Learning a Probabilistic Model of Event Sequences From Internet Weblog Stories

Mehdi Manshadi[1], Reid Swanson[2], and Andrew S. Gordon[2]

[1]Department of Computer Science, University of Rochester
P.O. Box 270226, Rochester, NY 14627
mehdih@cs.rochester.edu
[2]Institute for Creative Technologies, University of Southern California
13274 Fiji Way, Marina del Rey, CA 90292 USA
swansonr@ict.usc.edu, gordon@ict.usc.edu

Abstract

One of the central problems in building broad-coverage story understanding systems is generating expectations about event sequences, i.e. predicting what happens next given some arbitrary narrative context. In this paper, we describe how a large corpus of stories extracted from Internet weblogs was used to learn a probabilistic model of event sequences using statistical language modeling techniques. Our approach was to encode weblog stories as sequences of events, one per sentence in the story, where each event was represented as a pair of descriptive key words extracted from the sentence. We then applied statistical language modeling techniques to each of the event sequences in the corpus. We evaluated the utility of the resulting model for the tasks of narrative event ordering and event prediction.

Story Understanding Systems

Automated story understanding has proved to be an extremely difficult task in natural language processing. Despite a rich history of research in automatic narrative comprehension (Mueller, 2002), no systems exist today that can automatically generate answers to questions about the events described in simple narratives of arbitrary content. Part of the difficulty is the amount of commonsense knowledge that is necessary to adequately interpret the meaning of narrative text or the questions asked of it. Accordingly, progress in this area has been made by limiting the story understanding task to specific domains and question types, allowing for the hand-authoring of the relevant content theories needed to support interpretation. Mueller (2007) describes a state-of-the-art story understanding system that follows this approach, where a reasonably large number of questions about space and time can be answered with stories involving people eating in restaurants. By limiting the scope of the problem to restaurant stories, all of the relevant expectations about the activity of going to a restaurant could be formalized and integrated into a natural language processing pipeline.

Although promising, approaches like this suffer from problems of scalability. A significant amount of knowledge engineering effort is required to encode expectations about this one activity context, and others have demonstrated that many hundreds (or more) of these schemas are included in our commonsense understanding of everyday activities (Gordon, 2001). Accordingly, there is a strong need for new methods to acquire expectations about narrative events on a much larger scale.

A promising alternative was explored by Singh & Barry (2003), where commonsense expectations about everyday activities were acquired from thousands of volunteers on the web as part of the Open Mind Commonsense Project. Contributors to this knowledge base (StoryNet) authored these expectations as sequences of natural language sentences. This allows for more scaleable knowledge engineering, but requires additional processing to transform natural language expressions into knowledge that can be manipulated in automated story comprehension applications. In many respects, the (fictional) stereotypical activity descriptions that are contributed by volunteers to this knowledge base are not so different from the real (nonfiction) stories that people write in their Internet weblogs, describing the experiences of their daily lives. Given the tens of millions of Internet weblogs in existence, we considered the question: Can the stories that people write in Internet weblogs be used to acquire expectations about everyday event sequences?

In this paper, we describe an approach for acquiring a probabilistic model of event sequences through the large-scale processing of stories in Internet weblogs. This approach begins with the processing of an existing corpus of stories automatically extracted from hundreds of thousands of Internet weblogs. Events within these stories are represented as a predicate-argument pair, one pair for each sentence, consisting of a main verb in the sentence and the head word of its patient argument. We then describe the novel application of existing language modeling technologies to create a probabilistic event model. Multiple variations of this model are then evaluated

for their utility in two narrative comprehension tasks: narrative event ordering and event prediction.

Event Representation in Weblog Stories

At the time this paper was written, it was estimated that there were over 70 million Internet weblogs in existence (Technorati, 2007). However, not all the content of a weblog entry consists of narrative descriptions of peoples' daily lives. By annotating random weblog entries, Gordon et al. (2007) estimated that roughly 17% percent of weblog text belongs to the story genre, the rest consisting of news article quotations, commentary, opinion, and lists, among others. To exploit weblogs as a corpus of narrative text, Gordon et al. developed a number of automated story extraction techniques, incorporating automated part-of-speech tagging, confidence smoothing, and various machine learning algorithms. Using a method that favored recall performance over precision performance, they applied their system to 3.4 million Internet weblog entries. From these, 4.5 million segments of story text were extracted, a corpus consisting of 1.06 billion words.

