Tracking Dragon-Hunters with Language Models

Anton Leuski
Institute for Creative Technologies
University of Southern California
13274 Fiji Way, Suite 600
Marina del Rey, CA 90292, USA
leuski@ict.usc.edu

Victor Lavrenko
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
140 Governor's Drive
Amherst, MA 01003, USA
lavrenko@cs.umass.edu

ABSTRACT

We are interested in the problem of understanding the connections between human activities and the content of textual information generated in regard to those activities. Massive online collaborative environments, specifically online virtual worlds, where people meet, exchange messages, and perform actions can be a rich source for such an analysis. In this paper we study one of such virtual worlds and the activities of its inhabitants. We explore the existing dependencies between the activities and the content of the chat messages the world's inhabitants exchange with each other. We outline three experimental tasks and show how language modeling and text clustering techniques allow us to explore those dependencies successfully.

1. INTRODUCTION

Information Retrieval, specifically, search deals with retrieving documents that are topically-similar to a user query from relatively static collections. Topic Detection and Tracking (TDT) focuses on locating and following interesting topics in a continuous and constantly changing stream of stories. Data Mining (DM) and Information Extraction (IE) focus on extracting well-defined properties or features of entities from static collections. In this paper we explore another research area that deals with analyzing a continuous stream of textual information that is linked to a parallel and also continuous stream of data (see Table 1). For example, consider a stream of news stories coming from a newswire and consider a stream of data from the financial market such as stock prices and trading volumes of individual companies. Both the news stories and the changes in financial market are influenced by the same world events and therefore linked. Exploring the existing dependencies between the streams opens fascinating opportunities [9].

Another relevant example would be a stream of data collected from a seismograph and the text that contains annotations, research articles, and news publications that deal with the recorded earthshakes. However, we are interested in cases when both the text and the data are the result of human conversations and activities. In this paper we talk about them as streams of words and actions.

	static	dynamic
text vs. text	search	TDT
text vs. data	DM/IE	

Table 1: A comparative classification of text-related research fields. The problems we discuss in this paper occupy the bottom right corner of the table.

In the domain of Language Grounding researchers study how language understanding and language learning is connected to every day activities and human ability to perceive and explore the world [11]. For example, they observe toddlers recording streams of everything a child hears, sees, and does. Then they attempt to reconstruct how the humans acquire language knowledge. The knowledge of how the words and actions link together makes possible development of successful language training systems. Johnson and his colleagues [8] created an interactive virtual environment that simulates student's presence in a foreign country. The students hear and read new words together with both observing and performing actions in the simulations. On the other hand, computers also can benefit from a clearly defined link between words and actions. For example, Fleischman and Hovy [7] studied a virtual environment where users converse with computer-generated characters. They demonstrated that taking into account the situational context – predicting what kind of language the system should expect from the user based on the current state of the virtual world, the user's task, and her progress through the task, – may significantly improve system's natural language understanding. In other words, the system predicts the content of the text stream from the content of the action stream.

Most of the current research deals primarily with one-on-one interactions where either two humans talk to each other or a human converses with a character in a virtual world. We are interested in analyzing word and action dependencies in large collaborative environments where multiple people organize, perform actions together, and exchange information regarding those actions. Note a bi-directional nature of these dependencies: we may focus on detecting, tracking, and predicting activities from the text messages; or detect and track text messages relevant to a specific activity.

2. VIRTUAL WORLDS

A MMORPG (Massively Multiplayer Online Role-Playing Game) is an online computer role-playing game in which a large number of players can interact together or against one another in the same game at the same time. An MMORPG follows a client-server model in which players, running the client software, are represented in the game world by an avatar – this is usually a graphical representation of the character they play. Providers, usually the game's publisher, host the persistent worlds these players inhabit. This interaction between a virtual world, always available for play, and an ever-changing, potentially worldwide stream of players characterizes the MMORPG genre [12].

Once a player enters the game world he or she can engage in a variety of activities with other players ranging from chat with their friends or guild members to teaming up in order to kill large enemies or to complete complex tasks or quests that are not achievable alone. Killing these enemies (typically referred to as mobs by gamers) yield the players experience points and equipment or loot such as armor and weapons. Both the experience points (used to "upgrade" the character or his abilities) and the loot gained from slaying mobs, help to improve the character so he can handle fighting in more adverse situations.

Players interact with each other using both the textual chat and through the avatar actions. Some more advance games have elaborate avatars that may represent a wide variety of gestures and emotions.

MMORPGs (sometimes the term Virtual Worlds is also used) are immensely popular, with several commercial games reporting millions of subscribers. Some analysis suggests that there are at least 35-40 million MMORPG subscribers around the world [13]. The demographics analysis conducted by Lee [14] shows that 40% of the subscribers are spending more than 20 hours per week on-line.

