

Universidade Federal do ABC  
Centro de Matemática, Computação e Cognição (CMCC)  
Pós-Graduação em Ciência da Computação

Jonathan Ohara de Araujo

AGENTES INTELIGENTES PARA BATALHAS POKÉMON

Dissertação

Santo André - SP

2016

Jonathan Ohara de Araujo

## AGENTES INTELIGENTES PARA BATALHAS POKÉMON

Qualificação

Qualificação apresentada ao Curso de Pós-Graduação da Universidade Federal do ABC  
como requisito parcial para obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Fabrício Olivetti de França

Santo André - SP

2016

## Ficha Catalográfica

de Araujo, Jonathan Ohara.  
Agentes inteligentes para batalhas Pokémon / Jonathan Ohara  
de Araujo.  
Santo André, SP: UFABC, 2016.  
3p.

Jonathan Ohara de Araujo

## AGENTES INTELIGENTES PARA BATALHAS POKÉMON

Essa Qualificação foi julgada e aprovada para a obtenção do grau de Mestre em Ciência da Computação no curso de Pós-Graduação em Ciência da Computação da Universidade Federal do ABC.

Santo André - SP - 2016

---

Prof. Dr. João Paulo Góis

Coordenador do Curso

### BANCA EXAMINADORA

---

Prof. Dr. Fabrício Olivetti de França

---

Prof.Dr.C

---

Prof.Dr.B

---

Prof.Dr.D

# Resumo

Esse trabalho explora a criação de agentes inteligentes competitivos em um ambiente onde dezenas de milhares de jogadores humanos competem diariamente para melhorar sua colocação no jogo Pokémon Showdown.

Escolher o melhor algoritmo e a melhor forma de aprendizado para esse agente é um dos grandes desafios desse trabalho. Outro grande aspecto que o agente precisa se adaptar é a grande variedade de composição de times que o agente e o seu adversário pode montar. Para obter a melhor flexibilidade o agente será submetido somente a batalhas randômicas onde as composições de equipes são todas aleatórias e a avaliação de performance do agente será feita através do sistema de ranqueamento do próprio jogo.

# Abstract

*To Do.*

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Principais contribuições . . . . .	2
<b>2</b>	<b>Agentes Inteligentes</b>	<b>3</b>
2.1	Classificações de agentes . . . . .	3
2.2	Agentes em jogos . . . . .	5
2.3	Agentes inteligente contra jogadores . . . . .	5
<b>3</b>	<b>Metodologia</b>	<b>7</b>
3.1	Treino e aprendizado . . . . .	7
3.2	Avaliação de resultado . . . . .	8
<b>4</b>	<b>Plano de Trabalho</b>	<b>9</b>

# Lista de Figuras

2.1	Visão parcial da tipologia de um Agente . . . . .	4
-----	---	---



# Lista de Tabelas

4.1 Cronograma da Dissertação . . . . .	11
---	----

# Capítulo 1

## Introdução

### 1.1 Motivação

Existem diversas finalidades para criação de agentes inteligentes para jogos. Podemos enumerar algumas como: adaptação a diferentes tipos de jogador, gerar diferentes experiências em cada nova partida, gerar grandes dificuldades entre outras.

Esse trabalho explora a criação de agentes inteligentes que compitam com jogadores humanos no sistema de batalhas *Pokémon* no ambiente *Pokémon Showdown*.

Um dos grandes desafios na criação desses agentes é a adaptabilidade e a competitividade. Diferentes de jogos como xadrez, damas, reversi entre outros jogos de tabuleiro, em batalhas *Pokémon* nada se sabe do adversário até que comece o jogo, ou seja, invalidando qualquer tipo de técnica prevendo possíveis movimentos antes do jogo começar. Para acentuar ainda mais essas propriedades os agentes irão treinar e competir no modo randômico, nesse modo a escolha do seu time assim como de seu adversário é feito pelo próprio sistema do jogo.

Por causa da característica de desconhecimento do time adversário, dificulta bastante a técnica criação de árvores de possíveis jogadas do adversário, pois o agente precisaria prever os possíveis *Pokémon*s adversários assim como suas características assim, prejudicando a utilização de técnicas baseadas em grafos de jogadas como a usada no famoso sistema *Deep Blue* que segundo o trabalho *Deep Blue System Overview* [Hsu et al., 1995] "O *Deep Blue* é um massivo sistema paralelo para realização de busca em árvores de jogos de xadrez".

