UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Reconhecimento de Entidades Nomeadas em Processamento de Linguagem Natural

Aplicadas no contexto jurídico

Pedro Augusto de Almeida Mattos

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

PEDRO AUGUSTO DE ALMEIDA MATTOS

RECONHECIMENTO DE ENTIDADES NOMEADAS EM PROCESSAMENTO DE LINGUAGEM NATURAL

Aplicadas no contexto jurídico

Goiânia
2024

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): **PEDRO AUGUSTO DE ALMEIDA MATTOS**

Título do trabalho:

RECONHECIMENTO DE ENTIDADES NOMEADAS EM PROCESSAMENTO DE LINGUAGEM NATURAL

Aplicadas no contexto jurídico

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [**X**] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Pedro Augusto De Almeida Mattos**, **Discente**, em 15/02/2024, às 19:21, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 12/09/2024, às 11:07, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4383420** e o código CRC **209D6A03**.

Referência: Processo nº 23070.008396/2024-91 SEI nº 4383420

PEDRO AUGUSTO DE ALMEIDA MATTOS

RECONHECIMENTO DE ENTIDADES NOMEADAS EM PROCESSAMENTO DE LINGUAGEM NATURAL

Aplicadas no contexto jurídico

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.
Orientador: Prof. Dr. Fernando Marques Federson

Goiânia
2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

MATTOS, PEDRO AUGUSTO DE ALMEIDA
RECONHECIMENTO DE ENTIDADES NOMEADAS EM PROCESSAMENTO DE LINGUAGEM NATURAL [manuscrito] : Aplicadas no contexto jurídico / PEDRO AUGUSTO DE ALMEIDA MATTOS. - 2024.
64 f.

Orientador: Prof. Dr. FERNANDO MARQUES FEDERSON.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2024.

1. inteligência artificial. 2. entidades nomeadas. 3. processamento de linguagem natural. I. FEDERSON, FERNANDO MARQUES, orient. II. Título.

CDU 004

PEDRO AUGUSTO DE ALMEIDA MATTOS

RECONHECIMENTO DE ENTIDADES NOMEADAS EM PROCESSAMENTO DE LINGUAGEM NATURAL

Aplicadas no contexto jurídico

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 08 de fevereiro de 2024.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Vinícius Sebba Patto
Coordenador do BIA (INF-UFG)

Prof. Dr. Iwens Gervasio Sene Junior
(INF-UFG)

PEDRO AUGUSTO DE ALMEIDA MATTOS

RECONHECIMENTO DE ENTIDADES NOMEADAS EM PROCESSAMENTO DE LINGUAGEM NATURAL

Aplicadas no contexto jurídico

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Processamento de Linguagem Natural (NLP)**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, entidades nomeadas, processamento de linguagem natural.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Natural Language Processing (NLP)**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: artificial intelligence, named entities, natural language processing.

Goiânia

2024

Minha Jornada

Pedro Augusto de Almeida Mattos

Especialista em: Processamento de Linguagem Natural (NLP)

MINHA JORNADA

Nome:

Especialidade:

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha jornada começou na **Semana 1** com atividades para definir a área de conhecimento da minha especialização. A formação obtida nas Disciplinas Aprendizado de Máquina e Processamento de Linguagem Natural do Bacharelado, além da participação em projetos de pesquisa, foram importantes para a minha escolha. Detalhes sobre essa experiência podem ser encontrados no **Apêndice 1**. Com a leitura de alguns artigos e blogs, pude decidir na **Semana 2** que gostaria de me aprofundar na área de Processamento em Linguagem Natural (NLP). As referências lidas, assim como algumas observações que considero importantes, podem ser obtidas em detalhes no material disponibilizado no **Apêndice 2**.

Nas **Semanas 3 e 4**, meu foco direcionou-se para estudos mais aprofundados de técnicas avançadas em NLP. Realizei uma Revisão Sistemática da Literatura sobre embeddings e técnicas de pré-processamento de texto, identificando métodos relevantes publicados entre 2015 e 2020. Mais detalhes podem ser encontrados no **Apêndice 3**. Simultaneamente, realizei uma pesquisa de frameworks, visando identificar as ferramentas

[1] Dez semanas, entre setembro de 2023 e janeiro de 2024.

mais apropriadas para as principais tarefas de NLP. Essa pesquisa foi realizada antes do início da implementação de qualquer algoritmo, com o objetivo de estabelecer uma base sólida para o desenvolvimento subsequente. Os resultados dessa etapa podem ser encontrados no **Apêndice 4**.

Com uma base sólida em conceitos de Processamento de Linguagem Natural (NLP) e técnicas exploradas, avancei para a etapa prática da minha jornada de pesquisa durante a **Semana 5**, implementando diferentes representações de palavras. Comecei utilizando técnicas básicas, como o método *Bag of Words* (BoW), para descrever textos de um conjunto de dados em inglês. O objetivo era explorar as nuances e impactos dessas representações na tarefa de classificação, considerando a presença ou ausência de pré-processamento. Nessa semana, a teoria se converteu em prática, consolidando a compreensão das complexidades envolvidas nas escolhas de representações de palavras e seus impactos na tarefa de classificação. Mais detalhes sobre o experimento se encontram no **Apêndice 5**.

Durante a **Semana 6**, dei continuidade à implementação e exploração de representações de palavras já estudadas. Além da abordagem inicial com o método *Bag of Words* (BoW), minha pesquisa estendeu-se para outras técnicas, incluindo o *Term Frequency-Inverse Document Frequency* (TF-IDF), *Word to Vector* (Word2Vec) e *Bidirectional Encoder Representations from Transformers* (BERT). O desenvolvimento destas atividades está no **Apêndice 6**. Além disso, decidi direcionar meu foco para o desenvolvimento de um modelo de Reconhecimento de Entidades Nomeadas (NER). Essa decisão foi fundamentada na necessidade de aplicar as técnicas aprendidas em um contexto, considerado por mim, prático e relevante.

Durante a **Semana 7**, meu objetivo era reproduzir o estudo apresentado no artigo *Recognizing Pharmacovigilance Named Entities in Brazilian Portuguese with CoreNLP* e fazer uma comparação com os trabalhos. Este artigo, centrado na área de aplicação da Saúde, serviria como uma base sólida para o desenvolvimento do meu próprio modelo de Reconhecimento de Entidades Nomeadas (NER). No entanto, ao enfrentar desafios técnicos durante o processo, tornou-se evidente que realizar uma comparação seria inviável nas circunstâncias apresentadas. Essa constatação durante a execução do projeto ressalta a complexidade e os desafios de se adaptar metodologias propostas em ambientes de

desenvolvimento diferentes. No **Apêndice 7**, é possível encontrar uma descrição sobre os obstáculos enfrentados.

Essa percepção orientou minha abordagem na **Semana 8**. Diante das dificuldades, optei por mudar minha área de aplicação. Embasado nos conhecimentos adquiridos ao longo da Residência, percebi que essa mudança não representaria um obstáculo, uma vez que os conceitos assimilados em relação a NER e Processamento de Linguagem Natural (NLP) poderiam ser aplicados independentemente do domínio de atuação. Decidi direcionar meus esforços para o campo jurídico. Destaco, nesse contexto, o artigo *LeNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text*, que serviu como alicerce fundamental para conduzir meus resultado subsequentes. O artigo propõe um conjunto de dados em língua portuguesa para o reconhecimento de entidades nomeadas no contexto jurídico. Para avaliar a eficácia desses dados, foi feito um *baseline* utilizando *Long Short-Term Memory* e *Conditional Random Field* (LSTM + CRF). Meu objetivo para a Semana era reproduzir esse trabalho implementando o *baseline*. Ao completar essa etapa, obtive um F1-score geral de 88,06% no conjunto de teste, enquanto o modelo de referência alcançou 92,53%. Além disso, busquei aprimorar o trabalho original através de uma adaptação utilizando a biblioteca Transformers, alcançando um F1-score geral de 88,3%. No **Apêndice 8**, é possível encontrar o artigo juntamente com algumas observações detalhando os resultados obtidos.

