A Synset-Based Recommender Method for Mixed-Initiative Narrative World Creation

Perez, Mijael R.Bueno; Eisemann, Elmar; Bidarra, Rafael

DOI
10.1007/978-3-030-92300-6_2

Publication date
2021

Document Version
Final published version

Published in
Interactive Storytelling - 14th International Conference on Interactive Digital Storytelling, ICIDS 2021, Proceedings

Citation (APA)
Perez, M. R. B., Eisemann, E., & Bidarra, R. (2021). A Synset-Based Recommender Method for Mixed-Initiative Narrative World Creation. In A. Mitchell, & M. Vosmeer (Eds.), *Interactive Storytelling - 14th International Conference on Interactive Digital Storytelling, ICIDS 2021, Proceedings* (pp. 13-28). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 13138 LNCS). Springer. https://doi.org/10.1007/978-3-030-92300-6_2

A Synset-Based Recommender Method for Mixed-Initiative Narrative World Creation

Mijael R. Bueno Perez(✉), Elmar Eisemann, and Rafael Bidarra

Delft University of Technology, Delft, The Netherlands
{M.R.BuenoPerez,E.Eisemann,R.Bidarra}@tudelft.nl

Abstract. A narrative world (NW) is an environment which supports enacting a given story. Manually creating virtual NWs (e.g. for games and films) requires considerable creative and technical skills, in addition to a deep understanding of the story in question. Procedural generation methods, in turn, generally lack in creativity and have a hard time coping with the numerous degrees of freedom left open by a story. In contrast, mixed-initiative approaches offer a promising path to solve this tension. We propose a mixed-initiative approach assisting an NW designer in choosing plausible entities for the locations, where the story takes place. Our approach is based on a recommender method that uses common and novel associations to narrative locations, actions and entities. Our method builds upon a large dataset of co-occurrences of disambiguated terms that we retrieved from photo captions. Building on this knowledge, our solution deploys entity (un)relatedness, offers clusters of semantically and contextually related entities, and highlights novelty of recommended content, thus effectively supporting the designer's creative task, while helping to stay consistent with the story. We demonstrate our method via an interactive prototype called TaleForge. Designers can obtain meaningful entity suggestions for their NWs, which enables guided exploration, while preserving creative freedom. We present an example of the interactive workflow of our method, and illustrate its usefulness.

Keywords: Narrative world · Recommender method · Authoring tool · Mixed initiative · Synset vectors

1 Introduction

Narrative Worlds (NW) are environments designed to support enacting a given story [2]. In the fast-paced industry of games and films, NWs are becoming more realistic and complex in nature. NW designers need new techniques and workflows to rapidly explore their creative ideas while preserving story consistency and integrity. Tools such as the game engines *Unity* or *Unreal Engine* have become increasingly popular for the creation of virtual worlds in form of game levels or virtual film production. Yet, these tools lack any narrative understanding and leave designers with full creative responsibility.

A. Mitchell and M. Vosmeer (Eds.): ICIDS 2021, LNCS 13138, pp. 13–28, 2021.
https://doi.org/10.1007/978-3-030-92300-6_2

Procedural content generation (PCG) techniques support designers in automating the generation of specific types of content [29]. PCG methods often involve a trial and error process due to their stochastic nature, and their frequent lack of intuitive control [1]. Also, most PCG methods fail to accurately capture the creative intent of a designer, as well as to properly combine and harmonize their generative process with other types of content [14], e.g., a story. In contrast, *mixed-initiative approaches* have proven to support designer creativity in different domains [7]. Mixed-initiative methods combine the strengths of PCG approaches with the input of a human designer, often in an iterative process. Thus, rather than randomly generating content, mixed-initiative methods try to adapt to the creative intent and methodology of a designer, following their chosen directions, preferences and choices.

Due to the complexity of NWs and their strong dependency on a story, a mixed-initiative approach can use both the story and choices of a designer to provide suggestions throughout the creation process. In this paper, we propose a mixed-initiative approach to assist a designer in choosing appropriate entities for an NW. Our approach is based on a recommender method that uses the story and designer guidance to suggest entities based on learned real-world associations. Our method takes into account narrative locations and actions as well as additional entities chosen by the designer.

At its core, our method relies on an embedding of synset vectors. A synset is a set of synonyms that share a common meaning. The embedding encodes their contextual associations. This representation is similar to word vectors but instead of using words, we use previously disambiguated terms annotated with synsets from WordNet[1]. We learned these synset vectors by extracting the co-occurrences of synsets found in photo captions obtained from *Shutterstock* and by using the unsupervised learning algorithm GloVe [20]. A vector representation allows us to perform several operations such as relatedness/similarity search, vector negation for unrelatedness and hierarchical clustering. In addition, a synset representation allow us to take advantage of WordNet's semantic knowledge for categorization and highlighting novelty over the recommended content. With our solution TaleForge, designers can easily populate NWs with relevant entities with little effort, while maintaining creative control.

2 Related Work

In this section, we examine mixed-initiative approaches, as well as strategies to encode contextual associations, as they form the basis of our solution.

Mixed-Initiative Content Creation. A mixed-initiative approach is based on a human-computer collaboration paradigm; a creative dialogue to co-create content [7,27,34]. Sketchaworld facilitates non-technical users to create complete 3D worlds through the integration of procedural techniques [30]. Tanagra uses human-computer collaboration to produce levels of a 2D platformer

[1] WordNet is an open lexical database of synsets, wordnet.princeton.edu.

by respecting user-specified constraints [31]. Sentient Sketchbook assists in the creation of maps for strategy or roguelike games by providing suggestions to a designer and improving upon their choices with novelty search [15]. Ropossum helps users to create and test puzzle levels by using grammatical evolution [26]. Linden et al. generate dungeons that fulfill high-level designer-specified gameplay requirements [16].

