The BladeMistress Corpus: From Talk to Action in Virtual Worlds

Anton Leuski[†], Carsten Eickhoff[‡], James P. Ganis[§], Victor Lavrenko[§]

† USC Institute for Creative Technologies, 12015 Waterfront Drive, Playa Vista, CA 90094 USA, leuski@ict.usc.edu
‡ Delft University of Technology, Mekelweg 4, 2628 CD Delft, Netherlands, c.eickhoff@tudelft.nl
§ University of Edinburgh, 10 Crichton Street Edinburgh, EH8 9AB, UK, {j.p.ganis,vlavrenk}@sms.ed.ac.uk

Abstract

Virtual Worlds (VW) are online environments where people come together to interact and perform various tasks. The chat transcripts of interactions in VWs pose unique opportunities and challenges for language analysis: Firstly, the language of the transcripts is very brief, informal, and task-oriented. Secondly, in addition to chat, a VW system records users' in-world activities. Such a record could allow us to analyze how the language of interactions is linked to the users actions. For example, we can make the language analysis of the users dialogues more effective by taking into account the context of the corresponding action or we can predict or detect users actions by analyzing the content of conversations. Thirdly, a joined analysis of both the language and the actions would empower us to build effective modes of the users and their behavior. In this paper we present a corpus constructed from logs from an online multiplayer game BladeMistress. We describe the original logs, annotations that we created on the data, and summarize some of the experiments.

Keywords: Virtual Worlds, User modeling, Text classification

1. Introduction and Motivation

Virtual worlds (VWs) are quickly emerging as a new channel for social interaction. They are at once very similar to, and very different from the real world. These worlds are populated by the same people we interact with at work, and offer many of the activities we are used to – shopping, entertainment, socializing. The inhabitants take on the familiar roles of leaders, educators, craftsmen and salesmen. In addition, virtual worlds offer many activities that the participants cannot regularly experience in real-life, such as taking part in a military raid or coordinating the economy of a city-state.

The size of present-day virtual worlds makes it possible to study human behaviors on an unprecedented scale. Far from being a niche phenomenon, the population and monetary power of today's virtual worlds exceeds that of many real-world countries. Experts estimate that the number of VW subscribers approaches 47 million people in USA alone, surpassing the populations of California, Canada and Spain. The yearly revenue exceeded 2 billion dollars in 2010[1], placing the virtual world economy ahead of 40 developing nations.

As participation in these VWs broadens and deepens, many see a practical need to understand the nature of interactions and behaviors in these worlds. VW developers and maintainers are looking for insights on the players' motivation and behavior to create more engaging and attractive VWs. Sociologists and economists study the effects of the VW rules and constraints on the players behavior and they are searching for tools to construct accurate social models. Advertisers are attempting to use in-world advertisement to boost awareness of the real-life products and they require technology to make the ads more focused and effective.

Virtual worlds present a unique environment for studying the relation between human communications and actions in a task-oriented environment. Observations from a virtual world present a nearly-complete picture of behavior of large crowds: we can observe the exact location of every individual, who they are talking to, what they are saying, but also what they are doing at any particular moment. It is this last factor that makes virtual world observations particularly useful: it allows us to explore the connections between words spoken by one individual and actions performed by another.

We believe that a detailed analysis of virtual world transcripts can lead to development of a joint statistical model of language and actions. Such a model could help us to detect, predict, or explain users' actions in the virtual world by analyzing the content of their conversations on a large scale. Such an analysis also opens doors for determining the roles assumed by the users in various activities. We hypothesize that integrating the patterns of a user's roles and her in-world interactions along a significant period of time could give us an insight into the user's relationships to her peers, her experience, and her status; it would help us to build an accurate behavioral profile of the user and assess her personality and culture.

We expect that the joint language-action modeling, perfected in virtual environment, can be applied in real world, – e.g., to a collection of Twitter messages, – to help with consumer market analysis or predicting a poll results. We also see real-life applications anywhere where people closely interact in a task-oriented environment that can range from an online study group for a large internet course to a military chatter on a battlefield.

Developing such a technology requires access to a sufficiently large collection of data from a virtual world for training and testing the models. In this paper we present a new corpus of annotated communications and activities that took place in a virtual world over an extended period of time. Our goal is to describe the initial dataset and the annotations that we created on a portion of the data. Additionally, we briefly summarize some of the experiments we

[1] http://www.gamesindustry.com/about-newzoo/todaysgamers_graphs_MMO

conducted with the corpus to illustrate possible research directions and unique experimental questions that can be examined using this data.

