Deep Dungeons and Dragons: Learning Character-Action Interactions from Role-Playing Game Transcripts

Annie Louis
University of Edinburgh
10 Crichton Street
Edinburgh EH8 9AB
alouis@inf.ed.ac.uk

Charles Sutton
University of Edinburgh
10 Crichton Street
Edinburgh EH8 9AB
sutton@inf.ed.ac.uk

Abstract

An essential aspect to understanding narratives is to grasp the interaction between characters in a story and the actions they take. We examine whether computational models can capture this interaction, when both character attributes and actions are expressed as complex natural language descriptions. We propose role-playing games as a testbed for this problem, and introduce a large corpus[1] of game transcripts collected from online discussion forums. Using neural language models which combine character and action descriptions from these stories, we show that we can learn the latent ties. Action sequences are better predicted when the character performing the action is also taken into account, and vice versa for character attributes.

1 Introduction

Imagine a giant, a dwarf, and a fairy in a combat situation. We would expect them to act differently, and conversely, if we are told of even a few actions taken by a character in a story, we naturally start to draw inferences about that character's personality. Communicating narrative is a fundamental task of natural language, and understanding narrative requires modelling the interaction between events and characters.[2]

In this paper, we propose that collaboratively-told stories that arise in certain types of games provide a natural test bed for the problem of inferring interactions between characters and actions in narratives. We present a corpus of role-playing game (RPG) transcripts where characters and action sequences are described with complex natural language texts. Table 1 shows an example character description, and an action text for the same character. This example shows how the ties between characters and their actions are subtly present in the text descriptions, and learning the latent ties between them is a difficult task. Based on our corpus, and using neural language models, this work demonstrates an initial success on this problem.

Character description
Name: Ana Blackclaw; Age: 27; Gender: Female Appearance: Standing at a mighty 6'5, she is a giant among her fellow humans. Her face is light, though paler than the average man or woman's, and is marked by scars. ... Her body is muscular, as it would have to be to carry both her armor and the hammer. Her light grey eyes nearly always keep a bored expression. Her canines seem a tad larger than the normal person's. Preferred Weapon: Hammer. Preferred Armor: Heavy. Gift: Binoculars. Darksign: No.
Action description
She stopped dead in her tracks as the hissing began. A grumble escaped her as it did so, and she looked over to make sure the other woman was doing fine. Seeing that all was not entirely well, she allowed herself to slide down, her hand gripping the slope side once more to slow herself. Once that was accomplished, she reached out and grabbed the back of the girl's neck, pulling her back to steady herself. The giant remained silent as she did so, and then glanced over to the nearby skeletons. They would be upon them soon. Her grip tightened on the hammer as she glanced from side to side. It would not be a fun fight.

Table 1: Example descriptions from our RPG corpus

The ability to understand and generate narratives is a useful skill for natural language systems, for example, to plan a coherent answer to a question, or to generate a summary of a document. Prior work on narrative processing has focused on inducing disjoint sets of character and event types (as topic models), capturing the relationship between characters in the same story, or extracting character-action pairs as low level noun-verb tuples. However, these models do not aim to match or infer characters and actions from each other.

We make two contributions towards closing this gap. We introduce a corpus of thousands of RPG

[1] http://groups.inf.ed.ac.uk/cup/ddd/

[2] In this paper, the word *character* will always be used in the sense of "character in a story" rather than the sense of "character in a token".

transcripts and demonstrate predictive cues between characters and actions by building neural language models with facility for adding side information. We show that a language model over action text obtains lower perplexity when we also make available a representation of the character who produced each token. Likewise, a language model for character descriptions benefits from information about the actions the character made. Our findings open up new possibilities for making sophisticated inference over narrative texts.

2 Related work

In work on narratives, both characters and actions have received significant attention, albeit separately. There is work on inducing *types* of characters (Bamman et al., 2013, 2014) or relationships between characters (Chang et al., 2009; Elson et al., 2010; Chaturvedi et al., 2016; Iyyer et al., 2016). Often these approaches are based on probabilistic topic models or more recently distributed word representations computed by neural networks. Others focus on learning regular and repetitive event sequences in stories (Chambers and Jurafsky, 2009; McIntyre and Lapata, 2009), together with some information about the agent of the actions. These extractions are fairly low-level, in the form of noun-verb pairs. There are also models for clustering stories either based on their characters (Frermann and Szarvas, 2017), or sentiment and topic (Elsner, 2012, 2015).

