Abordagem não supervisionada para criação e manutenção da base de dados de chatbots.

Anthony Miranda Vieira, RA 229058
Rodolfo de Nadai, RA 208911

Resumo – Este trabalho apresenta uma abordagem para automatizar a criação e manutenção da base de dados de *chatbots*. O processo de criação e manutenção da base de perguntas de um assistente virtual emprega tempo e esforço de mão de obra especializada no assunto abordado, pois toda classificação das perguntas da base em intenções é feita de forma manual e individual. Empregando técnicas de aprendizagem não supervisionada, este trabalho tem como objetivo automatizar o trabalho manual dos especialistas na classificação das perguntas e, com isso, ganhar rapidez e economizar mão de obra. Apesar das dificuldades encontradas no idioma português, através da comparação do emprego de algumas técnicas, os resultados obtidos neste trabalho foram satisfatórios e indicam um caminho promissor para resolução deste problema.

Palavras-chave – *chatbot*, aprendizagem não supervisionada, clusterização, K-Means, Agglomerative Clustering, ground truth, corpus, Word2Vec, TD-IDF

1. Introdução

Este projeto oferece uma solução para o problema de tempo e esforço na criação e manutenção da base de dados dos *chatbots* (assistente virtual). Atualmente no mercado existem diversos serviços em *Cloud* disponíveis para criação da estrutura dos *chatbots* (IBM Watson, Google DialogFlow, Microsoft Azure Bot, etc), tornando assim a tecnologia de NLP (*Natural Language Processing*) e hierarquia dos diálogos uma tarefa menos complexa de ser executada. Isto faz com que um dos grandes desafios para a criação de chatbots sejam a criação da base inicial (*ground truth*), pois todos os assistentes necessitam de uma base de dados rotulada para seu funcionamento, e a manutenção da base (*corpus*) com as novas interações que estão sendo executadas no *chatbot* já existente.

Normalmente o processo de criação inicia-se através da seleção de perguntas históricas de um determinado tema (tema o qual será abordado pelo assistente) através de perguntas frequentes (FAQs) ou dos *logs* existentes de interações anteriores (*tickets* históricos em caso de uso de *HelpDesk*, *e-mails* antigos, etc.). Cada pergunta selecionada deve ser manualmente rotulada em uma intenção específica (classe) para possibilitar posteriormente a criação do diálogo.

Assim, as perguntas depois de selecionadas são manualmente revisadas e rotuladas por um grupo de especialistas que entendem do assunto. Esse processo é demorado e custoso pois requer atividade humana especializada proporcional a quantidade de perguntas selecionadas para serem usadas no *chatbot*.

O processo de manutenção do *corpus* segue um sistema similar, onde toda a base de novas perguntas é revisada manualmente por um especialista. Durante essa revisão, é verificado se a nova pergunta pertence ao domínio do *chatbot* (caso positivo, ela é incluída em uma intenção já existente) ou se trata de uma pergunta de um tema novo, ou seja, se uma nova intenção deve ser criada para acomodar essa pergunta.

A seguir, o diagrama que demonstra o processo normal de criação e manutenção de um *chatbot*:

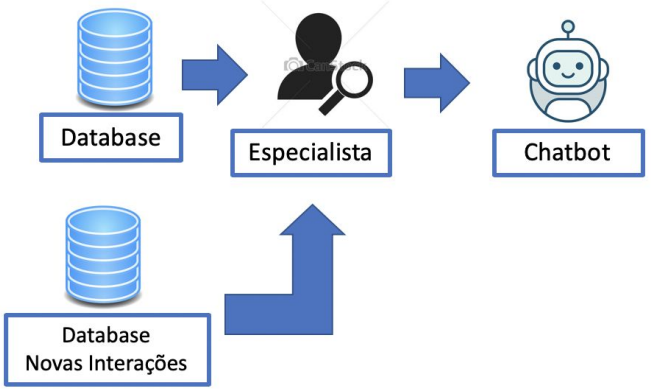

Figura 1. Processo manual de criação e manutenção de *chatbots*.