Most of the MMORPGs have well-developed economy rules. Players collect or purchase resources, produce items such as swords or magic potions and sell those items to other players. Castronova [4] did a thorough economic analysis of the game called *EverQuest* and concluded that the virtual goods (items, loot, experience points, etc) produced by the players have a noticeable monetary value and can be exchanged for real-life money at places such as eBay (`http://www.ebay.com/`), ige (`http://www.ige.com/`), etc. His analysis showed that the players generated quite significant $2,266 per capita yearly. The currency exchange rates for the most popular games can be found on the web [6].

We give such an extended introduction into the domain of MMORPGs to highlight two important points: The first, massive multiplayer online games are a very serious human activity. This activity is primarily recreational, but it does not make it less serious. A significant number of people spending a significant amount of time playing and potentially accumulate a noticeable amount of wealth doing so. We expect that as technology develops, these games are going to attract more and more participants. We also observe the appearance of non-recreational virtual worlds, e.g., – games oriented towards learning. Making sense of the things that are happening in these environments is becoming a very important task.

The second point is that these on-line games are a very good model of social processes existing in the real world. We have a massive record of what people were saying, who said what, where they said it, when, and what they were doing at that moment. Statistical analysis of this data creates exciting opportunities and novel challenges to the field of Information Retrieval.

We continue the paper by introducing a collection of logs from one of the small MMORPGs. We define three questions that we investigate on that collection: we study how well we can detect a presence of a particular player activity from the content of their conversations; we establish who of the players participated in the activity; and we consider how players' conversation depends on their geographic location in the virtual world. We describe our experiments, present the results of our analysis and conclude the paper with an extensive outline of possible direction for future work.

Our experimental data comes from *BladeMistress*, a small non-profit low-bandwidth fantasy-oriented MMORPG[1] [1]. As in much larger virtual worlds, this game has players collecting resources, exploring the world, killing dragons and other monsters, practicing magic, trading items and stories. The player avatars move around in a 3D virtual world which is divided into squares. Our data includes both chat and game logs from September 2002 to August 2003.

The chat log is the record of all chat messages exchanged in the game. Each message is tagged with the time of the message (with one second resolution), the grid coordinates of the speaker, the speaker name and the message addressee. There are several different modes of messaging that determine who is going to see it: a player can broadcast the message to the whole world, limit its scope to players in the same square, direct the message to a specific player or to a group of players.

The game log records a single game activity – players killing monsters. Each activity is tagged with its time (with one minute resolution), the name of the monster and the names of the people present at the same square at the moment of the kill.

3. PROBLEM FORMULATION

As we discussed in the introduction, the goal of this work is to understand the connections between collaborative activities of players in a social environment and messages they exchange in relation to these activities. In this section we will attempt to turn this informal description into a mathematical formalism which will ultimately guide us towards a solution.

We start by describing the observable variables. Our data consists of a set of messages $\{\mathcal{M}_i : i{=}1\ldots N_\mathcal{M}\}$ and a set of activities $\{\mathcal{A}_j : j{=}1\ldots N_\mathcal{A}\}$. Each message $\mathcal{M}$ is represented as a tuple $\{W, X, Y, T, S, R\}$. Here S and R represent the sender and recipient of the message; both are discrete random variables taking values in $\mathcal{V}_\pi$, the list of known players. $\mathcal{V}_\pi$ may also include special values representing groups of players, such as *'everyone'*. X, Y and T are integer-valued variables representing the location and the time when the message was produced by the sender S. Finally, W is a sequence of words representing message content, each word being a discrete random variables drawn from the vocabulary $\mathcal{V}_w$. An activity $\mathcal{A}$ is a tuple $\{A, X, Y, T, \Pi\}$, where A represents activity type (e.g. a *'monster kill'*), taking values in a discrete set $\mathcal{V}_a$. As before, X, Y and T represent the time and location of the activity. When the activity is

[1]The authors deny any first-hand knowledge of the game beyond the access to its logs and website.

stretched is space and time, we assume that X, Y, T marks an important event, such as the moment when the monster died. Π represents players directly involved in the activity, it is a set of discrete random variables drawn from $\mathcal{V}_\pi$. Note that it is possible to extend the framework to model the *role* of each participant in the activity. While straightforward, such extension is beyond the scope of this paper.

The aim of our research is to discover hidden "connections" between the messages $\mathcal{M}_i$ and observed activities $\mathcal{A}_j$. We will attempt to capture these connections by constructing statistical language models for the various activity types. We can then use these language models to tackle a wide array of mining and discovery problems. We are particularly interested in addressing the following tasks:

(A) **Activity detection.** Suppose we cannot observe activities directly. Given a collection of messages $\mathcal{M}_{1..m}$, try to predict times and locations of a specific type of activity (e.g. 'monster kills').

(B) **Player forensics.** Suppose we know the time when a specific activity happened, but do not know location or the participants. Can we find the likely participants by analyzing the messages $\mathcal{M}_{1..m}$? We call this task *forensics* because we can view message content W as *traces of evidence* hinting to potential involvement of a given player in an activity.