Other work has focused on mixed-initiative story authoring. Wide Ruled is an interactive interface that generates stories based on author's specified story world and goals [28]. WritingBuddy aims to help authors with the generation of story beats and actions by using the social simulation engine Ensemble [24]. TaleBox is a game that supports players to collaboratively create stories [5]. It uses GluNet, a semantic knowledge base that integrates several lexical databases for commonsense reasoning of narratives [12].

Only little work has focused on using a story as a basis for assisting an NW designer [13], and no solution has examined our goal: suggesting and supporting the exploration of suitable entities for an NW. GameForge is a PCG method that helps a designer to create NWs for computer role-playing games by using a genetic algorithm that balances story requirements, designer control and player preferences [10]. The designer controls the algorithm by setting adjacency probabilities between narrative locations. ScriptViz assists writers by executing their screenplays in real time with computer graphics [17]. It conveys a scene, with a location, camera position, characters and their poses.

Word Vectors for Contextual Associations. Hand-crafted semantic knowledge has been extensively used for expanding story-world domains [21,22]. In this paper, we extract semantic knowledge from automatically learned contextual associations among locations, objects and actions described in photo captions, which allows us to support a designer in their task of choosing NW content. Fortunately, word vector models have been effectively used for learning these associations. Obj-GloVe bridged language and vision by learning word vectors for common visual objects [33]. ViCo learned visual co-occurrences between objects and attributes in annotated photos [9] and `vis-w2v` learned co-occurrences of visual cues (e.g., objects, actions) from abstract scenes [11]. In addition, some models provide novel associations due to their ability to capture high-order relations (e.g., *frisbee* related to *ball*, due to *frisbee* to *dog* and *dog* to *ball*) [25].

3 Mixed-Initiative Approach

In this section, we describe the basis of our mixed-initiative approach, as well as its interactive workflow, both illustrated in Fig. 1.

First, assume that an NW designer has a working space (e.g., game engine or level editor) that allows to create virtual locations for an NW; we refer to this as NW *canvas*. We improve the designer's workflow with a recommender method, which also supports exploration. Our method suggests plausible entities for an NW based on common and novel associations to narrative locations, actions and other entities. Similar to a screenplay, the designer first selects a

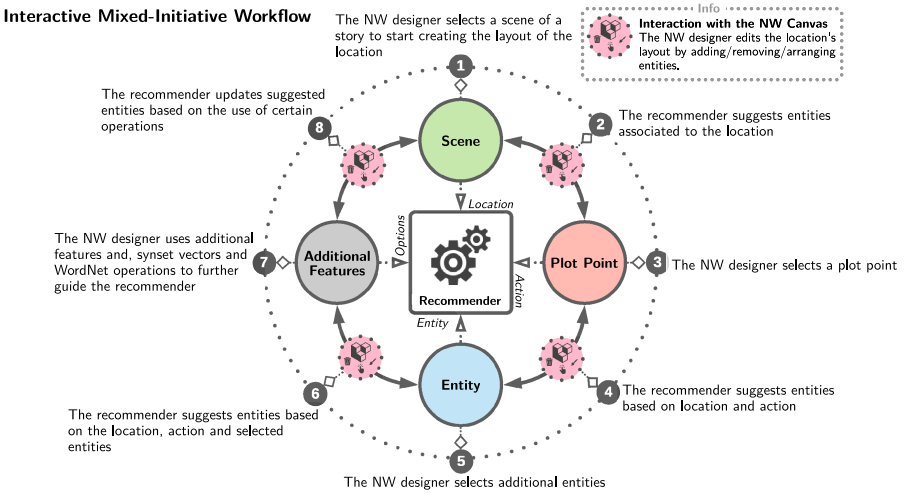

Fig. 1. A proposed interactive workflow for our mixed-initiative approach. We arbitrarily chose the steps shown in this illustration.

particular scene of the story and an NW canvas is shown with the location's layout of the scene; the recommender then suggests entities associated only to the location. The designer iteratively adds, removes and arranges entities by exploring recommendations or searching for specific content. The designer walks through each plot point or selects entities to further guide the recommender. At this point, the recommendations are based on the location, action and selected entities. The designer can select additional features to guide the exploration process. In the following subsections, we describe in detail the main components of our approach.

3.1 Story Elements, Structure and Representation

Our approach focuses on representing the lexical elements (terms in a vocabulary) and structure of a story. In a broad sense, a story can be seen as a sequence of events involving particular entities and their relations. We use the concept of *entity* to refer to the lexical and physical elements specified in the story and existing in the NW. An *entity* can be *abstract* (e.g., love) or *concrete* (e.g., desk). In addition, we adopted the hierarchy of two basic elements of a screenplay: *scene* and *action*. A *scene* is a story unit describing the particular *location* and sequence of actions at a specific *time*. A *location* refers to the particular space of the scene; it can be the *interior* or *exterior* of a *room* (e.g., bedroom), *natural area* (e.g., lake), or other. The *time* describes the point of the day (e.g., day, night, etc.) that hints about, for example, the potential illumination required for the location during the scene. An *action* describes what a character does in a given moment of the scene, as well as the involved entities. It can refer to a character (e.g., man) performing a certain activity (e.g., play tennis) and interacting with certain physical objects (e.g., racquet) and other characters (e.g., boy).