Este contexto demonstra que é necessário um método que automatize os processos explicados anteriormente. A classificação manual das perguntas para a criação e manutenção do *chatbot* precisa ser realizada de forma mais rápida e menos dependente do esforço de um especialista humano.

2. Proposta

Com base nas questões levantadas na introdução, este projeto propõe um método que emprega aprendizado de máquina não supervisionado para automatizar as etapas manuais e cria um fluxo de processo que, uma vez seguido, pretende diminuir a quantidade de tempo e esforço realizado por humanos.

O método faz uso de um algoritmo de clusterização não supervisionado para classificação, ou seja, para realizar a rotulação da base inicial de perguntas que irão compor o *ground truth* do *chatbot*.

Para a etapa de manutenção do conhecimento, uma vez que o assistente esteja criado, todos os *clusters* criados anteriormente são considerados como um único *cluster* que representa o conhecimento do assistente. Assim as novas perguntas que chegam são submetidas ao algoritmo de clusterização para detecção de anomalias, e caso esta nova pergunta seja de um tema já abordado pelo assistente, ou seja, se já possuir uma intenção correspondente, ela irá pertencer a este único *cluster*. Desta forma ela é automaticamente adicionada ao *cluster* que possui maior similaridade. Caso contrário, ou seja, se a nova pergunta for uma anomalia (*outlier*) para o *cluster* de conhecimento, ela é considerada uma pergunta fora do domínio de conhecimento do assistente, ou seja, uma pergunta que não possui intenção criada. Neste caso, as novas perguntas que não possuem intenções criadas representam um novo assunto e portanto devem ser armazenadas para futuramente serem submetidas ao processo automático de criação da base inicial, ou seja, serem clusterizadas pelo algoritmo de clusterização não supervisionado.

Segue o diagrama que demonstra o processo proposto de criação e manutenção de um *chatbot*:

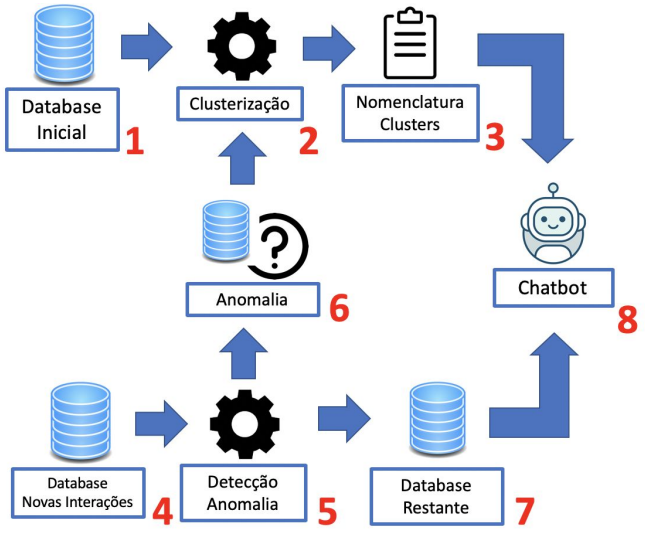

Figura 2. Processo automatizado de criação e manutenção de chatbots.

1- Database inicial: Base de dados de perguntas realizadas anteriormente que servirá de *ground truth* para criação do *chatbot*.
2- Clusterização: Algoritmo de classificação não supervisionado para clusterizar a base de dados desconhecida do *chatbot*.
3- Nomenclatura *Clusters*: Técnica para nomear automaticamente os *clusters* gerados pela etapa de clusterização para facilitar a criação do *chatbot*. Esta técnica consiste em utilizar as 2 palavras mais frequentes (preferencialmente um verbo e um substantivo) de um único cluster para rotular o cluster, por exemplo, "consultar_saldo" ou "pedir_férias".
4- Database Novas Interações: Base de dados com as novas perguntas realizadas no *chatbot* já em funcionamento.
5- Detecção de Anomalia: Algoritmo de clusterização não supervisionado capaz de identificar *outliers*.
6- Anomalia: Base de anomalias (*outliers*) identificados pelo algoritmo de detecção de anomalia. Na prática são perguntas de assuntos novos, ou seja, que não compõem a atual base de conhecimento do *chatbot*.
7- Database Restante: São as novas perguntas que não foram identificadas como anomalias e portanto estão dentro do domínio de conhecimento do *chatbot*, ou seja, já possuem intenções criadas para acomodá-las.
8- *Chatbot*: Sistema de atendimento automatizado sobre um domínio de conhecimento.