The above approaches mine *types* of actions or characters. This work focuses on infering the latent ties between actions and characters, and whether one aspect can help predict the other. Flekova and Gurevych (2015) present recent work related to this latter idea. They classify characters based on their speech and actions into an introvert or extrovert class. In contrast, we focus on attributes of characters and actions beyond such coarse traits, and when these attributes are expressed as complex descriptions.

3 A corpus of RPG transcripts

Traditionally, RPGs are played orally with players seated around a table. But there are also online forums where users play RPGs by posting text descriptions instead.

We collected a corpus of RPG threads from one such website `roleplayerguild.com`. Here each game play is recorded in two threads. In one of these, each player posts a detailed text description of the role (character) she is going to play in the game, which we call a *character description*. This description includes the character's physical appearance, personality, family background, as well as special and supernatural powers, and possessions. A second thread consists of the actual game play where each player contributes a post when his turn comes. Each post describes how the character that is assumed by that specific player responds to the game situation. Thus the story develops collaboratively. We call each post in the story thread an *action description*. An example from our corpus of a character description and an action description is shown in Table 1.

A noteworthy aspect of these RPGs is that character attributes are determined by writing the descriptions before the game starts. The story thread itself then focuses predominantly on the actions and does not reiterate character attributes. Moreover, we know unambiguously which character is associated with each action post. Such mapped pairs of clean character descriptions and associated actions would be difficult to obtain from novels or other stories without sophisticated analysis.

Our corpus contains 1,544 RPGs spanning a variety of themes—fantasy, apocalyptic, romance, anime, military, horror, and adventure. There are a total of 56,576 posts, comprising of 25.3M tokens. The maximum number of posts in a story is 753, minimum 2, and the average is 26. Note that many stories are in progress and some are long running. There are 9,771 unique characters in the corpus, and their descriptions amount to 8.5M tokens. There is a minimum of 1, average 6, and maximum 24 characters in a single story.

Even though each character or action description focuses on a single character, it nevertheless contains descriptions of background settings of the scene, and interactions of other characters (eg. descriptions of the parents of a character). Hence we preprocess the texts to only retain parts most related to the *character in focus*. To this end, in character descriptions, we only keep those sentences which mention the character's name or the personal pronouns 'he' or 'she'. The use of pronouns reflects an intuition that since the description is of one key character, the pronoun is most likely to refer to this salient entity. We also take sentences which mention personality describing words such as 'personality', 'skill', 'specialize', 'ability', 'profile', 'talent', etc., even when they do not contain the name. For action descriptions, we only keep sentences which start with the character's name. We do not use pronouns since action text may refer to other salient characters as well.

Finally, we replace the main character (contributor) of a post with an "ENT" (for entity) token. Other proper names in a post are replaced with a "NAME" token and numbers with a "NUM" token. We drop all punctuation and any text with less than 5 tokens. After these preprocessing steps, we have 1,439 stories containing 1.48M tokens for action descriptions and 2.95M for characters.

4 Learning character-action interactions

We examine the feasibility of inferring character-action interactions from text using neural language models (LM) with side information.

4.1 ACTION and CHAR language models

The *story line*, that is, the full text of a story is the token sequence $X = x_1...x_K$ created by concatenating the tokens across all the action descriptions of the story. The posts are taken in time order without any mark for post boundaries. Let C be the set of all characters in a story. For each character j, we denote the character description as the token sequence $C_j = c_{j1} \dots c_{jm}$.

We build separate language models for action sequences and for character descriptions. The action sequence model is over the story lines (the sequence of all action descriptions in a story), i.e. X as defined above. The character description model is over individual character descriptions i.e., C_j.

First we describe the language model $P(X)$ for the story line. We hypothesize in this work that a better model of X can be built by taking into account the character in focus for each individual action description. First, a baseline recurrent neural network (RNN) language model, which we denote ACTION-LM, would be

$$\begin{aligned} \mathbf{h}_i &= LSTM(\mathbf{h}_{i-1}, \mathbf{x}_{i-1}) \\ P(x_i|x_1 \dots x_{i-1}) &= softmax(W_{hv}\mathbf{h}_i + b_v) \end{aligned}$$

Here $\mathbf{x}_{i-1}$ is the embedding of the input token x_{i-1}, and $\mathbf{h}_{i-1}$ is the hidden state which summarizes the token sequence $x_1 \dots x_{i-2}$. $LSTM$ computes the next hidden state using an LSTM cell (Hochreiter and Schmidhuber, 1997). The output layer produces a probability distribution over the LM vocabulary using weight matrix $W_{hv} \in \mathbb{R}^{|V|*|h|}$ where $|h|$ is the hidden size and $|V|$ is the vocabulary size; b_v is the bias vector.