We represent an action and its involved entities by means of a *plot point* (*pp*); a semantically-coherent structure which describes an important event of a story. The argument structure of a plot point is determined by the verb of an action (e.g., eat, give) and involved entities, each slot denoted by a *semantic role* (SR). SR names can be customized to provide clarity about the role an entity has in the plot point. A general structure of the plot point representation with N slots is as follows:

$$\underset{\text{Subject}}{[\texttt{SLOT-0}]_{\text{SR}_0}}\ \underset{\text{Verb}}{[\texttt{SLOT-VERB}]}\ \underset{\text{Direct Object}}{[\texttt{SLOT-1}]_{\text{SR}_1}}\ \ldots\ \underset{\ldots}{[\texttt{SLOT-N}]_{\text{SR}_n}}$$

The number of slots and their corresponding SRs depend on the common argument structure of a verb. In our approach, we used the argument structure of verbs and SRs already provided by the hand-crafted lexical-semantic resource VerbAtlas[2] [8]. To illustrate our representation, consider a story with a scene happening in a *living room* in the *evening*, with this sequence of two plot points:

$$\begin{aligned} S_1.\ & \underset{\text{Space}}{[\texttt{INT.}]}\ \underset{\text{Location}}{[\texttt{LIVING ROOM}]}\ \text{-}\ \underset{\text{Time}}{[\texttt{EVENING}]} \\ pp_1.\ & \underset{\text{Subject}}{[\texttt{Bob}]_{\text{Agent}}}\ \underset{\text{Verb}}{[\texttt{Give}]}\ \underset{\text{Direct Object}}{[\texttt{Pizza}]_{\text{Theme}}}\ \underset{\text{Indirect Object}}{[\texttt{Sally}]_{\text{Recipient}}} \\ pp_2.\ & \underset{\text{Subject}}{[\texttt{Sally}]_{\text{Agent}}}\ \underset{\text{Verb}}{[\texttt{Eat}]}\ \underset{\text{Direct Object}}{[\texttt{Pizza}]_{\text{Patient}}} \end{aligned}$$

The first line is a scene description S_1, followed by the two plot points, pp_1 and pp_2. In pp_1, *give* is used as a transitive verb of three argument slots: the subject *Bob* is the Agent who initiates the action, the direct object *Sally* is the Recipient, and the indirect object *pizza* is the thing being transferred as Theme. In pp_2, *eat* is used as a transitive verb of two argument slots; the subject *Sally* is the Agent who performs the action, and the direct object *pizza* is the thing affected as Patient.

3.2 Narrative World Content

An NW can have several locations, each of them decorated with physical entities that are coherent with associations suggested by a story [2]. We identify two types of NW content: *explicit content* and *plausible content*. The *explicit content* consists of every entity specified in a plot point. For example, in *"Bob drinks coffee"*, *coffee* must exist in the location to support the action *drink*. In contrast, *plausible content* is every entity that potentially fits in the location, but their implicit nature is more of an open question. These entities might be required or only serve as decoration for the location. Traditionally, designers determine plausible content based on their own creative experience. Our method assists designers with suggestions based on the following learned associations:

- *Location-centric associations*. The entities associated to a location, e.g., *bed*, *closet* and *alarm clock* are commonly found in a *bedroom*.

[2] VerbAtlas provides semantically coherent structures of verbs, see verbatlas.org.

- *Action-centric associations*. The entities associated to an action/activity, e.g., *tv* and *popcorn* found when *watching a movie*.
- *Object-centric associations*. The entities associated with an object, e.g., *pen* to *paper*, *tennis racquet* to *tennis ball*, etc.
- *Character-centric associations*. The entities stereotypically associated to a character, like *film director* to *movie camera*, *doctor* to *stethoscope*, etc.

Plausible content can vary depending on the combination of location, action and even other entities involved. For example, entities during *sleep* vary if the location is a *mountain* or *bedroom*. On a *mountain*, *sleep* might suggest *tent* and *sleeping bag*. In contrast, *sleep* in a *bedroom*, might result in *bed* and *mattress*.

3.3 Learning Synset Vectors

Learning common and novel associations for recommending NW content given narrative locations, actions and entities is a challenging problem. If we assume that these associations resemble what co-occurs in the real world, then we can rely on many exemplars to cover the vast number of cases. To address this, we looked into textual knowledge found on the web, assuming that co-occurrence data within photo captions are a good basis for learning the aforementioned associations. We represented this co-occurrence data as an embedding of synset vectors, which allowed several vector operations for suggesting NW content.

Photo Captions Dataset. We created a dataset based on the photo captions and keywords provided by users of stock photography services. We found that in this way we could include a vast number of real-world associations. We extracted 99.7 million photo captions by using the *Shutterstock* API service[3].

Synset Representation. We identify entities as synsets instead of as words. A synset is a set of synonyms with shared meaning. Lexical databases such as WordNet and BabelNet, contain a rich glossary of synsets and semantic knowledge for defining their meaning [18,19]. We used WordNet and identified a synset entry with a WordNet ID, composed of a letter for *part of speech* (e.g., *n* for noun) followed by 8 digits (e.g., `n02958343` for a *car* synset).

Dataset Preprocessing. We preprocessed the dataset by removing or replacing characters that were not alphanumeric, tokenizing single-word (e.g., cat) and multi-word (e.g., tennis ball) expressions, and tagging words with a WordNet ID by using the *adapted lesk algorithm* for word sense disambiguation [3]. This algorithm compares the context (a photo caption) of a target word with WordNet's semantic knowledge of candidate synsets to determine a plausible synset for the word. Then, we extracted a *dependency tree* of each caption by using python's *spaCy*[4]. A *dependency tree* is a hierarchic representation of the syntactic structure of a sentence. From this, we captured verb and direct object as a

[3] see api.shutterstock.com

[4] spaCy has pre-trained models, which perform many NLP tasks (see spacy.io).

bigram term of an action/activity, e.g., *play tennis* is tagged with WordNet IDs of *play* and *tennis* separated by a semicolon `v01072949:n00482298`. In total, we tagged 3.75 billion tokens (an average of 38 tokens per caption).