A base de dados utilizada neste projeto para criação inicial do *chatbot* é composta de 240 perguntas em português. Para efeitos comparativos do cálculo de desempenho dos algoritmos não supervisionados, essa base foi classificada manualmente em 12 intenções (classes) onde cada intenção possui 20 perguntas. Para a etapa de detecção de anomalia, foram utilizadas 100 novas perguntas em português (sendo 50 perguntas similares a base inicial e 50 perguntas de temas distintos).

Todas as perguntas antes de submetidas aos algoritmos foram pré processadas com tokenização, lematização, remoção de acentos e *stopwords*.

Para a etapa de clusterização, optou-se realizar comparação de desempenho de duas técnicas de geração do espaço vetorial de documentos, o método TF-IDF (*Term Frequency-Inverse Document Frequency*), o qual propõe a criação de um espaço

vetorial baseado na contagem de palavras e a frequência com que aparecem no conjunto de todos os documentos apresentados, e o *Paragraph Vector*[1] (Doc2Vec), que cria um espaço vetorial a partir da execução de uma rede neural *shallow*, cujo resultado são vetores em espaços multidimensionais representando cada frase e/ou documento e sua relação com os outros. Por ser uma extensão do *Word Vector*[2] (Word2Vec), o método propõe encontrar contextos linguísticos (e consequentemente relação entre palavras) dentro do conjunto total de documentos.

Como métricas para o cálculo de distâncias dos documentos no espaço vetorial foram avaliadas a distância cossenoidal[3][4] e euclidiana.

O algoritmo *K-Means* foi escolhido para ser empregado como o algoritmo de clusterização não supervisionado devido a sua popularidade e simplicidade [5].

Na etapa de detecção de anomalia, optou-se realizar a comparação de desempenho entre os algoritmos *K-Means* e *Agglomerative Clustering*. Nesta etapa o algoritmo *Agglomerative Clustering* foi selecionado por apresentar uma abordagem de clusterização do tipo hierárquica, que colabora para classificação taxonômica das palavras. Com isto, apenas o método TF-IDF para geração do espaço vetorial e a métrica de distância cossenoidal foram empregadas nesta fase.

3. Resultados

Abaixo temos uma amostra da base de 240 perguntas utilizadas:

perguntas
Como posso registrar um cartão SIM?
Por favor adicione o roaming internacional na minha conta.
Fui assaltado, preciso acionar seguro.

Figura 3. Amostra da base de dados.

Como descrito anteriormente, esta base passou por um processo de pré processamento, tendo como resultado, alguns dos seguintes exemplos.

```
['servico', 'bot', 'oferecer'], tags=[1]),
['voce', 'um', 'nome'], tags=[2]),
['nao', 'recarregar', 'desativar', 'ativar'], tags=[3])
```

Figura 4. Amostra da base de dados pré processada.

Para a avaliar a melhor métrica de similaridade (distância) a ser utilizada no algoritmo *K-Means* foi realizado uma comparação manual supervisionada da base de dados. Neste sentido obteve-se as seguintes acurácias de acordo com o método e métrica de similaridade.

Método	Métrica	Acurácia
TF-IDF	cossenoidal	85.4% ± 2
	euclidiana	81.8% ± 2
Doc2Vec	cossenoidal	80.7% ± 2
	euclidiana	74.5% ± 2

Tabela 1. Resultado dos testes de acurácia.

