The AAAI-17 Workshop on
What's Next for AI in Games?
WS-17-15

Towards Automatically Extracting Story Graphs from Natural Language Stories

Josep Valls-Vargas,[1] Jichen Zhu,[2] Santiago Ontañón[1]
[1]Computer Science, [2]Digital Media
Drexel University
Philadelphia, PA 19104, USA
josep.vallsvargas@drexel.edu, jichen.zhu@drexel.edu, santi@cs.drexel.edu

Abstract

This paper presents an approach to automatically extracting and representing narrative information from stories written in natural language. Specifically, we present our results in extracting *story graphs*, a formalism that captures the entities (e.g., characters, props, locations) and their interactions in a story. The long-term goal of this research is to automatically extract this narrative information in order to use it in computational narrative systems such as story generators or interactive fiction systems. Our approach combines narrative domain knowledge and off-the-shelf natural language processing (NLP) tools into a machine learning framework to build story graphs by automatically identifying entities, actions, and narrative roles. We report the performance of our fully automated system in a corpus of 21 stories and provide examples of the extracted story graphs and their uses in computational narrative systems.

Introduction

Computational narrative studies how to algorithmically represent, understand, and most importantly, generate stories. Computational narrative has applications in areas of digital entertainment such as interactive fiction or computer games and can provide insights into computational creativity (Turner 1993) and the analysis and understanding of literature (Elson, Dames, and McKeown 2010).

Computational narrative systems, especially story generation systems, require a significant amount of domain knowledge encoded in some form of structured formalism in order to function. Currently this information is mostly hand-authored, a notoriously time-consuming task requiring expertise in both storytelling and knowledge engineering. This well-known "authorial bottleneck" problem could be alleviated if narrative information could be automatically extracted from natural language since we could leverage the content in the vast amount of existing written literature.

In this paper, we present a fully automated system that can extract *story graphs* from natural language stories. These story graphs encode information from the entities in the text (e.g., characters, props, locations) and their interactions (e.g., a character *moves* to a location or *obtains* a prop). Story graphs are similar to the background knowledge required in existing story generation systems such as *Tale-spin* (Meehan 1976) or *Riu* (Ontañón and Zhu 2010), or automatic game generation systems such as Game Forge (Hartsook et al. 2011). Thus this paper constitutes a first step toward allowing these kind of systems to exploit information contained in natural language stories.

The proposed approach has been implemented into a system called *Voz*, which pre-processes the text using off-the-shelf natural language processing tools, and then uses a collection of machine-learning modules that exploit narrative domain knowledge to extract the different pieces of information required to construct the story graphs. We report results using a dataset consisting of 21 Russian stories manually translated to English, evaluating the quality of the resulting story graphs, and discussing the feasibility of utilizing these graphs directly as input to map and story generation systems.

In the rest of the paper, we first discuss previous work on computational narrative systems and the formalisms used, and research on extracting narrative information from natural language. Next, we present our automatic narrative extraction system and the story graphs it extracts. We follow with an empirical evaluation of the quality of the extracted story graphs, provide samples of the output of our system, and show a sample application to content generation for games. The paper closes with conclusions and future work.

Related Work

Most computational narrative systems require a significant amount of domain knowledge which is mostly hand-authored. Several variations of the *Planning Domain Definition Language* (PDDL) have been proposed to formalize the plot of a narrative or narrative space in order to generate stories. *Tale-spin* (Meehan 1976) was a pioneer computational narrative system that generated stories using planning.Other approaches to story generation, such as those based on case-based reasoning, or analogical methods, require background knowledge and story examples annotated in a machine readable format. *ProtoPropp* (Gervás et al. 2005) uses annotated stories and an ontology to generate stories matching a user query. The *Riu* system (Ontañón and Zhu 2010) uses computational analogy between manually annotated stories during an interactive storytelling session. Systems like *Game*

Forge (Hartsook et al. 2011) or the work by Valls-Vargas et al. (Valls-Vargas, Ontañón, and Zhu 2013) augment plot points with annotations for spatial restrictions or graphical realization in order to generate game environments.

There have been some efforts to standardize the process of adding computer-readable annotations to natural language stories, which would allow computational narrative systems to exploit the information in these stories. The *Proppian fairy tale Markup Language* (PftML) project (Malec 2001) proposes an annotation scheme standardize a formal analytical model for stories based on Propp's work (Propp 1973). The *NarrativeML* (Mani 2012) is a proposed markup language that seeks to annotate several narrative primitives, discourse and character information.

Previous work on extracting narrative structures from text include the work of Finlayson (2008), who created the *Story Workbench*, a semi-automatic tool that facilitates story annotation. Similar work has been done by Elson (2012b) in *Scheherazade*. Elson proposed a graph-like semantic encoding of a story called *Story Intention Graphs* (SIG). SIGs are annotated using *Scheherazade* and have been used to detect story analogies (Elson 2012a). Rishes et al. (2013) use SIGs to generate different story tellings by automatically learning rules to convert SIG to the input required for a natural language generation system. Harmon and Jhala (2015) explored converting the output of Skald (a reconstruction of Minstrel) into SIG. While SIGs encode much richer information than the story graphs proposed in this paper, these are authored manually whereas our goal is to extract a story representation automatically from unannotated text.