Vocabulary and Co-occurrences. We extracted a vocabulary of all synsets V available in the dataset and kept only synsets that occur over 10 times. Our vocabulary contains 470,577 terms: 78,400 unigram WordNet IDs and 392,177 bigram WordNet IDs for actions/activities. Then, we aggregated the pairwise co-occurrences of all synsets within each photo caption to the global co-occurrences counts X_{ij}. We captured 457.2 million of such co-occurrence pairs.

GloVe Formulation. GloVe is a widely used unsupervised learning algorithm for learning word vectors [20]. It uses a log-bilinear model to optimize the d-dimensional embeddings $w_i \in \mathbb{R}^d$ by using the non-zero co-occurrence counts X_{ij} in the following objective:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2 \tag{1}$$

We optimized a single embedding layer for both w_i and w_j instead of an additional embedding layer for the context vectors $\tilde{w}_j$. This should not make a difference due to the symmetry in the objective, ideally both embeddings should be identical [9]. In addition, the original formulation [20] uses a weighting function $f(X_{ij})$, but offers just an empirical motivation for it; instead, we use $f(X_{ij}) = 1$. With these simplifications, we did not observe any impact on the quality of our synset vectors. We learned and tested synset vectors of different d-dimensions: 50, 100, 200 and 300 dimensions.

3.4 The Recommender Method

Our mixed-initiative approach is powered by a recommender method able to compute and deploy related entities by using several vector and WordNet operations. In this subsection, we discuss the different components of the recommender, such as input/output and operations (see Fig. 2).

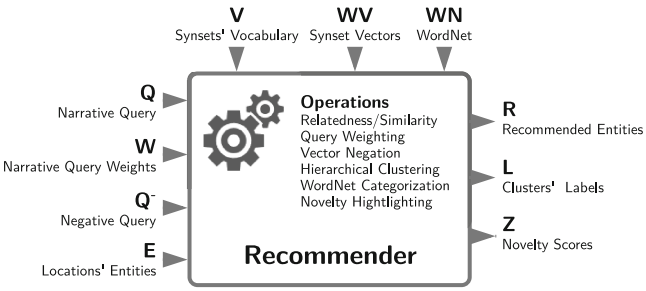

Fig. 2. Schematic of inputs/outputs of the recommender.

The Narrative Query. We define a *narrative query* as a set of synsets $Q = \{q_1, q_2, ..., q_n\}$ that prompt the recommender to output a set of k most related synsets $R = \{r_1, r_2, ..., r_k\}$. The argument k is set by the designer to control the number of results. A narrative query is formed by extracting the synsets of location, action and designer-selected entities (e.g. in the plot point or already placed in the NW canvas). If the plot point involves a direct object, the narrative query includes a combined term for verb and direct object; if, however, the combined term does not exist in our vocabulary, only the verb is included in Q.

Relatedness/Similarity Search. This search operation is based on the assumption that similar or related terms appear closer together for a certain distance metric in a semantic vector space. We use the *cosine similarity score* $\cos(\vec{A}, \vec{B})$ as a quantifiable measure of similarity between two synset vectors with values between 1 (similar directions) and -1 (opposite directions). Then, we compute the *angular cosine distance* φ_{dist} of values from 0 to 1 to respect the notion of similarity based on closeness:

$$\varphi_{dist}(\vec{A}, \vec{B}) = \frac{\arccos(\cos(\vec{A}, \vec{B}))}{\pi} \qquad \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\|\|\vec{B}\|} \tag{2}$$

We obtain a set of related entities R by computing the average angular distance $\bar{\varphi}$ of each synset vector $\vec{v_j}$ in the vocabulary V to every synset vector $\vec{q_i}$ in the narrative query Q:

$$\bar{\varphi}_{dists}(Q, V) = \{\bar{\varphi} \in \mathbb{R} : \bar{\varphi} = \frac{1}{|Q|} \sum_{\vec{q_i} \in Q} \varphi_{dist}(\vec{q_i}, \vec{v_j}),\ \forall \vec{v_j} \in V\} \tag{3}$$

Then, we sort the result according to the average angular distances $\bar{\varphi}_{dists}$ and select the k entities with the smallest distances.

Narrative Query Weighting. We determined that computing related entities by considering narrative query terms q_i of equal importance is not an optimal approach. The designer might be interested in controlling the association strength of location, action or selected entities. Thus, we propose a controllable weighting scheme with three possible weight values: the *location weight* w_l, *action weight* w_a and *selection weight* w_e. We empirically set $w_l = 0.5$, $w_a = 0.7$ and $w_e = 1$ by default. These values are included in a set $W = \{w_1, w_2, ..., w_n\}$, where $w_i \in W$ corresponds with the position of its query term $q_i \in Q$. We expand Eq. (3) to compute weighted averaged distances $\bar{\varphi}_w$ with W as follows:

$$\bar{\varphi}_{dists}(Q, W, V) = \{\bar{\varphi}_w \in \mathbb{R} : \bar{\varphi}_w = \frac{\sum_{i=1}^{n} w_i\, \varphi_{dist}(\vec{q_i}, \vec{v_j})}{\sum_{i=1}^{n} w_i},\ \forall \vec{v_j} \in V\} \tag{4}$$

Vector Negation for Unrelatedness. The designer can provide a *negative query* as a set of synsets $Q^- = \{q_1^-, q_2^-, ..., q_n^-\}$ that must appear unrelated to the

results, analogous to a boolean NOT query (e.g. *animal* NOT *bird*). Thus, we perform *vector negation* as Q NOT Q^- to extract a subspace Q^+ of vectors that has no features in common with Q^- [32]. We obtain $\vec{q_i}^+ \in Q^+$ by subtracting the vector projections of all synset vectors $\vec{q_j}^- \in Q^-$ on each $\vec{q_i} \in Q$:

$$Q^+ = \{\vec{q_i}^+ \in Q^+ : \vec{q_i}^+ = \vec{q_i} - \sum_{\vec{q_j}^- \in Q^-} \frac{\vec{q_i} \cdot \vec{q_j}^-}{||\vec{q_j}^-||^2} \vec{q_j}^-, \forall \vec{q_i} \in Q\} \tag{5}$$

We replace Q for Q^+ in Eq. (4) as $\bar{\varphi}_{dists}(Q^+, W, V)$ and retrieve the top k entities with shortest distance. Interestingly, this feature can be used to neutralize common human subjective biases (e.g., gender and cultural biases) prone to be found in models learned from text corpora [4].