Para tentar melhorar o F1-score, ao longo da **Semana 9**, me concentrei na otimização dos hiperparâmetros. Utilizando a biblioteca Optuna, realizei experimentos variando os parâmetros de *learning rates* entre 5e-5, 4e-5, 3e-5 e 2e-5, e os *batch sizes* entre 4, 8 e 16. Os valores de *learning rate* foram escolhidos com base no estudo *How to Fine-Tune BERT for Text Classification?*. Quanto ao *batch size*, as escolhas levaram em consideração as limitações de RAM durante o treinamento, já que valores acima de 16 resultaram em estouro de memória, interrompendo o processo de treinamento. Os melhores parâmetros obtidos foram *learning rate* de 2e-5 e um *batch size* de 8. Após identificar a melhor combinação, prossegui treinando o meu modelo Transformers utilizando essa configuração e o resultado obtido foi 89.7% de F1-score geral. Informações sobre o desenvolvimento das atividades e as referências podem ser encontradas no **Apêndice 9**.

Na última etapa da jornada, durante a **Semana 10**, concentrei-me na otimização final do treinamento do modelo. Implementei o conceito *Low Rank Adaptation* (LoRA) para aprimorar a performance, reduzindo significativamente o tempo de treinamento sem comprometer a qualidade dos resultados. Com o LoRA, ocorreu uma economia de 45 minutos no treinamento do modelo em comparação com o método anterior, e o F1-score geral permaneceu constante em 89,7%, indicando que a eficiência temporal não comprometeu a qualidade das predições do modelo. Os detalhes dessas melhorias foram registrados e documentados no **Apêndice 10**.

Em conclusão, esta jornada ao longo dessas dez Semanas proporcionou muito conhecimento e experiências enriquecedoras. Cada Semana dedicada a explorar, estudar técnicas e implementar modelos contribuíram significativamente para a minha compreensão prática e teórica do tema. Essas dez Semanas representam um alicerce fundamental para o meu crescimento profissional. O caminho para me tornar um especialista é um processo contínuo e estou comprometido a continuar me dedicando a esse percurso de aprendizado constante. Como próximo passo, planejo continuar os meus estudos em Processamento de Linguagem Natural (NLP), aprofundando e aprimorando minhas habilidades, especialmente em *Large Language Models (LLMs)*.

APÊNDICE 1

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 19 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Augusto de Almeida Mattos

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante esse primeiro gate, foram realizadas as seguintes atividades:

- Definição do tema central que será abordado durante a Residência
  - O objetivo deste projeto é a criação de um modelo de ML-NLP que analise as revisões dos pacientes para identificar e classificar os efeitos colaterais dos medicamentos. Para isso pretendo usar os princípios e técnicas de Processamento de Linguagem Natural (NLP). Isso incluirá o pré-processamento de texto, extração de recursos textuais, análise de texto descritivo, integração de dados tabulados e textuais, além da interpretação de resultados, estudos de caso e contextualização.
- Pesquisa de artigos relacionados ao assunto se encontra no link abaixo.
  - Artigos e estudos relacionados à farmacovigilância
- Classificação do trabalho de acordo com os temas da *Conference on Computational Science and Computational Intelligence (CSCI'23)*:
  - Languages and programming techniques for AI
  - Unsupervised and Supervised Learning
  - Aspects of natural language processing
  - Natural Language Processing
  - Feature Engineering

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Estudo de tópicos sobre farmacovigilância (abordagens a serem utilizadas).
- Analisar bases de dados existentes para fechar o escopo do projeto.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Gate 19/10/2023

Uma breve contextualização dos artigos focados na ideia de usar nlp na saúde, mais especificamente aplicada aos efeitos adversos de medicamentos

- Supervised machine learning techniques for predicting drug side effects

Em resumo, o artigo fala da importância da previsão de efeitos colaterais de medicamentos no desenvolvimento e design de drogas. Evidencia que os efeitos colaterais podem variar de questões menores a problemas potencialmente letais, enfatizando a necessidade de garantir que os benefícios de um medicamento superem os efeitos adversos conhecidos.
O texto também destaca a escassez de estudos específicos sobre a previsão de efeitos colaterais de drogas usando aprendizado de máquina supervisionado e argumenta que esse tipo de pesquisa é vital para melhorar a segurança dos medicamentos e acelerar o processo de desenvolvimento de drogas.

- Prediction and evaluation of combination pharmacotherapy using natural language processing

A pesquisa discute os desafios no desenvolvimento de terapias farmacológicas combinadas e destaca a limitação do conhecimento atual sobre mecanismos de doenças.

- Using Machine Learning for Pharmacovigilance

O artigo discute a aplicação do processamento de linguagem natural (NLP) para a análise de conteúdo gerado pelos usuários como uma fonte de evidência complementar eficaz na farmacovigilância, que envolve a monitorização contínua das reações adversas a medicamentos existentes. A revisão sistemática envolveu uma busca abrangente e multidisciplinar em quatro bancos de dados, resultando em 16 publicações relevantes que foram consideradas de média confiabilidade e validade. Essas publicações apresentaram evidências de que o NLP pode ser efetivamente usado para identificar reações adversas a medicamentos em conteúdo textual gerado pelos usuários publicado na internet, sem depender da notificação ativa dos usuários às autoridades. A análise de dados textuais tem o potencial de complementar o sistema tradicional de farmacovigilância, fornecendo uma abordagem mais econômica e eficiente. Além disso, o texto fornece informações sobre a farmacovigilância, os desafios na detecção de reações adversas a medicamentos e a evolução da análise de texto na área da saúde pública.

- Pharmacovigilance through the development of text mining and natural language processing techniques

O estudo discute a importância da farmacovigilância na detecção e prevenção de problemas relacionados a medicamentos, com foco na detecção precoce de eventos adversos a medicamentos (ADEs) e reações adversas a medicamentos (ADRs). A

farmacovigilância envolve uma série de atividades, incluindo a monitorização contínua de medicamentos existentes.

ADEs são efeitos negativos que acontecem enquanto o paciente toma um medicamento, sendo causados diretamente pelo medicamento ou indiretamente por erros na prescrição, dispensação, administração, adesão ou monitorização do medicamento. A detecção precoce de ADEs auxiliaria a segurança do paciente e a redução dos custos de saúde associados a esses eventos.

- A Systematic Review of Natural Language Processing in Healthcare

Este artigo descreve o uso de técnicas de Processamento de Linguagem Natural (NLP) na área da saúde em geral, mais especificamente na organização e análise de informações narrativas em sistemas de saúde eletrônicos.

APÊNDICE 2

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 26 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel, Héber Júnior e Pedro Augusto

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante essa semana e após os encontros da semana passada, compreendemos melhor o que se espera de nós dessa residência. Assim, realizamos a busca por surveys de NLP, tanto no aspecto geral quanto específico da parte de classificação, a fim de ter um histórico de técnicas e métodos da área, e também buscar quais dessas técnicas e métodos são os que possuem melhores resultados atualmente. Os surveys podem ser encontrado em:

- A Survey on Text Classification: From Traditional to Deep Learning (https://arxiv.org/pdf/2008.00364.pdf)

- A Survey on Text Classification Algorithms: From Text to Predictions (https://www.mdpi.com/2078-2489/13/2/83)

- Efficient Methods for Natural Language Processing: A Survey (https://arxiv.org/pdf/2209.00099.pdf)

Também foi montado um repositório compartilhado de artigos entre todos da turma que estão trabalhando com NLP e/ou LLM: https://drive.google.com/drive/folders/11igIGGlTPdB_qAXi71--XOu71Rxv3cWH

Junto com esse estudo, também fizemos um compilado de termos que julgamos ser mais importante para essa etapa inicial da residência, que constitui os fundamentos de NLP. Para isso foi utilizado de base tanto os surveys acima quanto os artigos que foram mencionados na entrega da semana passada. Esse compilado pode ser acessado em: gate 26/10/2023

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Consolidar o aprendizado praticando as abordagens de embeddings
- Busca de surveys e artigos específicos sobre na área de classificação
- Busca por datasets e benchmarks para classificação

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Gate 26/10/2023

Usando os surveys e artigos que foram citados como referência nos termos de entrega da semana passada e essa, separamos os principais tópicos sobre representações de palavras e modelos de NLP.

