Humor as Circuits in Semantic Networks

Igor Labutov
Cornell University
iil4@cornell.edu

Hod Lipson
Cornell University
hod.lipson@cornell.edu

Abstract

This work presents a first step to a general implementation of the Semantic-Script Theory of Humor (SSTH). Of the scarce amount of research in computational humor, no research had focused on humor generation beyond simple puns and punning riddles. We propose an algorithm for mining simple humorous scripts from a semantic network (ConceptNet) by specifically searching for dual scripts that jointly maximize overlap and incongruity metrics in line with Raskin's Semantic-Script Theory of Humor. Initial results show that a more relaxed constraint of this form is capable of generating humor of deeper semantic content than wordplay riddles. We evaluate the said metrics through a user-assessed quality of the generated two-liners.

1 Introduction

While of significant interest in linguistics and philosophy, humor had received less attention in the computational domain. And of that work, most recent is predominately focused on humor recognition. See (Ritchie, 2001) for a good review. In this paper we focus on the problem of humor generation. While humor/sarcasm recognition merits direct application to the areas such as information retrieval (Friedland and Allan, 2008), sentiment classification (Mihalcea and Strapparava, 2006), and human-computer interaction (Nijholt et al., 2003), the application of humor generation is not any less significant. First, a good generative model of humor has the potential to outperform current discriminative models for humor recognition. Thus, ability to generate humor will potentially lead to better humor detection. Second, a computational model that conforms to the verbal theory of humor is an accessible avenue for verifying the psycholinguistic theory. In this paper we take the Semantic Script Theory of Humor (SSTH) (Attardo and Raskin, 1991) - a widely accepted theory of verbal humor and build a generative model that conforms to it.

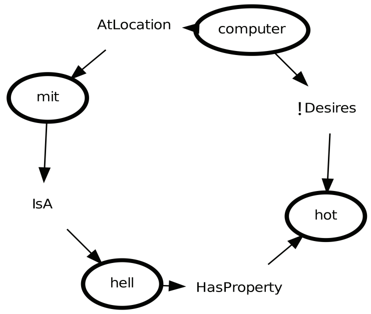

Figure 1: Semantic circuit

Much of the existing work in humor generation had focused on puns and punning riddles - humor that is centered around wordplay. And while more recent of such implementations (Hempelmann et al., 2006) take a knowledge-based approach that is rooted in the linguistic theory (SSTH), the constraint, nevertheless, significantly limits the potential of SSTH. To our knowledge, our work is the first attempt to instantiate the theory at the fundamental level, without imposing constraints on phonological similarity, or a restricted set of domain oppositions.

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 150–155,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

1.1 Semantic Script Theory of Humor

The Semantic Script Theory of Humor (SSTH) provides machinery to formalize the structure of most types of verbal humor (Ruch et al., 1993). SSTH posits an existence of two underlying scripts, one of which is more obvious than the other. To be humorous, the underlying scripts must satisfy two conditions: overlap and incongruity. In the setup phase of the joke, instances of the two scripts are presented in a way that does not give away the less obvious script (due to their overlap). In the punchline (resolution), a trigger expression forces the audience to switch their interpretation to the alternate (less likely) script. The alternate script must differ significantly in meaning (be incongruent with the first script) for the switch to have a humorous effect. An example below illustrates this idea (S_1 is the obvious script, and S_2 is the alternate script. Bracketed phrases are labeled with the associated script).

> ``Is the [doctor]$_{S_1}$ at home?''
> the [patient]$_{S_1}$ asked in his
> [bronchial]$_{S_1}$ [whisper]$_{S_2}$. ``No,''
> the [doctor's]$_{S_1}$ [young and pretty
> wife]$_{S_2}$ [whispered]$_{S_2}$ in reply.
> [``Come right in.'']$_{S_2}$ (Raskin, 1985)

2 Related Work

Of the early prototypes of pun-generators, JAPE (Binsted and Ritchie, 1994), and its successor, STANDUP (Ritchie et al., 2007), produced question/answer punning riddles from general non-humorous lexicon. While humor in the generated puns could be explained by SSTH, the SSTH model itself was not employed in the process of generation. Recent work of Hempelmann (2006) comes closer to utilizing SSTH. While still focused on generating puns, they do so by explicitly defining and applying script opposition (SO) using ontological semantics. Of the more successful pun generators are systems that exploit lexical resources. HAHAcronym (Stock and Strapparava, 2002), a system for generating humorous acronyms, for example, utilizes WordNet-Domains to select phonologically similar concepts from semantically disparate domains. While the degree of humor sophistication from the above systems varies with the sophistication of the method (lexical resources, surface realizers), they all, without exception, rely on phonological constraints to produce script opposition, whereas a phonological constraint is just one of the many ways to generate script opposition.

