Determining Environmental Context from Fictional Narratives

Abstract

Experiencing a narrative can be made more enjoyable and powerful by increasing the level of the immersion in the story. This can be achieved through contextual cues, such as sounds and images. In order to provide these cues automatically, we attempt to detect parts of the environment described in a fictional narrative. We present three machine learning approaches to this problem, compared to a rule-engineered baseline, and evaluate them over a dataset of fictional texts. In addition, we analyse the performance of this system over the output of a text-to-speech system taking the same narratives as input. This audio dataset is made available. The system is able to recognise environmental context and respond appropriately.

1 Introduction

Non-verbal cues are important to improving the experience of a narrative. For example, adding sound changes mood, and is useful for immersion and positive experiences (Madden and Logan, 2009; Huiberts, 2010). An immersive environment is most engaging when participants believe their actions affect the environment; reliable and responsive cues such as sounds are important for this (Bobick et al., 1999). Taking narratives as input, we attempt to determine what the appropriate environmental component of the context is, which can then be used to provide cues (such as ambient sounds).

There is a diverse range of potential environments from which cues can be generated, even in a constrained corpus of narratives. In addition, effects (e.g. sound or audio cues) are required for surprise events described in a story, like an approaching horse or a thunderstorm. This is similar to streamed topic extraction (Allan, 2002; Preotiuc-Pietro et al., 2012). We determine a corpus, frame the task by giving a specific set of environments for which cues are available, and then evaluate system performance over this dataset. Finally, we create a new language resource of audio recordings and automatic transcriptions, used to evaluate the system on input closer to that it might receive when operating in real-time.

2 Method

2.1 Dataset

For this dataset, we used paragraphs from a set of fantasy role-playing books (Sharp et al., 1989; Livingstone, 1987; Darvill-Evans et al., 1989; Jackson, 1984). Each paragraph is set in a distinct location, making manual environment annotation simple and distinct. The set of potential environmental items in this genre is:

> Mountain; Hill; Forest; Swamp; Windy; Blizzard; Rain; Lightning; Stream; River; Campfire; Night; Meadow; Road; Town; Crowd; Tavern; Underground; Trotting horse; Galloping horse

Paragraphs were labelled with one or more of these labels by a human annotator. In total, X paragraphs were labelled.

2.2 Spoken dataset

An eventual use of this system is to provide automatic sound effects in real-time for a story that is read out loud by a human narrator. As a result, it should operate well on the output of a speech recognition system. Paragraphs were read by an English native speaker, and recorded. A speech recognition system (Lamere et al., 2003) then interpreted these readings and generated a textual representation for each paragraph. The labels used in the text input corpus were then associated with these outputs. This constitutes the spoken dataset.

2.3 Baseline

Trigger words of the name of the environment (swamp, rain, etc)

2.4 Features and Classifier

method: bag of n-grams + multiple nbayes.

LD feature extraction (Lui and Baldwin, 2011) + nbayes; LD motivated by diversity in subsets of environments encountered per story (e.g. some stories set mostly in woodland, others in a town)

bag of word reprs (w2v) (Mikolov et al., 2013) + multi-svm.

3 Results

4 Related Work

ML doc classification (Sebastiani, 2002).

NN good at binary doc classification (Derczynski, 2006).

SVM doc classification (Isa et al., 2008).

5 Conclusion

References

James Allan. 2002. Introduction to topic detection and tracking. In *Topic detection and tracking*, pages 1–16. Springer.

Aaron F Bobick, Stephen S Intille, James W Davis, Freedom Baird, Claudio S Pinhanez, Lee W Campbell, Yuri A Ivanov, Arjan Schütte, and Andrew Wilson. 1999. The kidsroom: A perceptually-based interactive and immersive story environment. *Presence: Teleoperators and Virtual Environments*, 8(4):369–393.

Peter Darvill-Evans, Steve Jackson, and Ian Livingstone. 1989. *Portal of Evil*. Puffin.

Leon Derczynski. 2006. Machine learning techniques for document selection. Master's thesis, University of Sheffield.

Sander Huiberts. 2010. *Captivating sound the role of audio for immersion in computer games*. Ph.D. thesis, University of Portsmouth.

Dino Isa, Lam Hong Lee, V Kallimani, and Rajprasad Rajkumar. 2008. Text document preprocessing with the Bayes formula for classification using the support vector machine. *Knowledge and Data Engineering, IEEE Transactions on*, 20(9):1264–1272.

Steve Jackson. 1984. *Scorpion Swamp*. Puffin.

Paul Lamere, Philip Kwok, William Walker, Evandro B Gouvêa, Rita Singh, Bhiksha Raj, and Peter Wolf. 2003. Design of the CMU Sphinx-4 decoder. In *Proc. INTERSPEECH*.

Ian Livingstone. 1987. *Crypt of the Sorcerer*. Puffin.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In *Proc. IJCNLP*. ACL.

Neil Madden and Brian Logan. 2009. Collaborative narrative generation in persistent virtual environments. In *Proc. AAAI*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Daniel Preotiuc-Pietro, Sina Samangooei, Trevor Cohn, Nicholas Gibbins, and Mahesan Niranjan. 2012. Trendminer: An architecture for real time analysis of social media text. In *Proceedings of the workshop on real-time analysis and mining of social streams*.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Luke Sharp, Ian Livingstone, and Steve Jackson. 1989. *Fangs of Fury*. Puffin.