To the best of our knowledge our dataset is unique in that it combines observations of in-game events and activities with a detailed record of communications between the players. Currently, most of the VW studies are conducted using the datasets consisting of elementary actions: monetary transactions, a specific item being added to or removed from the player inventory, a new skill acquired, etc. Some of these datasets are quite large (Ahmad et al., 2009) but the lack of communications between the players means that we can never fully understand why the money changed hands or what prompted a given player to join a particular guild. On the other hand, there exist collections of online chat room transcripts and collections of Twitter messages (Elsner and Charniak, 2008; TREC, 2011). Here we have a language format similar to the one used in VWs, however, those collections do not record any activity information. Without knowing the users' actions, it is difficult to distinguish an ongoing operation from idle chatter. Furthermore, existing textual collections tend to contain *retroactive* messages that report events happening in the real world, or discuss their implications. They rarely contain *proactive* real-time discourse that takes place when people plan, organize and coordinate group activities.

Our collection of activity and chat message logs comes from a Massive Multiplayer Online Game (MMOG) called BladeMistress. We have permission to make the collection available for research purposes. The rest of the paper is organized as follows: we describe the collection and discuss the virtual world of BladeMistress. We then describe two sets of annotations that we created on the data and plan to package with the collection. Finally, we summarize research experiments we conducted using the dataset.

2. BladeMistress

BladeMistress is a small non-profit low-bandwidth fantasy-oriented MMOG. As in much larger virtual worlds, this game has players collecting resources, exploring the world, raiding against dragons and other monsters, practicing magic, trading items and stories. The player-controlled avatars move around in a 3D virtual world which is divided into squares. Our data includes a full year of both chat and game logs from September 2002 to August 2003. To preserve the privacy of the players, the data does not contain any player identification information beyond the avatar names.

After being taken off line in 2005 the game is now run again by a community of fans. This was extremely helpful as it enabled us to actually verify the correctness of our assumptions about the game mechanisms which would not have been possible from the log files alone.

The rather high age of the chat data is not detrimental to the quality of the corpus as the main features of MMORPGs have been largely constant over the past years. It does even have the positive side effect of providing greater completeness than a similar more recent log file could. Today, the use of third party Voice over Internet Protocol (VoIP) applications for VW user communication is very common. This makes textual log files less and less valuable as a growing proportion of the communication is not captured. In the early phase of BladeMistress, however, such technologies were not as readily available. We can therefore expect that the BladeMistress log files have few if any such information gaps.

2.1. Dataset contents

The BladeMistress dataset consists primarily of two log files. The *chat log* presents a complete record of communications between the users of the game. The *game log* contains a history of in-game events. Both logs come in the form of plain text files.

Each line in the *chat log* (see Figure 1) consists of the time at which the chat message was sent, the grid location of the speaker, the speaker name, the transmission mode of the message, and the content of the message. There are 6 transmission modes that determine who is going to see it: a player can broadcast the message to the whole world (*tells everyone*), limit its scope to players in the same square (*says*), to all players in 4 square radius (*shouts*), direct the message to a particular player (*tells you*), to her friends (*tells friends*), or to the members of her guild (*tells the guild*).

The *game log* (see Figure 2) records when players join the game, when the players leave the game, and when a monster is killed, including its name and the names of the players present at that location at the same time.

We pre-processed the logs as follows: We removed non-ASCII characters and empty messages. We normalized the time stamps of the messages and the activities. Specifically, we converted the times into seconds from a reference date and accounted for the time shift due to change from and to the daylight savings time. The latter proved to be a little confusing as chat logs showed that shift, while game log did not have the shift on the time line. The chat log has 5,514,173 messages that take approximately 310MB of disk space. There are 284,728 unique terms in the vocabulary and 19,144 unique player names.

The game log time line resolution is one minute and the log does not contain precise locations for the activities. We extrapolated the activity coordinates by considering the locations of all the messages from the players involved in the activity during that minute and averaging those coordinates. The log lists 447,874 raids (monster kills). Some monsters are stronger than others and require more people getting together to succeed at the task. Such activities might prove to be more interesting for analysis because they potentially require a more elaborate and intense discussion among the players. Figure 3 shows the number of individual raids as a function of the number players involved in the raid.