We used this existing corpus of Internet weblog stories from Gordon et al. as a starting point for our research on modeling event sequences. Our first task was to prepare the corpus for analysis by applying a maximum entropy based sentence delimiting algorithm (Reynar & Ratnaparkhi, 1997). Since the automated story extraction algorithm was not sentence aligned, very often the first and the last sentence of each segment were actually sentence fragments not a complete sentence; therefore we simply removed these fragments from the beginning and end of every segment. The resulting corpus consisted of 371,626 story segments and a total of 66,485,305 sentences.

Our assumption was that these stories consisted largely of time-ordered event descriptions. While very few of these stories described the stereotypical events of any given activity context, the sheer size of the corpus would allow us to acquire useful statistical regularities.

Our strategy was to borrow techniques that have been employed for statistical language modeling in other natural language processing applications. Statistical language modeling is most often used to estimate the relative probability of a given sequence of words, e.g. to improve the performance of automated speech recognition systems. In our own work, we sought to apply these techniques at the level of events, rather than words. Accordingly, it was necessary to devise a scheme for identifying the events in narrative text, and encoding them in a uniform manner as nodes in a transition network. Here we drew upon ideas that have been explored in the area of semantic role labeling (Palmer et al., 2005), where the propositional content of sentences is encoded as predicate-argument relationships, where both the predicates and arguments are simply words extracted from the sentence. Typically, the sense-disambiguated lemma of each verb in the sentence is used as a predicate, and spans of text are identified as arguments, as in the following example.

"I got a flat tire." → get("I", "a flat tire")

In order to create very compact representations of events in weblogs stories, we adapted this style of predicate-argument representation in the following manner. First, we select only one verb in each sentence (henceforth, the "main verb") as a compact expression of the event type. Second, we use only the patient of this verb (referred to using the PropBank convention of 'ARG1') to differentiate between events with the same main verb. Third, we represent this patient of the verb not as a substring of the sentence, but as a single word that is the syntactic head of this argument. With these simplifying principles, we can effectively encode the event described by a sentence as a pair of words: a verb for the event type and a single word for its patient. The following are some sentences that are representative of those found in our story corpus, with the main verb and patient indicated with single and double underlining, respectively.

1. I got a flat tire coming back from the concert.
2. I thought it was just the bad pavement at first.
3. The guy next to me pointed to my tire, so I pulled over.
4. I wish I had a better towing plan, though.
5. I blew a few hundred dollars to get the car home.

By casting narrative sentences as a pair of event words, narrative sequences in weblog stories can be compactly represented as sequences of these pairs. For a sequence consisting of the five sentences above, this encoding would be as follows: got-tire, thought-was, pointed-tire, wish-had, blew-dollars.

Representing event sequences in this manner allows for the efficient application of existing statistical language modeling tools. Here each event word pair can be treated as a single "word" in a "sentence" of consecutive story events, where the length of this event sentence is exactly the number of sentences in the extracted weblog story. In this manner, tools for learning a statistical language model from a corpus of sentences can be used to learn a statistical event model from a corpus of event sequences.

In the sections that follow, we describe our approach to extracting event word pairs from weblog sentences, applying language modeling tools to event sequences, and comparative evaluations of variations of this model on two story interpretation tasks.

Predicate-Argument Extraction

A key component to our approach is the ability to accurately extract the main verb and patient word from each sentence in the corpus. There exist a number of well-known techniques for finding the head verbs of sentences (Collins, 1999) and their semantic role arguments (Gildea & Jurafsky, 2002; Moschitti et al., 2006). However, these techniques typically operate on the syntactic parse trees of

sentences. Given the size of our corpus (over 66 million sentences) and the challenge of producing accurate parse trees from (often ungrammatical) weblog text, we chose instead to develop techniques that did not rely on the syntax of the sentence. Specifically, we developed a set of systems using machine learning algorithms that identified main verbs and patient words using only lexical and part-of-speech information for words in the sentence.