Hierarchical Clustering. We observed that recommended entities belong to meaningful clusters, which reflect similarity or contextual relatedness; for example, several types of animals for a forest. To this extent, we present the recommended content clustered, using a set of labels $L = \{l_1, l_2, ..., l_n\}$ that associate a cluster number to each recommended entity $r_i \in R$. To determine clusters, we used the *agglomerative clustering algorithm* to iteratively fuse clusters (starting from clusters containing a single entity) based on a distance measure (linkage) between clusters [6]. We chose the *complete-linkage*[5] criterion; the distance between two clusters is the maximum pairwise distance between their entities:

$$\varphi_{dist}(C_a, C_b) = \max\{\varphi_{dist}(\vec{a}_i, \vec{b}_j) : \vec{a}_i \in C_a, \vec{b}_j \in C_b\} \tag{6}$$

We chose this criterion due to the tendency to form smaller, compact clusters. The optimal clusters are determined via the *silhouette coefficient* [23]: a value from -1 (bad fit) to 1 (good fit) which measures the quality of the clusters in terms of how good each entity fits in its own cluster. Given the mean intra-cluster distance a_i and the mean nearest-cluster distance b_i, we compute the coefficient s_i for each entity and the average coefficient $\bar{s}$ as follows:

$$s_i = \frac{\bar{b_i} - \bar{a_i}}{\max(\bar{a_i}, \bar{b_i})} \qquad \bar{s} = \frac{1}{|R|} \sum_{i=1}^{|R|} s_i \tag{7}$$

We further constrain clusters by only considering the steps where their size is below the maximum threshold $T_{max} = 3$ to ensure compact clusters. For convenience, we sort clusters in ascending order by the average of the score obtained with Eq. (4) for entities $\vec{c_j}$ in each cluster $C_k \in C$:

$$\bar{\varphi}_{cdists}(Q, W, C) = \{\varphi_c \in \mathbb{R} : \varphi_c = \frac{1}{|C_k|} \sum_{\vec{c_j} \in C_k} \bar{\varphi}_{dists}(Q, W, C_k), \forall C_k \in C\} \tag{8}$$

[5] For hierarchical clustering visit nlp.stanford.edu/IR-book/completelink.html

WordNet Categorization. The designer might be interested in exploring certain categories of entities (e.g., *plants* for *forest*, *furniture* for *bedroom*, etc.). Fortunately, WordNet allows us to create custom sets of synsets $V_c \subset V$ based on *lexnames* and *hyponyms* of synsets. A *lexname* is one of 45 tags for each synset in WordNet (e.g., noun.animal, noun.plant, etc.) and *hyponyms* are less generic synsets below a synset's semantic tree, for example, *car* and *motorcycle* are hyponyms of *vehicle*. We can perform any previous operation with a custom set V_c instead of the whole vocabulary V. To showcase this feature, we created custom sets for *artifact*, *tool*, *furniture*, *food*, *substance*, *plant*, etc.

Novelty Highlighting. We highlight novelty of a recommended entity based on a novelty score that measures how new the entity is for the current location. First, we compute a dictionary of counts $X : H \rightarrow V$ of WordNet's *hypernyms* and *hyponyms* H of all synsets of entities $e_i \in E$ placed in the location. Then, we obtain $g_i \in G$ as the $\log_{10}$ of sum of counts $X(h)$ of only intersecting hypernyms and hyponyms $h \in (H_i \cap H)$ of the synset of a recommended entity $r_i \in R$. We use $\log_{10}$ to smooth out the influence of large counts. Lastly, we calculate a novelty score $z_i \in Z$ with min-max normalization in reverse order of G to get values from 0 (least novel) to 1 (most novel):

$$G = \{g_i \in \mathbb{R} : g_i = \log_{10} \sum_{h \in (H_i \cap H)} X(h)\} \quad Z = \{z_i \in \mathbb{R} : z_i = \frac{\max(G) - g_i}{\max(G) - \min(G)},\ g_i \in G\} \tag{9}$$

4 Interactive Prototype

We implemented TaleForge, an interactive prototype to showcase our mixed-initiative approach. For this, we use a story of three scenes in three locations: a *forest*, *beach*, and *tavern*; see Fig. 3. The story is about a *pirate* stranded in an uninhabited island. The *pirate* performs several actions to survive, and finds a *treasure* while trying to repair his *ship*. Eventually, he ends up celebrating in a *tavern*, plays a game against the *barkeeper* and gives his *treasure* away. Assume that a designer is looking for inspiration to further design an NW for this story.