The chat messages pose some known challenges for language processing: the frequency and extent of spelling errors significantly exceed those observed in other textual sources. We observe a high proportion of ungrammatical sentences which often lack any form of punctuation. The players make extensive use of abbreviations and game-specific jargon (Maness, 2008; Tagliamonte and Denis, 2008).

```
11/02/02, 16:46:21[1], 44, 105[2], RainStorm[3] says[4], u gettin exp?[5]
```

Figure 1: Chat log sample. We show a single chat message: (1) the time at which the chat message was sent, (2) the grid location of the speaker, (3) the speaker name, (4) the transmission mode of the message, and (5) the content of the message.

```
new connection:  johnny123, gg[2] 3/25, 16:00[1]
closing connection:  mark4val, lady[4] 3/25, 16:01[3]
Dragon Queen[6] killed!  3/25, 16:02[5] ----- attending:  Pheremone[7], -----
```

Figure 2: Game log sample. It shows: (1) when a player joins the game and (2) her name; (3) when a player leaves the game and (4) her name; (5) when a monster is killed, (6) its name, and (7) the names of the players present at that location at the same time.

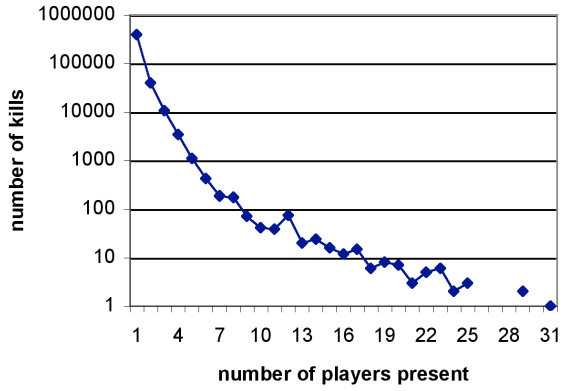

Figure 3: Count of raids vs. number of players involved.

2.2. VW Geography

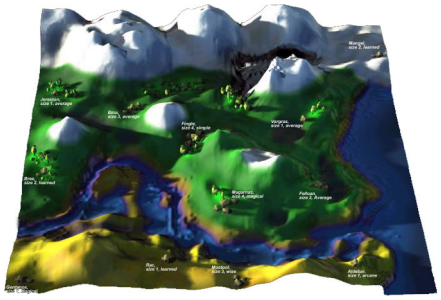

Figure 4: Map of the BladeMistress virtual world.

The BladeMistress world has an elaborate geography that includes mountains and valleys, a river and a desert, towns and guild towers. The geography features give rise to some interesting questions for language analysis, e.g., does the language of the players change as they move from one location to another? Figure 4 shows the map of the world. One unusual feature of the BladeMistress world is the existence of a secondary domain inside the game: the area the game calls *Spirit Realm*. It is presented to the players as an elaborate dungeon that occupies a quarter of the world space. To move between the worlds, players have to get together in a group of at least three avatars and type a special command into the chat window. The secondary domain may pose a challenge for a study of language and activities: its coordinates are mapped to the coordinates of the main world, so some analysis that tracks players over time is required to separate messages originated from the main world from the messages coming from the secondary one.

3. Annotations

We only have one type of players activity recorded in the BladeMistress game logs – monster killings (or raids). While this activity is important to the game process, we are also interested in analyzing other activities, e.g., quests, item exchanges, goods trading, resource farming, tutoring of new players, etc. There are two reasons for that: Firstly, any analysis or experiment with the raid activity should be confirmed on other activity types, so we can make sure our technology works across multiple activities. Secondly, considering a player's involvement in different activities would give us a deeper insight into the player's behavior, motivations, and personality.

Extending our analysis to other activity types requires a significant annotation effort. However, a virtual world is normally designed to support and encourage a fixed number of activities in the world, e.g., it would be difficult to arrange a football match in the World of Warcraft, while it might be easy to buy a sword from one of the players. There are numerous examples of players engaging in those few activities across the virtual world on multiple occasions. These examples should give us an ample training data to develop a classification approach capable of detecting the activity based on the content of the players' chat. We believe that we can annotate a small portion of the BladeMistress logs and use text classification to extend the annotation labels on the rest of the collection.

