

KeyGraph を用いたプレイヤー群特徴抽出のための アクションシンボル集約

Ruck THAWONMAS

理工学研究科

KeyGraph とはテキストデータ間の単語関係を発見する可視化ツールを指す。本稿では多人数大規模オンラインゲーム (MMOG) におけるプレイヤーの特徴発見のための KeyGraph の新しい応用について論じる。我々は高い可視化の能力を得るために前処理の手法を提案する。この前処理法ではプレイヤーのアクションシンボル列をより意味のある形に集約する。提案手法を適用した KeyGraph と適用しなかった KeyGraph の MMOG プレイヤータイプごとの識別力を評価する被験者実験を実施し、手法の有効性を確かめる。

Development of game engine and contents mining project for establishing "Kyoto Virtual Time Space" **Aggregation of Action Symbol Sub-sequences for Discovery of Online-Game Player Characteristics Using KeyGraph**

Ruck THAWONMAS

Graduate School of Science and Engineering

Keygraph is a visualization tool for discovery of relations among text-based data. This paper discusses a new application of KeyGraph for discovery of player characteristics in Massively Multiplayer Online Games (MMOGs). To achieve high visualization ability for this application, we propose a preprocessing method that aggregates action symbol sub-sequences of players into more informative forms. To verify whether this aim is achieved, we conduct an experiment where human subjects are asked to classify types of players in a simulated MMOG with KeyGraphs using and not using the proposed preprocessing method. Experimental results confirm the effectiveness of the proposed method.

1 まえがき

1997年に誕生したEA社の「ウルティマオンライン(UO)」の成功に始まってMMOGの市場は急速な広がりを見せている。米Zonaが市場調査会社Executive Summary Consultingと共同でMMOG市場に関して調査したレポート[1]では、2006年までにMMOGの市場規模が27億ドルに達するとしている。拡大するMMOG市場もいずれは飽和し、新たなプレイヤー取得よりも現在プレイしているユーザーの囲い込みに重きが置かれるようになるだろう。またMMOGのビジネスモデルの特性上[2]、既存のユーザをできる限りそのゲームから離れないようにしなければならない。プレイヤーの動向や傾向など有益な情報を得ることができたら、新たなコンテンツのヒントになったり、次の経営判断の支援ができる。このためのツールのひとつとして、KeyGraph[3]を用いて抽出した特長を可視化し、人の目で判断しやすくわかりやすい抽出結果を提供することを考えた。KeyGraphは単語のシーケンスである文を対象としている。しかし、MMOGのログデータはただの文字列である。これを意味のある部分ごとに単語のシーケンスに変換しないと、KeyGraphの能力を充分に引き出すことができない。本研究ではこのために、有効なMMOGログデータの変換とその結果の検証を行っている。

2 KeyGraph

KeyGraphとは(特に何かを主張するような論文などの)文章は主張とその根拠(基礎概念)を表すキーワードがあるという前提に、文章中のキーワードを抽出する手法である。性能については、KeyGraphのアルゴリズムと応用例を示した論文[3]を参照していただきたい。KeyGraphでは文章を建物に例えている。文章の基礎概念を土台、主張点を屋根、内容の主な展開を屋根を支える柱と考える。

2.1 アルゴリズム

KeyGraphのアルゴリズムを順を追って説明する。

2.1.1 STEP0:ノイズ除去

最初に原文からノイズになる部分を除去しておく。

2.1.2 STEP1:土台の形成

土台はノードとリンクから成る。まず単語を出現頻度順にソートする。このうち適当な閾値以上の単語をノードとする。次に共起の高いノード間をリンクとして結ぶ。ある単語の対 (w_i, w_j) の共起 $co()$ は以下の式で表される。ただし x_s は文 s における出現回数とする。