## 1.2 Objetivos

O Objetivo desse trabalho é o desenvolvimento de agentes inteligentes que joguem e aprendam com milhares de jogadores humanos. Será implementada distintas técnicas para criação e aprendizado desses agentes. No decorrer do trabalho será sumarizado a evolução dos agentes no sistema de ranqueamento do jogo e, essa posição será confrontada com a quantidade de treinamento que cada agente recebeu, podendo assim observar a curva de melhora em relação a quantidade de treinamento.

Para criação desses agentes foi desenvolvida uma *API* (Application Programming Interface) que permitirá a comunicação com o jogo *Pokémon Showdown*. Inicialmente a *API* está disponibilizada apenas para *JavaScript* mas durante o desenvolvimento do projeto será portada para *Java*

## 1.3 Principais contribuições

O trabalho irá explorar a criação de agentes inteligentes adaptativos, num ambiente que pouco se sabe sobre o adversário. Os agentes terão informações apenas durante a batalha e, essas informações são apenas aquelas que o adversário realizar. Por exemplo: o agente só saberá que adversário tem um *Pokémon* até o mesmo usá-lo, os movimentos que *Pokémon* tem serão apenas conhecidos a medida que o adversário utilizá-los e existem características que agente não tem como descobrir como por exemplo a quantidade de ataque e defesa distribuída no monstro.

Além disso a construção de uma *API* para acesso ao jogo contribuirá para que outros pesquisadores também possam desenvolver estudos e criar seus próprios agentes podendo criar-se uma cultura de competição entre agente inteligentes na plataforma *Pokémon Showdown*.

# Capítulo 2

## Agentes Inteligentes

Na computação, especialmente em inteligência artificial, é bem comum utilizar-se do termo agente ou agente inteligente. Por definição do dicionário [mic, 2016] agente significa "Que age, que exerce alguma ação; que produz algum efeito".

Na academia temos definições mais direcionadas como a de [Russell and Norvig, 2010] "Um agente é algo capaz de perceber seu ambiente através de sensores e agir sobre esse ambiente por meio de atuadores", ou a de [Smith et al., 1994] "Vamos definir um agente como uma entidade de *software* persistente dedicada para um propósito específico. 'Persistente' distingue agentes de sub-rotinas; agentes têm suas próprias ideias sobre como realizar tarefas, suas próprias agendas. 'Propósitos especiais' os distingue de aplicações multifuncionais inteiras; agentes são tipicamente muito menores."

Existem diversas definições de agentes. Mas qual seria a diferença de um agente para um programa de computador?. Os autores [Franklin and Graesser, 1997] propõem uma definição mais sólida para agente "Um agente autônomo é um sistema situado no dentro e como parte de um ambiente que sente esse ambiente e age sobre ele, ao longo do tempo, em busca de sua própria agenda e de modo a efetivar o que sente no futuro."ainda segundo os autores maioria dos programas comuns quembram ou mais regras dessa definição como, por exemplo, um sistema de folha de pagamento sente o ambiente através de suas entradas e age sobre ele gerando uma saída, mas não é um agente porque é sua saída normalmente não afeta o que ele sente mais tarde.

### 2.1 Classificações de agentes

Os agentes possuem também diferentes explicações quanto a suas características, classificações e taxonomia. Uma das definições mais antigas e mais adotadas é a de [Nwana, 1996] nesse trabalho o autor define três dimensões para o agente:

- **Mobilidade.** Capacidade de se mover em torno de alguma rede. Definindo como agentes fixos ou móveis;
- **Deliberativos ou reativos.** Agentes deliberativos possuem simbólicos internos, modelos de raciocínio e se envolvem com o planejamento e negociação a fim de conseguir uma coordenação com outros agentes. Já os reativos agem por estímulos/respostas que o estado atual do ambiente que está inserido o proporciona.
- **Ideais e atributos primários.** Os agentes podem ser classificados em três características: autônomos, cooperativos e aprendizes. Autonomia se refere ao princípio que o agente deve realizar ações sem intervenção humana. Cooperação é a habilidade de se comunicar com outros agentes. Aprendizagem é a capacidade de aprender conforme ações são lidas ou realizadas. Dessas definições surgem quatro tipos de agentes conforme mostra a Figura 2.1.