There is also research on automatically extracting narrative information. Goyal et al.'s AESOP system (Goyal, Riloff, and Daumé 2010) explored how to extract characters and their affect states from textual narrative in order to produce plot units (Lehnert 1981) for a subset of Aesop fables. The system uses both domain-specific assumptions (e.g., only two characters per fable) and external knowledge (word lists and hypernym relations in WordNet) in its character identification stage. Chambers and Jurafsky (Chambers and Jurafsky 2008) proposed using unsupervised induction to learn what they called "narrative event chains" from raw newswire text. In order to learn Schankian script-like information about the narrative world, they use unsupervised learning to detect the event structures as well as the roles of their participants without pre-defined frames, roles, or tagged corpora. In related work, Li et al. (2013) extract *plot graphs* to represent the events in a collection of stories describing a given theme (e.g., bank robbery). Also related is the body of work on text-to-scene conversion of Coyne et al. (Coyne and Sproat 2001) and Chang et al. (Chang, Savva, and Manning 2014).

Our past work involves the automatic identification of characters and their narrative roles in stories so the stories can be used as input for systems such as Riu (Ontañón and Zhu 2010). In this paper we focus on extracting a graph representation of a narrative that includes all entities and can later be used as input to computational narrative systems that require a structured story representation. Another possible application of our story graphs could be the automated analysis and visualization of literature works in terms of interactions between characters similar to the work of Elson et al. (Elson, Dames, and McKeown 2010). We also explore areas of application related to the spatial configuration of story worlds that could be used with systems like *Game Forge* (Hartsook et al. 2011) or the work by Valls-Vargas et al. (2013).

Automatically Extracting Story Graphs

In this section we describe our fully-automated narrative extraction system called *Voz* and the story graphs it extracts.

System Architecture

Voz is a narrative information extraction system. Given the text of a story, *Voz* uses off-the-shelf natural language processing (NLP) tools, commonsense knowledge, narrative domain knowledge, and machine learning approaches to extract, enrich, classify and finally compile narrative information into a graph representing the original story. Figure 1 illustrates the architecture of the system and the main processes described in this section.

Extraction: *Voz* uses the Stanford CoreNLP suite to segment the input text into sentences and annotate them with several layers of NLP information (*i.e.*, part-of-speech tags, syntactic parse trees, coreference information and typed dependencies). Then the *mention extraction* process identifies *referring expressions* (*i.e. mentions*) to entities in the text. *Voz* traverses the syntactic parse trees looking for "noun phrase" (NP) nodes. This process yields a set of mentions $E = \{e_1, ..., e_n\}$. After that, an additional *coreference resolution* process is run in order to improve the output from the Stanford Coreference Resolution system (Lee et al. 2013). Besides the pronominal coreference resolution information, our process uses semantic and contextual information to further group mentions in E into *coreference groups* (Lee et al. 2013). The output of this process is a coreference graph $G = \langle E, L \rangle$ where E is the set of mentions, and, $L \in E \times E$ is the set of edges between each pair of mentions which are believed to refer to the same entity. In the *verb extraction* process, *Voz* identifies actions linking the extracted mentions using the typed dependencies from the Stanford CoreNLP. Currently, we only consider actions represented by verbs. The output is a set of triplets: $V = \{v_1, ..., v_w\}$, where each triplet v_i is of the form $\langle$verb, subject, object$\rangle$ and subject and/or object may be empty.

Enrichment: For each extracted mention *Voz* computes a feature-vector that encodes linguistic features related to the extracted verbs and mentions combined with external common sense and domain knowledge (Valls-Vargas, Zhu, and Ontanon 2016). The features are computed from the parse tree of the sentence where the mention is found, the subtree representing the mention, the leaves of the subtree (e.g., word-level tokens with POS tags) and the dependency lists that contain a reference to any node in the mention's subtree, including verb arguments. We also query knowledge bases such as WordNet, ConceptNet and word lists (*i.e.*, dictionaries or gazetteers). Our features also include features for determining if a mention appears as a subject of a verb, which

