

# Релевантность текста и нейроинтерфейсы



# Проблема

- Во время учебы или работы, многим людям часто приходится работать с огромным количеством литературы.
- Важно то, что через некоторое время приходится к ней возвращаться.
- Не хочется все читать заново.



# Доступные технологии – нейроинтерфес



# Задача



- С помощью нейроинтерфейса научиться определять то, насколько пользователю важна читаемая в данный момент информация.

# Решение

- Бегущая строка
- Рекуррентные нейронные сети

# Результаты

Искусственный интеллект (слабая его форма) постепенно становится все более умелым. Компьютер с успехом решает даже те задачи, которые всего несколько лет назад считались доступными для понимания лишь человеку. Один из примеров — игра го, где стать чемпионом может лишь тот, у кого хорошо развита интуиция и логическое мышление. Го для машины считалась недостижимым «потолком». Сейчас же мы видим, что на Земле уже нет людей, способных выиграть у машины. Понятно, что ИИ не только в го играет, круг задач гораздо шире. Тем не менее, специалисты из разных сфер ИТ стараются обучить компьютер сражаться на равных с человеком и в компьютерных играх, с тем, чтобы проверить возможности своих программных платформ. В этом случае человек и бот находятся в одинаковых условиях. Обычно соперников сталкивают лбами на карте стратегии с «туманом войны», так что оба противника не в курсе, какие объекты и где на этой карте находятся (в обычных условиях игровой «ИИ» знает, где что находится). Именно в таких условиях человек победил машину в StarCraft. Речь идет о сражении профессионального игрока в StarCraft Сон Бён-гу (Song Byung-gu) с четырьмя различными StarCraft ботами. Один из них, «CherryPi» был разработан корпорацией Facebook. Другие боты созданы компаниями из Австралии, Норвегии и Кореи. Игра проходила в Университете Sejong, Сеул, Корея. С 2010 здесь проходят соревнования по StarCraft с участием людей и машин. Мероприятие поддерживают различные организации (как коммерческого толка, так и научно-исследовательские и общественные). Конкретно это соревнование было поддержано, в частности, Institute of Electrical and Electronics Engineers (IEEE). Понятно, что матчи по го и StarCraft нельзя сравнивать. В первом случае игрок видит всю доску и ему нужно найти выигрышную стратегию. В случае StarCraft тоже нужна выигрышная стратегия, но здесь приходится держать в уме многие объекты на карте, иметь неплохую реакцию и уметь действовать внезапно, непредсказуемо. Понимая все это, разработчики ИИ решили использовать среду StarCraft для тренировки машины. Известные игроки в StarCraft ранее уже заявляли, что очень хотели бы сразиться с компьютерным противником. Компании, которые работают в сфере ИИ, например, DeepMind, приветствовали это желание, заявив о намерении провести подобные соревнования в будущем. И это будущее наступило — люди смогли сразиться с ИИ. У компьютера не сложилось — человек победил всех четырех ботов за 27 минут в общем зачете. Максимальная продолжительность матча составила 10 минут и 30 секунд. Самый короткий матч продолжался 4 с половиной минуты. И это при том, что боты за единицу времени выполняли гораздо больше действий, чем человек. Например, система, созданная норвежскими специалистами, способна выполнять 19 000 действий в минуту. Киберчемпионы из мира людей в лучшем случае способны выполнять несколько сотен

Искусственный интеллект (слабая его форма) постепенно становится все более умелым. Компьютер с успехом решает даже те задачи, которые всего несколько лет назад считались доступными для понимания лишь человеку. Один из примеров — игра го, где стать чемпионом может лишь тот, у кого хорошо развита интуиция и логическое мышление. Го для машины считалась недостижимым «потолком». Сейчас же мы видим, что на Земле уже нет людей, способных выиграть у машины. Понятно, что ИИ не только в го играет, круг задач гораздо шире. Тем не менее, специалисты из разных сфер ИТ стараются обучить компьютер сражаться на равных с человеком и в компьютерных играх, с тем, чтобы проверить возможности своих программных платформ. В этом случае человек и бот находятся в одинаковых условиях. Обычно соперников сталкивают лбами на карте стратегии с «туманом войны», так что оба противника не в курсе, какие объекты и где на этой карте находятся (в обычных условиях игровой «ИИ» знает, где что находится). Именно в таких условиях человек победил машину в StarCraft. Речь идет о сражении профессионального игрока в StarCraft Сон Бён-гу (Song Byung-gu) с четырьмя различными StarCraft ботами. Один из них, «CherryPi» был разработан корпорацией Facebook. Другие боты созданы компаниями из Австралии, Норвегии и Кореи. Игра проходила в Университете Sejong, Сеул, Корея. С 2010 здесь проходят соревнования по StarCraft с участием людей и машин. Мероприятие поддерживают различные организации (как коммерческого толка, так и научно-исследовательские и общественные). Конкретно это соревнование было поддержано, в частности, Institute of Electrical and Electronics Engineers (IEEE). Понятно, что матчи по го и StarCraft нельзя сравнивать. В первом случае игрок видит всю доску и ему нужно найти выигрышную стратегию. В случае StarCraft тоже нужна выигрышная стратегия, но здесь приходится держать в уме многие объекты на карте, иметь неплохую реакцию и уметь действовать внезапно, непредсказуемо. Понимая все это, разработчики ИИ решили использовать среду StarCraft для тренировки машины. Известные игроки в StarCraft ранее уже заявляли, что очень хотели бы сразиться с компьютерным противником. Компании, которые работают в сфере ИИ, например, DeepMind, приветствовали это желание, заявив о намерении провести подобные соревнования в будущем. И это будущее наступило — люди смогли сразиться с ИИ. У компьютера не сложилось — человек победил всех четырех ботов за 27 минут в общем зачете. Максимальная продолжительность матча составила 10 минут и 30 секунд. Самый короткий матч продолжался 4 с половиной минуты. И это при том, что боты за единицу времени выполняли гораздо больше действий, чем человек. Например, система, созданная норвежскими специалистами, способна выполнять 19 000 действий в минуту. Киберчемпионы из мира людей в лучшем случае способны выполнять несколько сотен