In this section we describe two sets of annotation that we have done on a portion of the BladeMistress data. The first set of annotations attempted to enumerate in-game activities and informal player roles associated with each activity. The second set of annotations focused on "raids": several players gather together in a coordinated fashion to hunt down a particularly strong monster.

3.1. Activities and Roles

We examined the conversations recorded in the chat log and isolated 7 activity classes that are commonly observed. We also marked down the roles players take in those activities:

Raiding One or more players are trying to defeat a VW monster and gain any treasure it might have guarded. Roles: *leaders*, who coordinate the action, and *adventurers*, who carry the plan out.

Trading Players are buying, selling or exchanging virtual goods such as swords. Roles: *buyers* and *sellers*.

Crafting A player is creating some made-to-order virtual item, e.g., a sword. Relevant interactions involve specifying properties of the item. Roles: *craftsmen* and *customers*.

Social Players are organizing, forming implicit or explicit social group in the VW, e.g, a guild. Roles: *organizers* and *members*.

Teaching A player is helping another player, explaining the game mechanic and providing tips for better in-game performance. Roles: *teachers* and *learners*.

Storytelling A player is describing her past in-world activities to other players. Roles: *narrators* and *listeners*.

Other In-world or real-world chatter that does not fit into any of the categories described previously.

We used rule-based heuristics to pre-segment a portion of the chat log into individual conversations. We then hired Amazon Mechanical Turk annotators to refine the conversational segments and label them with one of the seven activity categories. The annotators also labeled the roles of each player engaged in the conversation. The annotated part of the BladeMistress dataset contains 11,227 chat messages organized in 849 conversations with activity and role labels.
In the following, we will explain the annotation process and the specific challenges imposed by the VW domain step by step. Our original research aim was to predict role labels for participants in VW conversations (Eickhoff and Lavrenko, 2012). This goal however is not directly achievable and will therefore be split into three sub tasks.
By analogy to the real world, virtual world roles depend strongly on the surrounding context. A person might take several very different roles depending on what he or she does and who else is around. In order to take this fact into account, roles are not assigned globally but rather for the context of each conversation. Global trends can then be inferred from the roles which a person regularly takes.
The first step towards role detection is grouping those messages from the chat log which belong to a single coherent conversation. Formally this means creating groups $G_1..G_K$ which consist of coherent messages $G_{i,1}..G_{i,n}$ and which are mutually exclusive

$$\forall m \epsilon\, G_i |\, m \epsilon\, G_j \rightarrow i = j.$$

In virtual and real worlds alike, roles tend to depend on the kind of activity which is carried out. Given the high degree of specificity in which our roles were defined, information about the conversation's topic is essential for role detection. Since we defined groups as sets of messages belonging to a single activity, this means assigning a single class label for each system-generated group H_j.

With the messages grouped into conversations and their activities annotated, the third sub task is to finally assign an activity-dependent role to every participating person in the conversation G_i.

3.1.1. Annotation process

The vast number of messages in the chat log makes an automated pre-grouping essential in order for annotators to handle the tasks in acceptable time. Therefore, a system has been devised which, according to a set of rules, assigns messages to clusters. This rule-based system makes two very basic assumptions to approximate topicality and coherence.

The single channel assumption A group of people talking to each other is unlikely to change the means of communication during the conversation.

For example, two people chatting to each other in a cafe are very unlikely to spontaneously switch to talking via mobile phones or writing letters. This notion has been transferred into the virtual world with its variety of chat channels. The single channel assumption therefore results in the pre-grouping system only taking into account messages from one channel for each message cluster.

The time to live assumption A conversation, independently of the means of communication, is very unlikely to contain major phases of silence without also encountering a change in topicality.

Figuratively speaking, this means that the above two people in the cafe will probably talk very avidly about a certain topic, for example a recent news story. Following intuition, we do not expect them to suddenly fall silent for five minutes before continuing to discuss the same topic. Such a major break is much more likely accompanied by a topic change, as the previous conversation has "run dry". Therefore, the clusters created by the pre-grouping system will be closed if they haven't received additional related messages for a threshold period of time.
Following our rules, the automatic pre-grouping system iterates over the chat log file and extracts each entry according to the log's structural properties. The first condition the system checks for is the means of communication. For some universal broadcasting channels (like *tells everyone*) this is already sufficient. For others which are position-dependent due to broadcast ranges (e.g. *says* or *shouts*) we have to consider further factors to determine the channel precisely. Once this has been achieved, the system checks for the existence of an active cluster belonging to this particular channel. Open in this case means a cluster which has not yet surpassed the threshold time. If no fitting active cluster is found, a new one is opened and the message is associated with this cluster.
These assumptions and the rules derived from them are necessarily generalized in order to remain practical. Another problem is the flat format in which the chat data is represented in the log files. We cannot access any virtual world information beyond the chat log entries. Therefore, the audiences of chat channels available to an exclusive group of people (like *tells guild* or *tells friends*) are unfortunately