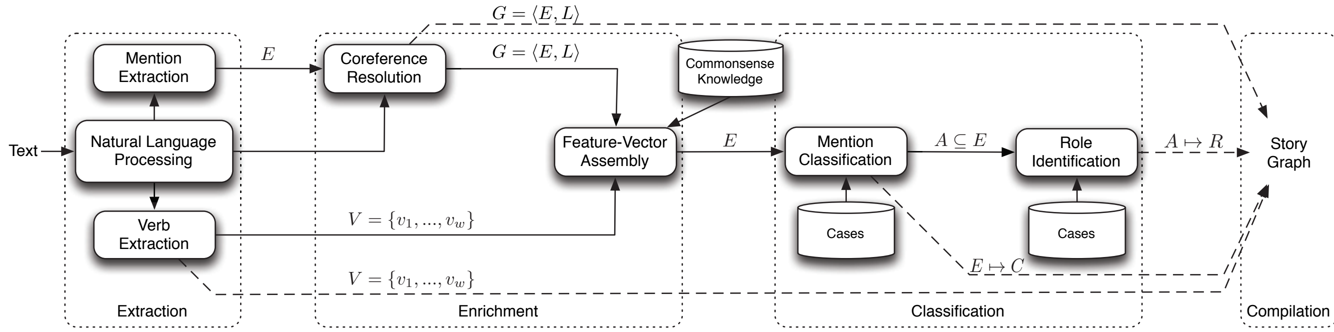

Figure 1: Architecture of the *Voz* system illustrating the processes in our automatic narrative information extraction pipeline. Dotted boxes identify the four steps in our story graph extraction methodology. Solid arrows represent information passed between the different processes. Dashed arrows represent the contribution to the extracted story graph.

argument of a verb a mention appears in, and, when a mention appears as the subject or object, we compute additional features for several individual verbs and conceptual action clusters. The output is a set of mentions $E = \{e_1, ..., e_n\}$ where each mention e is a feature vector.

Classification: In the next step, *Voz* uses a case-based reasoning (CBR) approach to classify each entity e into a set of classes S inspired by Chatman's taxonomy (1990): *happening* (e.g., rain), *male character*, *female character*, *anthropomorphic animal character*, *anthropomorphic object character*, *group or abstract set of characters* (e.g., people, pirates, all the devils), *magical being character* (e.g., Jack Frost, the devil), *part of a character* (e.g., her soul, her fingers), *animal* (non-character), *object* or *prop*, *locations* that the characters visit (e.g., the hill), *scenery* that is mentioned (e.g., the mountains in the distance, the fields surrounding the hill), *temporal references* (e.g., the day after, Winter), *part of a non-character* (e.g., the bed's blankets, the horse's back), and an additional "N/A" class label used mostly for parsing errors. As a CBR system, *Voz* contains a *case-base* $C = \{c_1, ..., c_l\}$, where each *case* $c_i = \langle e_i, s_i \rangle$ is composed of a mention e_i (represented by the feature vector described above) and a class $s_i \in S$. The case base is populated from the training set, described in the experimental evaluation section below. For experimentation purposes, when running the system for one story, only the annotated data for the remaining 20 stories is included in the case base. When classifying a new mention, the most similar instance to e from the case-base is selected, and, the class of e is predicted as that of the retrieved case. To determine the most similar case, *Voz* uses a continuous variant of the Jaccard distance (Valls-Vargas, Ontañón, and Zhu 2014). Once all the mentions have been classified, the output of coreference resolution is used to refine the results. Given a mention $e \in E$, we identify its coreference group $coref(e)$, that is, all the mentions that are linked to e in the coreference graph G. Then, the class assigned to e is replaced by the majority class in the group $coref(e)$.

Finally, the set of characters in the story are passed on to a *role identification* process that classifies each character into a set or roles R derived from the 31 Proppian functions and subfunctions (Propp 1973). The Proppian role labels in R are: *hero*, *false hero*, *sought-for-person*, *villain*, *helper* (includes magical helper since mostly correspond to the same character in our dataset), *other* (includes dispatcher, family members and other minor roles), and an additional "N/A" class label used mostly for misidentified characters. Roles are predicted in a similar way to entity classes (Valls-Vargas, Zhu, and Ontañón 2014).

Story Graph Compilation: The output of the different processes in *Voz* is finally compiled into a *story graph* $S = \langle N, V \rangle$, where N is the set of nodes in the graph, and V the set of edges. Each node n_i is a tuple (g, s, r), where g is a coreferenced entity group, $s \in S$ is the class of the entities in that group (happening, male character, female character, object, etc.), and $r \in R$ is the role of the entities in the group (hero, villain, etc.), which is N/A for those entities not being characters. The edges V correspond exactly to the set of verbs extracted from the story. There is an edge between two nodes $n_1 = (g_1, s_1, r_1), n_2 = (g_2, s_2, r_2) \in N$ if there is a verb $v \in V$ such that g_1 is the *subject* of the verb and g_2 is the *object* of the verb.

Edges, therefore, represent the relation between the entities, and the actions that each entity executes. However, notice that in the current version of *Voz*, no temporal information about the order of these actions is extracted. This will be part of our future work.

Experimental Evaluation

In order to assess the quality of the extracted story graphs, we report an empirical evaluation on a dataset containing 21 Russian stories. In this section, we first describe our dataset, then numerically evaluate the accuracy of the resulting story graphs, and finally illustrate the performance of the system showing some automatically extracted story graphs, and compare them with manually generated ones.