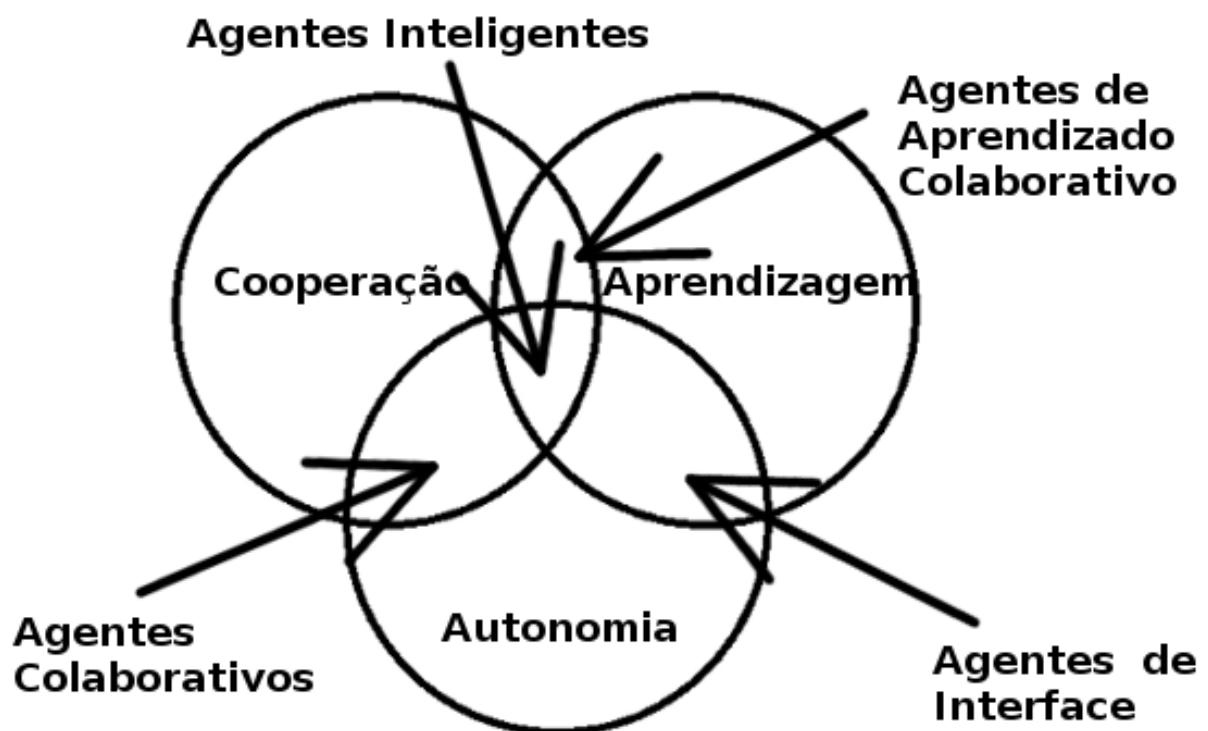


Figura 2.1: Visão parcial da tipologia de um Agente.

Já [Franklin and Graesser, 1997] classifica os agentes pelas propriedades que eles apresentam, são elas: reatividade, autonomia, orientada a objetivo, temporária ou contínua, comunicativa, aprendizagem, mobilidade, flexibilidade e caráter.

Emboras as diferentes definições muita coisas são comuns ou muito parecidas como a característica de autonomia, aprendizado, objetivos e comunicatividade.