1. Representação de Palavras: A representação de palavras é fundamental para a classificação de textos, pois transforma o texto em um formato que os modelos de aprendizado de máquina podem entender e, consequentemente, possibilitar seu processamento e realização de diversas tasks, como classificação e sumarização. Existem várias abordagens, incluindo:

a. Bag of Words (BoW): Nessa abordagem, um texto é representado como um conjunto de palavras, ignorando a ordem das palavras. Cada palavra é codificada como um vetor de recursos. As frequências das palavras são contadas e usadas como valores nos vetores.

b. TF-IDF (Term Frequency-Inverse Document Frequency): O TF-IDF atribui valores a palavras com base em sua frequência em um documento específico e sua importância geral em todo o corpus de documentos, o que ajuda a destacar palavras importantes.

c. Word Embeddings Estáticos (word2vec): Além das abordagens tradicionais, é importante mencionar o uso de word embeddings estáticos, como word2vec. Esses

embeddings representam palavras como vetores em um espaço de alta dimensão, capturando relações semânticas e contextualmente relevantes entre palavras. O word2vec é treinado em grandes conjuntos de texto e é uma maneira eficaz de capturar o significado das palavras para uso em tarefas de processamento de linguagem natural.

2. Classificadores Clássicos: Os classificadores clássicos podem ser usados com representações de palavras como BoW ou TF-IDF. Alguns exemplos de classificadores clássicos incluem:

a. Naive Bayes: É um classificador probabilístico que assume independência entre as palavras.

b. Regressão Logística: Uma técnica que modela a probabilidade de uma instância pertencer a uma classe em função das variáveis explicativas (representação de palavras).

c. SVM (Support Vector Machine): mapeia os vetores de palavras para um espaço de alta dimensão e encontra hiperplanos de separação entre classes.

3. Métodos Baseados em Deep Learning: Os métodos baseados em deep learning têm se destacado na classificação de textos, especialmente quando combinados com representações de palavras mais avançadas. Alguns métodos incluem:

a. Redes Neurais Artificiais (MLP): Redes neurais de feedforward, como perceptrons multicamadas (MLP), podem ser usadas para classificação de textos. Eles podem ser alimentados com representações vetoriais de palavras.

b. Redes Neurais Recorrentes (RNN): As RNNs são capazes de lidar com sequências de palavras e podem capturar dependências temporais. No entanto, elas têm dificuldade em lidar com sequências muito longas e sofrem de desvanecimento de gradientes (vanishing gradients).

c. Redes Neurais com Mecanismo de Atenção (Attention): As redes com atenção, como o mecanismo de atenção de seq2seq, são capazes de dar mais peso a certas partes do texto durante a classificação, o que é útil para entender o contexto.

d. Transformers: Os modelos baseados em Transformers revolucionaram o campo do processamento de linguagem natural (NLP) e a classificação de textos. Eles introduziram uma arquitetura de rede neural altamente paralelizável que se destaca em uma variedade de tarefas NLP, incluindo classificação de textos.

4. Arquitetura dos Transformers: A principal inovação dos Transformers é a arquitetura de atenção, que permite que o modelo considere todas as palavras do contexto ao mesmo tempo. Isso é fundamental para a compreensão do contexto e a captura de relações de longo alcance. Cada camada do Transformer contém duas partes principais:

a. Mecanismo de Auto-Atenção (Self-Attention): É a espinha dorsal do Transformer. Ele permite que o modelo avalie a importância de todas as palavras na sequência em relação a uma palavra de entrada específica. Isso é feito por meio de pesos de atenção que indicam a importância relativa de cada palavra para a palavra de referência.

b. Redes de Feedforward (Feedforward Networks): Após a auto-atenção, uma camada de redes neurais de feedforward é aplicada para combinar informações e gerar saídas finais.

Os modelos Transformers são altamente adaptáveis para tarefas de classificação de textos. Para usar um modelo Transformer para classificação, a camada de saída pode ser personalizada para se ajustar ao número de classes do problema de classificação. Geralmente, é adicionada uma camada de saída softmax no topo da arquitetura para calcular as probabilidades de pertencer a cada classe.

Uma das vantagens dos modelos Transformers é a capacidade de pré-treinamento em grandes quantidades de dados e, em seguida, ajustar o modelo para tarefas específicas por

meio do fine-tuning. Para classificação de textos, é possível pré-treinar um modelo Transformer em um grande corpus de texto e, em seguida, ajustá-lo em dados de treinamento de classificação específicos.

APÊNDICE 3

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 9 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Augusto de Almeida Mattos

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Na semana anterior, foi feita uma revisão de estudo sobre as técnicas utilizadas para fazer embeddings. Partindo desse ponto, como complemento, meus estudos para essa semana foram em torno de técnicas de pré-processamento de texto. Pesquisei artigos publicados entre janeiro de 2015 e de 2020 usando IEEE Xplore Digital Library. Também pesquisei nos servidores de pré-impressão (por exemplo, arXiv) por meio do Google Scholar e banco de dados como o Scopus para identificar os métodos mais relevantes.

O pré-processamento de texto é uma parte essencial de qualquer sistema de Processamento de Linguagem Natural (NLP) porque os caracteres, palavras e frases identificados nessa etapa são as unidades fundamentais que são passadas para todas as etapas de processamento subsequentes. Um breve resumo do estudo de Tecnicas de pré-processamento.docx

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Levantamento das principais ferramentas utilizadas nas etapas de desenvolvimento de um modelo, e também quais fundamentos vistos até o momento são empregados em conjunto a essas ferramentas.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Gate 09/11/2023

Breve resumo dos artigos

A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques

O artigo traz a importância do pré-processamento e do processamento de texto no contexto do Text Mining (mineração de texto). Ele destaca que a quantidade de texto gerada diariamente está aumentando drasticamente e que essa grande quantidade de texto, em sua maioria não estruturado, não pode ser facilmente processada e compreendida por computadores. Portanto, são necessárias técnicas e algoritmos eficientes para descobrir padrões úteis nesses dados. O texto mining é a tarefa de extrair informações significativas de texto, com destaque para tarefas fundamentais de mineração de texto, incluindo pré-processamento de texto, classificação e agrupamento. Além disso, o texto menciona que o text mining também é aplicado nos domínios biomédico e de cuidados de saúde.

A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing

Ele cita que o pré-processamento é particularmente importante na etapa inicial das estratégias de recuperação de informações, dado que erros nas técnicas iniciais de pré-processamento podem afetar as etapas posteriores do pipeline de NLP. As técnicas de pré-processamento incluem segmentação de frases, conversão para letras minúsculas, tokenização, remoção de stopwords, remoção de pontuações e lematização.

Preprocessing Techniques for Text Mining - An Overview

É sobre mineração de dados, com foco em mineração de texto, que é o processo de extrair informações úteis de documentos de texto. Ele destaca a importância do pré-processamento de texto, que inclui a remoção de palavras de parada (stop words) e a redução das palavras para seus troncos (stemming), a fim de tornar o texto mais manejável para análise. A mineração de texto é usada em diversas áreas de pesquisa, como processamento de linguagem natural, recuperação de informações, classificação de texto e agrupamento de texto. Além disso, o texto menciona técnicas de extração de recursos, como a técnica TF-IDF. No geral, ele enfatiza a importância do pré-processamento e das técnicas de mineração de texto para obter informações úteis de grandes conjuntos de dados de texto.

As técnicas de pré- processamento de cada um dos artigos gira em torno dessas listadas abaixo.

Segmentação de Sentenças:
Envolve dividir um documento de texto em sentenças individuais. Facilita a identificação de limites de palavras para análises subsequentes.

Conversão para Minúsculas:
Todos os caracteres do texto são transformados em minúsculas. Isso é útil para garantir consistência, já que a diferenciação entre maiúsculas e minúsculas pode levar a resultados distintos.

Tokenização:
Consiste em dividir um texto em unidades menores, chamadas tokens, como palavras ou caracteres. Facilita a filtragem de palavras desnecessárias e prepara o texto para análises mais avançadas.