3 System overview

ConceptNet (Liu and Singh, 2004) lends itself as an ideal ontological resource for script generation. As a network that connects everyday concepts and events with a set of causal and spatial relationships, the relational structure of ConceptNet parallels the structure of the fabula model of story generation - namely the General Transition Network (GTN) (Swartjes and Theune, 2006). As such, we hypothesize that there exist paths within the ConceptNet graph that can be represented as feasible scripts in the surface form. Moreover, multiple paths between two given nodes represent overlapping scripts - a necessary condition for verbal humor in SSTH. Given a semantic network hypergraph $G = (V, \mathcal{L})$ where $V \in Concepts$, $\mathcal{L} \in Relations$, we hypothesize that it is possible to search for script-pairs as semantic circuits that can be converted to a surface form of the Question/Answer format. We define a circuit as two paths from root *A* that terminate at a common node *B*. Our approach is composed of three stages - (1) we build a script model (SM) that captures likely transitions between concepts in a surface-realizable sequence, (2) The script model (SM) is then employed to generate a set of feasible circuits from a user-specified root node through spreading activation, producing a set of ranked scripts. (3) Ranked scripts are converted to surface form by aligning a subset of its concepts to natural language templates of the Question/Answer form. Alignment is performed through a scoring heuristic which greedily optimizes for incongruity of the surface form.

3.1 Script model

We model a script as a first order Markov chain of relations between concepts. Given a seed concept, depth-first search is performed starting from the root concept, considering all directed paths terminating at the same node as candidates for feasible script pairs. Most of the found semantic circuits, however, do not yield a meaningful surface form and need to be pruned. Feasible circuits are learned in a supervised way, where binary labels assign each candidate circuit one of the two classes {`feasible`, `infeasible`} (we used 8 seed concepts, with 300 generated circuits for each concept). Learned transition probabilities are capable of capturing primitive stories with events, consequences, as well as appropriate qualifiers of certainty, time, size, location. Given a chain of concepts S (from hereon referred to as a script) $c_1, c_2...c_n$, we obtain its likelihood $\Pr(S) = \prod \Pr(r_{ij}|r_{jk})$, where r_{ij} and r_{jk} are directed relations joining concepts $< c_i, c_j >$, and $< c_j, c_k >$ respectively, and the conditionals are computed from the maximum likelihood estimate of the training data.

3.2 Semantic overlap and spreading activation

While the script model is able to capture semantically meaningful transitions in a single script, it does not capture inter-script measures such as overlap and incongruity. We employ a modified form of spreading activation with fan-out and path constraints to find semantic circuits while maximizing their semantic overlap. Activation starts at the user-specified root concept and radiates along outgoing edges. Edge pairs are weighted with their respective transition probabilities $\Pr(r_{ij}|r_{jk})$ and a decay factor $\gamma < 1$ to penalize for long scripts. An additional fan-out constraint penalizes nodes with a large number of outgoing edges (concepts that are too general to be interesting). The weight of a current node $w(c_i)$ is given by:

$$w(c_i) = \sum_{c_k \in f_{in}(c_j)} \sum_{c_j \in f_{in}(c_i)} \frac{\Pr(r_{ij}|r_{jk})}{|f_{out}(c_i)|} \gamma w(c_j) \quad (1)$$

Termination condition is satisfied when the activation weights fall below a threshold (loop checking is performed to prevent feedback). Upon termination, nodes are ranked by their activation weight, and for each node above a specified rank, a set of paths (scripts) $S_k \in \mathcal{S}$ is scored according to:.

$$\phi_k = |S_k| \log \gamma + \sum_{i}^{|S_k|} \log \Pr_k(r_{i+1}|r_i) \qquad (2)$$

where ϕ_k is decay-weighted log-likelihood of script S_k in a given circuit and $|S_k|$ is the length of script S_k (number of nodes in the k^{th} chain). A set of scripts $\mathcal{S}$ with the highest scores in the highest ranking circuits represent scripts that are likely to be feasible and display a significant amount of semantic overlap within the circuit.