## 2.2 Agentes em jogos

Antigamente maioria do processamento do sistema que rodava um jogo (computador, video-game entre outros) era utilizada em sua grande maioria para processamentos gráficos. Com a evolução e custo baixo dos equipamentos (*hardware*) hoje em dia é possível dar mais atenção a outros aspectos de um jogo. Segundo [van Lent et al., 1999] "Como jogos de computador se tornam mais complexos e os consumidores exigem adversários mais sofisticados controlados pelo computador, os desenvolvedores de jogos são obrigados a colocar uma maior ênfase nos aspectos de inteligência artificial de seus jogos". Por isso hoje, existem um grande número de jogos que não são conhecidos por sua beleza gráfica, segundo [Millington and Funge, 2009] recentemente, um número cada vez maior de jogos tem como a Inteligência Artificial o ponto principal do jogo. Creatures [Cyberlife Technology Ltd., 1997] em 1997, e jogos como The Sims [Maxis Software, Inc., 2000] e Black and White [Lionhead Studios Ltd., 2001]. Ainda segundo [Millington and Funge, 2009] o jogo Creatures foi um dos sistemas mais complexos de inteligência artificial visto em um jogo, com um cérebro baseado em rede neural para cada criatura.

Agentes para jogos tem uma infinidade de propósitos. Ao associar esses dois termos é bem comum pensar em oponentes controlados por agentes, porém a utilização de agentes vai muito além disso. É possível utilização de agentes em outros aspectos como na geração de conteúdo procedural, criação de desafios e missões entre outros.

O artigo *Procedural Content Generation for Games: A survey* [Hendrikx et al., 2013] cita que agentes para criação de diferentes tipos de terrenos sem perder o controle do *design*.

Segundo [Doran and Parberry, 2011] a geração procedural de missões em jogos (especialmente para MMORPGs online) tem o potencial de aumentar a variedade e a longevidade dos jogos.

Existem também outras aplicações para agentes como busca de caminhos em ambientes complexos, geração de objetos, destruição de objetos entre outras aplicações.

## 2.3 Agentes inteligente contra jogadores

Existem várias abordagens para criação de agentes que enfrentam jogadores. Pode-se enumerar alguns deles como agentes adaptativos, evolutivos, competitivos entre outros.

Em alguns jogos o jogador não tem uma opção de dificuldade, ficando a critério do sistema do jogo controlar a dificuldade como em *Middle Earth: Shadow of Mordor*. Segundo [Ponsen and Spronck, 2004] no jogo Max Payne 2 é introduzido algo chamado opções dinâ-

micas de dificuldade. Informações do mundo jogo são extraídas para estimar a habilidade do jogador, ajustando a dificuldade da inteligência artificial.

Alguns agentes são criados para acompanhar a curva de aprendizado do jogador, para que o jogador sempre tenha um bom nível de desafio sempre que jogar. No jogo *Star Craft 2* existe o modo chamado pareamento contra Inteligência artificial, nesse modo o desempenho do jogador nos jogos anteriores define qual o nível do jogador, ou seja, quanto mais o jogador evolui mais difícil será seu oponente.

Outra abordagem interessante é a utilizada em Drivatar [Herbrich1, 2011] utilizada no jogo Forza Motorsport onde o sistema cria agentes "humanizados" que aprende com a pessoa que está jogando, incorporando suas características e utilizando para jogar contra o próprio jogador.

Segundo [Schaeffer and van den Herik, 2002] apenas na década de 90 os computadores foram capazes de competir com sucesso contra o melhor dos humanos. O primeiro jogo a ter um campeão mundial não humano foi o jogo de damas em 1994 seguido pelo Deep Blue em 1997.

# Capítulo 3

## Metodologia

Para atingir os objetivos propostos serão implementados três agentes. Dois desses agentes serão submetidos a uma grande quantidade de treino para que possam chegar a patamares estáveis. O terceiro será um agente baseado em grafos com diferentes profundidades onde, será categorizado a evolução no ranqueamento do jogo de acordo com o aumento de profundidade de árvore de decisão.

- **Agente 1 com neuroevolução.** O primeiro agente persiste em um rede neural onde os pesos de sua rede serão ajustados por um algoritmo evolutivo.
- **Agente 2 com aprendizado por reforço.** Esse agente irá se ajustar através de estímulos positivos e negativos que regularão sua decisão dentro do jogo através do algoritmo de aprendizado por reforço.
- **Agente 3 baseado em grafo de decisão.** Utilizará uma árvore de decisão onde os possíveis movimentos serão mapeados. O algoritmo tentará prever possíveis incógnitas do adversário assumindo o pior cenário possível para cada variável não conhecida.