Identificação de Partes do Discurso (POS Tagging):
Identificação de Partes do Discurso (POS Tagging): Atribui rótulos gramaticais, como substantivos, verbos, adjetivos, etc., a cada palavra em uma frase. Ajuda a compreender a estrutura gramatical do texto.

Remoção de Stopwords:
Envolve eliminar palavras comuns, como "o", "e" e "é", que geralmente não contribuem significativamente para o significado do texto. Isso reduz o ruído nos dados.

Remoção de Pontuações:
Elimina caracteres de pontuação, como vírgulas e pontos de exclamação. Ajuda a simplificar o texto, removendo elementos que geralmente não são informativos para análises.

Stemming:
Reduz palavras à sua forma base ou raiz, removendo sufixos. Isso ajuda a agrupar variações morfológicas da mesma palavra, simplificando a análise.

Lemmatização:
Similar ao stemming, mas produz palavras (lemmas) que têm significado linguístico real. Leva em consideração o contexto e a estrutura gramatical para produzir formas mais legíveis e interpretáveis das palavras.

APÊNDICE 4

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 16 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel, Héber Júnior e Pedro Augusto

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Cada integrante do grupo vai para uma sub-área diferente, porém todas estão dentro da task de classificação de texto. Assim, o geral da atividade realizada foi igual para todos os integrantes, mas alguns detalhes foram específicos para cada um.
Como mencionado no planejamento anterior, durante essa semana desenvolvemos um levantamento com os principais frameworks para realização da task de classificação de textos. O objetivo desse levantamento é que, como especialistas, ao se deparar com algum problema dessa área de NLP já sejamos capazes de, antes mesmo de começar a parte de programação, já identificar quais frameworks são mais adequados ao problema em questão. Esse levantamento pode ser encontrado em: Entrega - Frameworks
Vale ressaltar que, embora tenhamos realizado as atividades desta semana em conjunto, o escopo que estou abordando durante a residência é a tarefa de classificação. Portanto, algumas das ferramentas listadas no documento acima são específicas para essa subárea de NLP. No entanto, boa parte das ferramentas listadas é utilizada tanto na análise de sentimentos quanto em outras atividades de classificação de texto. Por isso, optamos por realizá-lo em grupo, visto que, quando chegar a hora de realizar a aplicação prática, todos recorreremos a esse levantamento para verificar as ferramentas que melhor se encaixam em nosso problema.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Próxima semana:

- Realizar a busca por datasets para classificação na área da saúde.
- Com o conjunto de dados em mãos, segue-se a ideia do pipeline na implementação prática com base no levantamento realizado nesta semana.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Gate 16/11/2023

Esse documento contém um levantamento dos principais frameworks para a tarefa de classificação de textos. Foram consideradas as etapas listadas abaixo de desenvolvimento de um modelo de classificação de textos. Dependendo da implementação pode ser que algumas etapas não sejam adotadas ou que outras não listadas sejam, porém de forma geral as etapas a serem consideradas são as descritas abaixo:

- Importação dos dados
- Pré-processamento / feature engineering
- Feature extraction
- Treino do modelo
- Obtenção de métricas
- Otimização de hiperparâmetros

Vale lembrar que todos os frameworks comentados abaixo são da linguagem Python.

Pandas: Oferece estruturas de dados poderosas, como DataFrames, que simplificam a manipulação e transformação de conjuntos de dados. Especificamente, no contexto de processamento de linguagem natural (NLP), o Pandas é frequentemente utilizado para a importação, pré-processamento e manipulação de dados textuais. Suas funcionalidades incluem operações eficientes para lidar com texto, como remoção de duplicatas, seleção de features e engenharia de características (feature engineering). Apesar de ser uma ferramenta de alto desempenho em tarefas de manipulação de dados, é importante notar que para volumes muito grandes de dados, o Pandas pode exigir uma quantidade considerável de memória. https://pandas.pydata.org/

PySpark: Desenvolvido como uma interface Python para o Apache Spark, o PySpark é uma biblioteca que oferece poderosas capacidades de processamento distribuído. Projetado para lidar com grandes volumes de dados, é especialmente valioso em ambientes de big data, onde o Spark é frequentemente utilizado. No contexto do processamento de linguagem natural

(NLP), o PySpark é aplicado para tarefas de importação de dados, pré-processamento e análise em larga escala. Sua arquitetura distribuída permite a execução eficiente de operações em clusters, tornando-o adequado para lidar com conjuntos de dados extensos e complexos em projetos de NLP em grande escala. Possui funcionalidade similares ao Pandas, porém ao contrário dele, que é mais eficiente para operações em conjuntos de dados que cabem na memória, o PySpark é projetado para processamento distribuído, tornando-o uma escolha poderosa em ambientes onde o dimensionamento é essencial. https://spark.apache.org/docs/latest/api/python/index.html

NLTK (Natural Language Toolkit): Capaz de auxiliar em diversas tarefas de pré-processamento textual, oferece uma ampla variedade de módulos para tokenização, lematização, análise sintática, entre outras operações. Sua versatilidade é evidente na capacidade de lidar com uma variedade de tarefas, desde simples manipulações de texto até análises linguísticas mais complexas. Além disso, o NLTK inclui recursos como corpora e modelos pré-treinados, facilitando a implementação de soluções em projetos de NLP. https://www.nltk.org/

VADER: Valence Aware Dictionary and sEntiment Reasoner (VADER) é uma biblioteca especificamente projetada para análise de sentimento em mídias sociais e inclui uma abordagem baseada em léxico (lexicon based) que é ajustada para a linguagem das mídias sociais. Inclui um léxico de sentimento pré-construído com medidas de intensidade para sentimento positivo e negativo, e incorpora regras para lidar com intensificadores de sentimento, emojis e outros recursos específicos que, via de regra, são utilizados em mídias sociais. https://github.com/cjhutto/vaderSentiment. Possui uma adaptação para português: https://github.com/rafjaa/LeIA

SpaCy: Oferece uma ampla gama de funcionalidades, desde tokenização e lematização até análise sintática e reconhecimento de entidades nomeadas. No âmbito do processamento de linguagem natural, o SpaCy é frequentemente utilizado para realizar diversas etapas do pré-processamento textual. Ele se destaca por sua capacidade de processar

grandes volumes de texto de forma rápida e eficiente. Além disso, o SpaCy possui modelos pré-treinados para várias línguas, o que facilita a incorporação de recursos linguísticos em projetos de NLP. Uma característica notável do SpaCy é a sua extensibilidade, permitindo que os usuários adicionem componentes personalizados para atender a requisitos específicos. Com documentação abrangente e uma comunidade ativa, o SpaCy é uma escolha popular em projetos que envolvem análise de texto e processamento de linguagem natural. https://spacy.io/

TextBlob: É uma biblioteca em Python que oferece uma gama de ferramentas para tarefas de pré-processamento e análise de texto. Embora possua menos recursos em comparação com outros frameworks, é mais recomendado para projetos que buscam soluções mais simples e eficazes. Facilita tarefas comuns de NLP, como tokenização, lematização e análise de sentimentos, possuindo uma interface mais simples que permite a extração rápida de informações relevantes de texto, enquanto a funcionalidade de análise de sentimentos incorporada oferece uma solução direta para avaliar a polaridade e subjetividade de expressões. Embora não seja tão extensivo quanto alguns frameworks especializados em NLP, o TextBlob é uma escolha prática para projetos nos quais a facilidade de implementação é valorizada. https://textblob.readthedocs.io/en/dev/

Scikit-learn: É uma ferramenta essencial para tarefas de aprendizado de máquina em Python, destacando-se por sua robustez e variedade de algoritmos para classificação, regressão, clustering e, especialmente, e pelo suporte sólido às tarefas de Processamento de Linguagem Natural (NLP). Ao integrar funcionalidades avançadas, o Scikit-learn oferece não apenas algoritmos de aprendizado de máquina, mas também ferramentas essenciais para pré-processamento de dados textuais. O Scikit-learn se destaca por sua capacidade de realizar extração de características, incluindo técnicas como Bag-of-Words (BoW) e Term Frequency-Inverse Document Frequency (TF-IDF). Além disso, o framework facilita a implementação de pipelines completos para tarefas complexas de NLP, desde a tokenização até a criação de modelos preditivos, e possui um design modular e documentação abrangente.