(C) **Investigative search.** Given an instance of activity, find all messages directly relevant to that activity. Note that this task is not as simple as it may seem. Some very relevant messages may have been generated by players who did not directly participate in the activity (e.g. messages inciting other players to join a monster kill). Conversely, players who did participate in the activity in question may send and receive messages completely unrelated to that activity and therefore non-relevant.

(D) **World mapping.** Given all messages from all players we explore if there is a correlation between what players saying and their location in the world, e.g., is the message content different for the area where monsters live from the conversations occurring in the other parts of the world.

Beyond the four tasks suggested above, one could certainly define other problems that would become feasible if we had an accurate model of what type of language is likely to be associated with specific activities. The scope of this paper will be limited to tasks (A), (B), and (D). While we are very interested in addressing (C), the absence of relevance judgments makes this task difficult to evaluate quantitatively and we leave this task for future work. In the following two sections we will describe our approach to constructing activity-specific language models and will discuss their performance on tasks (A) and (B).

4. EXPERIMENTAL DATA

We processed the chat log by removing non-ASCII characters and empty messages, and normalizing the time stamps of the messages and the log format – the original log format showed some variations over the collection time period. We have a chat log of 5,514,173 messages that take approximately 310MB of disk space. There are 284,728 unique terms in the vocabulary and 19,144 unique login names.

In the game log we normalized the timing of the activities to synchronize it with the chat log. We do not have up-to-a second accurate information from the game log, so we assume that each kill happens at the last second of the recorded minute. We also tag each record with an approximate location of the activity. We consider the recent locations of each player present at the kill from the chat log, – locations of all the messages from the players in the preceding minute, – and average those coordinates. There are 447,874 monster kills recorded. Some monsters are stronger than others and require more people getting together to succeed at the task. Such activities are more interesting to us because they potentially require a more elaborate and intense discussion among the players. Figure 1 shows the plot of the number of individual activities as a function of the number players involved. We focus our attention only on the activities with at least three players involved that makes it 16,337 recorded kills.

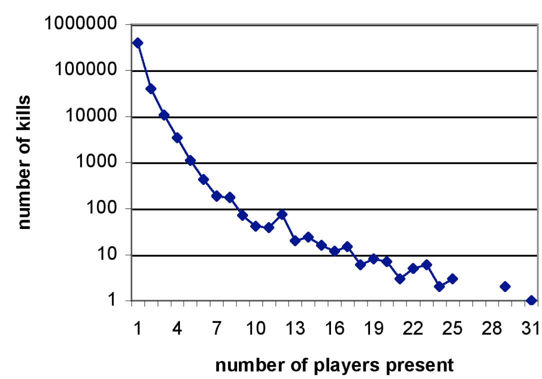

Figure 1: Shows the distribution of number of kill activities as a function of the number of people present. The plot is drawn on logarithmic scale.

We divided the data into training and testing sets at midnight of June 1st, 2003. We completely excluded a day full of data (May 31) to avoid contaminating the test set. Table 2 shows the size of the testing and training sets.

Dataset	Training	Testing
message count	4,230,126	1,264,586
# of kills with at least 3 people	12,129	4,174
— " — 5	1691	540
— " — 7	588	130

Table 2: The size of the training and testing subsets. We show the number of kills with at least 3, 5, and 7 players present.

5. ACTIVITY DETECTION

In the activity detection task, we are given a testing collection of messages $\{\mathcal{M}\}$ and asked to guess the time t and location x, y of all activities of a given type a. We do not have to predict the participants of each activity, and furthermore we will assume that the training set will not include any information about the participants. This is done intentionally, because in many non-virtual domains we will not know who participated in the activity of interest. However, our training data will include a set of training messages $\mathcal{M}_{1..m}$ and a set of activities $\mathcal{A}_{1..n}$ with known times and locations.

We approach the problem of activity detection as follows. First, we cluster the training and testing messages into a set of groups G_{xyt} by their proximity in space and time. The exact grouping procedure is described below. Then we use the training groups to estimate language models $P_a(\cdot)$ specific to each activity a. Finally, for each testing group we determine whether it is more likely to be a sample from activity-specific language model $P_a(\cdot)$, or from its opposite $P_{\neg a}(\cdot)$. We evaluate the quality of our models using the standard signal detection methodology.