To take the character descriptions into account when generating actions, we define a second model ACTION-LMS which estimates

$$P(X|C) = \prod_{i=1}^{K} p(x_i|z_i, x_1...x_{i-1}, z_1...z_{i-1}),$$

where z_l is a variable indicating which character produced the token x_l. For this model, we essentially augment the RNNs with the character descriptions as side information. For each token x_l, the side information is the character description indicated by z_l, i.e, C_{z_l}. We follow the approach by Mikolov and Zweig (2012), and Hoang et al. (2016), where a feature embedding vector $\mathbf{e}$ representing side information is input to both the RNN's hidden and output layers, or to one of them. During development, we found that concatenating the feature embedding with the token embedding at the input layer, and with the hidden state at output layer gave the best performance. More formally, ACTION-LMS computes:

$$\begin{aligned} \mathbf{h}_i &= LSTM\left(\mathbf{h}_{i-1}, \begin{bmatrix}\mathbf{x}_{i-1}\\ \mathbf{e}_i\end{bmatrix}\right) \\ P(\mathbf{x}_i|\mathbf{x}_1 \dots \mathbf{x}_{i-1}) &= softmax\left(W_{rv}\begin{bmatrix}\mathbf{h}_i\\ \mathbf{e}_i\end{bmatrix} + b_v\right) \end{aligned}$$

where $\mathbf{e}_i$ is a representation of the character which produced the token x_i. The hidden state $\mathbf{h}_{i-1}$ now summarizes both the action tokens up to $i-2$ and the character information up to $i-1$. The output layer weight matrix is $W_{rv} \in \mathbb{R}^{|V|*(|h|+|e|)}$ where $|h|$ is the size of the RNN hidden unit, and $|e|$ the feature embedding size.

In our work, the feature embedding itself comes from a feedforward neural network trained jointly within the LM. This feature network takes as input the average value of pretrained embeddings[3] for the tokens in the character description (we remove stopwords[4]). This initial vector is passed through hidden layers to yield the feature embedding $\mathbf{e}$ (reminiscent of deep averaging networks by Iyyer et al. (2015)).

The language models for character descriptions are similar in structure. First, we call the unconditioned model $P(C_i)$ for a character description C_i as CHAR-LM; this is again an LSTM language model. Second, we implement CHAR-LMS

[3]300 dimension *word2vec* (Mikolov et al., 2013) embeddings trained on the 1 billion word Google News Corpus.

[4]We remove stopwords for side information only

which estimates $P(C_i|X_{C_i})$, where X_{C_i} is the subsequence of X only containing the tokens produced by C_i. We obtain this conditional probability based on the same architecture as ACTION-LMS. Here the input to the feature neural network is the average pretrained embeddings of the tokens (without stopwords) in X_{C_i}.

4.2 Experiments

We randomly divide our corpus into 100 stories for testing, 20 for development, and the rest, 1319 for training. We compare the two ACTION language models, based on a vocabulary size of 20,000, and the CHAR models have a vocabulary of 10,000.

Some posts are long even after our filtering steps, and create a winding story line when concatenated. So we also explore whether limits on description lengths is useful. In ACTION models, a limit of g means that only the first g words of each post are concatenated to form X. For CHAR models, only the first g words of the description C_i is used as the sequence for the LM. The same limit g is given to both the models with and without side information. When using side information, we can restrict the conditioning text as well, to a maximum of h words. We tune these limit parameters, as well as the number of hidden layers, hidden unit sizes and dropout probability on a development set.

For the ACTION models, we set g to 100 words. ACTION-LM uses 2 layers with 256 hidden units each. ACTION-LMS has 1 layer with 256 hidden units for the feature network with h set to 25 words, and 1 layer with 50 units for the RNN part. For the CHAR models, $g = 200$ words. CHAR-LM has one hidden layer with 100 units. For CHAR-LMS, the best network was the same as ACTION-LMS but with $h = 100$ (the first 100 words of all the action posts by that character are combined as the side information). We apply a dropout probability of 0.65, clip gradients at 5.0, and use the Adam algorithm (Kingma and Ba, 2015) for optimization. All our models can trained in an hour, ACTION-LM with 14 epochs, CHAR-LM 62, ACTION-LMS 60 and CHAR-LMS 91 epochs. We implemented the models in TensorFlow[5].