### 3.1 Treino e aprendizado

O primeiro agente que será testado contra jogadores humanos é o agente 3(grafo). Será criado várias versões desse agente, cada versão será cablitrado com uma profundidade diferente em sua árvore. Cada versão do agente será submetido a uma série da batalhas que só terá fim quando seu ranqueamento se estabilizar.

Os agentes 1 e 2 terão comportamentos semelhantes em como serão treinados. Ambos agentes terão duas versões, cada versão fará um treinamento diferente:



- **Treino contra humanos:** O agente será submetido a jogar contra jogadores humanos que será escolhido pelo próprio jogo *Pokémon Showdown*, o sistema de pareamento de partidas é feito baseado no ranqueamento dos dois jogadores, ou seja, o adversário sempre vai ter uma pontuação no *rank* semelhante ao agente.
- **Treino contra agente 3:** Os agentes 1 e 2 jogarão apenas contra o agente 3, sempre que o agente que está treinando conseguir um grande número de vitória sobre o agente 3, será aumentado mais um nível de profundidade na árvore do agente 3.

## 3.2 Avaliação de resultado

Avaliar a situação da batalha será um recurso muito importante pois o agente 3 precisará avaliar cada nó de sua árvore para fazer a melhor escolha possível ou a escolha que gere menos boas escolhas para o adversário. Essa avaliação terá como base a quantidade de pokémons vivos e a situação dele (quantidade de pontos de vida, afetado por algum efeito negativo entre outros).

Ao fim de cada batalha também feita uma avaliação da vitória. Com essa avaliação será possível aferir o quão dispar foi a situação do vencedor. Além disso, com essa métrica podemos eliminar possíveis ruídos em resultados de batalhas como desistências ou desconexões por parte dos oponentes.

# Capítulo 4

## Plano de Trabalho

As atividades compreendidas no cronograma são:

- **Revisão de literatura (RDL):** Leitura e entendimentos de trabalhos que permeiam o assunto da dissertação dentre eles redes neurais, computação evolutiva, neuroevolução, aprendizado por reforço, agentes inteligentes para jogos entre outros.
- **Estudo de caso AIBirds (AIB):** Criação de um agente inteligente baseado em grafos para a competição AIBirds. AIBirds ou *Angry Birds AI Competition* é uma competição entre agentes que joguem o jogo Angry Birds. Segundo sua página oficial "A tarefa dessa competição é desenvolver um programa de computador que jogue Angry Birds autonomamente e sem intervenção humana."[aib, 2016] A competição foi sediada no congresso IJCAI 2015 e o agente desenvolvido foi nomeado UFAngry-BirdsC. O desenvolvimento desse agente foi muito importante para o trabalho, além de adquirir experiência para criação de agente para jogos serviu como inspiração para o tema do trabalho.
- **Desenvolvimento de API de comunicação com o jogo Pokemon Showdown (API):** Para que seja possível criação de agentes foi desenvolvida uma API em JavaScript que recupera todas informações e pode realizar as mesmas informações que um jogador humano pode fazer. Durante o desenvolvimento foi tomado muito cuidado para que a API não tenha mais informações ou ações que uma pessoa pode obter ou realizar.
- **Escever documento de qualificação (EDQ):** Escrever o documento para o exame de qualificação.
- **Desenvolvimento dos três agentes (DTA):** Como já citado anteriormente serão criados três agentes utilizando técnicas diferentes como: reuroevolução, aprendizado

por reforço e algoritmo de grafos baseado em *minimax*.

- **Treino inicial dos Agentes (TIA):** Como descrito na seção ?? será inicialmente treinado apenas o agente 3. Esse agente será muito importante para o treino dos próximos agentes e para sugerir uma categorização para jogadores humanos para cada faixa de ranqueamento.
- **Treino dos demais agentes (TDA):** Treino dos agentes 1 e 2 contra o agente 3 e contra jogadores humanos. Nessa fase também será sumarizada a subida no ranqueamento de acordo com a quantidade de treino.
- **Compilação de resultados (CDR):** Compilação dos resultados de cada agente e comparação da performance de cada agentes com jogadores humanos.
- **Escever documento da dissertação (EDD):** Escrever o documento para a defesa da dissertação.
- **Elaboração de artigo (EDA):** Escrever um artigo com o tema da dissertação.
- **Desenvolver WebSocket de comunicação com outras linguagens (DWS):** Será desenvolvido um WebSocket na API para que possa ser desenvolvida agentes em qualquer outra linguagem que tenha o recurso de WebSocket.
- **Escrever documentação da API (WIK):** Fazer uma página mostrando como utilizar os principais recursos da API e como utilizar ela em outra linguagens através do WebSocket.