5.1 Activity-specific language models

We are given a set of training messages $\mathcal{M}_{1..m}$ and a set of training activities $\mathcal{A}_{1..n}$. Our goal is to construct a model of language associated with all activities of a given type, e.g. a monster kill. The difficulty comes from the fact that even when messages and activities are fully observable, we do not know which messages are related to which activities. To resolve this problem, we are going to consider spatio-temporal proximity of messages and activities. We are going to assume that all messages $\mathcal{M}_i$ that are generated in a small radius around the activity $\mathcal{A}_i$ and around the same time are relevant to that activity. Upon close examination of the data we must admit that the assumption is false. There will always be bystanders – players that happen to be in the immediate vicinity of the activity without participating in it. Even more frequently, activity participants will exchange messages on topics that are not directly related to the activity. Occasionally there may also be remote participants – players who incite or coordinate the activity without being physically present at the site. Nevertheless, assuming that all nearby messages are relevant to the activity is not entirely unreasonable. First, from a brief analysis of our data a large proportion of nearby messages do appear to be relevant. Second, when we estimate activity-specific language models we will average word probabilities over a large number of activities of the same type. We hope that words that come from genuinely relevant messages will occur time after time, whereas words that come from unrelated messages will be different every time and their statistics will "wash out".

For a given activity type a, we estimate the corresponding language model in the following fashion. First, we aggregate the messages $\mathcal{M}_{1..m}$ into a set of groups G indexed by time and location:

$$G_{xyt} = \{M_i : g(X_i){=}x, g(Y_i){=}y, g(T_i){=}t\} \qquad (1)$$

Where the function $g(x){=}x{\cdot}\lfloor x/\delta \rfloor$ quantizes its argument to a given granularity δ. We use separate δ for space and time dimensions. The groups are arranged in such a way that they overlap by half along each dimension, so every message falls into $2^3{=}8$ distinct groups. Forcing the groups to overlap helps us to avoid boundary effects where an activity and a nearby message fall into different (neighboring) groups. After constructing the groups, we label them with activities that happen within the time-space region corresponding to the group, so that $a{\in}L_{xyt}$ if and only if there is an activity $\mathcal{A}_j$ of type a such that $q(X_j){=}x, q(Y_j){=}y$ and $q(T_j){=}t$. Once all the groups are labeled, we construct activity-specific word counts as follows:

$$N_a(w) = \sum_{k:a\in L_k} \sum_{i:M_i\in G_k} N(W_i;a) \qquad (2)$$

Here the first summation goes over all groups k labeled with activity a, and the second summation computes the total number of times the word w occurred in all messages falling into group k. After we have word counts for all activity types, we estimate the activity-specific probability of observing the word w as:

$$P_a(w) = \lambda^2 \frac{N_a(w)}{\Sigma_v N_a(v)} + \lambda(1{-}\lambda)\frac{N_{\neg a}(w)}{\Sigma_v N_{\neg a}(v)} + \frac{(1{-}\lambda)}{|\mathcal{V}_w|} \qquad (3)$$

Here λ is the smoothing parameter, which was set to 0.9 in our experiments. $N_{\neg a}(w)$ represents the overall count of w in groups *not* labeled by a, and $|\mathcal{V}_w|$ is the vocabulary size. The last term in equation (3) is necessary because we need to allocate non-zero probability mass to words that do not appear in any training messages.

5.2 Detecting activity from text

In this section we describe how we can predict the times and locations of activity a using the activity-specific language model $P_a(\cdot)$ derived in the previous section. Our predictions will be based on the time, location and content of testing messages. First, we aggregate the individual testing messages $\mathcal{M}_i$ into groups G_{xyt} employing exactly the same procedure that we used to cluster the training messages (equation 1). Then, for each testing group G_{xyt} we perform the likelihood ratio test:

$$\frac{P_a(G_{xyt})}{P_{\neg a}(G_{xyt})} = \frac{\prod_{\mathcal{M}_i\in G_{xyt}} \prod_{w\in W_i} P_a(w)}{\prod_{\mathcal{M}_i\in G_{xyt}} \prod_{w\in W_i} P_{\neg a}(w)} \qquad (4)$$

The numerator in equation (4) represents the likelihood that all messages in group G_{xyt} are i.i.d. random samples from the activity-specific language model $P_a(\cdot)$. Similarly, the denominator gives the likelihood of observing G_{xyt} as a random sample from $P_{\neg a}(\cdot)$, the language model not associated with activity of type a. Large values of equation (4) indicate that the language of messages around time t and location x, y closely resembles word statistics associated with activity a, and allows us to hypothesize that activity a took place around this time and location. Conversely, small values of the likelihood ratio indicate that most likely activity a did not take place around x, y, t.