S1 [EXT.] [FOREST] - [EVENING]
Space Location Time
pp_1. [Pirate] [Look_for] [Food]
AGENT Verb THEME
Subj DObj
pp_2. [Pirate] [Make] [Fire]
AGENT Verb PATIENT
Subj DObj
pp_3. [Pirate] [Prepare] [Food]
AGENT Verb PATIENT
Subj DObj

S2 [EXT.] [BEACH] - [MORNING]
Space Location Time
pp_4. [Pirate] [Cut] [Tree]
AGENT Verb PATIENT
Subj DObj
pp_5. [Pirate] [Repair] [Ship]
AGENT Verb PATIENT
Subj DObj
pp_6. [Pirate] [Find] [Treasure]
AGENT Verb THEME
Subj DObj

S3 [INT.] [TAVERN] - [NIGHT]
Space Location Time
pp_7. [Pirate] [Buy] [Drink] [Barkeeper]
AGENT Verb THEME SOURCE
Subj DObj IObj
pp_8. [Pirate] [Play] [Game] [Barkeeper]
AGENT Verb THEME CO-AGENT
Subj DObj IObj
pp_9. [Pirate] [Give] [Treasure] [Barkeeper]
AGENT Verb THEME RECIPIENT
Subj DObj IObj

Fig. 3. The story *"Once upon a time, a pirate..."*: with 3 plot point per scene/location.

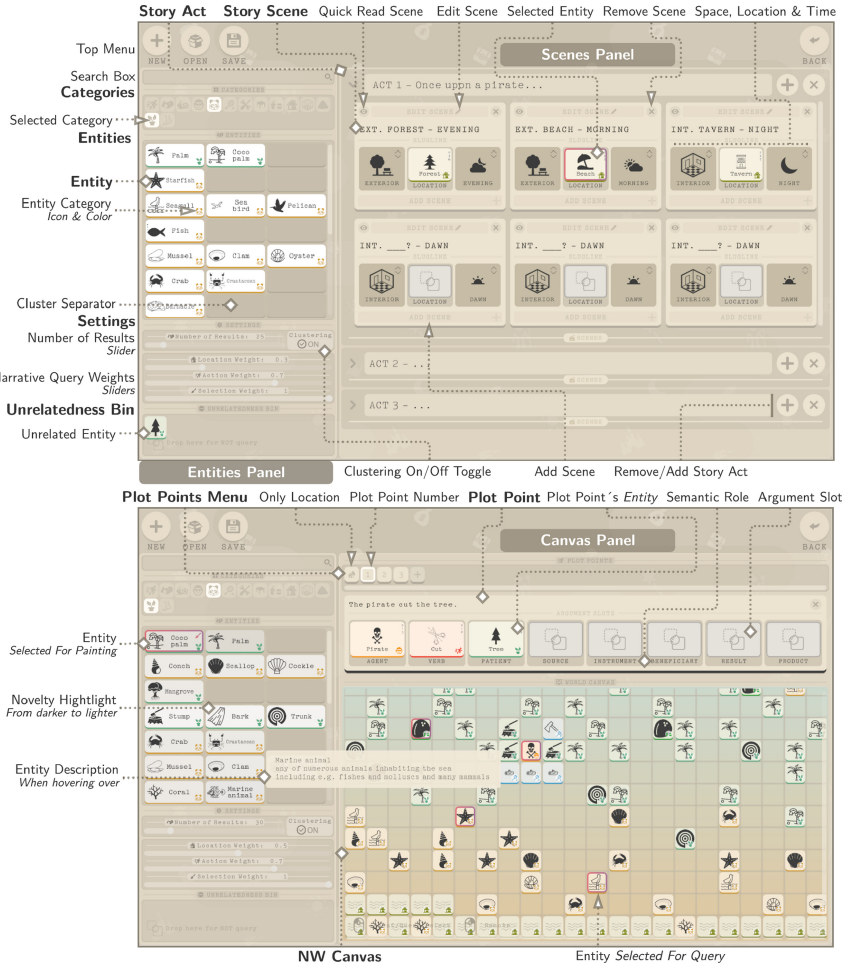

Fig. 4. TaleForge, an interactive prototype of our mixed-initiative approach.

In Fig. 4, we show three main panels of TaleForge: the *entities panel*, *scenes panel* and *canvas panel*. In the *entities panel*, the designer searches for entities, obtains recommendations and configures the recommender by setting categories, weights, number of results, clustering and an unrelatedness bin for NOT queries. An entity appears with a name, icon[6], category color and description when hovering over it. The novelty of a recommended entity is visualized as a lighter (more novel) to darker (least novel) background. In the *scenes panel*, the designer sees every scene of the story and selects one for editing the location's layout. In the *canvas panel*, the designer creates a location by placing entities into cells of the canvas. The designer can select individual plot points and entities to guide

