Journal of Intelligent & Fuzzy Systems xx (20xx) x–xx
DOI:10.3233/JIFS-169491
IOS Press

Statistical language modelling for automatic story generation

Marta Vicente*, Cristina Barros and Elena Lloret
Department of Software and Computing Systems, University of Alicante, Alicante, Spain

Abstract. This paper proposes an end-to-end Natural Language Generation approach to automatically create fiction stories using statistical language models. The proposed approach integrates the stages of macroplanning and the surface realisation, necessary to determine the content to write about together with the structure of the story, and the syntactic and lexical realisation of sentences to be generated, respectively. Moreover, the use of language models within the stages allows the generation task to be more flexible, as far as the adaptation of the approach to different languages, domains and textual genres is concerned. In order to validate our approach, two evaluations were performed. On the one hand, the influence of integrating position-specific language modelling in the macroplanning stage into the surface realisation module was evaluated. On the other hand, a user evaluation was performed to analyse the generated stories in a qualitative manner. Although there is still room for improvement, the results obtained from the first evaluation in conjunction with the user evaluation feedback shows that the combination of the aforementioned stages in an end-to-end approach is appropriate and have positive effects in the resulting generated text.

Keywords: Natural language generation, language modelling, document planning, surface realisation, automatic story generation

1. Introduction

The generation of stories is a difficult task in which complex cognitive processes intervene. The process of creating a story requires both knowledge of the world and ability to perform tasks such as planning events or resolving situations [34]. It has been pointed out that the production becomes especially challenging when the process has to be automatically performed by a machine [12]. Even if the main topic of the story has already been established, it is equally important to determine some other details, such as the number of characters, the entities involved, the locations where the action will take place, etc. together with the related events and the order in which they should take place. Therefore, this task implies to make decisions at different linguistic levels (lexical, syntactic, semantic and discourse level) that will certainly have a great impact on the quality of the generated text.

Natural Language Generation (NLG), as a subarea of Natural Language Processing, can provide, in turn, useful mechanisms to create a story automatically. In order to generate a text, the responsibilities of NLG include, to determine *"what to say"* and *"how to say it"*. This dual purpose is normally addressed as a pipeline of three stages [29]: macroplanning, microplanning and surface realisation. Macroplanning is in charge of determining the structure of the text to be produced, as well as deciding the content it will have. Microplanning and surface realisation are focused on taking that content as input, and transforming it into human language, using the appropriate vocabulary, syntactic structures, referring expressions, discourse markers, etc.

*Corresponding author. Marta Vicente, Department of Software and Computing Systems, University of Alicante, Apdo. de correos 99, E-03080 Alicante, Spain. E-mail: mvicente@dlsi.ua.es.

In this paper, we study whether a story could be automatically generated from scratch without having any clue about the characters, topics, or facts to describe, just having some previous information from other existing stories that could serve as an inspiration source. Therefore, we propose a statistical end-to-end NLG approach to create fiction stories that has to face these two challenges: i) dynamically determine the structure and content to be produced; and ii) perform a realisation that leads to a meaningful and coherent text. In light of this, our approach is built integrating macroplanning and surface realisation within the same NLG pipeline as two consecutive stages: the output of the macroplanning will constitute the input for the surface realisation. We devised a sequence of steps that allows the approach to extract the distribution of relevant information from a document which will be used as a starting point in order to guide the realisation of a text in terms of content and structure. To achieve this, a statistical perspective ws adopted. Specifically, macroplanning is addressed using Positional Language Models (PLM), which are able to account the position of the words and their relevance while, in the surface realisation stage, Factored Language Models (FLM) are employed. As output, the approach produces a set of sentences in the form of a story, inspired by the structure and linguistic elements of an existing one, but that may have a different realisation (different events, actions and vocabulary).

In this sense, the novel aspects of our approach are as follows:

- We propose a statistical story generation approach that does not need any human intervention at any stage (i.e. it is fully automatic), where the creation of a new text is inspired by the structure and most relevant content of an existing one. These elements will guide and shape the generation of individual sentences that will make up the final story.
- We contribute to a more flexible macroplanning method, since the structure of the generated text is not restricted to a set of predefined options. In contrast, the structure and content are learnt from existing stories (they are considered as inspiring stories), so it is possible to generate as many new stories as one may desire.
- The surface realisation method uses semantic information that helps to increase the vocabulary range, thus preventing from the excessive and unnecessary repetition of the same terms over and over again.
- In the current approach, both stages, macroplanning and surface realisation, build their models considering synsets[1] as a representation of the linguistic elements composing the text. Those synsets are featured by their grammatical category. However, due to the statistical nature of the framework, our proposed architecture would be able to carry out the generation process regardless which linguistic element is selected, as long as its distribution could be estimated.

Although there is still a lot of room for improvement, the preliminary results show that statistical language models can definitely contribute to the development of more adaptive and flexible story generation systems, thus providing mechanisms that can be extended to other kind of domains, languages and textual genres.

In the next section (Section 2), we refer to the related work on story generation and the approaches for macroplanning and surface realisation. In Section 3, we explain our story generation approach which integrates the macroplanning and the surface realisation stages. In Section 4, we describe the experimentation as well as the tools employed, whereas in Section 5, the results are discussed, including a detailed error analysis. Finally, in Section 6, the conclusion and directions for future work are provided.

