



IEC – Instituto de Educação Continuada
Pós-Graduação em Inteligência Artificial e Aprendizado de Máquina

Projeto Integrado de Aprendizado de Máquina- Projeto Final

Van Gogh - Analisador de sentimentos em contexto

Trabalho apresentado ao Instituto de Educação Continuada (IEC) da Pós-Graduação em Inteligência Artificial e Linguagem de Máquina da PUC Minas, como requisito parcial para obtenção de créditos na disciplina Projeto Integrado de Aprendizado de Máquina

Professor: Danilo Costa Marim Segura

Alunos: André Luiz Simões Gomes
Laércio Nogueira Júnior
Luiz Henrique Quevedo Lima
Luiz Otávio Matias da Luz
Rogério Pascini
Roque Alexandre Gonçalves
Vinícius Costa Pereira
Wagner da Silva Ferreira

Belo Horizonte, Julho - 2020

Documentação API Van Gogh

O módulo Van Gogh é um addon de análise de sentimento em contexto para chatbots. O seu objetivo é medir a temperatura de uma conversa entre um humano e um chatbot e sugerir que o atendimento seja encaminhado a um operador humano caso o humor do cliente mude de forma negativa.

O que pensamos em termos de solução foi ter um banco de dados para armazenar as interações e poder calcular o score (termômetro) da conversa em tempo real, podendo sugerir uma mudança se necessário.

Na modelagem do banco, a ideia foi reproduzir um cenário real de utilização de chatbot, com as seguintes entidades:

- **Organização:** Permite criar uma organização dentro do módulo Van Gogh. Essa organização posteriormente poderá ter múltiplas conversas de chat associadas a ela;
- **Usuários:** Permite criar usuários que terão acesso à aplicação. Usuários serão associados a organizações;
- **Chat:** Um chat é a conversa entre o cliente e um chatbot. Esse é nosso principal insumo para calcular o histórico das conversas. É nessa tabela também que armazenamos a configuração do chat. Como o Van Gogh faz o monitoramento e análise de sentimento em contexto, precisamos usar como base as definições de **contexto** e **threshold** cadastradas para o chat. Contexto é o número de mensagens que iremos olhar para calcular o score agregado. O objetivo desse conceito é evitar que uma pequena mudança de humor dispare o alerta de transferência, gerando uma experiência ruim tanto para o cliente quanto para a marca. O parâmetro threshold indica qual o limite que vamos aceitar de polaridade (negativa) até sugerir a transferência para um atendente humano;
- **Track:** Por fim, para monitorar as mensagens em tempo real, a tabela track mantém o histórico de cada mensagem trocada e sua polaridade, além do identificador do chat que originou aquela mensagem. A partir desses dados, calculamos o score agregado levando em consideração os valores definidos de **contexto** e **threshold**.

A modelagem proposta para o projeto é apresentada na figura abaixo.