5.3 Evaluation

If we set a decision threshold θ over the likelihood ratio and take all tuples x, y, t that scored above θ as positive, we will get a fixed set of hypotheses ($\{H_\theta\}$). We can then compare $\{H_\theta\}$ against the ground truth – the set $\{\mathcal{A}\}$ of activities that are known to have occurred in the testing set. Comparison can be carried out with many different metrics, for example average distance to the true activity, binary accuracy, etc. We are going to adapt signal detection methodology and use True Positive and False Positive rates as our evaluation measures. True positive rate (TP) is the proportion of real activities that were correctly identified in our list of hypotheses. False positive rate (FP) is the

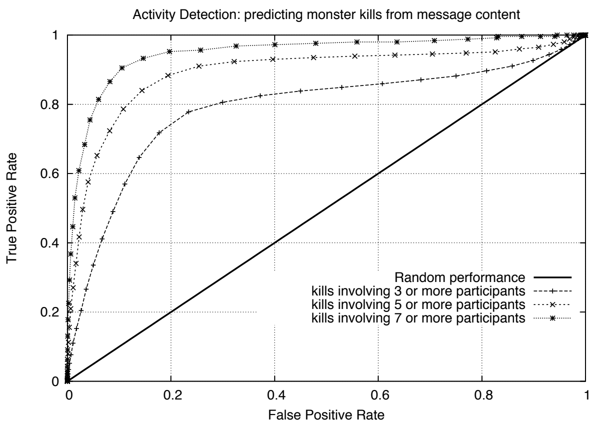

Figure 2: ROC curves for detecting a monster kill by analyzing message content. The system is more accurate on kills involving more players, achieving 90% recall with a 10% false positive rate.

proportion of non-activity locations that were erroneously included among the hypotheses. Formally the measures are defined as:

$$\mathrm{TP}_\theta = \frac{|\{H_\theta\} \cup \{\mathcal{A}\}|}{|\{\mathcal{A}\}|} \qquad \mathrm{FP}_\theta = \frac{|\{H_\theta\} - \{\mathcal{A}\}|}{|\neg\{\mathcal{A}\}|} \tag{5}$$

Different settings of the decision threshold θ will lead to different true positive and false positive rates. In general, different users exhibit different tolerance to false alarms, and consequently prefer different thresholds. A common way to evaluate performance for all users is through a Receiver Operating Characteristic (ROC) curve, which graphically shows a tradeoff between true positive and false positive rates for all possible settings of the decision threshold θ.

Figure 2 shows ROC curves for the task of detecting significant activities from the message content. In this case the activities we are detecting represent monster kills involving at least 3, 5 or 7 participants. Kills involving many participants are rare, but also more interesting because of the extensive collaboration required for success. Messages and kills were aggregated into regions covering 6x6 squares on the map and spanning 10 minutes. From looking at the ROC curves we immediately see that the system is substantially more accurate in detecting larger kills (7 participants), achieving an impressive 90% true positives with a false alarm rate of 10%. For users requiring higher levels of recall, the system would be able to cut the monitoring costs in half (50% false positives) while retaining 98% of true positives. Detection accuracy is somewhat lower for kills involving fewer participants, yielding 60% and 80% true positives at 10% false alarm rate for kills with 3 and 5 participants respectively.

An attentive reader may wonder how sensitive the system is to the way we aggregated messages into groups. Using a resolution of 6x6 squares and 10-minute intervals may not provide sufficient resolution for some applications. We address these questions in table 3, where we show how detection accuracy varies with the square size and time span. The numbers reported in table 3 represent the *area under*

square size	3-person kill	5-person kill	7-person kill
4x4	**0.7929**	**0.9049**	**0.9443**
6x6	0.7822	0.8984	0.9430
8x8	0.7672	0.8863	0.9428
16x16	0.7509	0.8686	0.9216
40x40	0.7064	0.8342	0.8751
time span			
3 min.	0.7597	0.8708	0.9099
10 min.	0.7634	0.8839	0.9396
30 min.	0.7672	0.8863	**0.9428**
5 hrs.	**0.7841**	**0.8948**	0.9206

Table 3: Accuracy of the detection system for different square sizes and time spans. Numbers represent the area under the corresponding ROC curve. The system is generally more accurate for small square sizes and longer time spans. However, detection on short (3-minute) time spans is not significantly worse.

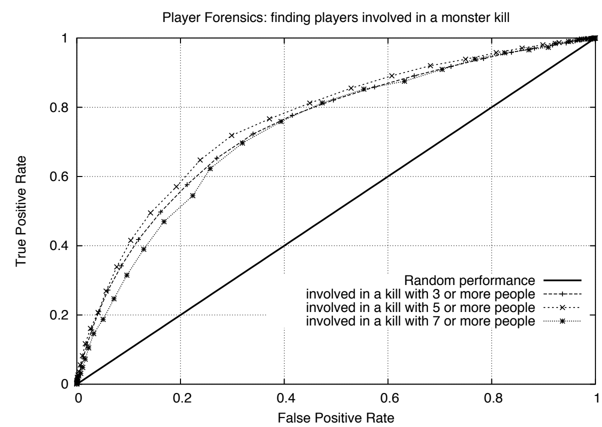

Figure 3: ROC curves for detecting the players involved in a monster kill. The system achieves similar performance detecting participants of 3-, 5-, and 7-person kills.