2. Related work

The task of automatic story generation has to face several challenges [12]. On the one hand, there is no clear definition of what the inputs and outputs should be, since these can be of various types, depending on the purpose of the story, the target reader, the theme, etc. On the other hand, the aspects that contribute to make a good story are still debatable and difficult to assess computationally.

Recent work in narrative and storytelling has been focused on regenerating stories from graphs of intentions [20] or approaching the task working with discourse and story planning simultaneously to differentiate levels of narrative [36]. Despite both are good attempts to automatically address this task, in the first case the graph has to be populated by humans while, in the second one, certain extra information to start with is required, such as the initial state, the set of action types or the conditions related to the goal.

[1] Set of cognitive synonyms related to a concept used in WordNet [8].

In some research works, the input is hand-made as in [33]. There, a causal network represents the actions of the characters in the story world. Our proposed approach differs from the previous ones in the sense that it attempts to minimise the human intervention in the NLG process in order to avoid the hand-coding of the story constraints, thus increasing the automation of the creation process.

In the literature, other NLG perspectives have been tried, for instance, the one proposed in [2]. It uses tree structures to organise both the story and its discourse, where the basic elements of such trees are events, concepts and relations derived from a dictionary of nouns and verbs. Then, using rules, different kinds of stories are created depending on the type of discourse relation desired. The authors proposed 6 types of discourse relations, such as "cause-effect", or "result", leading to a final story that is complemented with music and a graphical interface. However, the limitation of this approach is the lack of flexibility and variability when generating the stories, since the approach always produce the same type of sentence within a rigid structure, where only the nouns and verbs change.

Out of the scope of the creativity field, macroplanning and surface realisation have also been used for generating other types of texts. In this respect, one can find either rule-based [28] or trainable systems [13]. The first type of systems are more domain-dependent, but robust, whereas the second ones remain more transferable, although restricted by the amount of data to train.

Alternatively, some approaches rely on existing software (e.g., SimpleNLG [37]) due to its simplicity and versatility [1, 10]. However, this kind of techniques/software are not always easily adaptable to different domains, purposes, or languages aside from the ones they were originally designed for. In this regard, our model is designed to show more flexibility.

Statistical modelling, specifically n-gram language models, have also been used both for macroplanning and surface realisation in different tasks, such as producing sequences of discourse relations/markers to form sentences in [23], helping to order sentences from trigram-predicate probabilities in [7], or in [17], where language models are used to choose word transformations after applying generation rules.

Regarding the statistical models employed in this research, on the one hand, PLMs have been successfully used in some language-related areas such as summarisation [18] or information retrieval [21] in order to overcome the limitations of considering words but not locations. But, to the best of our knowledge, they have not been directly implemented within the generation framework. On the other hand, in recent years, FLM have been employed for NLG, such in BAGEL [22], where FLM are used to predict the semantic structure of the sentence to generate; or in [25], where FLM are used to rank sentences in Portuguese. In this sense, this kind of models are used within our approach to generate text based on the information given by the macroplanning stage, but also to apply a ranking strategy to determine the best sentence among a set of possible candidates.

Our proposal could be framed with the line of the trainable systems, but it differs from previous research work in the fact that it uses language models to dynamically learn the document plan directly from plain text, so the required content is first extracted, and then provided to the surface realisation stage, where sentences are generated word by word from the probabilities obtained through language models. In this manner, more flexible systems may be produced.

3. End-to-end story generation approach

This section describes our NLG approach for story generation which consists of two different stages: macroplanning and surface realisation. An overview of the process is illustrated in Fig. 1.

First, a document plan is generated using Positional Language Models (i.e., macroplanning). Then, based on the information given by the document plan, the surface realisation stage takes this information as input together with a Factored Language Model trained over an input corpus (i.e., a training corpus).

At the end of the surface realisation stage, the final output in the form of a text is obtained.

3.1. Macroplanning

In the NLG process, macroplanning is responsible for both selecting the content and providing the structure that articulates the output. In this approach, PLMs are used as the technique to address the two aforementioned tasks, being a document plan the output of this stage.

3.1.1. Positional Language Models fundamentals

Different from the common bag-of-words perspective, PLMs are able to take into account the positions of words together with the number of their