For the task of main verb extraction, we began by creating a training corpus of sentences where the main verb was annotated. Unfortunately we did not have the resources to annotate weblog story text for use as training data. Instead, we created a training corpus automatically by processing the Penn Treebank corpus with a slightly modified version of Collins' head-finding rules (Collins, 1999). Specifically, we modified the head-verb finding rules so as to annotate only the most dominating verb that was not an auxiliary verb. After annotating the main verb in this manner, the syntactic tree information was removed from this corpus, leaving only the lexical items and part-of-speech information in each of the Treebank sentences.

Given the Treebank corpus annotated with the main verbs, a head verb recognition module was developed using machine learning techniques. Using the part-of-speech tags in the Treebank corpus, we extracted each of the verbs in each sentence for use as training data in a binary classification task (main-verb or not main-verb). For each training instance, the lexical token of the verb and a window of words around the verb constituted the feature set. For each verb in the window, we use both the part-of-speech tag and the actual word as features, but for all other words we use just the part-of-speech tags. Although we believe that using the lexical tokens of the words around the verb would improve the performance of main verb extraction on the Treebank data, we limited these features to part-of-speech tags with the hope that these features would generalize well between the newswire text genre of Treebank data and our story corpus. On the other hand, the actual word of the verb is clearly indicative of whether it is the main verb of a sentence or not, and these features are especially helpful in distinguishing auxiliary from main verbs in a sentence.

Using these features, the task of head verb recognition is cast as a binary verb classification problem (main-verb or not main verb). We used the popular SVM-lite implementation of a Support Vector Machine learning algorithm (Joachims, 1999). For each sentence, the verb with the highest confidence level is considered to be the main verb of the sentence. 10,000 sentences of the Treebank corpus were used as the development set and the remaining sentences (around 40,000) were used as a cross validation train/test set. The average accuracy on the Treebank data is 91.7% with the standard deviation of 0.7%. A maximum entropy classifier was also tested for the classification task but performed slightly worse on this task.

Our second task was to identify the lexical head of the patient argument of the main verb in a sentence. The PropBank corpus (Palmer et al., 2005) provides semantic role annotations for a portion of the Treebank corpus, and is widely used for the task of automated semantic role labeling. Semantic role labeling systems typically build the parse tree of the sentence using rule-based or statistical parsers and use this syntactic information to assign semantic roles to the constituents of the sentence. As we discussed previously, we needed to develop a module to extract the patient argument of the main verb, what would be the argument labeled "ARG1" in PropBank annotations, for each sentence using only lexical and part-of-speech information. For this task, our approach was similar to main-verb extraction. We used the Propbank corpus to build a training corpus of sentences that were already annotated with the main verb as well as the lexical head of the main verb's ARG1 argument. As with the extraction of the main verb, a window of tags around each word that could potentially be the head of ARG1 (e.g. nouns, pronouns, verbs, adjectives, but not articles, conjunctions, etc), as well as a window of tags around the main verb, was used as features in a binary classification task. As before, the lexical token of each verb was used as a feature, but only the part-of-speech tags were used for all other words. Two more features for the task of patient word extraction seemed to have a great impact on the performance: a binary feature that indicated whether the word is located after or before the main verb in the sentence, and the sequence of part-of-speech tags between the word and the main verb, where the frequency of the sequence in the training data was at least three. SVM-light was used as the binary classifier to decide whether or not a word is the head of ARG1. The word with the highest confidence level was returned as ARG1 of the main verb.

Experiments showed that if two different classifiers are used for two different cases, one when the head of ARG1 is a verb (i.e. ARG1 is a VP or S) and one for all other cases, the accuracy of the combined system increased significantly. 10,000 sentences were used as the development set and the rest (around 40,000 sentences) were used for the cross validation evaluation of the system. The average accuracy of ARG1 extraction for VP/S case was 86.7% and for all other cases was 75.3%. A binary classifier was also used to decide whether or not to apply the VP/S classifier. The accuracy of this binary classifier was 96.1%.

Language Modeling of Event Sequences

As our event recognition modules (main verb and patient word) use part-of-speech tags as features, these tags needed to be assigned to each word in our story corpus. For this purpose we applied a maximum entropy based part-of-speech tagger (Ratnaparkhi, 1996) to the entire story corpus. We then applied the predicate-argument extraction module to the corpus, producing a new corpus where each story was represented as a sequence of word pairs (main verb and argument word).