O cronograma abaixo compreende o período de maio de 2015 data onde foram feitos as primeiras etapas relacionadas a dissertação, até setembro de 2016 mês que pretende-se defender a dissertação.

Tabela 4.1: Cronograma da Dissertação

	2015								2016								
	05	06	07	08	09	10	11	12	01	02	03	04	05	06	07	08	09
<b>RDL</b>	X	X	X	X	X	X	X	X									
<b>AIB</b>	X	X	X														
<b>API</b>							X	X	X								
<b>EDQ</b>									X	X							
<b>DTA</b>										X	X	X					
<b>TIA</b>												X	X				
<b>TDA</b>													X	X			
<b>CDR</b>														X	X		
<b>EDD</b>															X	X	X
<b>EDA</b>														X	X		
<b>DWS</b>													X	X			
<b>WIK</b>														X	X		

# Referências Bibliográficas

- [aib, 2016] (2016). AIBirds, Angry Birds AI Competition. <https://aibirds.org/>. Acessado em: 01-02-2016.
- [mic, 2016] (2016). Michaelis, Dicionários Michaelis. <http://michaelis.uol.com.br/>. Acessado em: 07-02-2016.
- [Doran and Parberry, 2011] Doran, J. and Parberry, I. (2011). A prototype quest generator based on a structural analysis of quests from four mmorpGs. In *Proceedings of the 2Nd International Workshop on Procedural Content Generation in Games*, PCGames '11, pages 1:1–1:8, New York, NY, USA. ACM.
- [Franklin and Graesser, 1997] Franklin, S. and Graesser, A. (1997). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, ECAI '96, pages 21–35, London, UK, UK. Springer-Verlag.
- [Hendrikx et al., 2013] Hendrikx, M., Meijer, S., Van Der Velden, J., and Iosup, A. (2013). Procedural content generation for games: A survey. *ACM Trans. Multimedia Comput. Commun. Appl.*, 9(1):1:1–1:22.
- [Herbrich1, 2011] Herbrich1, R. (2011). Drivatar - Microsoft Research. <http://research.microsoft.com/en-us/projects/drivatar/default.aspx>. Acessado em: 07-02-2016.
- [Hsu et al., 1995] Hsu, F., Campbell, M.S., H., and A.J.J (1995). Deep blue system overview. *Proceedings of the 9th ACM Int. Conf. on Supercomputing*, pages 240–244.
- [Millington and Funge, 2009] Millington, I. and Funge, J. (2009). *Artificial Intelligence for Games, Second Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition.
- [Nwana, 1996] Nwana, H. S. (1996). Software agents: An overview. *Knowledge Engineering Review*, 11:205–244.

- [Ponsen and Spronck, 2004] Ponsen, M. and Spronck, I. P. H. M. (2004). Improving adaptive game ai with evolutionary learning. In *University of Wolverhampton*, pages 389–396.
- [Russell and Norvig, 2010] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Discovering great minds of science. Prentice Hall.
- [Schaeffer and van den Herik, 2002] Schaeffer, J. and van den Herik, H. (2002). Games, computers and artificial intelligence. *Artificial Intelligence - Chips challenging champions: games, computers and Artificial Intelligence*, 134:1–8.
- [Smith et al., 1994] Smith, D. C., Cypher, A., and Spohrer, J. (1994). Kidsim: Programming agents without a programming language. *Commun. ACM*, 37(7):54–67.
- [van Lent et al., 1999] van Lent, M., Laird, J. E., Buckman, J., Hartford, J., Houchard, S., Steinkraus, K., and Tedrake, R. (1999). Intelligent agents in computer games. In Hendler, J. and Subramanian, D., editors, *AAAI/IAAI*, pages 929–930. AAAI Press / The MIT Press.