not resolvable from our perspective. Since we do not have server-side information about guild affiliations and friendship relations between players, all we could do would be to guess. To enable pre-grouping to be achieved in reasonable time, we introduce another assumption.

The global guild/circle of friends assumption Every conversation within the *tells guild* or *tells friends* channel is treated as if it were broadcast to the entire virtual world population.

In other words, this means that every avatar is assumed to be in the same guild and to be friends with everyone in the virtual world. In difference to our first two assumptions, which dealt with the nature of human communication this third one is an exclusively technical assumption made to ensure efficient processing during pre-grouping. It is obvious that this assumption is not correct. It is, however, also relatively harmless in our case. It has been introduced to offer an easy way to solve guild or friendship affiliations. For practical application this makes only little difference to knowing the actual social topology of the virtual world because of the small scale of BladeMistress. Where popular recent games with millions of users would probably not be treatable in such a way, we seldom encounter overlapping *tells guild* or *tells friends* conversations in the BladeMistress data.

The rules were intentionally kept simple as the pre-grouping step only aims to quickly create likely clusters. Human annotators will review and refine the grouping and will thus be able to make any corrections they see fit to create message clusters representing natural conversations.

To enhance the actual annotation process, an online system was provided through which the annotators could conveniently check, and, if necessary, alter the automatically-generated message groups. For the labeling tasks a set of tools was offered to help them with their decisions.

The annotators were hired through Amazon's Mechanical Turk service[2] and come from a variety of different backgrounds. Some of them had previous MMOG experience and seemed to cope better with the data's specifics than those without prior domain knowledge. In order to judge the complexity of the three annotation tasks and the quality of their results, a share of the data has been annotated redundantly. For a proportion of $\sim 20\%$ of the conversations there is one additional judgement, and for a number of at least 50 of these there are also third annotations. Table 1 shows the exact absolute and relative amounts of redundantly labelled data for the various tasks. Fleiss' κ agreement scores (Fleiss, 1971) across 3 judgements were 0.82, 0.55 and 0.49 for message grouping, activity and role labeling, respectively. Determining conversation boundaries appears to be a task that can be reliably solved by humans, while activity and role labels are more disputed.

3.2. Raids

For our second set of annotations we focused exclusively on the Raiding activity. Specifically, we looked at the cases where the players get together to hunt down a particularly powerful monster. We examined the conversations related to these hunts (or raids) and defined 8 stages a player may go through during a raid. Below we provide a brief description for each stage and indicate what percentage of messages were annotated with that stage label.

Table 1: Share of redundantly annotated data

Task	2nd (abs)	2nd (rel)	3rd (abs)	3rd (rel)
Grouping	190	22.38%	51	6.01%
Activity	179	21.08%	75	8.83%
Role	221	26.03%	62	7.3%

- 6% **Search**: the player is trying to find the location of a monster or an ongoing raid, possibly asking for directions.
- 14% **Invite**: the player has found the monster and is requesting reinforcements.
- 9% **Plan**: the player is preparing to attack the monster, negotiating the plan of attack with other players
- 38% **Fight**: the player is actively engaged in combat with the monster.
- 6% **Respawn**: the player's avatar is killed by the monster; he must return to the raid from a nearby "resurrection point".
- 2% **Win**: the monster is defeated, the player celebrates.
- 6% **Loot**: dividing the bounty left by the monster.
- 19% **Idle**: player not involved in the raid.

All annotations were produced according to the following procedure. First, we manually selected 16 raids from the *game log*. We tried to get a sampling of large and small raids. For each raid, we examined all messages in the *chat log* that occurred within two hours of monster's death, reasoning that a BladeMistress raid is unlikely to take over 4 hours. We discarded messages from players who were not present at the kill, assuming these players did not participate in the raid. Each remaining message was manually assigned one of the 8 state labels. Annotations were carried out by a single individual with extensive experience in MMORPGs. No adjudication was performed. The resulting dataset includes annotations for 1,608 messages from 122 different players participating in 16 raids. The smallest raid involved 5 players and lasted only 15 minutes. The largest raid involved 20 players and lasted almost an hour. For more detail see (Ganis, 2011).