ROC, which is a single-number measure commonly used to evaluate the quality of an ROC curve. The table suggests that our system is more accurate on smaller square sizes and longer time ranges. This means that the system will be able to pinpoint the location of a hypothesized activity, but may not be very accurate about the time when that activity will take place. However, detection accuracy is still very respectable on shorter time intervals, particularly if we are concerned with detecting larger kills.

6. PLAYER FORENSICS

We now turn our attention to the second task defined in section 3. This time, we are given a time and location of a particular activity of interest, but we do not know the players who were involved. We are also given a set of all messages observed within the same time span when the activity was recorded. We know the sender of each message, but do not

time span	3-person kill	5-person kill	7-person kill
20 sec.	0.6739	0.6355	0.6210
1 min.	0.6892	0.6616	0.6208
3 min.	0.6967	0.6726	0.6354
10 min.	0.6963	0.6896	0.6400
30 min.	0.7158	0.6923	0.6371
1.5 hr.	**0.727**	**0.7077**	**0.6629**

Table 4: Accuracy of participant detection for different time spans. Numbers represent the area under the corresponding ROC curve. The system is generally more accurate when provided with a longed stretch of messages from a particular player.

know the location where the message was sent from. Our goal is to figure out which players participated in the activity by analyzing the content of their messages. We approach this problem in the same manner as activity detection. The main difference is that this time we are not provided with message coordinates (if we were, the problem would become trivial). We aggregate all messages from a given player in a given time span, then label as positive the groups that correspond to activity participants. We use labeled training groups to estimate activity-specific language models as described in section 5.1. After the models are computed, we compute the likelihood ration (equation 4) for every player group in the testing set. We evaluate the detection accuracy using ROC curves as described in section 5.3.

Figure 3 shows performance of the system in identifying participants in 3-, 5-, and 7- person kills with the time span of 10 minutes. The results are pool-averaged over all players and all time spans containing a target kill. The overall performance is noticeably lower than what the system achieved on the activity detection task. However, performance is still substantially above the random baseline, and the higher false alarm rates may be tolerable due to a smaller overall number of negatives in this task. Another interesting observation is that detection accuracy appears to be insensitive to the size of a kill in question – the ROC curves for identifying participants in 3-person and 7-person kills are almost the same. Table 4 shows how much performance is affected by varying the time span around the activity. The numbers represent the area under the corresponding ROC curve and suggest that the system identifies participants most accurately when given longer spans of messages from a user. However, performance is reasonable for time spans as short as 20 seconds.

The task of finding activity participants can also be thought of as a ranked retrieval task – in some settings the goal may be to quickly find a few obvious participants, and then use additional information gained from them (e.g. alliances, guild membership, friend lists) to identify the remaining participants. In such precision-oriented setting, it would be appropriate to rank the hypothesized participants by the likelihood of their involvement and evaluate using precision at different ranks. Table 5 shows precision at ranks 5-100 for ranking hypothesized participants of 3-,5- and 7-person kills. We observe very high accuracies for 5- and 7-person kills: out of the top 100 hypotheses over 75 times the player in question was actually involved in a kill. The precision is somewhat lower for 3-person kills, especially at the very top of the ranked list. Overall, table 5 suggests that our system may be used to rapidly identify a few players involved in an activity of interest.

rank	3-person kill	5-person kill	7-person kill
5	0.4000	1.0000	1.0000
10	0.3000	0.8000	0.9000
15	0.4667	0.8667	0.9333
20	0.4000	0.8500	0.8500
30	0.5000	0.7333	0.7333
100	0.5900	0.7700	0.7500

Table 5: Precision at different ranks in a sorted list of hypothesized activity participants.

7. WORLD MAPPING

In the last set of experiments for this paper we explored dependencies between the content of the chat messages and the speaker location in the virtual world. We segmented the locations into half-overlapping squares of size 2×2. For each segment square we aggregated all chat messages from all players that originated from that location. We then constructed a feature vector for each message group using Inquery adjusted $tf \times idf$ score [2]. Finally, we computed pair-wise inter-vector similarities as cosine of the angle between the vectors and clustered the vectors using the Ward algorithm [10].

Each term vector corresponds to a square on the world map. The resulting clusters can be visualized as shaded regions on a grid. Figure 5 shows the grid corresponding to the map of the virtual world with clusters for messages collected from a 3-day period starting at midnight on August 1st, 2003. There were 83,129 messages at 5,474 different locations, and 237 monster killings occurred during that time. We terminated the algorithm when 100 clusters were produced, selected four largest clusters, and merged the rest into the fifth miscellaneous group. The clusters are labeled with numbers from 0 to 4 and each cluster is assigned a unique shade of gray starting with black for the largest cluster ("0") to a very light gray for the miscellaneous one ("4"). The color legend is at the top right corner of the picture. We also show the locations of the towns (squares) and the monster killings (circles) that occurred during that time period. The reader may compare the cluster grid to the actual map of the virtual world on Figure 4[2].