Dataset

Our dataset contains 21 Russian folk stories translated to English. We selected stories studied by Propp, 6 of which were collected by Malec (2010) and 15 by Finlayson (2012). To reduce NLP preprocessing issues at the discourse level, we removed quoted (*i.e.*, dialogue) and some instances of direct

	N/A	AA	AN	AO	FE	GR	HA	MA	MB	OB	PA	PO	SC	SS	ST	Recall	Prec.
N/A	**0**	24	1	7	8	17	30	37	4	166	11	0	9	150	47	0	0
AA	0	**39**	1	2	31	10	1	29	13	22	2	0	0	3	5	0.247	0.151
AN	0	4	**2**	6	0	2	2	8	2	49	0	0	1	3	7	0.023	0.133
AO	0	1	0	**0**	0	22	1	7	2	20	1	0	0	7	0	0	0
FE	0	14	0	0	**510**	3	8	9	0	24	0	0	0	17	4	0.866	0.765
GR	0	10	3	34	62	**56**	9	55	0	120	2	0	0	5	0	0.157	0.308
HA	0	3	2	1	2	2	**4**	7	1	60	4	0	6	21	10	0.033	0.033
MA	0	72	1	1	30	37	17	**799**	17	71	3	0	0	69	11	0.708	0.76
MB	0	34	1	9	1	5	0	57	**52**	58	0	0	21	1	2	0.216	0.433
OB	0	36	3	11	13	13	30	14	26	**375**	48	0	16	119	50	0.497	0.318
PA	0	5	1	8	7	5	5	4	0	56	**33**	0	1	16	0	0.234	0.308
PO	0	0	0	0	0	0	0	0	1	2	1	**0**	1	1	0	0	0
SC	0	8	0	0	0	2	2	1	1	19	0	0	**2**	21	6	0.032	0.032
SS	0	4	0	4	1	6	9	13	1	94	1	0	4	**283**	14	0.652	0.387
ST	0	5	0	0	2	2	2	12	0	42	1	0	2	16	**57**	0.404	0.268

Table 1: Confusion matrix for predictions in the 15 class labels in our classification process with counts for all the 21 stories using the leave-one-story-out protocol. The two letter labels stand for (from top to bottom): "N/A" for parsing errors, AA: anthropomorphic animal character, AN: animal (non-character), AO: anthropomorphic object character, FE: female character, GR: group of characters, HA: happening, MA: male character, MB: magical being character, OB: object or prop, PA: part of characters, PO: part of non-characters, SC: scenery that is mentioned, SS: locations that the characters visit, and ST: temporal references. Bold face indicates correct predictions (diagonal) and color gradient normalized on total count of instances for each class.

speech (4 passages where the narrator addresses the reader directly: "What could she do in this trouble?"). The edited dataset contains 914 sentences. The stories range from 14 to 69 sentences ($\mu = 43.52$ sentences, $\sigma = 14.47$). There is a total of 18,126 tokens (words and punctuation; $\mu = 19.83$ words per sentence, $\sigma = 15.40$)

Although the stories are relatively short, fully understanding them often requires significant inference based on commonsense knowledge and contextual information. For example, in one of the stories, a magical being called Morozko gave a young girl "a warm fur coat and downy quilts." In order to understand Morozko is *helping* her, the context of the forest in the winter is important. Furthermore, some actions need to be inferred. In the same story, the text only describes how the step-sister of the hero answered Morozko's question rudely. In the next scene, her mother "saw the body of her daughter, frozen by an angry Morozko," leaving out Morozko's direct actions to inference. Coreference in these stories can be difficult, even for the human readers at times. It is very common that a character's referring expression changes from "daughter" to "sister" or "girl" throughout the story. In one of the stories there are two young female characters. Besides the obvious pronominal coreference problems that may arise, they are both referred as "daughter" and "maiden" in different parts of the story.

To quantify the accuracy of the extracted story graphs and performance of *Voz*, we annotated different aspects of the story graph as the ground truth. First, we automatically identified noun phrases (NP) representing referring expressions. There are 4,791 annotated NP using the 15 class labels described in the previous section (including 511 parsing errors that are falsely reported as NP). There are 2,781 NP that represent characters (persons, anthropomorphic animals and other magical beings) that are further annotated with the 7 role labels also described in the previous section. The annotation were performed by 2 annotators independently and conflicts resolved by consensus. The coreference groups are annotated for characters and groups of characters (*e.g.*, "they") but are not included for the rest of the mentions (*i.e.*, we did not annotate coreference for props or locations). Finally, we manually annotated all the verb triplets present in the stories including the triplets where the subject and/or object may be empty. We used Finlayson's initial annotations to derive these and completed the annotations for the stories collected by Malec.

Story Graph Extraction Evaluation

In this section we provide a break down of the performance of different features of the automatically extracted story graphs averaged across all 21 stories in the dataset.

Mention Extraction: *Voz* identifies a total of 4,791 individual mentions, 4,280 correspond to noun phrases and 511 of which are not actual referring expressions but parsing errors, mostly adjectival phrases identified as nominal phrases. Our method has a recall of 1.000 (all of the annotated mentions were found) but a precision of 0.893 (due to parsing problems introduced by the Stanford CoreNLP system).