Também possui funcionalidades para avaliar o desempenho de modelos, como F1 e acurácia. https://scikit-learn.org/stable/

FastText: É uma biblioteca que se destaca pela capacidade de treinar modelos de embeddings de palavras e realizar classificação de texto de forma rápida e eficaz. Desenvolvido pelo Facebook, este framework aprende representações vetoriais de palavras e também leva em consideração a estrutura de subpalavra (subword) das palavras. É projetado para lidar com grandes conjuntos de dados textuais e oferece uma interface fácil de usar para tarefas comuns de NLP, como classificação de texto e análise de sentimentos. Além disso, sua funcionalidade de embeddings de palavras pré-treinados permite incorporar facilmente conhecimento linguístico em seus modelos. Dessa forma, o FastText se destaca como uma escolha sólida para projetos de NLP que demandam velocidade e desempenho. https://fasttext.cc/

Gensim: É especialmente destacado por sua eficiência na modelagem de tópicos e na criação de representações semânticas de documentos. Projetado para lidar com grandes conjuntos de dados textuais, o Gensim oferece uma implementação eficaz de algoritmos de modelagem de tópicos, como o Latent Dirichlet Allocation (LDA), permitindo a descoberta de padrões latentes em grandes coleções de documentos. Uma característica distintiva do Gensim é sua capacidade de treinar modelos de embeddings de palavras, como o Word2Vec e Glove. Além disso, o Gensim oferece funcionalidades para similaridade de documentos, indexação eficiente e manipulação de grandes quantidades de texto de maneira escalável.Ou seja, é um framework recomendado para fazer o uso de embeddings estáticos, como Word2Vec. https://radimrehurek.com/gensim/

Hugging Face: Reconhecido como um hub central para inovações em processamento de linguagem natural (NLP),também é uma plataforma que oferece uma ampla gama de modelos pré-treinados, datasets e ferramentas para tarefas diversas em NLP. Destaca-se por sua comunidade ativa e pela capacidade de compartilhar, descobrir e implementar modelos de última geração. Uma de suas características marcantes é a biblioteca Transformers, que

facilita o acesso e o uso de uma variedade de modelos pré-treinados de última geração, como BERT, GPT-3 e muitos outros. Essa abordagem simplifica significativamente a implementação de modelos de NLP avançados, estimulando a pesquisa e o desenvolvimento em larga escala. Assim, é a principal ferramenta para fazer uso de modelos baseados em Transformers, além de possuir funções para treino, fine-tunning e avaliação dos modelos. https://huggingface.co/

TensorFlow: Como uma das principais bibliotecas para aprendizado de máquina e processamento de linguagem natural (NLP), o TensorFlow se destaca por sua versatilidade e eficiência computacional. Desenvolvido pela Google, o TensorFlow oferece uma infraestrutura robusta para a criação e treinamento de modelos de aprendizado profundo, sendo amplamente adotado em projetos de NLP. Para tarefas específicas de NLP, o TensorFlow disponibiliza módulos e ferramentas, incluindo o TensorFlow Text, que oferece funcionalidades avançadas para processamento de texto, como tokenização, embeddings de palavras e camadas especializadas para tarefas como classificação de texto e tradução. A integração natural do TensorFlow com unidades de processamento gráfico (GPU) e suas capacidades de computação distribuída fazem dele uma escolha poderosa para treinamento de modelos em larga escala. Além disso, a comunidade ativa e a extensa documentação do TensorFlow contribuem para sua popularidade entre pesquisadores e desenvolvedores de NLP. https://www.tensorflow.org/?hl=pt-br

PyTorch: Similar ao TensorFlow, destaca-se como um dos principais frameworks para aprendizado de máquina e, em particular, para processamento de linguagem natural (NLP), o PyTorch é reconhecido por sua flexibilidade e interface intuitiva. Desenvolvido pelo Facebook, o PyTorch oferece uma abordagem dinâmica e orientada para o usuário, o que facilita a construção e experimentação rápida com modelos complexos. Para tarefas específicas de NLP, o PyTorch conta com o pacote torchtext, que simplifica o pré-processamento de dados textuais e fornece utilitários para carregamento eficiente de conjuntos de dados. Além disso, o PyTorch é a escolha preferida para muitas pesquisas e implementações de modelos de linguagem, especialmente em contextos nos quais a

exploração de arquiteturas personalizadas é crucial. Sua integração natural com a computação em GPU e o foco na experiência do usuário fazem dele uma ferramenta valiosa para projetos que demandam flexibilidade e eficácia no desenvolvimento de modelos de linguagem. https://pytorch.org/

MXNet: Destacando-se como uma biblioteca de aprendizado profundo escalável e eficiente, é especialmente reconhecido por sua arquitetura flexível e suporte eficaz para treinamento distribuído. Desenvolvido pela Apache Software Foundation, oferece uma plataforma robusta para projetos de processamento de linguagem natural (NLP) e outras tarefas de aprendizado de máquina. Com interfaces para Python e outras linguagens, o MXNet é acessível e versátil, sendo possível ao usuário montar suas próprias redes neurais. Além disso, é conhecido por sua eficiência em treinamento de modelos em ambientes distribuídos, tornando-o adequado para lidar com grandes volumes de dados. Seja para experimentação em pequena escala ou implementações em larga escala, o MXNet oferece uma variedade de recursos para modelagem e treinamento de redes neurais em projetos que envolvem NLP e outras tarefas complexas de aprendizado de máquina. https://mxnet.apache.org/

FastAI: Reconhecido por tornar o aprendizado profundo mais acessível, o FastAI é uma biblioteca construída sobre o PyTorch, que fornece uma interface de alto nível que simplifica o processo de criação e treinamento de modelos complexos. O FastAI oferece abstrações poderosas para tarefas comuns em aprendizado profundo, incluindo processamento de linguagem natural (NLP). Para tarefas específicas de NLP, ele conta com o módulo fastai.text, que simplifica o pré-processamento de texto, o carregamento de dados e a criação de modelos de linguagem. Além disso, a biblioteca inclui funcionalidades avançadas, como transferência de aprendizado e métodos de treinamento eficazes, que são especialmente úteis para experimentação rápida e desenvolvimento de modelos de NLP com desempenho superior. https://github.com/fastai/fastai

Ray Tune: Especializado em otimização de hiperparâmetros, é uma biblioteca eficiente para busca e ajuste sistemático de configurações ideais de modelos. Desenvolvido

sobre o ecossistema Ray, oferece uma interface intuitiva para experimentação escalável e distribuída. Ideal para projetos de processamento de linguagem natural (NLP) e aprendizado de máquina. Além disso, simplifica a seleção de hiperparâmetros, acelerando a descoberta de configurações otimizadas e melhorando o desempenho dos modelos. https://docs.ray.io/en/latest/tune/

Optuna: Similar ao Ray Tune, é focado em otimização de hiperparâmetros eficiente e automática, e oferece uma abordagem flexível para encontrar as melhores configurações de modelos. Faz uso de algoritmos de busca eficazes para explorar o espaço de hiperparâmetros, acelerando a descoberta de configurações otimizadas, além de possuir suporte para integração em diversas bibliotecas de aprendizado de máquina, como TensorFlow, PyTorch e Scikit-learn. https://optuna.readthedocs.io/en/stable/

WandB (Weights & Biases): Reconhecido como uma plataforma abrangente para rastreamento, visualização e colaboração em projetos de aprendizado de máquina, oferece uma solução completa para experimentação e monitoramento de modelos. Com suporte para várias bibliotecas, como TensorFlow e PyTorch, ele simplifica a análise e compartilhamento de resultados. Para projetos de processamento de linguagem natural (NLP) e outras tarefas, o WandB permite o registro de métricas, gráficos interativos e visualização de embeddings, fornecendo uma compreensão aprofundada do desempenho do modelo. Além disso, sua integração com fluxos de trabalho de aprendizado profundo facilita a compreensão e otimização contínua dos modelos. Ou seja, permite uma análise detalhada e compartilhamento transparente de experimentos. https://wandb.ai/

APÊNDICE 5

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 23 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Augusto de Almeida Mattos

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Meu objetivo principal para essa semana foi utilizar as representações de palavras já estudadas gate 26/10/2023 . Para isso, escolhi um conjunto de dados em inglês com a tarefa de inferir a classe à qual cada texto pertence, utilizando a representação gerada. Comecei com as representações de palavras básicas, como o método Bag of Words (BoW), para descrever o texto do meu conjunto de dados e categorizar seu conteúdo.