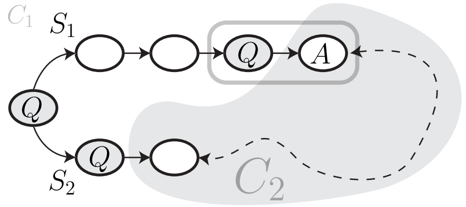

Figure 2: Question(Q) and Answer(A) concepts within the semantic circuit. Areas C_1 and C_2 represent different semantic clusters. Note that the answer(A) concept is chosen from a different cluster than the question concepts

3.3 Incongruity and surface realization

The task is to select a script pair $\{S_i, S_j\ i \neq j\} \in \mathcal{S} \times \mathcal{S}$ and a set of concepts $\mathcal{C} \in S_i \cup S_j$ that will align with some surface template, while maximizing inter-script incongruity. As a measure of concept incongruity, we hierarchically cluster the entire ConceptNet using a Fast Community Detection algorithm (Clauset et al., 2004). We observe that clusters are generated for related concepts, such as *religion, marriage, computers*. Each template presents up to two concepts $\{c_1 \in S_i, c_2 \in S_j\ i \neq j\}$ in the question sentence (Q in Figure 2), and one concept $c_3 \in S_i \cup S_j$ in the answer sentence (A in Figure 2). The motivation of this approach is that the two concepts in the question are selected from two different scripts but from the same cluster, while the answer concept is selected from one of the two scripts and from a different cluster. The effect the generated two-liner produces is that of a setup and resolution (punchline), where the question intentionally sets up two parallel and compatible scripts, and the answer triggers the script switch. Below are the top-ranking two-liners as rated by a group of fifteen subjects (testing details in the next section). Each concept is indicated in brackets and labeled with the script from which the concept had originated:

> `Why does the [priest]`$_{root}$ `[kneel]`$_{S_1}$ `in [church]`$_{S_2}$`? Because the [priest]`$_{root}$ `wants to [propose woman]`$_{S_1}$

Why does the [priest]$_{root}$ [drink coffee]$_{S_1}$ and [believe god]$_{S_2}$? Because the [priest]$_{root}$ wants to [wake up]$_{S_1}$

Why is the [computer]$_{root}$ [hot]$_{S_1}$ in [mit]$_{S_2}$? Because [mit]$_{S_2}$ is [hell]$_{S_2}$

Why is the [computer]$_{root}$ in [hospital]$_{S_1}$? Because the [computer]$_{root}$ has [virus]$_{S_2}$

4 Results

We evaluate the generated two-liners by presenting them as human-generated to remove possible bias. Fifteen subjects ($N = 15$, 12 male, 3 female - graduate students in Mechanical Engineering and Computer Science departments) were presented 48 highest ranking two-liners, and were asked to rate each joke on the scale of 1 to 4 according to four categories: *hilarious (4), humorous (3), not humorous (2), nonsense(1)*. Each two-liner was generated from one of the three root categories (12 two-liners in each): *priest, woman, computer, robot*, and to normalize against individual humor biases, human-made two-liners were mixed in in the same categories. Two-liners generated by three different algorithms were evaluated by each subject:

Script model + Concept clustering (SM+CC) Both script opposition and incongruity are favored through spreading activation and concept clustering.

Script model only (SM) No concept clustering is employed. Adherence of scripts to the script model is ensured through spreading activation.

Baseline Loops are generated from a user-specified root using depth first search. Loops are pruned only to satisfy surface templates.

We compare the average scores between the two-liners generated using both the script model and concept clustering (SM+CC) (*MEAN*=1.95, *STD*=0.27) and the baseline (*MEAN*=1.06, *STD*=0.58). We observe that SM+CC algorithm yields significantly higher-scoring two-liners (one-sided t-test) with 95% confidence.

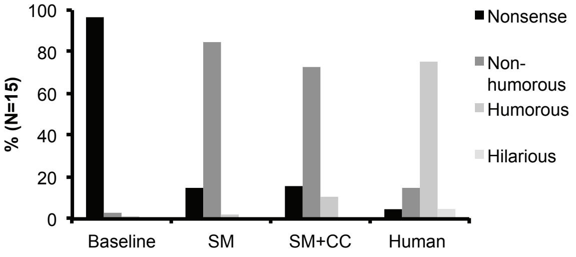

Figure 3: Human blind evaluation of generated two-liners

We observe that the fraction of non-humorous and nonsensical two-liners generated is still significant. Many non-humorous (but semantically sound) two-liners were formed due to erroneous labels on the concept clusters. While clustering provides a fundamental way to generate incongruity, noise in the ConceptNet often leads of cluster overfitting, and assigns related concepts into separate clusters.

Nonsensical two-liners are primarily due to the inconsistencies in POS with relation types within the ConceptNet. Because our surface form templates assume a part of speech, or a phrase type from the ConceptNet specification, erroneous entries produce nonsensical results. We partially address the problem by pruning low-scoring concepts (ConceptNet features a *SCORE* attribute reflecting the number of user votes for the concept), and all terminal nodes from consideration (nodes that are not expanded by users often indicate weak relationships).

5 Future Work

Through observation of the generated semantic paths, we note that more complex narratives, beyond questions/answer forms can be produced from the ConceptNet. Relaxing the rigid template constraint of the surface realizer will allow for more diverse types of generated humor. To mitigate the fragility of concept clustering, we are augmenting the ConceptNet with additional resources that provide domain knowledge. Resources such as SenticNet (WordNet-Affect aligned with ConceptNet) (Cambria et al., 2010b), and WordNet-Domains (Kolte and Bhirud, 2008) are both viable avenues for robust concept clustering and incongruity generation.