Como forma de visualização dos dados no espaço vetorial, foi utilizado o algoritmo de redução de dimensionalidade MDS (*Multidimensional Scaling*) com 3 dimensões, inicialização aleatória e um máximo de 500 interações.

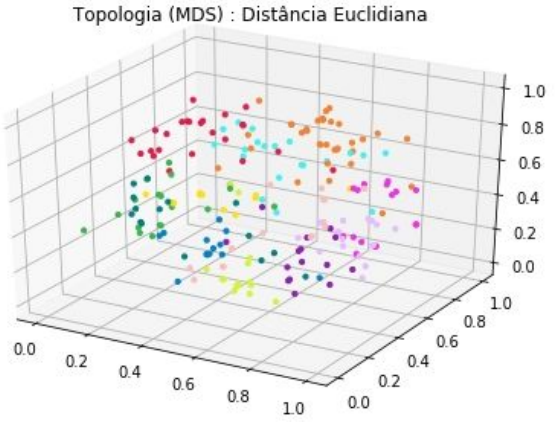

Figura 5. Visualização dos *clusters* TF-IDF Euclidiano

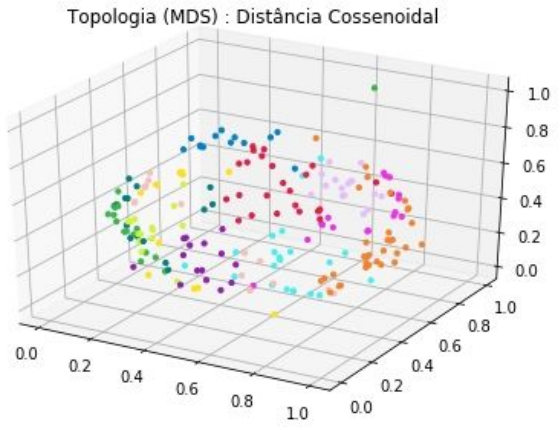

Figura 6. Visualização dos *clusters* TF-IDF Cossenoidal

Após os testes iniciais realizados, os parâmetros ótimos utilizados para o método Doc2Vec foram de 200 épocas, 1000 dimensões e 15000 épocas de inferência. No caso do TF-IDF, também foram utilizadas 1000 dimensões e os demais parâmetros foram os padrões do pacote *sklearn*.

Para determinar o número de *clusters* a serem utilizados pelo algoritmo *K-Means* foi empregada a técnica de *Elbow*.

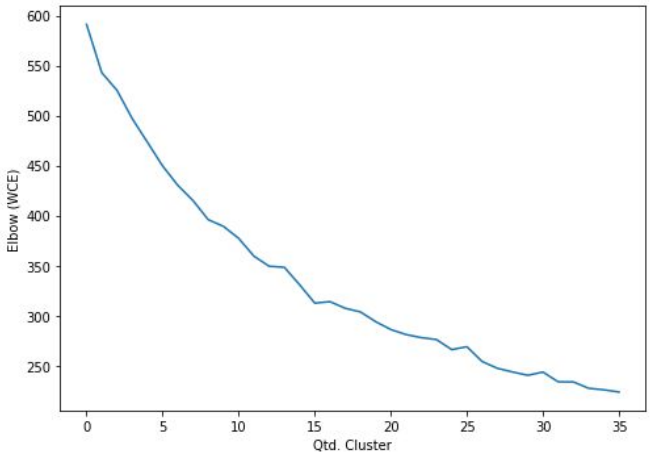

Figura 7. Resultado do método *Elbow*

A quantidade de *clusters* obtido pelo uso da técnica *Elbow* foram de 17 (resultado próximo as 12 classes existentes). Desta forma o algoritmo *K-Means* foi executado com os parâmetros de 17 *clusters* e inicialização aleatória. Além disso utilizou-se de outras duas métricas (*V-Measure*[6] e *Silhouette*[7]) para verificar o desempenho da clusterização e os resultados obtidos foram os seguintes:

	V-Measure	*Silhouette*
Treinamento	73% ± 2	0.188
Teste	83% ± 2	0.151

Tabela 2. Métricas da clusterização

Para o processo de detecção de anomalia ambos algoritmos *K-Means* e *Agglomerative Clustering* foram inicializados com 1 *cluster* em cima da base de 240 perguntas e, posteriormente, classificaram as 100 perguntas que representam as novas interações. Para ambos algoritmos as perguntas passaram pelo mesmo processo de pré processamento e foi utilizada a técnica de TF-IDF com distância cossenoidal (por apresentar melhores resultados na etapa anterior).