$$co(w_i w_j) = \frac{\cap(w_i, w_j)}{\cup(w_i, w_j)} \quad (1)$$

さらに、ノード間の経路が複数あるものの集合を土台と呼び、経路が一つもしくは一つもないノードは別の土台に属するものとする。

2.1.3 STEP2:屋根の抽出

ある土台を考慮しているときに用いられる可能性の高い単語の集合を屋根と考える。その語を w 土台を g とすると、その確率 $key()$ は以下の式で表される。

$$key(w) = \bigcup_{g \in D} P(w|g) \quad (2)$$

この確率 key のうち閾値より高いものの単語 w を抽出し、この集合を屋根とする。

2.1.4 STEP3:柱の定義とキーワードの抽出

すべての屋根に含まれる単語について、すべての土台に含まれる語との柱の強さの和を求める。すべての土台に含まれる語についても、すべての屋根に含まれる単語との柱の強さの和を求める。ある土台に含まれる語とある屋根に含まれる語の、柱の定義を以下の式で与える。

$$c(w_i w_j) = \sum_{s \in D} |w_i|_s |w_j|_s \quad (3)$$

これらの値のうち、閾値より上になる単語をキーワードとする。

2.2 グラフの見方

本研究では構造計画研究所[4]が提供するKeyGraphツールを使用する。図1に本論文をKeyGraphにか

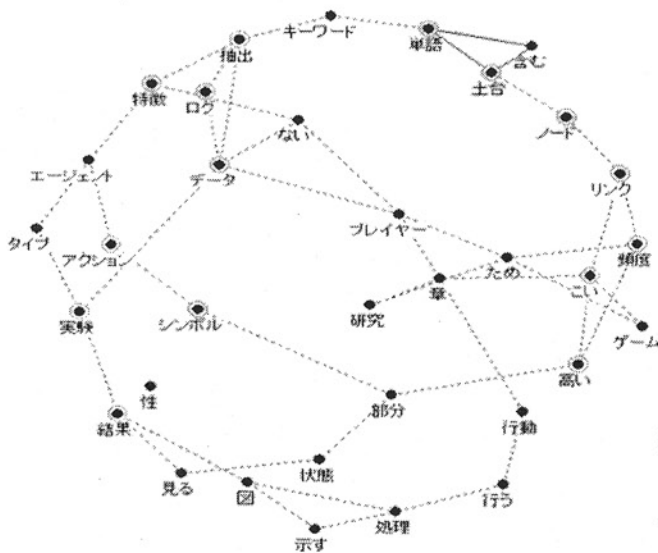


図 1: KeyGraph による出力結果：本論文

- エージェントは Killer, InexperiencedMarkovKiller, ExperiencedMarkovKiller の 3 種
- エージェントはそれぞれのタイプごとに 50 体
- モンスターは 50 体
- アイテムはそれぞれの種類ごとに 50 個
- 生成されるワールドは 16
- 300 タイムサイクル実行

3.2 エージェントのアクション

エージェントは異なる目的をもって行動するが、実行可能なアクションの種類は共通である。表 2 にその一覧と説明を示す。今回の実験においてはシンボル化の際にアクションの引数を考慮しない。PCA は Player Character Agent の略である。

表 2: Zereal におけるアクション一覧

アクション	シンボル	内容
walk	w	移動する
attack	a	PCA もしくは NPC を攻撃する
talk	t	PCA に話しかける
pick up food	f	food を拾う
pick up potion	p	potion を拾う
pick up key	k	key を拾う
leave the world	l	door を通ってワールドを離れる
enter the world	e	door を通ってワールドに入る
removed	r	HP が 0 になって取り除かれる

た結果を示す。

KeyGraph の出力結果であるグラフを見る上で注意しなければならないことがある。

1. グラフの形状は関係がない。
2. リンクの繋がり方が重要。
3. リンクされているノードは共起していることを示し、関係が強い。

また、グラフにおけるそれぞれの記号の意味は表 1 の通りである。

表 1: KeyGraph の記号

記号	説明
丸 (小)	ノード
囲い丸	抽出されたキーワード
実線	枝 (土台に属するノードを結ぶリンク)
点線	柱 (土台と屋根を結ぶリンク)

3 Zereal

3.1 実験環境

MMOG シミュレータ Zereal の環境は以下の通りで、これを各エージェントごとに 5 セット用意している。つまり今回の実験では合計で 800 体分のエージェントのログをまとめて処理している。