4. Experiments

In this section we will briefly summarize the findings from a number of empirical studies conducted on the BladeMistress dataset. In section 4.1. we argue that chat language patterns alone can be analyzed to pinpoint an ongoing raid and distinguish players participating in a raid from bystanders. Section 4.2. shows how this approach can be extended to detect other activities and assign roles to the participants in the raid. Finally, section 4.3. presents a statistical model of how players progress through the various stages of a raid. The studies were originally published in (Leuski and Lavrenko, 2006),(Eickhoff and Lavrenko, 2012) and (Ganis, 2011).

[2] *www.mturk.com*

4.1. Event Detection

Using the initial logs and without any additional annotation we studied the following three questions (Leuski and Lavrenko, 2006):

Activity Detection Is it possible to find out something about the users' actions in the VW by analyzing the content of chat messages? Suppose we cannot observe an activity in the VW, but can monitor the stream of all the chat messages. Can we predict the time and the place of the activity by looking just at the content of text messages? We showed that we can train a Naive Bayes-like text classifier that would detect the location and time of raids effectively. This result also supports our assumption that we can annotate a portion of the collection with activity labels and extend the annotation over the rest of the data using text classification.

Player Forensics Is it possible to find out something about the users themselves in the VW by analyzing the content of chat messages? Suppose we know the time when a specific activity happened, but do not know the location or the participants. Can we find the likely participants by analyzing the chat messages from all the players around that time moment? We showed that using the same text classification approach we can accurately pinpoint some of the participants in a raid by analyzing the message content, however detecting all of the participants proved to be a difficult task.

World Mapping How do VW features affect the users' conversations? For example, is there a correlation between what players saying and their location in the world? Is the message content different for the area where monsters live from the conversations occurring in the other parts of the world? We clustered the chat messages originating from each VW square by their content and overlaid the clusters on the VW map. We observed a clear correlation between the content of the chat messages and their origin in the VW. For example, the players tend to talk about the fighting in the areas inhabited by monsters and discuss "token" and "quests" on the vast planes between the towns.

4.2. Role Detection

Based on the activity and role labels introduced in Section 3.1., we devised an automatic means of solving the role detection task by means of a vector space model (VSM) approach. As indicated previously, to reduce complexity of decision boundaries, we split the task into 3 steps:

Message grouping
As a starting point, the chat log's submitted messages have to be grouped into coherent conversations. We employ a rule-based system that takes into account the senders' positions, their means of communication, as well as the time elapsed between messages m_i and m_{i+1}. We considered a combination of two types of rules: (1) Connectivity rules determine whether several users are able to communicate. Some of the VW's chat channels are position-dependent (e.g., *says* or *shouts*), others are directly sent to one or more players (e.g., *tells you* or *tells guild*), and some are indefinite broadcasts to the entire VW. Thus, we consider only messages from authors who were able to read a given message group's previous stream of communication. (2) Freshness rules are concerned with the delay between messages. If two messages are issued with a significant communication gap δ_t, of more than 140 seconds between them, maximum likelihood estimates suggest that they do not belong to the same conversation. Once connectivity and freshness are ensured, we expand the message group by m_i.

Activity labeling
For determining a conversation's dominant activity, we employ a vector space model approach using word-internal character n-grams and their frequencies as vector components. Each activity type is represented by a centroid. The closest centroid, according to Euclidean distance, defines the activity type that we assign to the conversation. The choice of n-gram length n is an important step towards obtaining good results. Best results could be achieved using $n = 4$.

Role Labeling
Given a conversation's estimated activity type, we finally employ a binary vector space model as a message group representation. It contains one vector component for each unique term in the group of messages, taking component values $c_i \in 0, 1$. Classification is based on a nearest neighbor method relying on Jaccard coefficient distance metrics.