The first thing to notice is that the players' discussions concentrate in the center of the map and the top right corner. From the descriptions on the web site we found out that the town in the center of the map is where a player normally starts in the game. The world's top right corner is where most of the guilds[3] have their towers or headquarters. The coordinates of that area are also mapped to what we believe is an underground realm where the most of the monsters live. There are also conversations in the town areas and few other locations on the map.

Second, we observe a good correlation between the circles (the monster killings) and the cluster labeled "1" – the circles occur near or on top of the squares colored with that

[2]This is the map of the BladeMistress world during the time period of our experimental collection. The present version of the world (January 2006) is noticeably different.

[3]A guild is a persistent group of players with well-defined membership and internal rules.

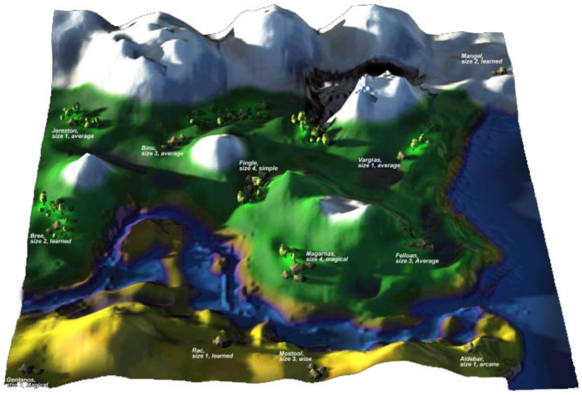

Figure 4: Map of the virtual world.

shade of gray. We analyzed the content of the clusters by looking at the top best terms from the cluster selected using word contribution to the clarity score [5]:

$$Sc(w) = P(w|C_i)\log\frac{P(w|C_i)}{P(w|C)} = \frac{N_{C_i}(w)}{|C_i|}\log\frac{N_{C_i}(w)|C|}{N_C(w)|C_i|}$$

where $P(w|C_i)$ is the probability of the word w occurring in cluster C_i, $P(w|C)$ is the probability of w occurring in the whole collection, and $N_{C_i}(w)$ and $N_C(w)$ represent the overall count of w in cluster C_i and collection C. We processed the resulting list of terms to remove all words shorter than 4 characters, standard stopwords, adverbs and adjectives.

Table 6 shows the top best terms from the three largest clusters "0", "1", and "2". What is interesting to note is that the words like "kill", "dragon", and some monster names (e.g. "dokk") appear in cluster "0", while the cluster "1" is characterized by the words "village", "dungeon", "help". It looks like the players do not use the monster names where they are killing them, which is reasonable – they are in the close proximity to the monster, maybe they can see it on the screen, so it not necessary for them to refer to the monster explicitly.

The third cluster picked up quite a few German words – it is possible that some German-speaking people were playing the game during those 3 days.

8. DISCUSSION AND FUTURE WORK

These are our first experiments with the MMORPG domain. We see several possible improvements for the current study and many research questions (ranging from text processing to other activity detection and player classifications) remain open.

Our analysis of spatial patterns in Section 7 was limited to a 3 days period. The choice of the time interval was motivated mostly by time and resources required to process and analyze the data in time for the submission. We plan to repeat the spatial analysis on the whole experimental data set for the final version of the paper.

Our model of time and space dependencies was quite simple – we segmented the time and the space into blocks with well-defined boundaries and all words collected from the chat messages inside those blocks had the same weight. We plan to investigate more elaborate models of those dependencies. For example, we may consider the words that occur in close

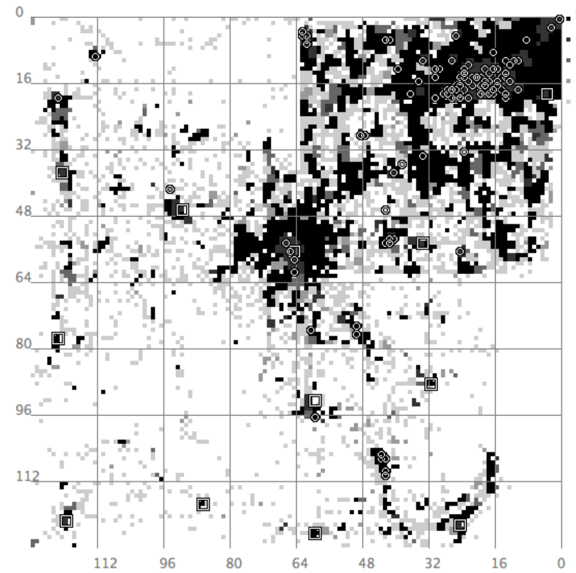

Figure 5: Message content clusters on the world map. Also the towns are shown as squares and the monster killings represented by circles.