Our next step was to develop a probabilistic model of the event sequences in this corpus. In selecting a type of model to use, our goals were to find a model that would allow for a good fit to the data, but also one that was computationally feasible, i.e. fast enough to process the 66 million events extracted from our corpus. With these concerns in mind, we chose the SRI Language Modeling toolkit (SRILM) to build a model for the sequence of events in the form of a traditional language model (Stolcke, 2002).

Language modeling has proven that it is able to model sequential data like sequences of words, tags, and phonemes very well in other natural language processing tasks. There is an intuition for using a language model for sequences of story events. Given the nature of the narrative genre, many events tend to be described in sequential order. For example, if you get a speeding ticket while driving on the highway, it is probable that you subsequently pay the fine; we expect that the probability of "get/ticket" preceding "pay/fine" is much higher than the probability of "have/dinner" preceding "pay/fine".

Language modeling also has a number of characteristics that make it well suited for modeling event sequences in our corpus. Tools like the SRI Language Modeling toolkit are extremely fast, allowing us to analyze significant portions of our corpus. Furthermore, there has been a lot of research on smoothing techniques and other issues related to language modeling that can be applied to story event modeling, improving the utility of such a model in practical applications. Practically speaking, sequences of story events can be analyzed by language modeling tools by treating them as a special type of sentence. Each event in a story (word pair) is treated as a single word, and each story can be represented as a sequence of words. In this manner, the SRILM tool can be used to extract the probability of event unigrams, event bigrams, event trigrams, and so on, from our corpus of 3.7 million stories (sentences in language modeling terminology). To use this language model, one can estimate the probability of a sequence of events by calculating n-gram probabilities. These probabilities can then be used in many different story understanding applications, such as event ordering, event prediction, story classification, story similarity evaluation and automated story generation.

Following this approach, we built a language model using the SRILM toolkit, with an n-gram order of five, and using the default smoothing parameters. In order to evaluate the comparative performance of this model, we also constructed a number of alternative models using this approach.

The first alternative model was built after applying a large-coverage dictionary (Courtois, 2004) to group nouns and verbs that differed only in their inflection or plurality. The idea was to use a dictionary in order to filter some noisy events and to decrease the sparsity of the data. In addition to filtering unknown main verb and patients, the dictionary was also used to replace each word with its lemma. The reason for using lemmatization is to further decrease the sparsity of the data. Here we are looking for a more general model of a story; e.g. no matter what the tense of the verb 'flip' is, we want to have the same predicate for both cases "flip/coin" and "flipped/coins."

A second alternative model was built by using just the main verb of the sentence (without a patient word). The motivation is to find out how much the procedure of extracting the argument contributes to the performance of the model. A third alternative model was built for the case where a random verb was selected (without a patient word) instead of applying our main-verb extraction approach. A fourth alternative was built where both a random verb from the sentence and a random (potential) patient word were extracted. These last two models provide comparisons for measuring how much the main verb extraction and patient word extraction modules contribute to the quality of the model.

Finally, a fifth alternative model was built in which all of the verbs of a sentence were extracted (without patient words). The reason behind building this model was that the performance of the sentence delimitation module on the weblog corpus is not nearly as high as for the newswire data that it was trained on, and many of the weblog sentences actually contain more than one complete sentence. Therefore, by extracting one verb from each delimited sentence, we are actually losing some of the event information that is described in the story. Comparing this model with the model using only the main verb gives us the opportunity to compare the effect of sentence delimitation errors on the performance of the overall model. In the next section we describe an evaluation of all of these different models on two tasks, event ordering and sentence prediction.

Evaluation of Event Models

To evaluate how well our event model and its five alternatives captured the regularities present in this corpus, we conducted a comparative evaluation of all of these models on two simple story understanding tasks. In each case, the models were trained on a subset of 900,000 stories and a different subset of 100,000 stories was used to test the system.

Our first evaluation was based on an event ordering task, designed to evaluate how well these models capture the expected order of events in narrative text. The task was formulated as a binary decision regarding the true ordering of events in a test story. For each story in the test set, we asked our model to decide which of two event sequences were more likely: the events of the story in their original order, or a random ordering of the same events. As an illustration, the example story presented earlier in this paper would be presented to our primary model as the following two options:

1. (*correct*) got-tire, thought-was, pointed-tire, wish-had, blew-dollars
2. (*incorrect*) wish-had, got-tire, blew-dollars, thought-was, pointed-tire

As a baseline, a random selection would yield an accuracy of 50% on this task. The comparative accuracy for each of the six models on this task is presented in Table 1.