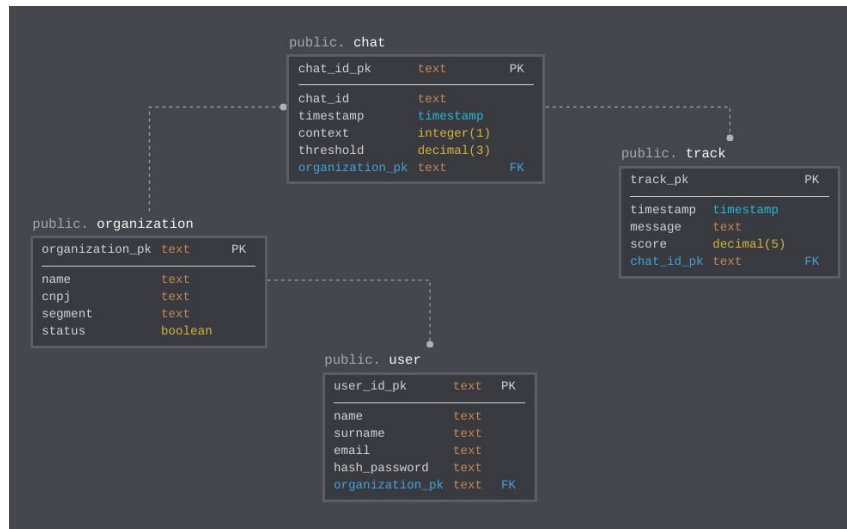


Figura 1 - Proposta de modelagem de dados Van Gogh

A API do módulo Van Gogh atua no meio de campo entre o chatbot e a aplicação de transferência de chamados. A mensagem é primeiramente enviada para o analisador de sentimentos em contexto Van Gogh e também persistida no banco. Para calcular a polaridade da mensagem recém enviada, o Van Gogh faz a requisição no **Sentiment Scorer** que nesse projeto foi utilizada o serviço de análise de sentimentos do Google por motivos de agilidade. Feito isso, o score de polaridade da mensagem é persistido no banco como um metadado da própria mensagem. Por fim, o Van Gogh calcula o score agregado das mensagens no contexto (definido previamente) para aplicação e caso ultrapasse o threshold (também definido previamente), o módulo sugere para a aplicação externa que uma ação seja tomada, pois identificou humor negativo na conversa em andamento.

O Van Gogh é responsável por orquestrar esse fluxo e o seu funcionamento é exemplificado na figura 2.

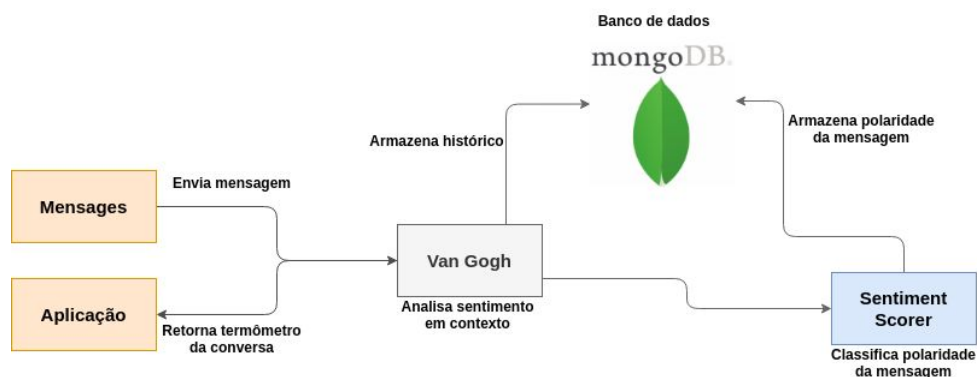


Figura 2 - Fluxo de funcionamento Van Gogh

Exemplo de retorno do API do Van Gogh com o termômetro de uma conversa específica, o chat de id `5f1384f8d798c58779ff1c34`. Como pode ser visto, o teor da conversa era negativo, e após 3 mensagens, que foi o contexto definido, e ao superar o threshold, o módulo sugere que uma ação seja tomada, caso contrário a experiência do consumidor pode ser prejudicada e o cliente perdido.

```
{
  "chat": [
    {
      "chat_id": "5f1384f8d798c58779ff1c34",
      "message": "quero falar com alguém que vai resolver o meu problema, isso é inaceitável",
      "score": -0.4000000059604645,
      "created_at": "2020-07-18T23:30:53.211000"
    },
    {
      "chat_id": "5f1384f8d798c58779ff1c34",
      "message": "nunca mais volto a fazer serviço com vocês, muito ruim",
      "score": -0.699999988079071,
      "created_at": "2020-07-18T23:30:01.920000"
    },
    {
      "chat_id": "5f1384f8d798c58779ff1c34",
      "message": "o atendimento de vocês é péssimo, estou esperando há mais de 1 hora",
      "score": -0.6000000238418579,
      "created_at": "2020-07-18T23:29:17.592000"
    }
  ],
  "agg_score": -0.5666666726271311,
  "conversation_tone_meter": "ACTION_REQUIRED",
  "context": 3,
  "threshold": -0.3
}
```

Trecho do código 1 - Retorno da API de tracking do Van Gogh para um chat

Testando a API em modo standalone

É possível testar a API em modo standalone, simulando um ambiente de conversa real. A API está em produção em: <https://van-gogh-api..com/docs>. As seguintes rotas estão disponíveis:

- **/organization:** permite criar uma organização para associar conversas de chat. Ao criar uma organização é possível gerenciar as conversas de chat por organização;
- **/chat:** para criar uma conversa de chat, retornar as configurações de um chat e deletar um chat existente. Chats são associados a uma organização para melhor gerenciamento das conversas;
- **/tracking:** essa rota permite trackear as mensagens enviadas e medir a polaridade das mesmas. É possível enviar cada mensagem enviada pelo agente humano (cliente) e calcular o seu score de polaridade. Por fim, podemos também monitorar mudanças de direção no humor da conversa.