3.3 自律エージェントの特徴

Zereal におけるワールド空間には、異なる実装によって生み出された 3 種類のタイプの自律エージェントが存在し、各々異なる目的を持って行動する。これは、MMOG のワールド空間には様々なプレイヤーが存在しているという事をシミュレートしている。その 3 種類のタイプの自律エージェントは、以下のように定義される。Markov Killer とは定義されたマルコフモデルに従って行動するエージェントを意味する。

- Killer : 攻撃を非常に好む。PCA にも monster にも攻撃する。チャットは一切行わない。他のキャラクタを見つけると、攻撃するエージェント。

- Inexperienced Markov Killer : どの行動かをランダムに決め、その行動が実施不可能な場合、さらにランダムに選択を試行する。初心者を模したエージェント。
- Experienced Markov Killer : 近くに monster がいたら攻撃することを選ぶ傾向が強い。また、一度実行したアクションをもう一度実行しようとする傾向が強い。戦闘をやや好むが、比較的同じアクションを続けて行うことが多くなる。自己防衛し、必要なアイテムを回収する上級者を模したエージェント。

4 本手法を用いない KeyGraph

今回提案した手法を用いずに KeyGraph にかけた場合の結果 (図 2 の左から順に Killer, IMK, EMK) は総じて次のようになった。"r" は "a" と "w" にリンクされ、"e", "p", "k" がそれぞれリンクされ、またそれ以外の部分が土台として連結される。出現頻度が多いの "r" 以外のシンボルがお互いにリンクしているのはいいが、お互いのリンクからこれらのシンボルの関係を見ることができない。こういった結果は 3 種のエージェント全てに見られ、エージェントタイプによるこれらの差異はほとんどみられない。

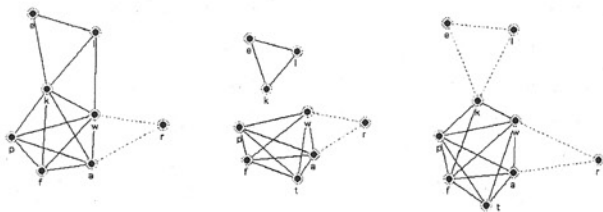


図 2: KeyGraph による出力結果 : 本手法適用なし

5 提案手法

KeyGraph に対して生の MMOG ログデータを投入すると、第 4 章で説明したような結果になってしまった。MMOG ログの形式が、実際の文章のように意味のある単語が並んでいるデータではないからだ。アクションシンボルとは、MMOG ログデータにおけるプレイヤーの単一行動をシンボルに置き換えたものである。本論文で提案する手法では、ログデータにおけるアクションシンボルの高連鎖部分が MMOG におけるプレイヤーの行動にある一定の意味のある行動連鎖であると仮定し、このシンボルを連結・集約する。

5.1 アルゴリズム

CAS のアルゴリズムは大きく分けると、1-2 に相当する連鎖のカウント部分と、3-4 に相当する連結したシンボルをシンボル列として置き換える部分の 2 つになる。後はそれを繰り返し、適切な形式で出力する。

Step 1 各文のシンボルを Bigram にかけて、Bigram によって得られたシーケンス? の出現回数をかぞえる。

Step 2 前記のシーケンス? のうち、全ての文において頻度の特に高いものを連鎖性が高いシーケンス? として登録する。

Step 3

文ごとに前記のシーケンス? を探し、つなげて 1 つの連鎖シンボルとする。この連鎖シンボルは以後の処理で他のシンボルと同等の扱いをする。

Step 4 前記で置き換えられた状態のログデータに対して、Step 2 で登録されるべき連鎖性の高いシーケンス? が見つからなくなるまで 1-3 の処理を行う。

Step 5 連鎖シンボル内で 2 つ以上同じシンボルが連続する部分は集約して大文字を 1 字だけにする。

Step 6 連鎖シンボル/シンボルと連鎖シンボル/シンボルの間をスペースで区切って出力する。

例えば次のような対象のログデータがあったとする。ここでは見やすくするためにシンボルの間に大括弧を入れて表記する。

[w] [w] [w] [w] [a] [w] [w] [p] [w] [w] [a]

最も頻度の高いシーケンス? を探すと (Step 1), は [w w] であるので (Step 2), これを次のように連結する (Step 3).