Non-verbal features
The models introduced above exclusively used words or character n-grams as features. In order to not only consider *what* people say but also *how* (where, when, etc.) they say it, the previous models can be expanded by a range of non-verbal features generated from the message groups. These features are:

- Mean message size in words/ size variance
- Total number of messages/ words
- Most frequently used communication channel's ID
- Mean x and y coordinates of senders
- Mean/ variance time interval between messages
- Total number of involved players
- Travelled geographical distance during conversation

These features bear information about shallow message properties, means of communication and movement patterns. Each of which might contain information about the role the sender takes. They are injected as additional dimensions in the vector space. In order to do this, we have to take into account the very different scales of the values. Previously, we only had to process term frequencies which are on the same scale. Now, however, we introduce new dimensions that might exceed those counts by far and that would therefore give a too high overall importance to the feature. We prevent this by normalizing the features' scales and mapping them onto word frequency scale in the following way:

$$c_i = \frac{x_i - min_i}{max_i - min_i} max_{freq} \lambda_i$$

Where

Table 2: Detection results on unseen test data

Method	Activity labels	Role Labels
SVM Baseline	31.51%	33.04%
VSM approach	36.47%	36.08%
Human performance	47.29%	44.96%

c_i is the rescaled value of the i-th feature,
x_i is the feature's original value,
min_i is the smallest observed value for feature i,
max_i is the largest observed value for feature i,
max_{freq} is the largest observed term frequency.

The minima and maxima needed for this mapping were calculated over the whole corpus of messages. To tune the features' individual contributions to the overall classification, the model is finally expanded by a vector Λ of weights λ_i for each feature component c_i.

Performance

Table 2 details the final performance on the previously-unseen held-out portion of the data set. As a point of comparison, we included the performance of a standard SVM-based text classification approach as well as human annotator performance as determined on the redundantly-annotated share of data. The differences between both systems and human performance were found to be statistically significant at $\alpha < 0.05$-level.

4.3. Raid Modeling

We used the annotations described in section 3.2. to develop a statistical model of how players behave during a raid. We hypothesized that during different stages of a raid, players will talk about different issues, move with different patterns and vary the stylistic aspects of their messages. For example we expect that during the *planning* phase (stages 1,2,3) the players will be more verbose, ask questions, discuss tactics and will cover great distances seeking out a monster to attack. In the *fighting* phase (stages 4,5) we expect the players will remain stationary and mostly silent as they focus on the monster. We also believe that time must be an important component of the model: there is a natural sequence for progressing through the stages, e.g. the player should not start plundering the loot before actually fighting the monster.

Figure 5 illustrates our approach with a mock example of three players. Player 1 encounters a monster (a *minotaur*) and calls for help. Player 3 hears the call but declines to get involved. Player 2 answers the call and asks for directions. The two players agree a strategy and engage the minotaur, with player 1 repeatedly dying and re-spawning during the fight. After defeating the monster, player 2 collects the bounty carried by the minotaur, while player 1 immediately takes off.

The nodes and arrows in Figure 5 represent the sequence of states each player goes through. States connected with dashed lines indicate that two players are involved in a common activity (a raid). Note that players don't have to be in

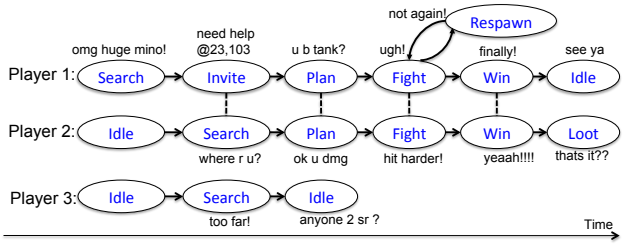

Figure 5: Example of stages in a raid with three players. The horizontal axis represents time. We show the states each player goes through and the messages they exchange. Dashed lines indicate *entanglement*: two players participating in the same raid.

the same state at the same time. Messages above and below the state sequence show the dialogue between the players.

The states shown in Figure 5 are never observed in the data naturally – they represent the hypothesized *latent* state of each player at the moment when he is typing a message (an observation). This setup makes it natural to use Hidden Markov Models (HMMs) for capturing the structure of the data (Rabiner, 1989). We chose to model each player with a separate Markov chain – this allows each player in the raid to have his own state (e.g. one is re-spawning while the other is fighting). We believe the above setup is more flexible than trying to model the entire raid with a single HMM, but it does raise a challenging issue of *entanglements* (dependencies) between the chains of individual players participating in the same raid. We addressed entanglement by augmenting each player's observation with *neighborhood* features as described below.