Testando a API simulando uma conversa

Para testar o fluxo da API, é possível simular um cenário real de um sistema de interação entre agentes via chatbot.

O primeiro passo é criar uma organização para associar conversas de chat a ela. Para criar uma organização, podemos emitir a seguinte requisição.

```
curl -X POST "https://van-gogh-api.herokuapp.com/api/organization/" -H
"accept: application/json" -H "Content-Type: application/json" -d
"{\"name\": \"PUC\", \"cnpj\": \"17.178.195/0001-67\", \"segment\": \"Education\", \"active\": true}"
```

Alternativamente, passe no corpo da requisição os dados necessários para fazer a requisição POST via interface do swagger.

```
{
  "name": "PUC",
  "cnpj": "17.178.195/0001-67",
  "segment": "Education",
  "active": true
}
```

Se tudo ocorrer bem, uma mensagem de sucesso será retornada no corpo da resposta com o id da organização recém criada. Iremos precisar desse identificador para associar um chat à organização PUC.

```
{
  "message": "organization created",
  "organization_id": "5f18dba22c1e301d0001630b"
}
```

O próximo passo é iniciar um chat para essa organização que acabamos de criar e configurar as definições de "contexto" e "threshold".

Para esse exemplo, iremos definir contexto como 4 e threshold como -0.3. O id da organização é o que foi gerado na requisição anterior.

```
curl -X POST
"https://van-gogh-api.herokuapp.com/api/chat/start/?organization_id=5f18dba
22c1e301d0001630b&context=4&threshold=-0.3" -H "accept: application/json"
-d ""
```

Em caso de sucesso, o corpo da resposta da API irá conter uma mensagem intuitiva informando do sucesso da operação e o identificador único do chat recém iniciado. Esse id também vai ser necessário para começarmos a monitorar as mensagens desse chat. A resposta da API é mostrada abaixo.

```
{
  "message": "chat created",
  "chat_id": "5f18dd618b290b3dcc5c7e60"
}
```

Por fim, para começarmos a monitorar as mensagens, precisamos enviar essa mensagem para o módulo Van Gogh e associá-la a um chat existente. Podemos fazer isso com a seguinte requisição.

```
curl -X POST "https://van-gogh-api.herokuapp.com/api/track/{chat_id}/" -H
"accept: application/json" -H "Content-Type: application/json" -d
"{\"chat_id\": \"5f18dd618b290b3dcc5c7e60\", \"message\": \"olá, gostaria de
resolver o meu problema com o cartão, estou tentando há dias sem
sucesso.\"}"
```

É preciso enviar mais de uma mensagem para essa mesma rota mudando somente a mensagem enviada para simular uma conversa real. Após enviar múltiplas mensagens, podemos monitorar a temperatura da conversa com a seguinte requisição.

```
curl -X GET
"https://van-gogh-api.herokuapp.com/api/track/5f18dd618b290b3dcc5c7e60/" -H
"accept: application/json"
```

A API irá retornar no corpo da resposta o histórico de mensagens, o score agregado, a configuração de contexto e threshold do chat e a temperatura da conversa. ACTION_REQUIRED sugere que o atendimento seja transferido para um operador humano a fim de não prejudicar a experiência do consumidor. O retorno da API é mostrado no trecho abaixo.

```
{
  "chat": [
    {
      "chat_id": "5f18dd618b290b3dcc5c7e60",
      "message": "gosto do serviço de vcs mas o atendimento está deixando a desejar",
      "score": -0.30000001192092896,
      "created_at": "2020-07-23T00:50:36.064000"
    },
    {
      "chat_id": "5f18dd618b290b3dcc5c7e60",
      "message": "aguardo uma resposta.",
      "score": -0.20000000298023224,
      "created_at": "2020-07-23T00:50:01.932000"
    },
    {
      "chat_id": "5f18dd618b290b3dcc5c7e60",
      "message": "olá, gostaria de resolver o meu problema com o cartão, estou tentando há dias sem sucesso.",
      "score": -0.6000000238418579,
      "created_at": "2020-07-23T00:48:10.792000"
    }
  ],
  "agg_score": -0.36666667958100635,
  "conversation_tone_meter": "ACTION_REQUIRED",
  "context": 3,
  "threshold": -0.3
}
```

Relatório API Van Gogh

O relatório da API Van Gogh permitirá avaliar os atendimentos que passaram pela análise de sentimentos e foram redirecionados ao atendente humano. O relatório foi desenvolvido utilizando o Google Data Studio e pode ser acessado a partir da URL: <https://datastudio.google.com/u/0/reporting/1b0908af-c15b-4fc6-96eb-01a475028ca1/page/mlXZB>

Os indicadores avaliados pelo relatório da API Van Gogh são:

Quantidade de Atendimentos Redirecionados ao Atendente Humano: quantidade de atendimentos que tiveram seu sentimento avaliado como negativo que dispararam o alerta de transferência e foram efetivamente transferidos.