[ww] [ww] [a] [ww] [p] [ww] [a]

Step 4 に従って繰り返すと, [ww][a] の連鎖性が高いので連結することになる。

[ww] [wwa] [ww] [p] [wwa]

この時点で Step 4 にある高い連鎖性がないものとする。次の処理に移る。[ww] を集約して [W] にする (Step 5)。

[W] [Wa] [W] [p] [Wa]

スペースで区切った状態で出力する (Step 6)。

W Wa W p Wa

6 実験結果・考察

次小節から今回提案した手法を用いて最適化したログデータを KeyGraph にかけた結果と考察を示す。なお、本実験では「Bigram で取り出したシーケンス? の出現回数の平均の十倍」を連鎖性の高いシーケンス? としている。リンクされているシンボルは同時期に起こったアクションであり、ある一連の行為に属するか、その行為の直後に行われたものである。MMOG におけるログデータでは移動する行動が極端に多いので、今回 KeyGraph にかけた結果でも "w" を含むノードが多かったり、"w" に多くリンクがされていたりする。次小節からの全ての結果で、walk や key が leave や enter にリンクされているのは、鍵を拾ってその鍵でドアを開けて出たり (leave) 入ったり (enter) しているからである。また、どのグラフもやはり極めて頻度の高い walk が中心にあるが、それ以外の部分で大きな差異がみられた。

6.1 Killer

グラフから得られるこのエージェントの特徴は次のような事である。(図 3)

1. 攻撃を非常に好む。
2. 戦闘の前後で回復アイテムもちゃんと回収している。
3. 戦闘の状態と回復アイテムを回収に行ってる状態と別の世界に移動している状態がある。
4. チャットは一切しない。

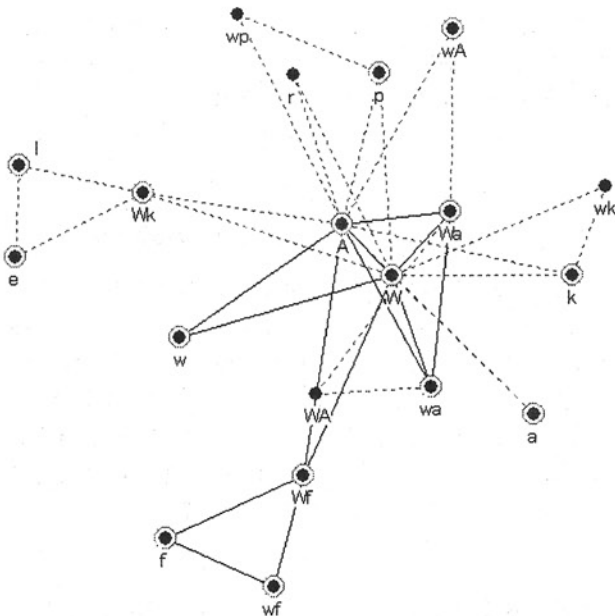


図 3: KeyGraph による出力結果 : CAS あり killer

まず最も注目すべきは土台 (実線で結ばれたノード群) だろう。attack が綺麗に "A" にリンクされているのがわかる。よって 1 は誰にでも容易にわかるであろう。

2 については、図 3 で下の部分の事を示している。ただむやみに攻撃しつづけるだけでなく、pick up potion や pick up food の行動をとっている。

そしてその部分は 1 の attack がリンクされた部分とは切り離された状態で存在している。そのため 3 のようなことが言える。key や leave と enter との繋がりの部分は、低頻度にもかかわらずキーワードとされ、確かに特徴と言える。しかし、他のエージェントも鍵を拾ったら使うように設計されているので、これは別段めずらしいことではない。

4 については KeyGraph を使わなくてもわかることである。talk が一切見られない。

6.2 Inexperienced Markov Killer

グラフから得られるこのエージェントの特徴は次のような事である。(図 4)