4.3 Results

First, we provide examples of the patterns captured by ACTION-LMS and CHAR-LMS by sampling from the models (Table 2). For side information, we use simple words (taken from the descriptions in our test corpus) for closer examination.

Action-LMS Model Prime text: ⟨bos⟩ **ENT** called . . .	
Char. context	Generated continuation
small girl cheerful	. . . her name ⟨eos⟩
bulky male	. . . out to the group ⟨eos⟩
hunter bow forest	. . . over and walked over to the large king had been making sure
fear afraid	. . . her her brother ⟨eos⟩
angry irritated	. . . back at name with her thick road with disappointment ⟨eos⟩
brutal violent	. . . out of **ENT** hard to help **ENT** help **ENT** help **ENT** help ⟨eos⟩
school student romantic	. . . out in the way of the conversation ⟨eos⟩

Char-LMS Model Prime text: ⟨bos⟩ **ENT** is . . .	
Action context	Generated continuation
appeared disappeared flew	. . . a very young man who has a few scars on his body ⟨eos⟩
walked looked stayed	. . . a very friendly person ⟨eos⟩
waited	. . . a little girl who is a little girl who is a little
pause stare	. . . a very very young woman ⟨eos⟩
strike slap	. . . a bad boy ⟨eos⟩
follow creep	. . . a slim and slim but slim physique ⟨eos⟩

Table 2: Samples from our language models

Model	**Train**	**Dev**	**Test**
ACTION-LM	82.56	106.83	105.06
ACTION-LMS	57.38	94.95	96.91
CHAR-LM	69.45	118.78	106.12
CHAR-LMS	61.84	110.13	100.86

Table 3: Perplexities of our models

For ACTION-LMS, we seed the story line with the priming text "⟨bos⟩ **ENT** called", where **ENT** is the token in our vocabulary referring to the main character of a post. ⟨bos⟩ is a beginning of sentence marker. Different inputs for the conditioning character description are shown under "Char. context". We then sample from the LM following a greedy approach taking the most likely token at each step until either the end token ⟨eos⟩ or a maximum of 12 tokens is reached. The sample is shown under "generated continuation". Similarly, we sample from CHAR-LMS where the sequence is first primed with "**ENT** is a". We find that both models capture interesting ties between character attributes and actions. However, there is much scope for improved models of generation.

In this work, we have focused on the possibility of capturing the interactions. For that, we compare the impact of side information using perplexity on held-out data (Table 3). For both charac-

[5] https://www.tensorflow.org

ter and action LMs, adding side information leads to a significant decrease in perplexity showing that the interdependence between the two aspects can be learned computationally. Again, there is a lot of scope for improving the language models given that the development and test perplexities are much higher than those during training.

5 Conclusions

We have proposed and demonstrated the feasibility of capturing interactions between characters and their actions in stories. While our neural models show that the data can be better modeled by combining both aspects, one might eventually want to infer a missing modality by sampling or generation from the model. We plan to work on these improvements for future work, and also explore evaluation methods which go beyond language model perplexities, and capture model aspects closer to the task and domain.

References

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. pages 352–361.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. pages 370–379.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. pages 602–610.

Jonathan Chang, Jordan Boyd-Graber, and David M. Blei. 2009. Connections between the lines: Augmenting social networks with text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pages 169–178.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daumé III, and Chris Dyer. 2016. Modeling evolving relationships between characters in literary novels. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. pages 2704–2710.

Micha Elsner. 2012. Character-based kernels for novelistic plot structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. pages 634–644.

Micha Elsner. 2015. Abstract representations of plot structure. *Linguistic Issues in Language Technology* 12(5):1–31.

David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. pages 138–147.

Lucie Flekova and Iryna Gurevych. 2015. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pages 1805–1816.

Lea Frermann and György Szarvas. 2017. Inducing semantic micro-clusters from deep multi-view representations of novels. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pages 1873–1883.

Cong Duy Vu Hoang, Trevor Cohn, and Gholamreza Haffari. 2016. Incorporating side information into recurrent neural network language models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1250–1255.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1534–1544.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. pages 1681–1691.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Neil McIntyre and Mirella Lapata. 2009. Learning to tell tales: A data-driven approach to story generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. pages 217–225.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems*.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *Proceedings of IEEE Spoken Language Technology Workshop*. pages 234–239.