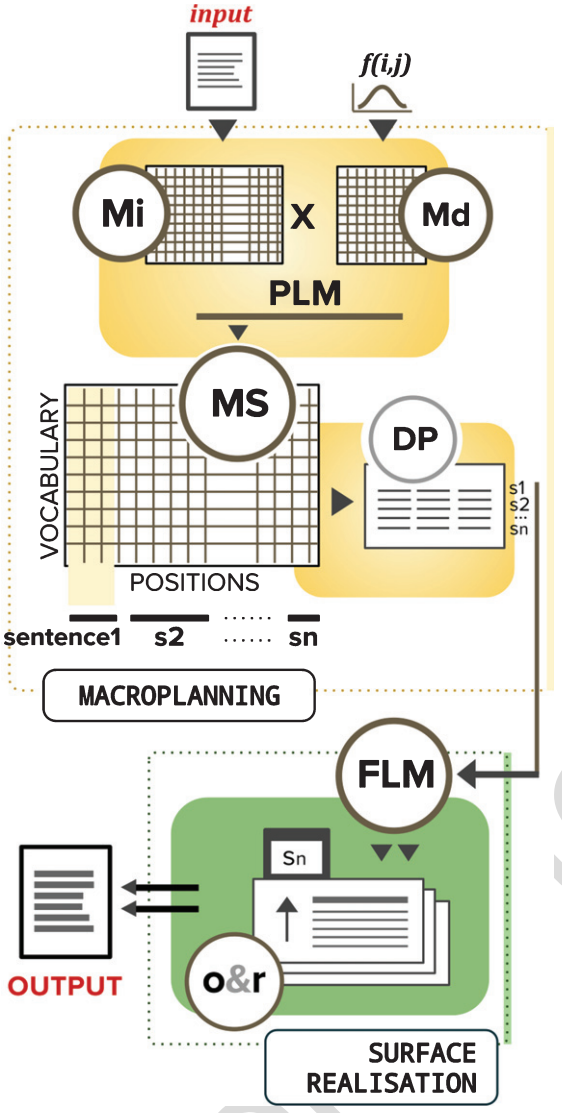

Fig. 1. Overview of the proposed approach, where macroplanning produces a document plan from the knowledge provided by the PLMs, and surface realisation generates an output leveraging on FLMs and over-generation and ranking techniques.

occurrences. On this basis, for each position i of a text, we compute a model $P(w \mid i)$ as formulated in Equation 1. As a result, every word of the vocabulary gets a value for that location conditioned by the distance of other appearances of that word along the text.

$$P(w \mid i) = \frac{\sum_{j=1}^{|D|} c(w, j) \times f(i, j)}{\sum_{w' \in V} \sum_{j=1}^{|D|} c(w', j) \times f(i, j)} \quad (1)$$

Here, $c(w,j)$ indicates the presence of term w in the position j, $|D|$ refers to the length of the document, V is the vocabulary and $f(i,j)$ is the propagation function that rates the distance between i and j. The greater the distance, the lowest the value. The process can be better understood through Fig. 2.

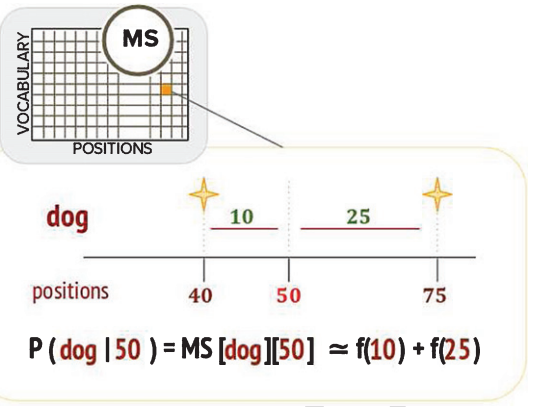

Fig. 2. The value for a term, e.g., "*dog*", given a position, e.g., 50, depends on the distances between that position and the ones were the word appears. The value P is calculated as a function of the distances, with a later normalisation, as shown in Equation 1. Those values are then stored in a matrix of scores (MS) (Section 3.1.2).

Using Equation 1, we can obtain the distribution of the elements along the text, considering those with higher values, the most relevant ones. We translate this information into a document plan.

3.1.2. *Implementation of the model*

Regarding the implementation of the model, the first step to be undertaken is the population of the matrix of importance (Mi), that corresponds to $c(w, j)$ in Equation 1. With as many rows as elements in the vocabulary, and as many columns as positions in the source text, the value stored in Mi[*w,j*] is either 1 or 0 depending on the presence of w in j. A matrix of distances (Md) is filled by means of the propagation function. Afterwards, the value of $P(w \mid i)$, is calculated in a matrix of scores (MS), as the result of the product Mi*x*Md, normalising the values per position/column.

At this point, the document plan is created line by line from the MS. The matrix is segregated into groups of columns, and from each of these resulting submatrices, one line of the document plan is composed, containing those elements that score the best in each set.

3.2. *Surface realisation*

From the document plan provided by the macroplanning stage, the surface realisation of

the final output is performed relying on Factored Language Models (FLM).

3.2.1. Factored Language Models fundamentals

FLM are an extension of language models, presented in [4], where a word w is seen as a collection of K parallel factors, so that $w \equiv \{f^1, f^2, f^3 \dots f^k\}$. In this sense, a factor f of a given word can be anything, including morphological classes, stems, lemmas, the word itself, and any other linguistic features that might correspond to or decompose a word. The main objective of these models is to build a statistical model over the individual factors selected: $P(f|f_1, \dots, f_N)$, where the prediction of the factor f is based on N parents $\{f_1, \dots, f_N\}$.

Therefore, there are two main issues to consider in the development of this kind of models: 1) choose an appropriate set of factors, and 2) find the best statistical model over these factors.

3.2.2. Implementation of the model

For the purpose of this research, we selected several factors (i.e. lemma, Part-of-Speech tag and synset) to train the FLM over a training corpora (that has not necessarily to be the same collection of documents used as input for the end-to-end system). This FLM is then employed for the generation of the text during the surface realisation stage.

The technique employed in this stage for generating sentences is based on over-generation and ranking, where a set of candidate sentences is first generated and subsequently ranked according to the probabilities given by a language model (i.e. the FLM in our case) so that the best sentence can be selected.