Model	*Percent correct*
Main verb + patient word	63.4%
a1: Lemmatized verb + patient	63.9%
a2: Main verb only	64.7%
a3: Random verb only	60.5%
a4: Random verb + random patient	55.4%
a5: all verbs only	61.5%

Table 1. Evaluation on event ordering task

The results presented in Table 1 indicate that our primary model (main verb + patient word) is outperformed by two of the alternative models. The first alternative model (a1) sees a slight improvement, most likely due to the decrease in data sparsity due to the application of the dictionary. However, the best result overall is found in the second alternative model (a2) where only the main verb extraction module is applied, representing events as single word entities (without any patient word). In addition to further diminishing the data sparsity problem, the high performance of this model can be attributed to the relative difficulty of the task: ordering the next verb in a sequence is comparatively easier than ordering the verb and its patient. Considering the models that select random verbs and patients (a3 and a4), it is clear that our extraction approach provides some utility to the resulting model, but the biggest contributor is the selection of an appropriate main verb. Also, low performance of the fifth alternative model (a5) demonstrates that selecting only a single verb to represent events is more effective than using all verbs in a sentence.

Our second evaluation was based on an event prediction task, designed to evaluate how well these models could make accurate predications about the next events in a narrative sequence. In this task, the model is asked to decide which of two story sequences is correct. The first is an unmodified sequence from the test corpus, but the second replaces the last event in the sequence with a different event selected at random from the entire corpus. As an illustration, the example story presented earlier in this paper would be presented to our primary model as the following two options:

1. (*correct*) got-tire, thought-was, pointed-tire, wish-had, blew-dollars
2. (*incorrect*) got-tire, thought-was, pointed-tire, wish-had, add-dimension

Here "blew-dollars" is replaced with a random event picked from the corpus, "add-dimension". This evaluation approximates the task of predicting the next event (the last event in the sequence) when an event sequence is given (the preceding events). As with the first evaluation, a random choice baseline would yield an accuracy of 50%. Table 2 shows the comparative results for the second evaluation.

Model	*Percent correct*
Main verb + patient word	52.5%
a1: Lemmatized verb + patient	52.3%
a2: Main verb only	53.2%
a3: Random verb only	51.5%
a4: Random verb + random patient	50.3%
a5: all verbs only	51.8%

Table 2. Evaluation on event prediction task

The results presented in Figure 2 show that the event prediction task is much more difficult than the event ordering task. However, the relative performance of the models on these two tasks is very similar. Again, our primary model (main verb + patient word) is outperformed by the model consisting of the main verb only (a2), although in this task there is no advantage to be gained by applying a dictionary (a1). As in the previous task, selecting random verbs or patients (a3 and a4) or all verbs (a5) yields lower performance, validating the utility of our main verb extraction module.

Overall, these two evaluations demonstrate that language modeling is an appropriate technology for modeling event sequences, but only for the task of event ordering. Using only our main verb extraction module to represent events, we achieve performance that is 14.7% above a random baseline. In contrast, performance on the task of event prediction is much lower, only a few percentage points above the random baseline. For this task, it is clear that incremental improvements and alternative approaches will need to be explored in future work.

Discussion

In this paper we have explored a novel approach to the development of broad-coverage knowledge resources for use in automated story-understanding systems. We focused specifically on the development of an event model, a resource for encoding expectations about event sequences in everyday narrative contexts. Our approach was to develop a means of extracting event sequences from narrative text documents, apply this technology to an extremely large corpus of Internet weblog text, and use language modeling techniques to build a probabilistic model. Our evaluation of this model, and several alternative models, demonstrated some utility of our approach for the task of ordering narrative events, but not for the task of event prediction.

As a first effort in the research area of event model extraction from story corpora, we are encouraged by our results. However, there are a number of challenges that must be addressed in future work in order to achieve levels of performance that have practical utility.