1. チャットを非常に好む。
2. チャットをしている状態と攻撃している状態と別の世界に移動している状態がある。
3. それ以外はバラバラ。

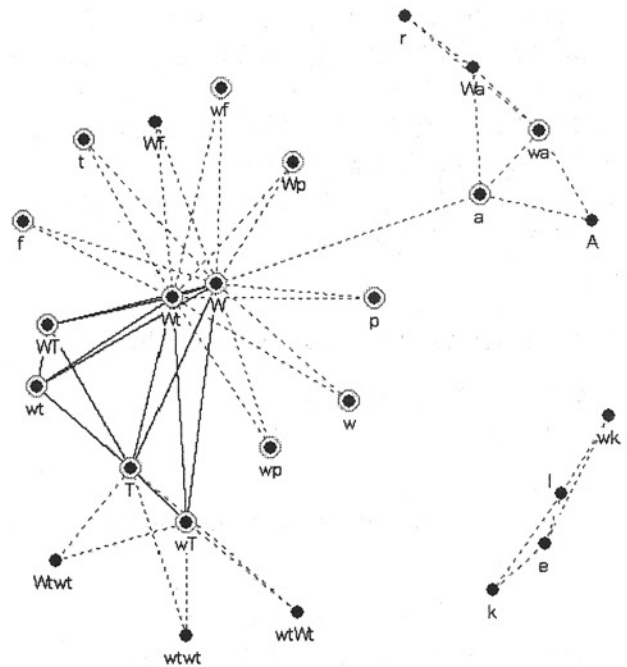


図 4: KeyGraph による出力結果 : CAS あり IMK

"T"や"t"を含むノードが多いので目立つ。土台に属しているのもそのようであるし、図4の下側の部分もそのようである。一部は少しはみ出ているが、これらはおおよそ環状になっている。つまり局所的にある程度連続してtalkしている行為(つまりチャットである)を示している。チャットしている状態は前述の通り。図4の右上側部分でattackが独立し、右下側には隔離されたkey, leave, enterが伺える。またそれ以外のノードはシンボルの種類もバラバラなものももっとも頻度の高い"W"と"wt"にリンクしているの、ここはランダムな行動であることがわかる。

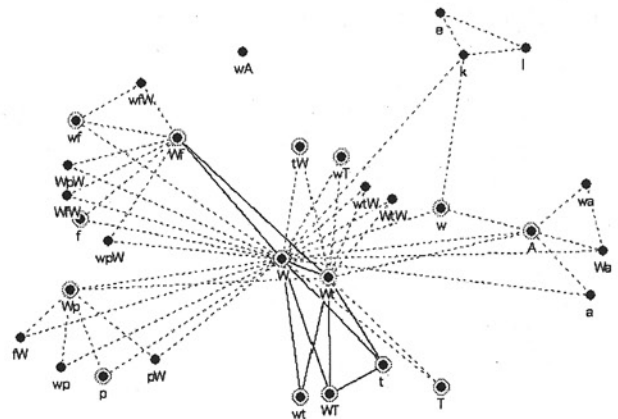


図5: KeyGraphによる出力結果：CASありEMK

Zerealのシステム上、条件がゆるいwalkやtalkが選ばれやすい。例えば、他のキャラクタがいなければattackはできないし、他のPCAが視界にいないとtalkはできない。マルコフモデルによって設計されたIMKの理念は「明確な目標を持たないで、ランダムな行動を行う」ことである。さらに「決定した行動が実行不可能な場合、再度ランダムに行動選ぶ」ことを考慮すると、上の結果はIMKのランダムな行動とジリアルな特性上の行動を抽出したことがわかる。

attackが独立しているのは、このアクションが「隣にキャラクタがいるときだけ」しか実施できないという制約が要因である。IMKはランダムに次の行動を決めるので、隣にキャラクタがいようとまいと次の行動をattack決める確率は変わらない。しかし、この制約のため次の行動をattackに決めても実際に実施することが出来ない。つまりattackのように実施するための条件が厳しいアクションについては、エージェントの思考基準(IMKの場合ランダム)ではなく、実際に実行できる条件(システム上の条件)が基準となってKeyGraphが特徴抽出しているのである。leave, enterが独立しているのもこのためである。