To generate the set of candidate sentences using a FLM, we select the words with the highest probabilities with respect to the grammar shown in Fig. 3. This grammar is used to guarantee that the generated sentence contains the same type of elements enclosed in the document plan. Moreover, at the time of selecting words, in addition to the FLM, the approach also takes into account the information specified in the document plan, so it tries to prioritise the selection of the words contained in it.

$$S \rightarrow NP\ VP$$
$$NP \rightarrow D\ N$$
$$NP \rightarrow D\ Adj\ N$$
$$VP \rightarrow V\ NP$$
$$VP \rightarrow V\ Adv$$
$$VP \rightarrow V\ Adv\ Adj$$

Fig. 3. Basic clause structure grammar.

When a set of candidate sentences is finally generated, these candidate sentences are ranked to select the final sentence, which is the one with the highest probability. The sentence probability is computed by the chain rule (shown in Equation 2), where the probability of a sentence is calculated as the product of the probability of all its words.

$$P(w_1, w_2 ... w_n) = \prod_{i=1}^{n} P(w_i|w_1, w_2 ... w_{i-1}) \quad (2)$$

The probability of a sentence can be computed in different ways depending on the language model used. In our case, the probability of a word when using a FLM, is calculated as the linear combination of FLMs, as suggested in [14], where a weight λ_i, which was empirically determined, is assigned to each of the FLM used, with 1 as the total sum of the weights, as shown in Equation 3:

$$P(f_i|f_{i-2}^{i-1}) = \lambda_1 P_1(f_i|f_{i-2}^{i-1})^{1/n} + \dots + \lambda_n P_n(f_i|f_{i-2}^{i-1})^{1/n} \quad (3)$$

where f is the selected factors to train the FLMs.

4. Experimental scenario

This section describes the experimentation carried out in the scenario of fiction stories generation, and more specifically, children stories. The corpora and tools employed are described in Section 4.1 while the experiments performed are detailed in Section 4.2.

4.1. Corpora and tools

A collection of 779 English children stories was used as corpora, including the Lobo and Matos corpus [19] and other stories automatically gathered from *Bedtime stories*[2] and *Hans Christian Andersen: Fairy Tales and Stories*[3]. Table 1 shows the statistics of the corpora used to obtain the document plans in the macroplanning stage and also used to train the FLM used during the surface realisation.

These corpora were preprocessed using Freeling [26], an open-source multilingual language processing library, to obtain linguistic information necessary to build the models. Specifically, for the purpose of this research, lexical, grammatical and semantic information was extracted. In particular, the

[2]https://freestoriesforkids.com/

[3]http://hca.gilead.org.il/

Table 1
Statistics of the collection of English children stories used as corpora

# of documents	779
# of total sentences	26959
average # of sentences per document	35
# of total words	745783
average # of words per document	720

following information was obtained: the lemma, Part-of-Speech(POS) tag and synset. WordNet 3.0 in conjunction with JWI [9], a library for interacting with the tool, was employed in order to manage the use of the synsets as well as to obtain the words enclosed in them.

In addition, the training of the FLM employed during the surface realisation stage was performed using SRILM [32], a software which allows to build and apply different probabilistic language models that includes an implementation of FLM.

4.2. Experiments

To evaluate our end-to-end NLG approach, a series of experiments were performed. First, each stage was empirically adjusted to achieve an optimal configuration. Then, a subset of the corpora was used to produce the document plans. They would become the input for the surface realisation module, responsible of generating from them the final stories.

At this point, it has to be noted that in this research work, from each document plan only one story was generated. However, since we are relying on semantic knowledge in the surface realisation stage, our approach could be able to generate different story realisations from the same document plan. For instance, let's assume that the document plan provides the following information: *01382086-a; 00107416-r; 00339934-v; 07428954-n*. Then, we could lexicalise (i.e. realise) a sentence in various ways, such as *"A big earthquake took place recently"*, *"A large seism was produced recently"*, or *"A big seism occurred lately"*. As it can be observed, due to the grammar restrictions, the proposed approach is currently limited to the generation of simple short sentences, but this could be refined, as long as the grammar could be expanded in order to create more elaborated sentences like *"A big earthquake of magnitude 8.2 took place in the south of Mexico recently."*.

Regarding the macroplanning, some decisions were made about the vocabulary, the propagation function and the production of the document plan itself. First of all, the vocabulary was formed by the synsets corresponding to every element of the text belonging to any of these four categories: nouns, verbs, adjectives and adverbs (one synset per word). In this manner, we were able to incorporate grammatical and semantic features to the process. The propagation function was computed using a Gaussian kernel, following the work of [18], through Equation 4 in order to obtain the values to be stored in the Md matrix.

$$f(k, j) = e^{\frac{-(k-j)^2}{2\sigma}} \quad (4)$$

As it can be observed, there is a parameter σ that needs to be tuned. This parameter determines the semantic scope of the term that can be found in the position modelled. Regarding the distance parameter, the performance of the kernel with σ values of 25, 75 and 125 is represented in Fig. 4. If σ has an high value, the kernel will cause the model to behave as a bag-of-words approach, were every occurrence of the word is always rated with the same value, regardless of the existing distance, thus loosing the location information.