Quantidade de Atendimentos Finalizados pelo Chatbot: quantidade de atendimentos que não foram redirecionados para o atendente humano.

Quantidade de Atendimentos Com Sentimento Negativo Revertidos Para Sentimento Positivo: quantidade de atendimentos que foram redirecionados ao atendimento humano que tiveram seu sentimento revertido para o positivo.

Avaliação do Redirecionamento Pelo Atendente Humano: feedback do atendimento dado pelo atendente que recebeu o redirecionamento. Ao receber o atendimento o atendente dará um feedback informando se concorda ou não que o redirecionamento foi realizado no corretamente.

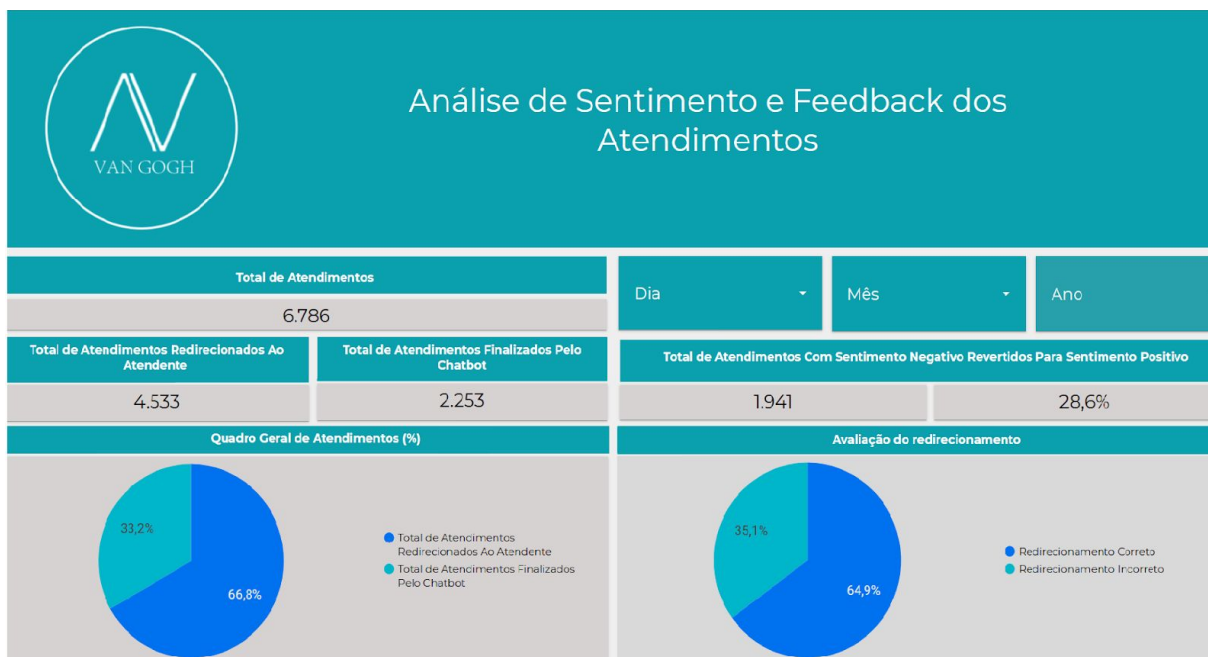


Figura 3 - Relatório Análise de Sentimentos e Feedback dos Atendimentos Van Gogh

Gerenciamento do Projeto

O Desenvolvimento e Implementação do Projeto teve início a partir de uma primeira reunião de Kickoff, onde foram discutidos o escopo e entregáveis do projeto, com o objetivo de:

- Obtenção de uma visão unificada por parte de todos os membros da equipe,
- Alinhamento de todos com relação ao cronograma do projeto
- Definição de responsabilidades de cada time.

Nesta reunião de Kickoff a pauta final foi:

- O acompanhamento e gerenciamento do projeto com a utilização da Ferramenta JIRA;
- Projeto conduzido tomando-se como referência a metodologia Scrum;
- Identificação das tarefas a serem cumpridas, designando-se os responsáveis de cada tarefa bem como o seu respectivo cronograma;
- Definição da quantidade de sprints necessárias para a execução do projeto.