アイテム回収状態について、データセット1-5(図5上)では"Wp"まわりと"Wp"まわりの2箇所について集まっているが、データセット6-10(図5下)では"pW"まわりにしか集まっていない。これは、pick up potionとpick up foodが、アクションとして大きな差がなく、わずかな発生頻度の差がこれらのノードとリンクの関係を作った結果である。

6.4 アンケートによる分類検証

本研究は特徴抽出を目的としているが、その抽出された特徴が果たして人の目で正しく分類されるものかアンケートによって検証した。被験者には前提知識を与えるために本論文の2.2, 3.2, 3.3を見せ、順番を入れ替えた図2と図3, 図4, 図5を提示した。これらがそれぞれのタイプに属するものかを被験者の主観で選択してもらった結果が表3, 表4である。

6.3 Experienced Markov Killer

グラフから得られるこのエージェントの特徴は次のような事である。(図5)

1. チャットを好む。
2. チャット, 戦闘, 回復アイテム回収, 別世界への移動の状態がある。

talkは単純にその回数が多く、土台になっている。

Zerealにおける全ての状態があるのは、図5においてそれぞれのリンクが"Wp", "Wp", "A"に集まっていることからわかる。keyとleaveとenterの関係は他の同じである。

6.4.1 本手法を用いたKeyGraphの結果を見て分類

killerはtalkがないので、説明を本手法のあるなしに関わらず説明をきちんと読めば正答できる。一方IMKとEMKは、本手法ありの場合は高い正答率になった。誤答の理由として、二人の被験者はEMKの「戦闘をやや好む」という特徴が該当すると考えたため、図4の右上のattackのかたまりを見て、IMKとEMKを取り違えた。もう二人の被験者は理由なしと答えている。

6.4.2 本手法を用いないKeyGraphの結果を見て分類

killerに関しては手法を用いた場合と同じである。IMKとEMKに関しては「わからない」という回答が多く正答率が低くなっている。

表 3: 本手法を用いた KeyGraph の正誤

回答 \ 正答	killer	IMK	EMK
killer	9	1	0
IMK	0	7	3
EMK	1	2	7
未回答	0	0	0
正答率	0.9	0.7	0.7

表 4: 本手法を用いない KeyGraph の正誤

回答 \ 正答	killer	IMK	EMK
killer	9	0	0
IMK	0	3	3
EMK	0	3	3
未回答	1	4	4
正答率	0.9	0.3	0.3

[3] 砂山渡, 大澤幸生, 谷内田正彦: KeyGraph キーワード抽出ツールから発見ツールへの展開, 2000

[4] (株) 構造計画研究所 創造工学部 KeyGraph コミュニティ
<http://www2.kke.co.jp/keygraph/>

7 結論と今後の課題

killer の主な行動はattack, IMK はランダム, EMK は状況に応じた多彩な行動をとっていることが確認できた. 6.4にあるようにこれらの特徴をうまく分類することもできる. 今回提案した手法を用いた例では, それぞれのエージェントの特徴の抽出に成功したと言える. 逆に本手法用いない例では特徴抽出として十分な結果を示すことができなかった.

今回は構造計画研究所が提供する KeyGraph[4] と, MMOG シミュレータの Zereal を用いて実験した. 今後は, 実際の MMOG のログデータを対象とした場合でも十分な結果を出せるか検証することや, さらに明瞭な特徴抽出ができるように KeyGraph 自体のアルゴリズムを MMOG ログの解析に適したように改良する必要があると考えている.

参考文献

- [1] FOR IMMEDIATE RELEASE ZONA, INC., AND EXECUTIVE SUMMARY CONSULTING, INC., RELEASE STUDY DETAILING CURRENT AND FUTURE WORLDWIDE MARKET PROJECTIONS OF MASSIVELY MULTI-PLAYER ONLINE GAMES
- [2] International Game Developers Association, IGDA オンラインゲーム白書, 2003