Entity Classification: *Voz* achieves an average precision of 0.567 and recall of 0.507 in the entity classification process. These are micro-averaged results, which is, a weighted average by the number of entities in each of the 15 class labels. The confusion matrix for this classification is shown in Table 1. When considering only whether the entity is correctly classified as a character or non-character, the precision is 0.929 and recall 0.934, which shows that our approach is very good at identifying which entities are characters and

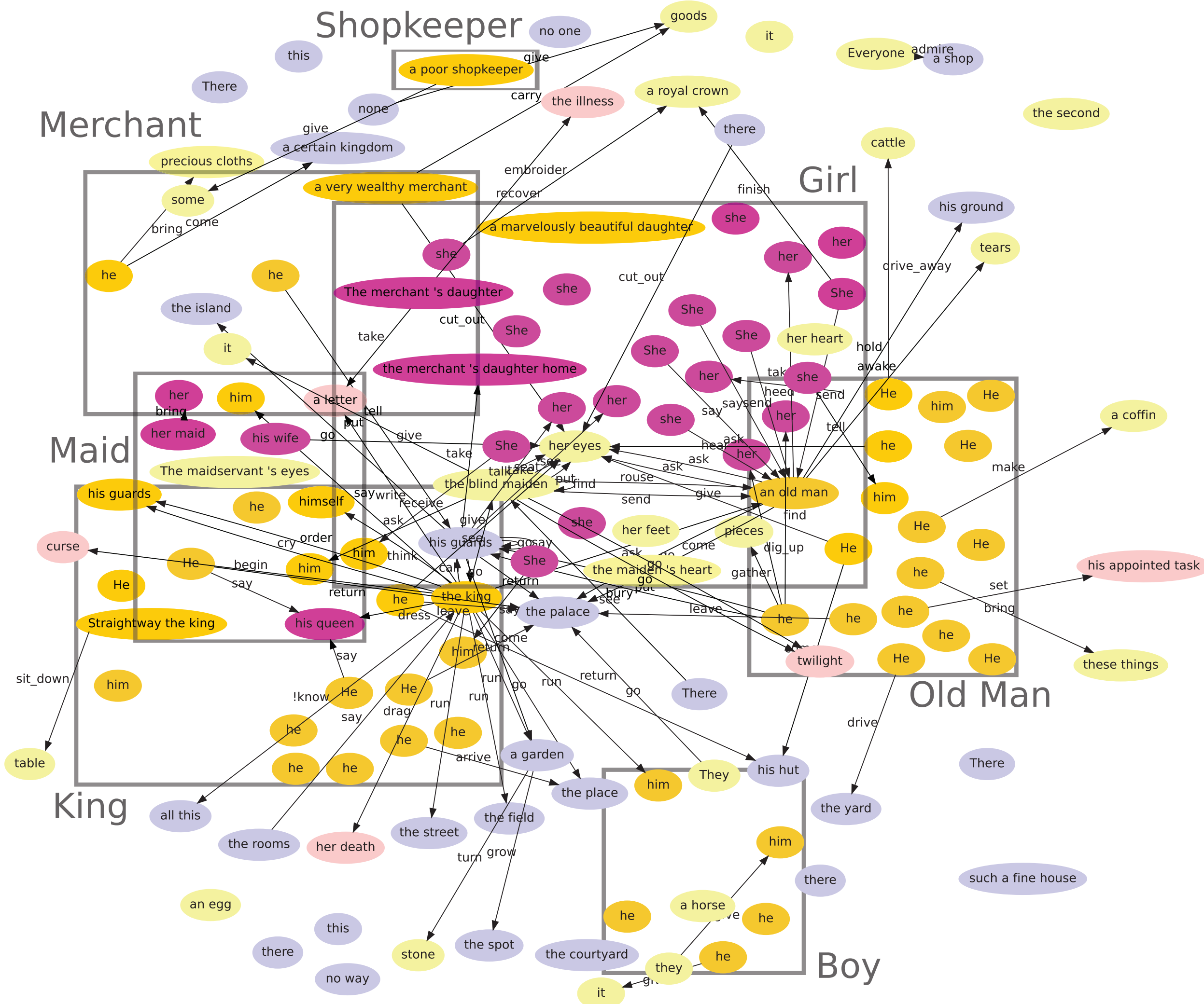

Figure 2: Example story graph extracted automatically from one of the stories in our dataset. This graph encodes a story with its entities (nodes) and interactions (edges). Orange nodes are male characters, magenta are female characters, yellow are objects, props and characters without a role, red are happenings and blue are locations visited by the characters. The arrows represent verb triplets and the boxes represent nodes that should have been merged into a single node by coreference resolution.

which are not. Concerning role classification, *Voz* achieves a precision of 0.425 and a recall of 0.661. With an f-measure of 0.517, the performance is substantially higher than a random baseline (0.143) or an informed baseline that always predicts a hero (0.349), which is the most common role.