O dataset possui 5 classes (negócios, entretenimento, política, esporte, tecnologia). Ao longo desta e da próxima semana, estou investindo tempo na análise das representações de palavras, tanto com quanto sem pré-processamento, e utilizando-as como entrada para um classificador. A intenção é observar de perto o impacto dessas abordagens nos resultados obtidos.
Pretendo finalizar o Colab durante a próxima semana, consolidando os resultados obtidos até então.
Classificação de notícias - BBC

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para o planejamento da próxima semana, o objetivo é concluir o Colab, colocando as principais formas de representação que foram estudadas. Além disso, vou listar as principais metodologias que serão empregadas daqui para frente.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Gate 23/11/2023

Atividade: Classificação de Texto em Categorias de Notícias

O conjunto de dados da BBC News em inglês consiste em artigos de notícias distribuídos em diferentes categorias. A atividade proposta é a classificação desses artigos em suas respectivas categorias usando técnicas de aprendizado de máquina para processamento de linguagem natural (NLP).

- A ideia central é colocar em prática os conhecimentos gerados no decorrer da disciplina, aplicando-os a situações do mundo real para consolidar a compreensão teórica adquirida.
- A atividade consistiu em utilizar o mesmo classificador em diversas representações dos dados buscando entender como cada uma dessas representações influencia o processo de classificação nesse cenário específico.

O colab Classificação de notícias - BBC contém o desenvolvimento da atividade.

APÊNDICE 6

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 30 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Augusto de Almeida Mattos

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante essa semana, finalizei a implementação e exploração de representações de palavras estudadas a partir do dia gate 26/10/2023. Escolhi como cenário um conjunto de dados em inglês, desafiando-me a inferir a classe de cada texto por meio das representações geradas. Além da abordagem inicial com o método Bag of Words (BoW), minha pesquisa estendeu-se para outras técnicas, incluindo o TF-IDF, Word2Vec e BERT. Foi utilizado o mesmo classificador para todas as representações com o intuito de ver como cada técnica impacta o desempenho do classificador, promovendo uma compreensão mais aprofundada das nuances associadas a cada método.
A descrição da atividade realizada nessas duas últimas semanas juntamente com o Colab utilizado está no documento: Semana 23/11/2023 - 30/11/2023

Além disso, realizei a definição do escopo que abordarei até o final da residência.
Desenvolvimento de um modelo NER para farmacovigilância.
A problemática abordada nesse contexto está relacionada à necessidade de extrair informações relevantes sobre os efeitos colaterais de medicamentos a partir de diversas fontes textuais. Essas fontes são importantes porque podem auxiliar na detecção de eventos adversos não previstos para um determinado medicamento. Meu objetivo central é a criação de um modelo de Reconhecimento de Entidades Nomeadas. Este modelo visa identificar entidades relevantes para a problemática levantada (melhor explicadas na Formalização do objetivo).

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Planejo seguir e implementar a metodologia do artigo citado na formalização do objetivo e finalizar a leitura de outro artigo (BioBERTpt) que pretendo também usar como base.
Fundamentando-me em trabalhos anteriores, quero replicar e adaptar metodologias que tenham demonstrado eficácia nesse contexto.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Neste gate, o Professor Aldo André Díaz Salazar esteve na banca avaliadora substituindo a

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

Professora Luana.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Em análise!

Gate 30/11/2023

Atividade: Classificação de Texto em Categorias de Notícias

O conjunto de dados da BBC News em inglês consiste em artigos de notícias distribuídos em diferentes categorias. A atividade proposta é a classificação desses artigos em suas respectivas categorias usando técnicas de aprendizado de máquina para processamento de linguagem natural (NLP).

A ideia central é colocar em prática os conhecimentos gerados no decorrer da disciplina, aplicando-os a situações do mundo real para consolidar a compreensão teórica adquirida.

A atividade consistiu em utilizar o mesmo classificador em diversas representações dos dados buscando entender como cada uma dessas representações influencia o processo de classificação nesse cenário específico.

Link do colab com os experimentos:
Classificação de notícias - BBC

Efeitos adversos de medicamentos

A problemática abordada nesse contexto está relacionada à necessidade de extrair informações relevantes sobre os efeitos colaterais de medicamentos a partir de diversas fontes textuais. Tais fontes são importantes porque podem auxiliar na detecção de eventos adversos não previstos para um determinado medicamento.

Para extrair informações importantes automaticamente, é necessário identificar com precisão entidades específicas no texto, como medicamento, sintoma e reação adversa. Essa identificação acurada permite estabelecer conexões e aprimorar o monitoramento, ampliando a capacidade de detectar eventos não previstos.

O meu objetivo é criar um modelo NER capaz de identificar as entidades relevantes para a problemática levantada replicando a metodologia adotada neste trabalho:
Recognizing pharmacovigilance named entities in BrazilianPortuguese with CoreNLP

A abordagem proposta busca otimizar o processo de farmacovigilância, tornando-o mais eficiente e preciso através da automação na extração e análise de informações provenientes de diversas fontes textuais. Este aprimoramento contribui significativamente para fortalecer a segurança dos medicamentos, ao identificar precocemente potenciais riscos à saúde associados ao seu uso.

APÊNDICE 7

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 7 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Augusto de Almeida Mattos

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

A ideia inicial era reproduzir o estudo do artigo Recognizing pharmacovigilance named entities in BrazilianPortuguese with CoreNLP para essa semana Formalização do objetivo. O CoreNLP é feito em java, mas conforme mencionado no site oficial, é possível usar um pacote Python chamado Stanza para treinar os modelos. Tive muitos problemas até então, principalmente relacionado a variáveis de ambiente, visto que, estou utilizando o windows.

Atualmente estou conseguindo treinar o modelo NER seguindo o pipeline da biblioteca, mas por padrão os embeddings utilizados são em inglês. É citado (a referência está na Entrega 07_12_2023) que podemos obter embeddings de palavras pré-treinados de diversas línguas no fasttext.cc e, usando um código que eles disponibilizam na biblioteca, converter esses embedding para um formato aceitável e continuar com a representação escolhida. Porém, a representação não está sendo reconhecida e o modelo não é treinado.

Mesmo se ignorar esse ponto e continuar treinando com a representação em inglês, não consigo utilizar o modelo ner para fazer inferência. Além disso, meu objetivo final era treinar um modelo com embeddings diferentes do word2vec e conseguir comparar os trabalhos. Alterando um parâmetro durante o treinamento padrão do modelo ner é possível usar representações BERT da hugginface. Porém, ao concluir o treinamento, também não consigo usá-lo para inferência

Documento com as referência sobre o que foi citado juntamente com um colab de exemplo demonstrando os problemas: Entrega 07_12_2023

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Ao invés de reproduzir o artigo com o stanza, quero utilizar outra maneira para conseguir treinar o modelo ner.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Neste gate, o Professor Aldo André Díaz Salazar esteve na banca avaliadora substituindo a Professora Luana.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Em análise!

Gate 07/12/2023

Documento com o resumo das atividades realizadas na semana

O CoreNLP é feito em java, mas conforme mencionado no site oficial https://stanfordnlp.github.io/CoreNLP/other-languages.html é possível usar um pacote Python chamado Stanza para treinar os modelos.
Atualmente estou conseguindo treinar o modelo ner seguindo o pipeline da biblioteca, mas por padrão os embeddings utilizados são em inglês. É citado no https://stanfordnlp.github.io/stanza/new_language_ner.html#word-vectors que podemos obter embeddings de palavras pré-treinados de diversas línguas no fasttext.cc e, usando um código que eles disponibilizam na biblioteca, converter esses embedding para um formato aceitável e continuar com a representação escolhida. Porém, a representação não está sendo reconhecida e o modelo não é treinado.

