the total of 40 sentences. All evaluators are university graduates and native Spanish speakers. They were asked to score each sentence on a scale of $[0-4]$, where 0 = very bad, 1 = bad, 2 = acceptable, 3 = good and 4 = very good. The following criteria were used in the evaluations.

- **Grammar:** spelling, correct conjugations and agreement between gender and number;
- **Coherence:** legibility, perception of a general idea;
- **Context:** relation between the sentence and the query.

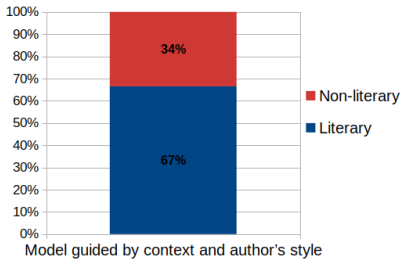
We also asked the evaluators to indicate if, according to their own criteria, they considered the sentences as literary (0 = NOT or 1 = YES). Finally, the evaluators were to indicate which emotion they associate to each sentence (0 = Fear, 1 = Sadness, 2 = Hope, 3 = Love, and 4 = Happiness). We compared our results with the evaluation made in [11] as a *baseline*. The evaluated criteria are the same in that and in this work. However, the evaluation scale in [11] is in a range of $[0-2]$ (0 = bad, 1 = acceptable, 2 = very good). Another difference is that, for the current evaluation, we have calculated the mode instead of the arithmetic mean, since it is more feasible for the analysis of data evaluated in the *Likert* scale.

In Figure 3a, it can be seen that the **Grammar** criterion obtained good results, with a general perception of *very good*. This is similar to the average of 0.77 obtained in the evaluation of the model proposed in [11]. The **Coherence** was rated as *bad*, against the arithmetic mean of 0.60 obtained in [11]. In spite of being an unfavourable result, we can infer that the evaluators were expecting coherent and logical phrases. Logic is a characteristic that is not always perceived in literature and we inferred, noting that many readers considered the sentences as not being literary (Figure 3b). The evaluators perceived as *acceptable* the relation between the sentences and the **Context** given by the *query*. This score is similar with the average of 0.53 obtained in [11]. Although one might consider that this rate should be improved, it is important to note that our goal in the current work is not only to approach the context, but to stay close enough to the original sentence, in order to simulate the author's style and to reproduce the psychological trait of the characters in the literary work.

In Figure 3b, we observe that 67% of the sentences were perceived as literary, although this is a very subjective opinion. This helps us understand that, despite



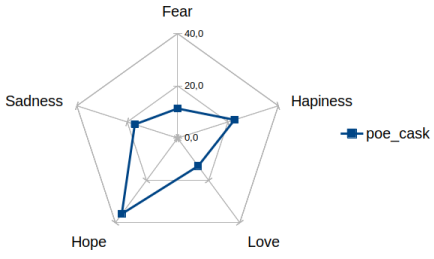
(a) Mode calculated from the evaluation made for the sentence generation model.



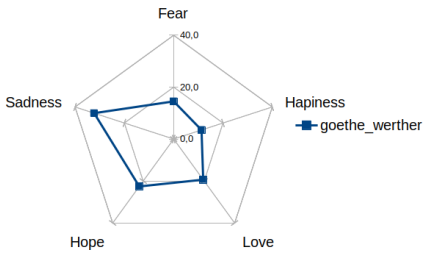
(b) Perception of literary nature of sentences.

Fig. 3: Evaluation of our GLT model.

the low rate obtained for **Coherence**, the evaluators do perceive distinctive elements of literature in the generated sentences. We also asked the evaluators to indicate the emotion they perceived in each sentence. We could thus measure to what extent the generated sentences maintain the author's style and the emotions that he wished to transmit. In Figure 4a, we observe that **hope**, **happiness** and **sadness** were the most perceived emotions in phrases generated with **cCask** and **cPoe**. Of these, we can highlight **Sadness** which is characteristic of much of Poe's works. Although, **Hapiness** is not typical in Poe's works, we may have an explanation for this perception of the readers. If we analyze *The Cask of Amontillado*, we observe that its main character, *Fortunato*, was characterized as a happy and carefree man until his murder, perhaps because of his drunkenness. The dialogues of *Fortunato* may have influenced the selection of the vocabulary for the generation of the sentences and the perceptions of the evaluators. In Figure 4b, we can observe that sentences generated with **cWerther** and **cGoethe** transmit mainly **sadness**, **hope** and **love**, which are psychological traits easily perceived when reading *The Sorrows of Young Werther*.



(a) Perceived emotions in sentences generated with Allan Poe's corpora



(b) Perceived emotions in sentences generated with Goethe's corpora

Fig. 4: Comparison between the emotions perceived in sentences.

6 Conclusions and Future work

We have proposed a model for the generation of literary phrases, which is influenced by two important elements: a context, given by a word input by the user, and a writing mood and style, defined by the training corpora used by our models. Results are encouraging, as the model generates grammatically correct sentences. The phrases also associate well with the established context, and perceived emotions correspond, in good part, to the emotions transmitted in the literature involved. In the case of *The Cask of Amontillado*, the perceived emotions seem not to resemble the author's style. This may be due to the fact that, in this short story, the dialogues of the main character are happy and carefree, the tragic murder occurring only in the end. When characters in a literary text have different psychological traits, the semantic analysis of the generated phrases may show heterogeneous emotional characteristics. Experiments using works with characters showing more homogeneous psychological traits, such as *The Sorrows of Young Werther*, more easily detect a dominant psychological trend portrayed by the author.