Acknowledgement

This paper is for my Babishan - the most important person in my life.

Huge thanks to Max Kelner - those everyday teas at Mattins and continuous inspiration.

This work was supported in part by NSF CDI Grant ECCS 0941561. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of the sponsoring organizations.

References

S. Attardo and V. Raskin. 1991. Script theory revis (it) ed: Joke similarity and joke representation model. *Humor: International Journal of Humor Research; Humor: International Journal of Humor Research*.

K. Binsted and G. Ritchie. 1994. A symbolic description of punning riddles and its computer implementation. *Arxiv preprint cmp-lg/9406021*.

K. Binsted, A. Nijholt, O. Stock, C. Strapparava, G. Ritchie, R. Manurung, H. Pain, A. Waller, and D. O'Mara. 2006. Computational humor. *Intelligent Systems, IEEE*, 21(2):59–69.

K. Binsted. 1996. Machine humour: An implemented model of puns.

E. Cambria, A. Hussain, C. Havasi, and C. Eckl. 2010a. Senticspace: visualizing opinions and sentiments in a multi-dimensional vector space. *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 385–393.

E. Cambria, R. Speer, C. Havasi, and A. Hussain. 2010b. Senticnet: A publicly available semantic resource for opinion mining. In *Proceedings of the 2010 AAAI Fall Symposium Series on Commonsense Knowledge*.

A. Clauset, M.E.J. Newman, and C. Moore. 2004. Finding community structure in very large networks. *Physical review E*, 70(6):066111.

F. Crestani. 1997. Retrieving documents by constrained spreading activation on automatically constructed hypertexts. In *EUFIT 97-5th European Congress on Intelligent Techniques and Soft Computing. Germany*. Citeseer.

L. Friedland and J. Allan. 2008. Joke retrieval: recognizing the same joke told differently. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 883–892. ACM.

C.F. Hempelmann, V. Raskin, and K.E. Triezenberg. 2006. Computer, tell me a joke... but please make it funny: Computational humor with ontological semantics. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11*, volume 13, pages 746–751.

S.G. Kolte and S.G. Bhirud. 2008. Word sense disambiguation using wordnet domains. In *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*, pages 1187–1191. IEEE.

H. Liu and P. Singh. 2004. Conceptneta practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

R. Mihalcea and C. Strapparava. 2006. Learning to laugh (automatically): Computational models for humor recognition. *Computational Intelligence*, 22(2):126–142.

M.E.J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

A. Nijholt, O. Stock, A. Dix, and J. Morkes. 2003. Humor modeling in the interface. In *CHI'03 extended abstracts on Human factors in computing systems*, pages 1050–1051. ACM.

V. Raskin. 1998. The sense of humor and the truth. *The Sense of Humor. Explorations of a Personality Characteristic, Berlin: Mouton De Gruyter*, pages 95–108.

G. Ritchie, R. Manurung, H. Pain, A. Waller, R. Black, and D. OMara. 2007. A practical application of computational humour. In *Proceedings of the 4th. International Joint Workshop on Computational Creativity, London, UK*.

G. Ritchie. 2001. Current directions in computational humour. *Artificial Intelligence Review*, 16(2):119–135.

W. Ruch, S. Attardo, and V. Raskin. 1993. Toward an empirical verification of the general theory of verbal humor. *Humor: International Journal of Humor Research; Humor: International Journal of Humor Research*.

J. Savoy. 1992. Bayesian inference networks and spreading activation in hypertext systems. *Information processing & management*, 28(3):389–406.

S. Spagnola and C. Lagoze. 2011. Edge dependent pathway scoring for calculating semantic similarity in conceptnet. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 385–389. Association for Computational Linguistics.

O. Stock and C. Strapparava. 2002. Hahacronym: Humorous agents for humorous acronyms. *Stock, Oliviero, Carlo Strapparava, and Anton Nijholt. Eds*, pages 125–135.

I. Swartjes and M. Theune. 2006. A fabula model for emergent narrative. *Technologies for Interactive Digital Storytelling and Entertainment*, pages 49–60.

J.M. Taylor and L.J. Mazlack. 2004. Humorous wordplay recognition. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3306–3311. IEEE.

J. Taylor and L. Mazlack. 2005. Toward computational recognition of humorous intent. In *Proceedings of Cognitive Science Conference*, pages 2166–2171.

J.M. Taylor. 2009. Computational detection of humor: A dream or a nightmare? the ontological semantics approach. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 03*, pages 429–432. IEEE Computer Society.