Mesmo se ignorar esse ponto e continuar treinando com a representação em inglês, não consigo utilizar o modelo ner para fazer inferência. Além disso, meu objetivo final era treinar um modelo com embeddings diferentes do word2vec e conseguir comparar os trabalhos. Alterando um parâmetro durante o treinamento padrão do modelo ner é possível usar representações BERT da hugginface. Porém, ao concluir o treinamento, também não consigo usá-lo para inferência.

Como mencionado, tive problemas para implementar o modelo NER com o stanza. Colab com um toy dataset para demonstrar o problema relacionado com o treino do modelo ner e o embedding em português:
Modelo NER - Stanza

GitHub com código base de como começar a usar o stanza para treinar o modelo ner
https://github.com/stanfordnlp/stanza-train.git

APÊNDICE 8

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 14 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Augusto de Almeida Mattos

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Na última semana, tive problemas durante a implementação do meu projeto. Diante das dificuldades, optei por redirecionar minha área de aplicação. Embasado nos conhecimentos adquiridos ao longo da minha residência, percebi que essa mudança não representaria um obstáculo, uma vez que os conceitos assimilados em relação a Named Entity Recognition (NER) e Processamento de Linguagem Natural (PLN) poderiam ser aplicados independentemente do domínio de atuação. Decidi direcionar meus esforços para o campo jurídico, tomando como referência o artigo LerNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text.

O artigo propõe um conjunto de dados em língua portuguesa para o reconhecimento de entidades nomeadas no contexto jurídico. Para avaliar a eficácia desses dados, foi feito um baseline utilizando (LSTM + CRF). Meu objetivo para a semana era reproduzir esse trabalho implementando o baseline. Ao completar essa etapa, obtive um F1-score geral de 88,06% no conjunto de teste, enquanto o modelo de referência alcançou 92,53%. Além disso, busquei aprimorar o trabalho original através de uma adaptação utilizando a biblioteca Transformers, alcançando um F1-score geral de 97,87% , superando significativamente o baseline citado no artigo.

Informações sobre o desenvolvimento da atividade e as referências podem ser encontradas no documento Gate 14/12/2023.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Otimização do modelo transformers com o objetivo de melhorar f1-score e/ou diminuir o tempo de treinamento e inferência

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Gate 14/12/2023

DESCRIÇÃO DA SEMANA

Nas últimas semanas, tive problemas durante a implementação do meu projeto. Diante das dificuldades, optei por redirecionar minha área de aplicação. Embasado nos conhecimentos adquiridos ao longo da minha residência, percebi que essa mudança não representaria um obstáculo, uma vez que os conceitos assimilados em relação a Named Entity Recognition (NER) e Processamento de Linguagem Natural (PLN) poderiam ser aplicados independentemente do domínio de atuação.

Decidi direcionar meus esforços para o campo jurídico, tomando como referência o artigo LerNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text [1]. Minha abordagem consistiu em reproduzir o baseline proposto por eles, utilizando a arquitetura LSTM-CRF. Além disso, elaborei um estudo próprio, pautado na implementação de modelos baseados em transformers.

O artigo propõe um conjunto de dados em língua portuguesa para o reconhecimento de entidades nomeadas, composto integralmente por documentos legais anotados manualmente. Adicionalmente, introduz duas novas categorias, "LEGISLAÇÃO" (para entidades nomeadas relacionadas a leis) e "JURISPRUDÊNCIA" (para entidades nomeadas referentes a casos legais), visando aprimorar a extração de conhecimento jurídico.

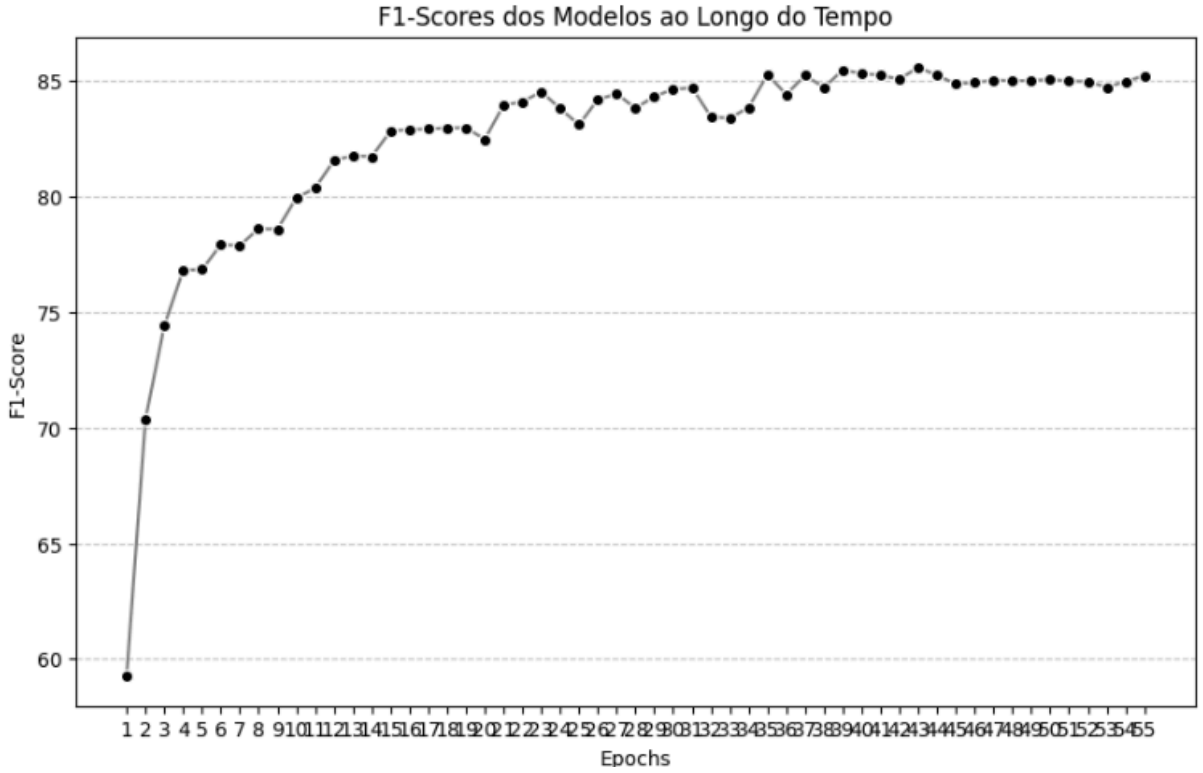
O modelo referenciado no artigo, juntamente com a descrição da arquitetura utilizada, encontra-se disponível no Github sequence tagging [2]. Meu objetivo para a semana em questão era reproduzir esse trabalho e estabelecer o baseline.

OBS: Vale ressaltar que utilizei os mesmos hiperparâmetros usados no trabalho de referência para reproduzir o baseline.

Table 4. Model hyper-parameter values.

Hyper-parameter	Value
Word embedding dimension	300
Character embedding dimension	50
Number of epochs	55
Dropout rate	0.5
Batch size	10
Optimizer	SGD
Learning rate	0.015
Learning rate decay	0.95
Gradient clipping threshold	5
First LSTM layer hidden units	25
Second LSTM layer hidden units	100

F1-Score durante treinamento



No conjunto de teste o modelo descrito no artigo alcançou 92.53% de f1-score geral. Por sua vez, minha reprodução alcançou 88.06 % f1-score demorando cerca de 3 horas para finalizar o treinamento.
A minha adaptação desse trabalho pode ser encontrada no Baseline reproduzido:

Além disso, realizei uma adaptação do trabalho original utilizando a biblioteca Transformers, alcançando um F1-score geral de 97,87% , superando o baseline citado no artigo. Na reconfiguração do modelo LeNer com transformers, o processo envolveu a tokenização dos textos, incorporando ao dataset as características essenciais de input_ids, token_type_ids e attention_mask. Essas características são necessárias para a arquitetura Transformers. Em seguida, desenvolvi uma função para estabelecer as métricas a serem utilizadas, configurei os hiperparâmetros usando TrainingArguments, defini os parâmetros para a função Trainer e, por fim, executei o ajuste fino do modelo Bertimbau [3] (modelo BERT pré-treinado para português brasileiro) para o conjunto de dados. No desfecho desse processo, a inferência foi realizada no conjunto de teste, uma vez que as métricas durante o treinamento foram avaliadas com base no conjunto de validação. O tempo de treinamento para 5 épocas foi de aproximadamente 1 hora e 55 minutos. Mais detalhes sobre o treinamento podem ser vistos no LeNer - Reprodução com transformers.