Verb Extraction: *Voz* extracts 1,335 verbs out of the 1,586 annotated in the ground truth across the 21 stories. The verb extraction process then expands the extracted verbs into verb triplets (edges in the story graph) with an average precision of 0.260 and recall of 0.204. As we will show later, this is one of the major bottlenecks in our system. *Voz* fails to identify the subject and object of many verbs, which results in many missing links in the resulting story graphs.

Coreference: Coreference is responsible for identifying which mentions refer to the same characters of objects. This is important, since, without it, each individual mention to a character would be considered a separate character in the graph. To evaluate the performance of the coreference resolution process, we compute the average number of different characters found in each coreference group (C/Gr), and, the average number of different groups a single character is spread across (Gr/C). Perfect coreference would score C/Gr = 1.00, and, Gr/C = 1.00 meaning that each group only contains mentions to one character and a character is mentioned in only one group respectively. Our process groups the 2781 mentions into 1,359 coreference groups and yields C/Gr = 1.07 and Gr/C = 6.00. This means that while our process is relatively good at separating mentions from different characters, it performs poorly at merging different mentions of the same character. The implication is that the resulting story graphs have in average 6 times more nodes representing characters than they should have. This observation is aligned

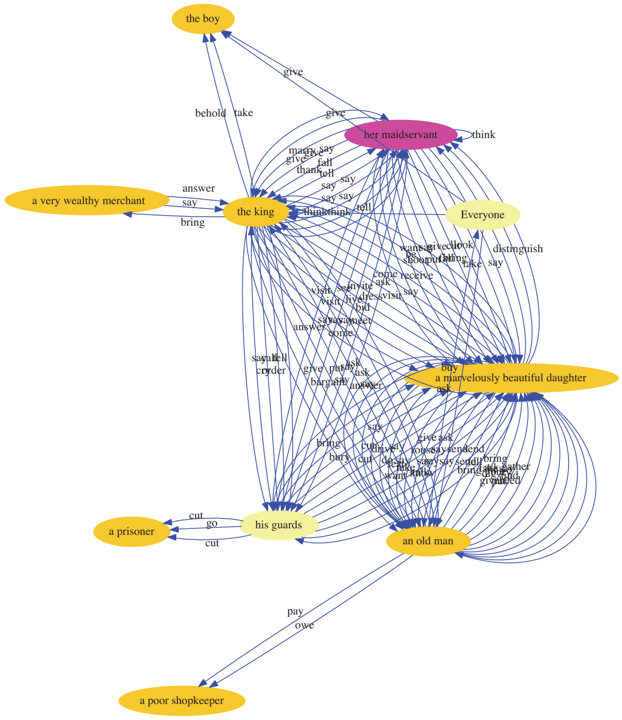

Figure 3: Subgraph from the same story represented in Figure 2 with the character interactions extracted from the ground truth story graph. Orange nodes are male characters, magenta are female characters, yellow are groups of characters. The arrows represent annotated verb triplets, one for each interaction. This graph elicits the coreference resolution problems in our current automatic approach.

with results in the NLP community where coreference tends to be conservative in terms of grouping coreference groups as it prefers precision over recall. Our method improves the groups of the Stanford Coreference Resolution system in this domain but is still missing many groupings and several pronouns are still unresolved.

Overall, *Voz* performs well at extracting and classifying entities but there is significant room for improvement in the verb argument extraction and coreference resolution tasks.

Visualizing Story Graphs

This section presents some visual examples of story graphs extracted by *Voz* in order to provide an illustration of the numerical performance reported in the previous section. Figure 2 shows an example of the story graph extracted for one of the stories while Figure 3 shows the ground truth. Recall that the ground truth does not contain any props or locations, so it has many less nodes than the automatically extracted node, which includes all types of mentions.

When comparing the automatically extracted story graph against the story graph from the ground truth we can see that the automatic coreference in the automatically extracted story graph is not grouping several mentions representing a single character. For example, while all mentions to the shopkeeper were properly merged into a single node, mentions to the King, or the maid were not properly coreferenced, resulting in a collection of separate nodes. Moreover, even if coreference resolution makes the resulting graph have more nodes than necessary, the performance of the entity classification yields nodes with adequate labels.

Temporal Information Although *Voz* does not currently extract temporal information, many story generation systems require this information. For example, *MEXICA* (Pérez and Sharples 2001) uses sequences of actions in its story representation, and *Riu* (Ontañón and Zhu 2010) uses sequences of *scenes*. In order to test the feasibility of employing our story graph extraction approach once temporal information is extracted by *Voz* we split the text of a story in three segments (the beginning of the story, the middle, and the end), and generated story graphs containing the entities of all the story, but highlighting the entities mentioned in the segment and drawing only the verbs of the corresponding segment. The resulting story graphs adequately capture the events in each of the three parts of the story, as shown in Figure 4, indicating that *Voz* could be used for extracting information for different time frames. For this visualization, we split the story in three parts manually, but part of our future work will consist on identifying the different parts of a story automatically by analyzing references to locations or temporal anchors in the text. As part of our future work, we would like to experiment with providing the output of *Voz* directly to a story generation system, and evaluate the quality of the resulting stories as compared with those generated when the system is given manually authored story graphs.