Os resultados obtidos, levando-se em consideração a classificação manual das 100 perguntas realizadas anteriormente, foram os seguintes:

		Anomalias	Não anomalias
Kmeans	**Anomalias**	16	34
	Não anomalias	12	38
Acurácia	54%	**Recall**	57%
Agglomerative	**Anomalias**	41	9
	Não anomalias	12	38
Acurácia	79%	**Recall**	77%

Figura 8. Matriz de confusão detecção de anomalias

4. Conclusões

Este trabalho apresentou uma solução para o problema de criação e manutenção das base de dados de *chatbots* através do emprego de técnicas de clusterização e detecção de anomalias não supervisionadas. O objetivo é substituir as etapas manuais destes processos e, com isto, tornar a criação e manutenção de *chatbots* mais rápida e independente de ação humana.

Durante o projeto constatou-se que o processo de clusterização não supervisionado de frases no idioma português apresenta um desafio extra devido à complexidade semântica e sintática do idioma e ao tamanho da dimensionalidade dos vetores que são submetidos ao modelo.

Apesar destas dificuldades, para as base de dados utilizadas, percebeu-se que com pré processamento dos exemplos, com a utilização da técnica de vetorização TF-IDF e a métrica de distância cossenoidal do espaço vetorial, o algoritmo *K-Means* apresentou um resultado considerado satisfatório visto que nenhuma interferência humana foi realizada. O mesmo pode ser dito da fase de detecção de anomalias que obteve acurácia de 79% empregando o algoritmo de *Agglomerative Clustering*.

Com estes resultados pondera-se que um processo híbrido, utilizando uma abordagem semi supervisionada para a pré-filtragem do conteúdo de frases a serem utilizadas, e posteriormente empregando a mão de obra especializada para classificação e/ou revisão em determinadas etapas deste processo para correção de eventuais falhas ocasionadas, pode possivelmente melhorar os resultados obtidos. Consequentemente, o tempo e esforço empregados poderão ser menores que no processo original.

A utilização de demais algoritmos de clusterização não supervisionados, técnicas de vetorização e métricas de distância diferentes das abordadas neste trabalho e também a aplicação do processo aqui sugerido em outras bases de dados (com mais exemplos e temas mais variados) podem contribuir para melhor validação e melhorias do processo automatizado.

Referências

[1] Le, Quoc and Mikolov, Tomas. “Distributed Represenations of Sentences and Documents”. In Proceedings of ICML 2014.

[2] Mikolov, Tomas, Chen, Kai, Corrado, Greg and Dean, Jeffrey. “Efficient Estimation of Word Representations in Vector Space”. In ICLR Workshop Papers, 2013.

[3] Farouk, Mamdouh. “Measuring Sentences Similarity: A Survey.” Indian Journal of Science and Technology 12.25 (2019): 1–11. Crossref. Web.

[4] Gomaa, Wael and Fahmy, Aly. “A Survey of Text Similarity Approaches”. International Journal of Computer Applications, 2013. 68. 10.5120/11638-7118.

[5] Dundar, Murat; Kou, Qiang; Zhang, Baichuan; He, Yicheng and Rajwa, Bartek. “Simplicity of Kmeans Versus Deepness of Deep Learning: A Case of Unsupervised Feature Learning with Limited Data.” IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015.

[6] Rosenberg, Andrew, and Julia Hirschberg. “V-measure: A conditional entropy-based external cluster evaluation measure.” Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL). 200

[7] Rousseeuw, Peter J. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." Journal of computational and applied mathematics 20 (1987): 53-65.