Referências

[1] LUZ DE ARAUJO, P. H. et al. LeNER-Br: a dataset for named entity recognition in Brazilian legal text. In: Computational Processing of the Portuguese Language: 13th International Conference, PROPOR 2018, Canela, Brazil, September 24–26, 2018, Proceedings 13. Springer International Publishing, 2018. p. 313-323.

[2] Genthial, G.: Sequence tagging - named entity recognition with Tensorflow. GitHub repository https://github.com/guillaumegenthial/sequence_tagging (2017)

[3] SOUZA, Fábio; NOGUEIRA, Rodrigo; LOTUFO, Roberto. BERTimbau: pretrained BERT models for Brazilian Portuguese. In: Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I 9. Springer International Publishing, 2020. p. 403-417.

APÊNDICE 9

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 21 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Augusto de Almeida Mattos

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Na semana anterior, compartilhei meu progresso no gate 8, no qual comparei o desempenho do meu modelo BERT com o baseline (LSTM + CRF), uma arquitetura destacada no artigo "LerNER-Br: a Dataset for Named Entity Recognition in Brazilian Legal Text" utilizado como referência. No baseline que reproduzi, obtive um F1-score geral de 88,06%, enquanto meu modelo transformers alcançou um F1-score de 97,87%. No entanto, posteriormente percebi um equívoco durante a transformação dos dados para o formato BIO ao treinar o modelo transformers, resultando na duplicação dos conjuntos de treino, teste e validação. Isso causou um overfitting no modelo, inflando artificialmente o F1-score geral. Ao corrigir essa questão, o score do modelo reduziu para 88,3%, superando a minha reprodução do baseline, mas ainda ficando aquém do score mencionado no artigo, que atingiu 92,53%.

Para tentar minimizar a diferença de f1-scores, ao longo da semana, me concentrei na otimização de hiperparâmetros. Utilizando a biblioteca Optuna, realizei experimentos variando os parâmetros de learning rates entre 5e-5, 4e-5, 3e-5 e 2e-5, e os batch sizes entre 4, 8 e 16. Os valores de learning rate foram escolhidos com base no estudo "How to Fine-Tune BERT for Text Classification?". Quanto ao batch size, a escolha foi empírica, levando em consideração as limitações de RAM durante o treinamento, já que valores acima de 16 resultavam em estouro de memória, interrompendo o processo de treinamento. Os melhores parâmetros foram learning rate de 2e-5 e um batch size de 8. Essa configuração demonstrou ser a mais otimizada.
Após identificar a melhor combinação, prossegui treinando o meu modelo transformers utilizando essa configuração, o resultado obtido foi 89.7%de f1-score geral.

Informações sobre o desenvolvimento da atividade e as referências podem ser encontradas no documento
Gate 21/12/2023

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Otimização do modelo transformers com o objetivo de melhorar f1-score e/ou diminuir o tempo de treinamento e inferência

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Gate 21/12/2023

Para melhorar a pontuação, ao longo da semana, me concentrei na otimização de hiperparâmetros. Utilizando a biblioteca Optuna, realizei experimentos variando os parâmetros de learning rates, e os batch sizes. Os valores de learning rate foram escolhidos com base no estudo "How to Fine-Tune BERT for Text Classification? [1]. Quanto ao batch size, a escolha foi empírica, levando em consideração as limitações de RAM durante o treinamento, já que valores acima de 16 resultavam em estouro de memória, interrompendo o processo de treinamento. Os melhores parâmetros foram learning rate de 2e-5 e um batch size de 8. Essa configuração demonstrou ser a mais otimizada.

Realizei uma adaptação no meu treinamento, incorporando o Optuna para fazer 30 trials, onde cada trial representa uma única execução da função de otimização. Essas execuções consistiram em uma seleção aleatória das combinações de learning rates (5e-5, 4e-5, 3e-5 e 2e-5) e batch sizes (4, 8 e 16). Cada trial compreendeu 4 épocas, e para determinar as melhores combinações de hiperparâmetros, escolhi o conjunto que alcançou o maior F1-score geral entre as execuções da função de otimização.
Após identificar a melhor combinação de hiperparâmetros dentre as 30 trials, prossegui treinando o meu modelo transformers utilizando essa configuração. O resultado melhorou, alcançando um F1-score geral de 89.7%, superando a performance anterior que de 88.3% de F1-score geral.

Vale ressaltar que para conseguir fazer a otimização com sucesso e em um tempo viável, utilizei a GPU V100 do Colab Pro. Os experimentos feitos estão nos colabs abaixo:
LeNer - Reprodução com transformers
LeNer - Otimização

Referências

[1] SUN, Chi et al. How to fine-tune bert for text classification?. In: Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18. Springer International Publishing, 2019. p. 194-206.

APÊNDICE 10

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 11 de jan. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Augusto de Almeida Mattos

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Na última semana, trabalhei em cima de melhorar a performance do meu modelo. Para isso utilizei o loRA (low rank adaptation). Utilizei como referência o modelo otimizado treinado na entrega anterior, para comparar o desempenho. Com a implementação do LoRA o tempo de treinamento foi reduzido, alcançando uma economia de 45 minutos em comparação com o método anterior. Além disso, o F1_score permaneceu constante com 89.7%. O LoRA resultou em uma modificação mais eficiente de parâmetros, proporcionando não apenas ganhos temporais, mas também uma economia de recursos computacionais.
Nessa recente experiência de aprimoramento de modelos que tive com o loRA,, identifiquei três aspecto explorando as suas funcionalidade:
Compressão Eficiente de Parâmetros:
O LoRA utiliza decomposição de baixa classificação, reduzindo o número de parâmetros ajustáveis.
Ênfase em Características Relevantes:
Ao capturar a dimensão intrínseca baixa de modelos grandes, o LoRA enfatiza apenas características relevantes durante o ajuste fino, resultando em modificações mais direcionadas.
Menor Sobrecarga Computacional:
A estratégia do LoRA de ajustar um número menor de parâmetros diminui a sobrecarga computacional, tornando o processo mais eficiente e economizando recursos durante o treinamento.

Informações sobre o desenvolvimento da atividade e as referências podem ser encontradas no documento Gate 11/01/2024

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Preparativos para o TCC

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Gate 11/12/2023

Continuando os meus estudos sobre o aprimoramento do modelo transformer, durante esta semana, foquei em melhorar a performance do modelo em desenvolvimento. Para alcançar esse objetivo, utilizei a técnica LoRA (Low Rank Adaptation). Essa abordagem provou ser eficaz na otimização do treinamento do modelo, resultando em uma redução significativa no tempo de treinamento, economizando aproximadamente 45 minutos em comparação com métodos anteriores. Além de manter o F1_score geral de 89.7% e diminuir seu tamanho em aproximadamente 3 vezes:
Tamanho do modelo original: 1654.25 MB
Tamanho do modelo com LoRA: 424.64 MB
Uma das vantagens notáveis do LoRA é sua capacidade de efetuar ajustes finos em modelos grandes de forma mais eficiente, enfocando uma baixa classificação e, portanto, ajustando um número menor de parâmetros em comparação com métodos tradicionais. Essa abordagem não apenas resultou em economia de tempo, mas também em uma significativa redução na sobrecarga computacional, contribuindo para a eficácia do modelo.

O experimento feito está no colab abaixo:
https://drive.google.com/file/d/1zVKXwTfbbNLxzNUrdS6r0WnhTbncfcUo/view?usp=sharing