First, we expect that significant improvements could be obtained if the story corpus was preprocessed with more

care. Compared to the newswire text on which the extraction models were trained, the story corpus suffers from ungrammatical sentences, fragments, and spelling errors. These problems hinder the performance of the automated sentence delimitation, part-of-speech tagging, and word extraction modules that we employ. Also, the corpus was created using automated story extraction algorithms, favoring high recall over high precision. Although this decision might be appropriate for the task of story retrieval from Internet weblogs (its original purpose), an approach that favored precision over recall would be preferable for the purpose of event modeling.

Second, our approaches to main verb and patient word extraction could be significantly improved. Certainly, some advanced natural language processing techniques (e.g. anaphora resolution) could be appropriately applied to this corpus to aid in argument extraction. However, we expect that the most significant gains could be achieved by working with a large amount of annotated Internet weblog text. Having annotated weblog data would allow us to train new sentence delimiters, part-of-speech taggers, main verb extractors, and patient word extractors that were specifically tuned to this genre of data. More importantly, an annotated corpus would allow us to evaluate the performance of each of these components on the test data, giving us a better idea of which modules most need improvement. At the very least, it would be good to have an annotated corpus of weblog text where the main verb and patient words were tagged. This would be comparatively less expensive to produce than a corpus of full syntactic parses and semantic role labels.

Third, we believe that there are suitable alternatives to language modeling as a framework for modeling event sequences. One of the problems with this type of n-gram model is that it does not optimally handle missing events in a sequence. The nature of storytelling is about the selective description of events to compose an effective narrative, not simply the reporting of all events in an episode. Statistical models that more robustly handle missing events (e.g. Hidden Markov Models) may be more suitable for this task, if challenges of scalability could be overcome.

Finally, we believe that future work in this area must be conducted in the context of a specific story understanding task in order to evaluate the utility of different approaches. A system that achieves 70% or 80% on either the event ordering or event prediction task may be a remarkable technical achievement, but if practical automated story understanding systems require accuracy of 90% or more, then alternatives to learning these models from data will need to be pursued.

Acknowledgments

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

Collins, M. (1999). Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania.

Courtois, B. (2004) Dictionnaires électroniques DELAF anglais et français. In Leclère, Laporte, Piot & Silberztein (eds.) Syntax, Lexis and Lexicon-Grammar: Papers in Honour of Maurice Gross. Amsterdam: John Benjamins.

Gildea, D. & Jurafsky, D. (2002). Automatic labeling of semantic roles. Computational Linguistics, 28(3):245-288.

Gordon, A., Cao, Q, & Swanson, R. (2007) Automated story capture from Internet weblogs. 4th International Conference on Knowledge Capture, Whistler, Canada.

Gordon, A. (2001) Browsing image collections with representations of commonsense activities. Journal of the American Society for Information Science and Technology, 52(11):925-929.

Joachims, T. (1999) Making large-Scale SVM Learning Practical. In Schölkopf, Burges, & Smola (eds.) Advances in Kernel Methods: Support Vector Learning. Cambridge, MA: MIT Press.

Moschitti, A., Pighin, D. & Basili, R. (2006) Semantic role labeling via tree kernel joint inference. Proc. of Computational Language Learning, New York.

Mueller, E. (2002) Story understanding. In Nadel (ed), Encyclopedia of Cognitive Science, vol. 4. London: Nature Publishing Group, pp. 238–46.

Mueller, E. (2007) Modelling space and time in narratives about restaurants. Literary and Linguistics Computing 22(1):67-84.

Palmer, M., Gildea, D., & Kingsbury, P. (2005) The Proposition Bank: An Annotated Corpus of Semantic Roles. Computational Linguistics 31(1):71-106.

Ratnaparkhi, A. (1996) A maximum entropy model for part-of-speech tagging. Proc. of Empirical Methods for Natural Language Processing, University of Pennsylvania.

Reynar, J. & Ratnaparkhi, A. (1997) A maximum entropy approach to identifying sentence boundaries. Proc. of Applied Natural Language Processing, Washington, D.C.

Singh, P. & Barry, B (2003) Collecting commonsense experiences. 2nd International Conference on Knowledge Capture, Sanibel Island, FL.

Stolcke, A. (2002) SRILM: An Extensible Language Modeling Toolkit, International Conference on Spoken Language Processing, Denver, Colorado.

Technorati (2007) State of the Blogosphere / State of the Live Web. Retrieved August 2007 from http://www.sifry.com/stateoftheliveweb.