From previous experiments, it was empirically determined that the variation of the vocabulary in the resulting document plan is affected by the fluctuation of σ parameter. We measured that feature considering that a document plan should enclose a degree of variety yet allowing certain repetition of their elements. The intuition behind this idea is that a coherent text would exhibit certain term/entity continuity, and therefore, repetition (this is known as coreference). Finally, the empirical results revealed that 25 was the best σ value to reach this goal.

Next, we completed the macroplanning process by conforming the document plan. To do so, it was first decided to select the values that will appear in the plan

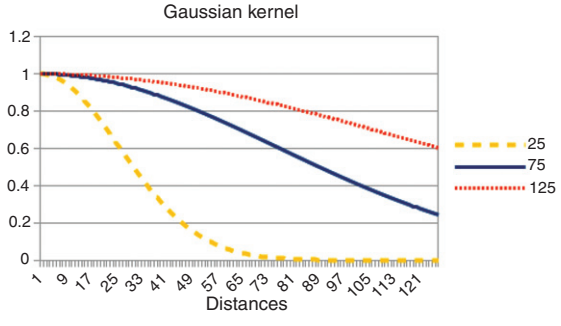

Fig. 4. Gaussian kernel performance for several values of σ.

Table 2
Example of the first lines of a document plan represented by synsets. To facilitate the understanding of the scheme, the lemmas that produce them have been included in this table. With the same aim of clarifying the example, second and third occurrences of same part-of-speech category have been rewritten as "-"

1	00439252-a,-,-	00047534-r,-,-	01009240-v,-,-	13384557-n,-,-
2	01332386-a,-,-	00047534-r,-,-	00056930-v,-,-	09917593-n,-,-
3	00217728-a,-,-	00048739-r,-,-	00941990-v,-,-	07544647-n,-,-
4	01943406-a,-,-	00047534-r,-,-	00829107-v,-,-	08329453-n,-,-
5	00754682-a,-,-	00101323-r,-,-	02624263-v,-,-	09466280-n,-,-
6	01391351-a,-,-	00020759-r,-,-	00120316-v,-,-	04236377-n,-,-
7	01391351-a,-,-	00031899-r,-,-	01009240-v,-,-	09988063-n,-,-
8	01889256-a,-,-	00024073-r,-,-	00720063-v,-,-	03221720-n,-,-
9	01592857-a,-,-	00017445-r,-,-	00720063-v,-,-	03221720-n,-,-
10	00173391-a,-,-	00117620-r,-,-	00668099-v,-,-	06333653-n,-,-
1	clever,-,-	also,-,-	say,-,-	money,-,-
2	intellectual,-,-	also,-,-	bear,-,-	child,-,-
3	beautiful,-,-	now,-,-	speak,-,-	heart,-,-
4	sensible,-,-	also,-,-	learn,-,-	court,-,-
5	industrious,-,-	far,-,-	rise,-,-	world,-,-
6	little,-,-	never,-,-	make,-,-	arm,-,-
7	little,-,-	very,-,-	say,-,-	papa,-,-
8	proud,-,-	not,-,-	look,-,-	door,-,-
9	lowly,-,-	even,-,-	look,-,-	door,-,-
10	great,-,-	then,-,-	stand,-,-	name,-,-

from sets of consecutive columns of the MS, which correspond to consecutive positions of the terms in the input document sentences. As a result, we obtained one line of the document plan per each sentence of the source text. Since the surface realisation should also generate one sentence per line contained in the document plan, we devised the lines of the document plan to contain synsets of verbs, nouns, adjectives and adverbs. More specifically, each line provides three synsets of each of the part-of-speech categories previously mentioned (i.e., each line of the document plan will contain 12 synsets in total). We show an example of several lines in Table 2.

Once the document plans were created, the surface realisation was in charge of generating the final stories following the guidelines the plans had established while being also consistent with the grammar. In a first step, the surface realisation produced a sentence in the form of synsets. This intermediate output can be considered as pre-sentence that acts as abstract representation of the candidate sentences that could be generated from it. As a result of their abstract configuration, the richness of the vocabulary was increased, since each synset could be turned into several words to produce a final combination.

Therefore, for a generated pre-sentence consisting of synsets, each of these synsets was translated into a maximum of 3 words, thus obtaining the possible combinations of those words coming from each synset in order to produce different variations of candidate sentences. It was decided to select only 3 words due to the fine granularity of WordNet. Some synsets can be associated to a large number of words, and this would be computationally too expensive and time-consuming to process.

At this point, the candidate sentences were ranked following the linear combination of FLMs explained in Section 3.2, with the aim to select the most probable combination of words in accordance to the trained FLM. Since we do not perform inflection at this stage, the final sentence was made up of lemmas.

This process was then sequentially and iteratively repeated for each line in the document plan to finally create a new story.

5. Evaluation, results and discussion

In order to evaluate how the document plan impacts in the generated text as well as the generated fiction story, we performed two tests. On the one hand, we evaluated how the synsets and their distribution in the document plan are reflected in the resulting stories. This manner, we could check to what extent the surface realisation is indeed taking into account the information given by the document plan. On the other hand, a user evaluation with three participants was performed to analyse the general problems and errors of the approach.