Cada sprint teve início a partir da sua Reunião de Kickoff (sprint planning), com a participação de todos os membros da equipe. Por questões de adaptação à disponibilidade de tempo dos membros, optou-se pela realização, em cada Sprint, de uma reunião intermediária de acompanhamento do projeto e outra de encerramento.

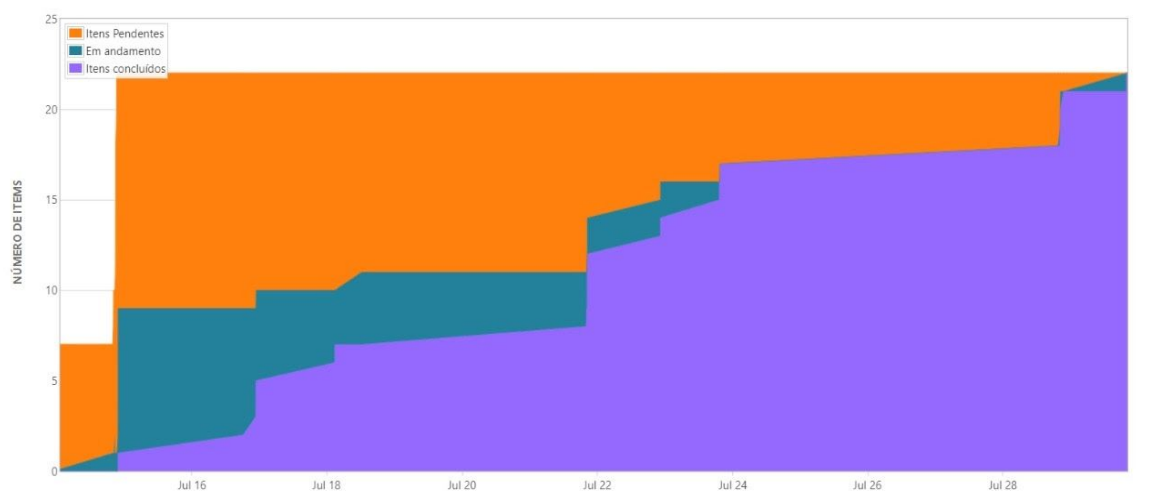
Conforme pode ser observado na documentação do projeto, apenas 2 tarefas sofreram atraso e foram estendidas para a sprint seguinte.

O projeto foi concluído dentro do prazo previsto, com todas os times gerando os seus respectivos entregáveis.

Evidências do Gerenciamento do Projeto

Apresentamos abaixo alguns gráficos que representam a evolução dos trabalhos:

Diagrama de fluxo cumulativo



A figura acima demonstra a evolução do número de tarefas realizadas ao longo das sprints, ao passo que o gráfico abaixo demonstra a subdivisão das sprints, ressaltando o esforço aplicado em cada uma delas.

Projeto: Projeto Final - Van Gogh (Sprint)

Gráfico

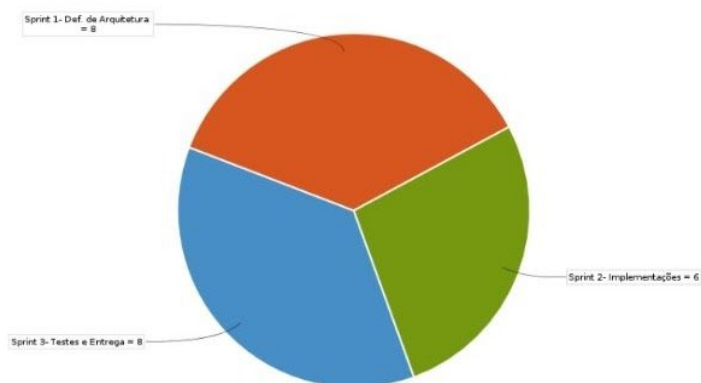


Tabela de Dados

	Itens	%
Sprint 3- Testes e Entrega	8	36%
Sprint 1- Def. de Arquitetura	8	36%
Sprint 2- Implementações	6	27%

Juntamente com a presente documentação estamos apresentado o Anexo I contendo o Relatório de Evidência do Gerenciamento do Projeto.

A documentação total do Gerenciamento do Projeto pode ser acessada através do seguinte link: <https://projetointegrado.atlassian.net/secure/BrowseProjects.jspa>