[6] Entities' icons are populated automatically from thenounproject.com

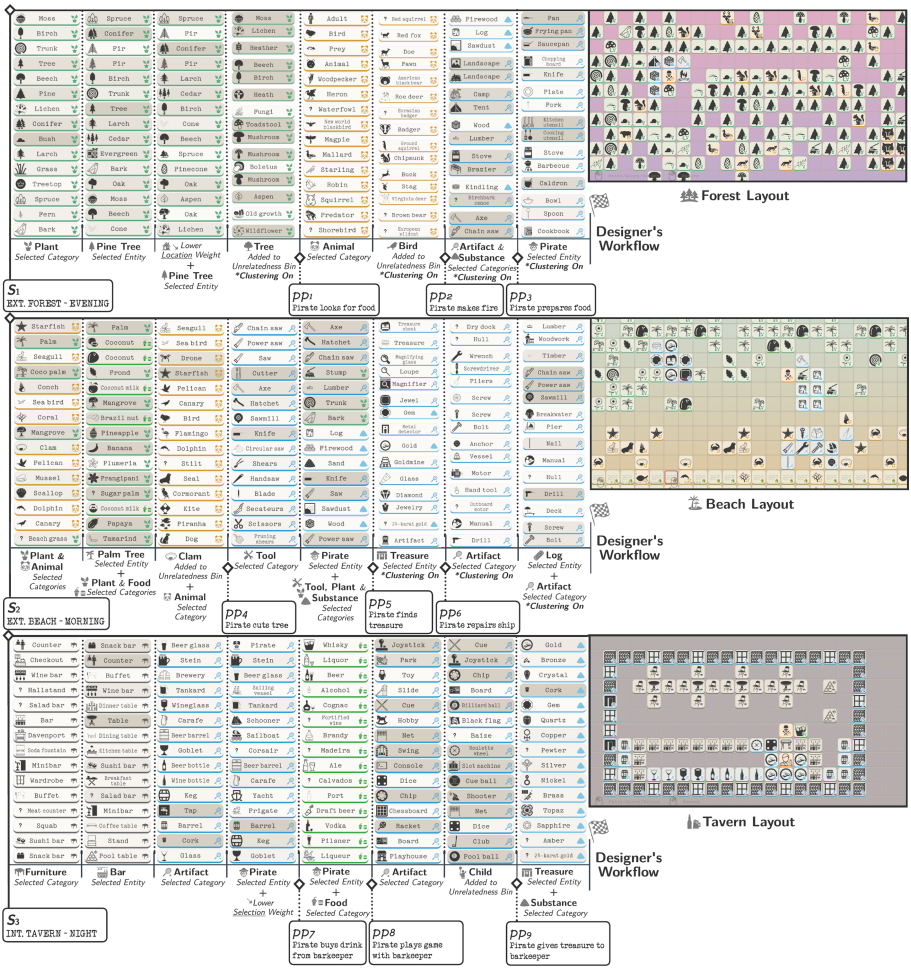

Fig. 5. Designer's workflow while creating an NW for *"Once upon a time, a pirate..."*.

the recommender. The front end of TaleForge was implemented in *Unity*, and the recommender method is served by a Python's *flask* API.

In Fig. 5, we show the recommender's output at various steps of the designer's workflow, together with the final layouts of the locations for our showcase story. To explain the designer's workflow, consider the first scene S_1 as an example. Here, the designer creates a layout for a *forest*. When selecting this scene, a plot points menu (top) and an empty NW canvas (bottom) is shown. The designer adds entities by searching for them and dragging/dropping them. Further, it is possible to *paint* on the canvas, e.g., to place several trees, by selecting the object and moving the mouse over the canvas, while holding the mouse button pressed. The recommender is prompted when interacting with any element. At first, it

only considers entities related to the location (the *forest*). The designer selects the category *plant* and adds recommended entities like *pine trees* and *bushes*. When entities are added to the canvas, the novelty scores are updated, thus, recommended entities such as *bark*, *trunk* and *lichen* appear lighter (more novel). The designer can enforce a different output of the recommender by changing the narrative query weights or using the unrelatedness bin. The designer prompts the recommender by selecting a *pine tree* and lowers the weight of the location to put more emphasis on the selected object, causing *pinecone* to appear in the top results. Next, the designer might be interested in entities that are not related to *tree*, so the designer drags and drops *tree* into the unrelatedness bin and activates clustering. Consequently, the recommender suggests and groups different types of *mushrooms*.

The designer selects a plot point to consider the influence of an action. In pp_1, the designer selects the category *animal* to add animals in the location for *finding food*. The output shows different *birds* which might not be interesting for the designer. Thus, the designer drags and drops *bird* to the unrelatedness bin and activates clustering to obtain groups of animals unrelated to *bird*, like *chipmunk*, *squirrel*, *deer*, and so on. In pp_2, the designer selects the category *artifact* and *substance* to find entities related to *making a fire*, such as *wood*, *firewood*, *axe*, and others. In pp_3, the recommender outputs entities for *preparing food*, however, these are common cooking utensils and might not be interesting enough for a story, where a *pirate* is involved. Thus, the designer selects *pirate* from the plot point and discovers *caldron* in the recommendations.

Figure 5 presents similar designer workflows for Scenes S_2 and S_3.

5 Conclusion

We have introduced a novel mixed-initiative approach based on a recommender method that assists designers throughout the creation of a narrative world (NW) for a given story. Our method discovers and suggests NW content based on previously-learned common and novel associations to current narrative locations, actions and entities. Our method provides several vector and WordNet operations that further expand the creative exploration of a designer. In addition, we have shown the usefulness of our approach through TaleForge, an interactive prototype that assists a designer with recommendations of entities for an NW.

Bringing mixed-initiative approaches into the workflows of NW designers can significantly improve the consistency between stories and NWs. The design of procedural tools to simplify artistic and technical tasks, as well as solutions to enhance designers exploration, are considered very important and challenging endeavours. However, there is a strong lack of research work on using a computational narrative as a basis for assisting NW designers. We believe this work provides a valuable step to a more complete NW authoring tool.