Our HMMs had 8 states (one for each stage of a raid). The observation vector consisted of four sets of features:

1. **Lexical features** convey the meaning of a message typed by the user. They include the set of words in the message, augmented with word collocations (phrases) and character bi-grams (to alleviate misspellings).

2. **Stylistic features** reflect the player's sentiment. We included the amount of capitalization (indicates emotion), numeric patterns (hit-points, coordinates), and punctuation patterns (exclamations, question marks, smiley faces).

3. **Motion features** reflect how the player was moving at the time of writing. Features included approximate location, direction of movement, velocity, acceleration, change in direction and relative verbosity.

4. **Neighborhood features** reflect our guess of what the nearby players are doing at this moment (if they're fighting, we're likely to be fighting as well). We include the label of the *most-probable* state of each neighbor in the observation (estimated using the Viterbi (Viterbi, 1967) algorithm).

We initialized the transition and emission probabilities of the HMM using a small amount of labeled training data. Then we used the EM algorithm (Dempster et al., 1977)

to iteratively re-estimate the model from a large amount of un-labeled data. After training, we extracted most-probable state for each message and compared it to the human annotation described in section 3.2.. We conducted a number of experiments with the above setup and made the following observations:

1. The final HMM correctly identified the state 58% of the time (macro-average). This is significantly more accurate than the majority baseline (14%) and the simple language model (36%), similar to the one used in (Leuski and Lavrenko, 2006). The improvement in accuracy was due to three factors: temporal component of the HMM, use of EM for unsupervised re-estimation, and the use of non-verbal features. Each of the three factors brought statistically significant improvements in accuracy.

2. The *neighborhood* features were particularly promising, but the way we used them caused an instability in the EM algorithm. We believe a better model of entanglement will be a very fruitful direction for future research.

3. The HMM is relatively insensitive to the amount of training data used to initialize the parameters, but it is very sensitive to state priors. Dominant states (e.g. *fighting*) have a tendency to absorb all of the probability mass.

4. The HMM is not capable of identifying the *idle* state. In other words, it cannot be used for separating raid messages from the background chatter. For this purpose the solution outlined in section 4.1. offers a better alternative.

5. Conclusions

In this paper we presented a corpus of annotated communications and activities that took place in online world of BladeMistress. We described the virtual world, the language and activity data recorded by the game server. We described two sets of annotations we constructed on a portion of the BladeMistress logs and plan to package with the collection. The first set of annotations focuses on identifying in-world activities and roles the users assume during those activities. The second set of annotations examines a single activity in more details.

We also described a number of experiments performed on the collection. We used a statistical text classification approach to show that we can detect and localize a raid and the players participating in the raid using only the chat message content. We studied dependencies between the chat topic and the players' location in the world (Leuski and Lavrenko, 2006). We extended the text classification approach to detect and mark other activities (see Section 4.2.) and player roles. Finally, we developed an HMM-based approach that learns to detect and annotate a player's progression through various stages of a raid (see Section 4.3.) using both verbal and non-verbal features.

Acknowledgments

We are extremely grateful to Aggressive Game Designs and Michael Steele for providing us the logs from BladeMistress.

6. References

Muhammad A. Ahmad, Brian Keegan, Jaideep Srivastava, Dmitri Williams, and Noshir Contractor. 2009. Mining for gold farmers: Automatic detection of deviant players in mmogs. In *Proceedings of International Conference on Computational Science and Engineering*, pages 340–345.

Arthur Dempster, Nan Laird, and Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Carsten Eickhoff and Victor P. Lavrenko. 2012. Towards Role Detection in Virtual Worlds. *ACM Computers in Entertainment, To Appear*.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 834–842. The Association for Computer Linguistics.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

James P Ganis. 2011. Modeling raids in multiplayer games. Honours thesis, The University of Edinburgh, Edinburgh.

Anton Leuski and Victor Lavrenko. 2006. Tracking dragon-hunters with language models. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 698–707, New York, NY, USA. ACM Press.

Jack M. Maness. 2008. A Linguistic Analysis of Chat Reference Conversations with 18–24 Year-Old College Students. *The Journal of Academic Librarianship*, 34(1):31–38.

Lawrence R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Sali A. Tagliamonte and Derek Denis. 2008. Linguistic ruin? lol! Instant messaging and teen language. *American Speech*, 83(1):3.

TREC. 2011. TREC microblog track. `http://sites.google.com/site/trecmicroblogtrack/`. Last checked on Sep 29, 2011.

Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, Apr.