Spatial Information Finally, we looked at the location and spatial information we can extract from our graphs, and how this spatial information can be used for generating content for games. We filtered the graph by selecting only locations and character nodes and edges labeled with verbs related to movement (e.g., go, come) and copular verbs (used to link adjectives and nouns). Then we select the biggest connected subgraph, such as the one shown in Figure 5 extracted from Figure 2. If we consider that locations that the same character is related to must be connected in order to allow the character to travel between them, we can generate a graph representing the spatial relationships between locations in the story. To show the usefulness of this graph, we provided this graph as input to the graph embedder and realizer by Valls-Vargas et al. (Valls-Vargas, Ontañón, and Zhu 2013) to generate a two-dimensional map that may be suitable for this particular story to happen (shown in Figure 6). This shows a first step toward automatically generating spaces from stories written in natural language. These spaces could be further populated by the characters and objects in the first location at which they are mentioned in order to obtain a complete spatial representation of the story world.

Conclusions and Future Work

This paper presented an approach to automatically extract story graphs from unannotated natural language text. The

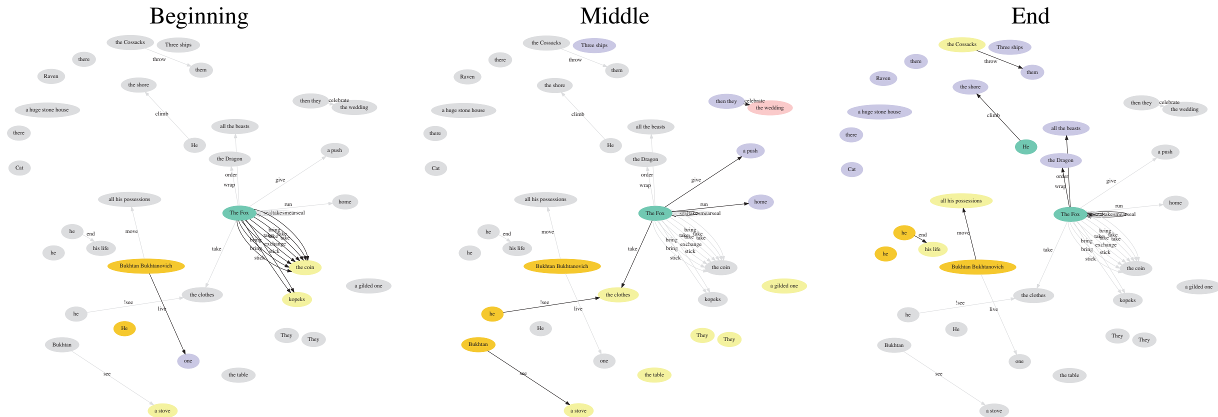

Figure 4: Story graph generated from three segments of a story. In the beginning of the story there is an anthropomorphic animal (the fox) that interacts with some coins (kopeks). In the middle of the story, the fox and the main character visit some locations (home), some props are mentioned (clothes) and an event happens (the wedding). At the end of the story the fox interacts with several beasts (raven, dragon).

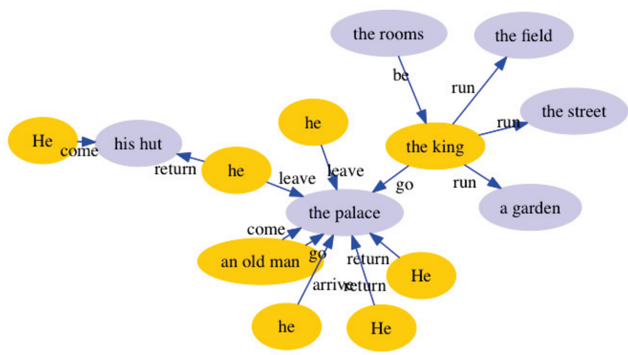

Figure 5: Subgraph with character and location information filtered from Figure 2.

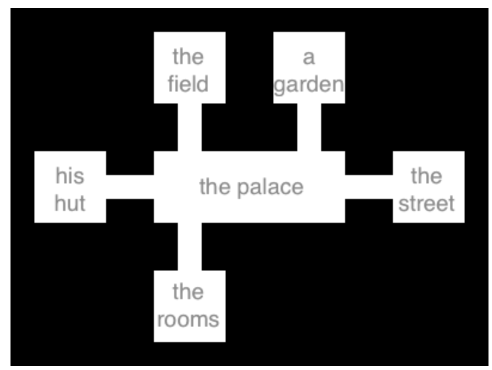

Figure 6: Two-dimensional embedding and realization of the locations in Figure 5.

long term goal of this line of research is to allow computational narrative systems, and in particular story generation systems, to automatically exploit stories represented in natural language, thus alleviating the "authorial bottleneck" problem. We presented *Voz*, our automated narrative information extraction system, and evaluated the accuracy of the extracted story graphs using an annotated ground truth on our corpus of 21 Russian stories. Finally we provided examples of the story graphs our system is capable of automatically extracting and briefly discussed how this could be used for both feeding story generation as well as map and game world generation systems.