Although there was a poor evaluation for the **Coherence** criterion, it is possible to argue that coherence is not a dominant feature of literature in general, and most of the generated sentences were perceived as literary. The model is thus capable of generating grammatically correct sentences, with a generally clear context, that transmits well the emotion and psychological traits portrayed by the content and style of an author. For future work, it is proposed to extend the length of the generated text. This could be achieved by joining several generated sentences together. In that sense, The introduction of the rhyme can be extremely interesting when produce several sentences to constitute a paragraph or a stanza. The coupling with a generator of assonant and consonant rhymes [9] is planned in the near future.

References

1. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1798–1828 (2013)
2. Bracewell, D.B., Ren, F., Kuriowa, S.: Multilingual single document keyword extraction for information retrieval. In: 2005 International Conference on Natural Language Processing and Knowledge Engineering. pp. 517–522. IEEE, Wuhan, China (2005)
3. Clark, E., Ji, Y., Smith, N.A.: Neural text generation in stories using entity representations as context. In: Conference of the NAC ACL-HLT. vol. 1, pp. 2250–2260. ACL, New Orleans, Louisiana (2018)
4. van Deemter, K., Theune, M., Krahmer, E.: Real versus template-based natural language generation: A false opposition? Computational Linguistics **31**(1), 15–24 (2005)
5. von Goethe, J.W.: The Sorrows of Young Werther. Penguin, London, England (2006, First edition 1774)

6. Mairesse, F., Walker, M.A., Mehl, M.R., Moore, R.K.: Using linguistic cues for the automatic recognition of personality in conversation and text. Journal of Artificial Intelligence Research **30**, 457–500 (2007)
7. Majumder, N., Poria, S., Gelbukh, A., Cambria, E.: Deep learning-based document modeling for personality detection from text. IEEE Intelligent Systems **32**(2), 74–79 (2017)
8. McRoy, S., Channarukul, S., Ali, S.: An augmented template-based approach to text realization. Natural Language Engineering **9**, 381 – 420 (2003)
9. Medina-Urrea, A., Torres-Moreno, J.M.: Rimax: Ranking semantic rhymes by calculating definition similarity. arXiv (2020)
10. Molins, P., Lapalme, G.: JSrealB: A bilingual text realizer for web programming. In: 15th ENLG. pp. 109–111. ACL, Brighton, UK (2015)
11. Moreno-Jiménez, L.G., Torres-Moreno, J.M., Wedemann, R.S.: Generación automática de frases literarias en español. arXiv pp. arXiv–2001 (2020)
12. Oliveira, H.G.: Poetryme: a versatile platform for poetry generation. In: Computational Creativity, Concept Invention and General Intelligence. vol. 1. Institute of Cognitive Science, Osnabrück, Germany (2012)
13. Oliveira, H.G., Cardoso, A.: Poetry generation with poetryme. In: Computational Creativity Research: Towards Creative Machines. vol. 7. Atlantis Thinking Machines, Paris, France (2015)
14. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: 8th LREC). Istanbul, Turkey (2012)
15. Pérez y Pérez, R.: Creatividad Computacional. Larousse - Grupo Editorial Patria, México (2015)
16. Poe, E.A.: The Cask of Amontillado. The Creative Company, Mankato, US (2008, First edition 1846)
17. Riedl, M.O., Young, R.M.: Story planning as exploratory creativity: Techniques for expanding the narrative search space. New Generation Computing **24**(3), 303–323 (2006)
18. Sharples, M.: How We Write: Writing as creative design. Routledge, London, UK (1996)
19. Siddiqui, M., Wedemann, R.S., Jensen, H.J.: Avalanches and generalized memory associativity in a network model for conscious and unconscious mental functioning. Physica A: Statistical Mechanics and its Applications **490**, 127–138 (2018)
20. Sridhara, G., Hill, E., Muppaneni, D., Pollock, L., Vijay-Shanker, K.: Towards automatically generating summary comments for java methods. In: IEEE/ACM International Conference on Automated Software Engineering. p. 43–52. ACM, Antwerp, Belgium (2010)
21. Szymanski, G., Ciota, Z.: Hidden markov models suitable for text generation. In: Mastorakis, N., Kluev, V., Koruga, D. (eds.) WSEAS. pp. 3081–3084. WSEAS - Press, Athens, Greece (2002)
22. Torres-Moreno, J.: Beyond stemming and lemmatization: Ultra-stemming to improve automatic text summarization. arXiv **abs/1209.3126** (2012)
23. Torres-Moreno, J.M.: Automatic Text Summarization. ISTE, Wiley, London, UK, Hoboken, USA (2014)
24. Wedemann, R.S., Plastino, A.R.: Física estadística, redes neuronales y freud. Revista Núcleos **3**, 4–10 (2016)
25. Zhang, X., Lapata, M.: Chinese poetry generation with recurrent neural networks. In: 2014 EMNLP. pp. 670–680. ACL, Doha, Qatar (2014)