5.1. Evaluating the impact of macroplanning

To estimate the influence of the document plan in the generation of the new tales, we analysed the relation between both documents. First, we confirmed that all the sentences created in their synset form contained at least one synset from the document plan. Since the three synsets of each part of speech were provided in descending order of probability, we could detect that, on average, 81% of the sentences included at least one of the highest rated synsets. However, a sentence can contain more than one synset, so we analysed this in detail by measuring the proportion of the synsets in each tale coming from the previous stage, macroplanning. On average, we found that 83% of the synsets were in the document plan, and 40% of them were the first option provided. These scores were expected since the FLM used to generate the sentences was trained on a corpus where the number of synsets and elements was far larger than the set conveyed by the document plan. Even though, we consider those results as clear indicators of the positive effect that macroplanning has into the surface realisation.

5.2. User evaluation

As it is usual in language generation, evaluating the system directly from the output, the story conveyed in our case, becomes a challenge itself. The result cannot be compared with some standard text. In light of this, 25% of the generated tales were randomly extracted, 45 stories in total, and three users manually read and analysed them one by one in order to assess the system performance and provide feedback based on the detection of problems. They shed light on the limitations of our approach. We used a 3 point score scale, where the meanings of 3, 2 and 1 were related to the potential of the output. If the text showed high potential to become a story itself, the text was rated with 3. If this not happened, but any set of consecutive sentences, seemed likely to become a more complex sentence or paragraph, the score to assign was 2. Finally, if no single set of sentences could make sense without adding information, then the score was 1. Some examples are shown in Table 3.

In addition, some aspects were considered in order to help classify the documents and also to detect some relationships between them and the score obtained: repetition of elements along the text, strong presence of some entities that could be transformed to characters, the consecutive sequence of sentences sharing meaning, the detection of a theme beneath the produced text, and the capacity of some parts to produce a description or a narration of events. From the stories analysed, and taking into account the indicated criteria, we obtained positive reviews on 21 generated stories. It was found that 8 of them had potential to inspire a story while from 13 documents, it would be possible to extract series of sentences susceptible to produce a paragraph or an episode of a larger narration. A total of 24 stories were reported to need deep changes. Nonetheless, the evaluation and feedback

Table 3
Examples of fiction stories generated with our approach

Tale (score 3)	**Tale fragment #1 (score 2)**
the two time fly the domestic_fowl.	[...]
the domestic_fowl fly the Eden.	the night ride the Moon.
the hare be the companion.	the night state the white hind.
the blind man perform the hare.	the full hour state the pale hyacinth.
the man perform not certain.	the night be the one light.
the man state then dry.	the wing give_birth the peace.
the big discipline snog the hare.	the cold wind give_birth the fire.
Tale (score 3)	**Tale fragment #2 (score 2)**
the sea be the rampart.	[...]
the mighty king look the sea.	the day be the brook.
the ship sail the sea.	the water run then clear.
the sky arrive the wood.	the bright sun travel the water.
the branch look the blue curtain.	the water achieve the bright sunlight.
the bird fly the full thing.	the bright star glitter the water.
the bad idea arrive the expression.	the water induce the thing.
the idea arrive the difficult expression.	the water give_birth the bright thing.
the bird arrive the small son.	the bright thing know the water.
the bad weather give_birth the bird.	the water state then strange.

Table 4
Effects of the grammar on sentence generation and possible improvements

the small time travel the Rome.
could become
the *small-time crook* travel *to* Rome
the three time saw the blind Queens.
could become
(the) three times *he* saw the blind Queens

obtained was profitable to detect the weaknesses to be resolved in the future.

5.3. Error analysis from user evaluation feedback

At a word level, the users highlighted the appropriate variety and richness of the vocabulary. Even though, for some examples they remarked that it would be adequate to use more synonyms, in order to prevent finding consecutive sentences repeating exactly the same terms. On the one hand, it brings forward the necessity of reaching a better command of semantic tools as WordNet, but on the other hand, it offers the possibility of using aggregation techniques to obtain richer statements.

At a sentence level, the evaluation revealed that, independently of the relation with their neighbours, sentences, in general, would become more meaningful once inflected. To understand why some excerpts present an oddly shape, we should attend to the grammar beneath, on the one side, and to the semantic aspect, on the other. As it was explained in Section 3.2, in the current development state of our approach, a grammar is mandatory in order to guarantee certain structure in the production of a sequence of terms that will form a sentence. However, this entails restrictions in the generation. Progress on this subject would permit the improvement of some formations (see Table 4). Regarding the semantic requirement, reading some sentences as "*the thousand sleep_together the early one-half*" or "*the small forest put the Moon*", what become apparent is that the system would benefit from a broader common knowledge source together with the application of richer techniques to improve the relations shown between the elements.

Thirdly, at the level of discourse, adapted to our defined framework, the users were able to identify themes and guiding threads. They also noted the presence of characters as being involved in possible actions. They stated that the repetition of elements was essential to detect those underlying features. This reinforces our initial consideration regarding the relevance of a proper distribution of the elements through the text. Along with these comments, the users indicated that in some of the examples the absence of enough information impeded the assumption of any of those elements (themes, characters), but mostly this difficulty was related to the semantics of the generated tales.

Table 5
The tale to which this fragment belongs was rated with 1, meaning that it could take profit from several refinements

	Tale fragment #3 (score 1)
	[...]
1	the old town state the time.
2	the thing happen the old town.
3	the thing necessitate a_bit different.
4	the child saw a_bit large.
5	the manner necessitate the large town.
6	the day survive long white.
7	the other people survive the three town.
8	the mind answer then friendly.