As part of our current work, we are working on improving the quality of the generated story graphs by improving the verb extraction process (our current focus is on automatically parsing dialog to capture additional interactions), coreference resolution (by feeding back information from later stages of *Voz*'s pipeline back to coreference resolution (Valls-Vargas, Zhu, and Ontañón 2015)), and extracting temporal information from text. In our future work, we would like to experiment with of feeding the story graphs extracted by *Voz* to actual map and story generation systems.

References

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the 2008 Anniversary Meeting of the Association for Computational Linguistics (ACL)*, 789–797.

Chang, A. X.; Savva, M.; and Manning, C. D. 2014. Learning spatial knowledge for text to 3d scene generation. In *EMNLP*, 2028–2038.

Chatman, S. 1990. What can we learn from contextualist narratology? *Poetics Today* 309–328.

Coyne, B., and Sproat, R. 2001. Wordseye: an automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 487–496. ACM.

Elson, D. K.; Dames, N.; and McKeown, K. R. 2010. Extracting social networks from literary fiction. In *Proceedings of the Fourty-Eighth Annual Meeting of the Association for*

Computational Linguistics, 138–147. Association for Computational Linguistics.

Elson, D. K. 2012a. Detecting story analogies from annotations of time, action and agency. In *Proceedings of the 2012 Workshop on Computational Models of Narrative*.

Elson, D. K. 2012b. *Modeling Narrative Discourse*. Ph.D. Dissertation, Columbia University.

Finlayson, M. A. 2008. Collecting semantics in the wild: The story workbench. In *Naturally Inspired Artificial Intelligence, Technical Report FS-08-06, Papers from the 2008 AAAI Fall Symposium*, 46–53.

Finlayson, M. A. 2012. *Learning narrative structure from annotated folktales*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Gervás, P.; Díaz-Agudo, B.; Peinado, F.; and Hervás, R. 2005. Story Plot Generation based on CBR. *Knowledge-Based Systems*.

Goyal, A.; Riloff, E.; and Daumé, III, H. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 77–86.

Harmon, S., and Jhala, A. 2015. Imaginative recall with story intention graphs. In *Proceedings of the 2015 Workshop on Computational Models of Narrative*.

Hartsook, K.; Zook, A.; Das, S.; and Riedl, M. O. 2011. Toward supporting stories with procedurally generated game worlds. *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)* 297–304.

Lee, H.; Chang, A.; Peirsman, Y.; Chambers, N.; Surdeanu, M.; and Jurafsky, D. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules.

Lehnert, W. G. 1981. Plot units and narrative summarization. *Cognitive Science* 5(4):293–331.

Li, B.; Lee-Urban, S.; Johnston, G.; and Riedl, M. 2013. Story generation with crowdsourced plot graphs. In *AAAI*.

Malec, S. A. 2001. Proppian structural analysis and xml modeling. *Proceedings of Computers, Literature and Philology (CLiP 2001)*.

Malec, S. 2010. Autopropp: Toward the automatic markup, classification, and annotation of russian magic tales. In *Proceedings of the First International AMICUS Workshop on Automated Motif Discovery in Cultural Heritage and Scientific Communication Texts*, 112–115.

Mani, I. 2012. *Computational Modeling of Narrative*, volume 5 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers.

Meehan, J. 1976. *The Metanovel: Writing Stories by Computer*. Ph.d., Yale University.

Ontañón, S., and Zhu, J. 2010. Story and text generation through computational analogy in the riu system. In *Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Pérez, R. P. Ý., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

Propp, V. 1973. *Morphology of the Folktale*. University of Texas Press.

Rishes, E.; Lukin, S. M.; Elson, D. K.; and Walker, M. A. 2013. Generating different story tellings from semantic representations of narrative. In *Interactive Storytelling*. Springer. 192–204.

Turner, S. R. 1993. Minstrel: a computer model of creativity and storytelling.

Valls-Vargas, J.; Ontañón, S.; and Zhu, J. 2013. Towards story-based content generation: From plot-points to maps. In *Proceedings of the 2013 IEEE Conference on Computational Intelligence in Games (CIG)*, 1–8. IEEE.

Valls-Vargas, J.; Ontañón, S.; and Zhu, J. 2014. Toward automatic character identification in unannotated narrative text. In *Proceedings of the Seventh Workshop in Intelligent Narrative Technologies (INT 7)*.

Valls-Vargas, J.; Zhu, J.; and Ontañón, S. 2014. Toward automatic role identification in unannotated folk tales. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Valls-Vargas, J.; Zhu, J.; and Ontañón, S. 2015. Narrative hermeneutic circle: Improving character role identification from natural language text via feedback loops. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2517–2523.

Valls-Vargas, J.; Zhu, J.; and Ontanon, S. 2016. Error analysis in an automated narrative information extraction pipeline. *IEEE Transactions on Computational Intelligence and AI in Games* PP.