0	1	2
kill	village	anubis
dragon	guard	gestern
bone	possy	aber
dokk	suit	meiner
time	help	crossing
tyrant	mice	computers
guardian	skelles	gefunden
maze	teens	entrance
tree	patches	dokk
look	colors	musashi
quest	dungeon	crickets
memory	time	hewre
warriors	levelling	schau
dungeon	verses	einfach
movie	babe	dimms
queen	place	back
personna	corners	poses
boss	crap	stirbt
lord	quest	sarges
might	home	water

Table 6: Top best terms in the 3 largest clusters.

proximity to the activity to be more important than those that occur at some distance. We can use a bell-shaped weighting function on the word probability estimations in the language models.

We observed that the language of chat messages is rather different from the traditional well-formed text we can expect from newspaper articles or web pages. Messages are very reach on typos, acronyms, and domain-specific lexicon. They are informal and ungrammatical. Often important and unusual information is expressed using punctuation characters, e.g., the author's emotion is conveyed with the emoticons. It is clear that traditional text processing techniques such as stemming will not be successful without significant effort on adapting them to this environment. Even the process of word tokenization is an open question.

We only have one type of players' activity recorded in our data set – monster killings. While this activity is important to the game process, we are also interested in analyzing other activities, e.g., quests, item exchange, goods trading, resource farming, tutoring of new players, etc. Such an analysis may require an extensive annotation effort. However, we can automatically detect when several players meet and stay together for a significant period of time. We hope that such gatherings are noticeable events in the players' life and carry important meanings. We may attempt to cluster the conversations that happen during those meetings, e.g., to isolate when people trading items from the cases when one player coaches another.

Another area of analysis that remains unexplored is the classification of players. Suppose you meet an unknown person in the virtual world and start chatting with her. How quickly can you estimate her level of experience? To setup such an experiment we can approximate the skill of each player by analyzing the time she spent on-line and the number of activities in which she has participated. Now given a random sample of text messages from a player can we predict her experience? How large that chat sample has to be?

Bartle [3] does an extensive analysis of player types and concludes that there four major types that should be attracted to a game for it to flourish: achievers, explorers, socializes and killers. He also believes that these players have distinct language patterns. We can construct individual language models for the players, cluster them, and attempt to verify his statement.

9. CONCLUSIONS

In this paper we considered the domain of massive collaborative environments where people meet, discuss various topics and do things together as an exciting area for IR studies. Our focus was on the dependencies that exist between people's activities in those environments and the content of the messages people generate in regard to the activities. One example of such an environment is a massive multiplayer online role-playing game. We showed how such a game can serve as a rich source for experimental data. We have defined three research tasks to analyze some aspects of the activity-content dependencies and demonstrated the success of language modeling and text clustering techniques in solving those tasks. The final contribution of this paper is the outline of potential directions for exploring this domain in the future.

10. 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.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the sponsor.

11. REFERENCES

[1] BladeMistress web site. `http://www.blademistress.com/`. Accessed January 24, 2006.

[2] J. Allan, J. Callan, B. Croft, L. Ballesteros, J. Broglio, J. Xu, and H. Shu. Inquery at TREC-5. In *Fifth Text REtrieval Conference (TREC-5)*, pages 119–132, Gaithersburg, Maryland, USA, 1997.

[3] R. Bartle. *Designing Virtual Worlds*. New Riders Games, 2003.

[4] E. Castronova. Virtual worlds: A first-hand account of market and society on the cyberian frontier. *The Gruter Institute Working Papers on Law, Economics, and Evolutionary Biology*, 2(1), 2001.

[5] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *In the proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299–306, Tampere, Finland, August 2002.

[6] P. Dhingra. MMO markets. `http://www.mmomarkets.com/`. Accessed January 24, 2006.

[7] M. Fleischman and E. Hovy. Taking advantage of the situation: Non-linguistic context for natural language interfaces to interactive virtual environments. In *Proceedings of International Conference on Intelligent User Interfaces (IUI)*, January 2006.

[8] W. L. Johnson, H. Vilhjalmsson, and M. Marsella. Serious games for language learning: How much game, how much AI? In *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 2005.

[9] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan. Language models for financial news recommendation. In *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM)*, pages 389–396, New York, 2000. ACM Press.

[10] A. Leuski. Evaluating document clustering for interactive information retrieval. In H. Paques, L. Liu, and D. Grossman, editors, *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM'01)*, pages 41–48, Atlanta, Georgia, USA, November 2001. ACM Press.

[11] D. Roy and E. Reiter. Connecting language to the world. *Artificial Intelligence*, 167(1-2):1–12, September 2005.

[12] Wikipedia. MMORPG — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=MMORPG&oldid=36508728. Accessed January 24, 2006.

[13] B. S. Woodcock. An analysis of MMOG subscription growth. http://www.mmogchart.com/. Accessed January 24, 2006.

[14] N. Yee. The Daedalus project. http://www.nickyee.com/daedalus. Accessed January 24, 2006.