To conclude the analysis, Table 5 shows a fragment of a tale that received score 1, which is the type of story that would need more transformation. Through the example, it can be detected that more semantic information of the elements would help to improve their composition. For instance, it would be the case of subjects and actions that do not properly match (sentences 1: *town*/*state* and 5: *manner*/*necessitate*) or objects that do not correspond to the verb (sentence 6: *survive*/*long white*). At the same time, wider frames of meaning or the inclusion of events would be helpful to provide temporal or spacial context that would lead to locate the actions (*happen*, *saw*, *survive*) and its participants (*town*, *child*, *day*) in more consistent episodes.

6. Conclusion and future work

This paper proposed and evaluated a novel end-to-end NLG approach for generating fiction stories that integrates the macroplanning and the surface realisation stages within a statistical framework. The macroplanning stage employs Positional Language Models to automatically extract the distribution of the important elements of a text to create a document plan. This document plan is taken as the input for the surface realisation stage, where with the use of Factored Language Models and, on the basis of the content and structure provided by the document plan,

a new text is generated. To assess our approach, two types of evaluation were performed. On the one hand, we evaluated how the structure and the content of the document plan influenced the generation of text, and, on the other hand, we also performed a user evaluation to analyse the generated text (i.e. the final output of the approach).

Regarding the first evaluation, all the sentences generated by the approach contained elements from the document plan, 81% of the sentences having at least one of the highest rated elements. Furthermore, as there are not only one type of element within the document plan, on average, 83% of the content of each generated tale came from the document plan. Concerning the results of the second evaluation, we received positive feedback from 47% of the generated stories. We consider these results to be an indicator of the positive effects of combining the two stages (i.e. macroplanning and surface realisation).

Although the results are still preliminary, there are some issues which need to be improved and some interesting research lines opening new directions for future work. Regarding the macroplanning stage, we could first analyse if there are common synsets to all the texts in the corpora in order to see their frequency and in what context they appear. Moreover, it would be also worthwhile to analyse the generated document plans in an independent manner to detect if the story could be divided into several parts (i.e. sections, paragraphs, themes, etc.). In addition, we plan to expand the study to different types of textual genres.

In the case of surface realisation, it would be interesting to introduce semantic information through diverse resources (e.g. FrameNet [3], ConceptNet [31], BabelNet [24], etc.) in order to know if a specific word could be used in that sentence context when generating the content of the sentence, or if that word has any specific complement (e.g. prepositions, related verbs or action, etc.).

Acknowledgments

This research work has been partially funded by the Generalitat Valenciana, through the grant ACIF/2016/501 and the project "DIIM2.0: Desarrollo de técnicas Inteligentes e Interactivas de Minería y generación de información sobre la web 2.0" (PROMETEOII/2014/001); by the Spanish Government through projects RESCATA (Representación canónica y transformaciones de los textos aplicado a las Tecnologías del Lenguaje Humano, Ref. TIN2015-65100-R) and REDES (Reconocimiento de Entidades Digitales: Enriquecimiento y Seguimiento, Ref TIN2015-65136-C2-2-R), as well as by Ayudas Fundación BBVA a equipos de investigación científica, through the project "Análisis de Sentimientos Aplicado a la Prevención del Suicidio en las Redes Sociales (ASAP)". We also would like to thank Pedro Pablo Sacristán, the author of "Cuentos para dormir" (BedTime Stories), for kindly let us use his tales to conduct research into automatic storytelling.

References

[1] S. Acharya, B. Di Eugenio, A.D. Boyd, R. Cameron, K.D. Lopez and P. Martyn-Nemeth, Generating Summaries of Hospitalizations: A New Metric to Assess the Complexity of Medical Terms and Their Definitions, in: *Proceedings of the 9th International Natural Language Generation Conference*, Association for Computational Linguistics (2016), 26–30.

[2] T. Akimoto, J. Ono and T. Ogata, Narrative Forest: An Automatic Narrative Generation System with a Visual Narrative Operation Mechanism, in: *The 6th International Conference on Soft Computing and Intelligent Systems and The 13th International Symposium on Advanced Intelligence Systems* (2012), 2164–2167.

[3] C. Baker, C.J. Fillmore and J.B. Lowe, The Berkeley FrameNet Project, in: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics – Volume 1*, Association for Computational Linguistics (1998), 86–90.

[4] J.A. Bilmes and K. Kirchhoff, Factored Language Models and Generalized Parallel Backoff, in: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2003–Short Papers – Volume 2* (2003), 4–6.

[5] N. Bouayad–Agha, G. Casamayor and L. Wanner, Natural Language Generation in the Context of the Semantic Web, *Semantic Web* **5**(6) (2014), 493–513.

[6] S. Demir, S. Carberry and K.F. McCoy, Summarizing Information Graphics Textually, *Computational Linguistics* **38**(3) (2012), 527–574.

[7] D. Duma and E. Klein, Generating Natural Language from Linked Data: Unsupervised Template Extraction, in: *Proceedins of the 10th International Conference on Computational Semantics* (2013), 83–94.

[8] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[9] M.A. Finlayson, Java Libraries for Accessing the Princeton Wordnet: Comparison and Evaluation, in: *Proceedings of the 7th International Global WordNet Conference*, Tartu, Estonia (2014), 78–85.