# Результаты

\$\$\$1\$\$\$ Само по себе программирование подразумевает не только процесс написания кода, но зачастую на это тратится большая часть времени при разработке. Только представьте, сколько усилий приходится тратить на то, чтобы держать в голове множество правил и спецификаций к конкретному языку программирования, вместо того, чтобы сосредоточиться на решаемой проблеме. Особенно может раздражать разнообразие синтаксиса в языках: где-то нужна точка с запятой, где-то не нужны фигурные скобки, где-то вообще ни одно выражение не обходится без скобок. Что уж и говорить о холивах, напоминающие религиозные споры. \$\$\$0\$\$\$ Часто советуют начать разработку ПО с графического описания будущей системы, ее компонентов и связей между ними, чтобы на ранних стадиях определить более выгодную структуру системы и минимизировать возможные проблемы в будущем. Графическое представление легче для понимания, чем текстовый вариант, но может иметь свои ограничения, к тому же это все равно придется переводить в понятный компилятору код. Конечно, на маленькие приложения (какими они могут казаться вначале) это не распространяется, можно сразу приступить к написанию кода, но проблема все равно остается — нужно думать в рамках определенного языка программирования. Тем более, когда вы это делаете в давно приевшемся вам императивном стиле. \$\$\$1\$\$\$ Программисты по своей сути должны быть ленивыми, чтобы находить выгодные способы решения задач и не тратить силы на рутину, тем более, глядя на тенденции увеличения сложности ПО. Именно это стимулирует рождение парадигм, языков программирования и абсолютно новых, казалось бы, и малоизвестных инструментов визуального программирования.

\$\$\$1\$\$\$ Само по себе программирование подразумевает не только процесс написания кода, но зачастую на это тратится большая часть времени при разработке. Только представьте, сколько усилий приходится тратить на то, чтобы держать в голове множество правил и спецификаций к конкретному языку программирования, вместо того, чтобы сосредоточиться на решаемой проблеме. Особенно может раздражать разнообразие синтаксиса в языках: где-то нужна точка с запятой, где-то не нужны фигурные скобки, где-то вообще ни одно выражение не обходится без скобок. Что уж и говорить о холивах, напоминающие религиозные споры. \$\$\$0\$\$\$ Часто советуют начать разработку ПО с графического описания будущей системы, ее компонентов и связей между ними, чтобы на ранних стадиях определить более выгодную структуру системы и минимизировать возможные проблемы в будущем. Графическое представление легче для понимания, чем текстовый вариант, но может иметь свои ограничения, к тому же это все равно придется переводить в понятный компилятору код. Конечно, на маленькие приложения (какими они могут казаться вначале) это не распространяется, можно сразу приступить к написанию кода, но проблема все равно остается — нужно думать в рамках определенного языка программирования. Тем более, когда вы это делаете в давно приевшемся вам императивном стиле. \$\$\$1\$\$\$ Программисты по своей сути должны быть ленивыми, чтобы находить выгодные способы решения задач и не тратить силы на рутину, тем более, глядя на тенденции увеличения сложности ПО. Именно это стимулирует рождение парадигм, языков программирования и абсолютно новых, казалось бы, и малоизвестных инструментов визуального программирования.

# Что удалось?

- Успешно реализовано считывание данных.
  - где читаем
  - сигналы мозга
- Успешно реализована RNN.





# Что не удалось?

- Плохое разграничение блоков данных.
  - мало данных
  - сложно получить данные (тренировочные)
- Не реализованно одно, обобщающее приложение.



# Планы на будущее

- Сделать больше данных для тренировки.
  - с человека
  - с группы людей
- Создание итогового приложения.



**СПАСИБО ЗА ВНИМАНИЕ!**