References

1. Amato, A.: Procedural content generation in the game industry. In: Korn, O., Lee, N. (eds.) Game Dynamics, pp. 15–25. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53088-8_2
2. Balint, J.T., Bidarra, R.: Procedural generation of narrative worlds (2021). Submitted for publication
3. Banerjee, S., Pedersen, T.: An adapted Lesk algorithm for word sense disambiguation using WordNet. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45715-1_11
4. Caliskan, A., Bryson, J.J., Narayanan, A.: Semantics derived automatically from language corpora contain human-like biases. Science **356**(6334), 183–186 (2017). https://arxiv.org/abs/1608.07187
5. Castaño, O., Kybartas, B., Bidarra, R.: Talebox-a mobile game for mixed-initiative story creation. In: Proceedings of DiGRA-FDG (2016). https://graphics.tudelft.nl/Publications-new/2016/CKB16/
6. Day, W.H., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. J. Classif. **1**(1), 7–24 (1984). https://doi.org/10.1007/BF01890115
7. Deterding, S., et al.: Mixed-initiative creative interfaces. In: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, pp. 628–635 (2017). https://doi.org/10.1145/3027063.3027072
8. Di Fabio, A., Conia, S., Navigli, R.: VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 627–637 (2019). https://doi.org/10.18653/v1/D19-1058
9. Gupta, T., Schwing, A., Hoiem, D.: ViCo: word embeddings from visual co-occurrences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7425–7434 (2019). https://arxiv.org/abs/1908.08527
10. Hartsook, K., Zook, A., Das, S., Riedl, M.O.: Toward supporting stories with procedurally generated game worlds. In: 2011 IEEE Conference on Computational Intelligence and Games (CIG 2011), pp. 297–304. IEEE (2011). https://doi.org/10.1109/CIG.2011.6032020
11. Kottur, S., Vedantam, R., Moura, J.M., Parikh, D.: Visual word2vec (vis-w2v): learning visually grounded word embeddings using abstract scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4985–4994 (2016). https://arxiv.org/abs/1511.07067
12. Kybartas, B., Bidarra, R.: A semantic foundation for mixed-initiative computational storytelling. In: Schoenau-Fog, H., Bruni, L.E., Louchart, S., Baceviciute, S. (eds.) ICIDS 2015. LNCS, vol. 9445, pp. 162–169. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27036-4_15
13. Kybartas, B., Bidarra, R.: A survey on story generation techniques for authoring computational narratives. IEEE Trans. Comput. Intell. AI Games **9**(3), 239–253 (2016). https://doi.org/10.1109/TCIAIG.2016.2546063
14. Liapis, A., Yannakakis, G.N., Nelson, M.J., Preuss, M., Bidarra, R.: Orchestrating game generation. IEEE Trans. Games **11**(1), 48–68 (2018). https://doi.org/10.1109/TG.2018.2870876
15. Liapis, A., Yannakakis, G.N., Togelius, J.: Sentient sketchbook: computer-aided game level authoring. In: FDG, pp. 213–220 (2013). http://julian.togelius.com/Liapis2013Sentient.pdf

16. Linden, R.V.D., Lopes, R., Bidarra, R.: Designing procedurally generated levels. In: Proceedings of the second AIIDE workshop on Artificial Intelligence in the Game Design Process, November 2013
17. Liu, Z.Q., Leung, K.M.: Script visualization (ScriptViz): a smart system that makes writing fun. Soft. Comput. **10**(1), 34–40 (2006). https://doi.org/10.1007/s00500-005-0461-4
18. Miller, G.A.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998). https://doi.org/10.7551/mitpress/7287.001.0001
19. Navigli, R., Ponzetto, S.P.: BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. **193**, 217–250 (2012). https://doi.org/10.1016/j.artint.2012.07.001
20. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014). https://doi.org/10.3115/v1/D14-1162
21. Porteous, J., Ferreira, J.F., Lindsay, A., Cavazza, M.: Extending narrative planning domains with linguistic resources. In: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), pp. 1081–1089. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS) (2020). https://dl.acm.org/doi/abs/10.5555/3398761.3398887
22. Porteous, J., Lindsay, A., Read, J., Truran, M., Cavazza, M.: Automated extension of narrative planning domains with antonymic operators. In: Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), May 2015. https://doi.org/10.5555/2772879.2773349
23. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987). https://doi.org/10.1016/0377-0427(87)90125-7
24. Samuel, B., Mateas, M., Wardrip-Fruin, N.: The design of writing buddy: a mixed-initiative approach towards computational story collaboration. In: Nack, F., Gordon, A.S. (eds.) ICIDS 2016. LNCS, vol. 10045, pp. 388–396. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48279-8_34
25. Schlechtweg, D., Oguz, C., Walde, S.S.I.: Second-order co-occurrence sensitivity of skip-gram with negative sampling. arXiv preprint arXiv:1906.02479 (2019). https://arxiv.org/abs/1906.02479
26. Shaker, N., Shaker, M., Togelius, J.: Ropossum: an authoring tool for designing, optimizing and solving cut the rope levels. In: Ninth Artificial Intelligence and Interactive Digital Entertainment Conference (2013). https://ojs.aaai.org/index.php/AIIDE/article/view/12611
27. Shaker, N., Togelius, J., Nelson, M.J.: Procedural Content Generation in Games. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-42716-4
28. Skorupski, J., Jayapalan, L., Marquez, S., Mateas, M.: Wide ruled: a friendly interface to author-goal based story generation. In: Cavazza, M., Donikian, S. (eds.) ICVS 2007. LNCS, vol. 4871, pp. 26–37. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77039-8_3
29. Smelik, R.M., Tutenel, T., Bidarra, R., Benes, B.: A survey on procedural modelling for virtual worlds. Comput. Graph. Forum **33**(6), 31–50 (2014). https://doi.org/10.1111/cgf.12276
30. Smelik, R.M., Tutenel, T., de Kraker, K.J., Bidarra, R.: Interactive creation of virtual worlds using procedural sketching. In: Eurographics (Short papers), pp. 29–32 (2010). https://doi.org/10.2312/egsh.20101040

31. Smith, G., Whitehead, J., Mateas, M.: Tanagra: a mixed-initiative level design tool. In: Proceedings of the Fifth International Conference on the Foundations of Digital Games, pp. 209–216. ACM (2010). https://doi.org/10.1145/1822348.1822376
32. Widdows, D.: Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 136–143 (2003). https://doi.org/10.3115/1075096.1075114
33. Xu, C., Chen, Z., Li, C.: Obj-GloVe: scene-based contextual object embedding. arXiv preprint arXiv:1907.01478 (2019). https://arxiv.org/abs/1907.01478
34. Yannakakis, G.N., Liapis, A., Alexopoulos, C.: Mixed-initiative co-creativity. In: FDG (2014). https://www.um.edu.mt/library/oar/handle/123456789/29459