[10] P. Genest and G. Lapalme, Fully abstractive approach to guided summarization, *Proceedings of the 50th AnnualMeeting of the Association for Computational*

Linguistics: Short Papers-Volume 2. Association for Computational Linguistics, 2012.

[11] P. Gervás, Story Generator Algorithms, *The Living Handbook of Narratology*, Hamburg University Press 19, 2012.

[12] P. Gervás, Computational Approaches to Storytelling and Creativity, *AI Magazine* **30**(3) (2009), 49–62.

[13] D. Gkatzia, H.F. Hastie and O. Lemon, Comparing Multi–Label Classification with Reinforcement Learning for Summarisation of Time–Series Data, in: *Association for Computational Linguistics* (1) (2014), 1231–1240.

[14] A. Isard, C. Brockmann and J. Oberlander, Individuality and Alignment in Generated Dialogues, in: *Proceedings of the International Natural Language Generation Conference*, Association for Computational Linguistics, (2006), 25–32.

[15] I. Konstas and M. Lapata, A GlobalModel for Concept–to–Text Generation, *Journal of Artificial Intelligence Research* **48** (2013), 305–346.

[16] G. Lampouras and I. Androutsopoulos, Using Integer Linear Programming in Concept–to–Text Generation to Produce More Compact Texts, in: *Association for Computational Linguistics* **(2)** (2013), 561–566.

[17] I. Langkilde and K. Knight, Generation That Exploits Corpus–Based Statistical Knowledge, in: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics–Volume 1*, Association for Computational Linguistics, (1998), 704–10.

[18] S. Liu et al., Positional Language Modelling for Extractive Broadcast News Speech Summarization, in: *INTERSPEECH* (2015), 2729–2733.

[19] P.V. Lobo and D.M. De Matos, Fairy Tale Corpus Organization Using Latent Semantic Mapping and an Item-to-Item Top-N Recommendation Algorithm, in: *Language Resources and Evaluation Conference*, 2010.

[20] S.M. Lukin, L.I. Reed and M.A. Walker, Generating Sentence Planning Variations for Story Telling, in: *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue* (2015), 188.

[21] Y. Lv, Yuanhua and C. Zhai, Positional Language Models for Information Retrieval, in: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in: Information Retrieval*, New York (2009), 299–306.

[22] F. Mairesse and S. Young, Stochastic Language Generation in Dialogue Using Factored Language Models, *Computational Linguistics*, **40**(4), 2014, 763–799.

[23] K. Mckeown, Discourse Planning with an N-Gram Model of Relations, *Empirical Methods on Natural Language Processing (September)* (2015), 1973–1977.

[24] R. Navigli and S.P. Ponzetto, BabelNet: Building a Very Large Multilingual Semantic Network, in: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics (2010), 216–25.

[25] E.M. Novais and I. Paraboni, Portuguese Text Generation Using Factored Language Models, *Journal of the Brazilian Computer Society* **19**(2) (2012), 135–46.

[26] Ll. Padró and E. Stanilovsky, FreeLing 3.0: Towards Wider Multilinguality, in: *Proceedings of the 8ht International Conference on Language Resources and Evaluation*, European Language Resources Association, 2012.

[27] N. Pourdamghani, K. Knight and U. Hermjakob, Generating English from Abstract Meaning Representations, in: *Proceedings of the 9th International Natural Language Generation Conference*, Association for Computational Linguistics (2016), 21–25.

[28] A. Ramos-Soto, A.J. Bugarín, S. Barro and J. Taboada, Linguistic Descriptions for Automatic Generation of Textual Short-Term Weather Forecasts on Real Prediction Data, in: *IEEE Transactions on Fuzzy Systems* **23**(1) (2015), 44–57.

[29] E. Reiter, R. Dale and Z. Feng, Building Natural Language Generation Systems Resumen ar Xivpreprint, http://arxiv.org/pdf/cmp-lg/9605002v1.pdf, 2000.

[30] R.M. Sayed, L. PerezBeltrachini and C. Gardent, Category-Driven Content Selection, in: *The 9th International Natural Language Generation Conference*, (2016), 94.

[31] R. Speer and C. Havasi, Representing General Relational Knowledge in ConceptNet 5, *in Language Resources and Evaluation Conference* (2012), 3679–3686.

[32] A. Stolcke, SRILM - An Extensible Language Modelling Toolkit, in: *Proceedings International Conference on Spoken Language Processing*, **2** (2002), 901–904.

[33] M. Theune, N. Slabbers and F. Hielkema, The Automatic Generation of Narratives, in: *Proceedings of the 17th Meeting of Computational Linguistics in: The Netherlands, LOT Occasional Series* (2007), 131–146.

[34] S.R. Turner, Minstrel: A Computer Model of Creativity and Storytelling, Ph. D. Dissertation. University of California at Los Angeles, 1993.

[35] M. White, Towards Surface Realization with CCGs Induced from Dependencies, in: *Proceedings of the 8th International Natural Language Generation Conference*, Association for Computational Linguistics (2014), 147–151.

[36] D.R. Winer and R.M. Young, Discourse-Driven Narrative Generation with Bipartite Planning, in: *The 9th International Conference on Natural Language Generation* (2016), 11–20.

[37] Simple NLG: A Realisation Engine for Practical Applications, http://www.aclweb.org/anthology/W09-0613, 2014.