CENTRO UNIVERSITÁRIO FEI

AMANDA MACIEL DE LIMA

APLICAÇÃO DA ARQUITETURA TRANSFORMER PARA SUMARIZAÇÃO DE ARTIGOS CIENTÍFICOS

São Bernardo do Campo

2023

AMANDA MACIEL DE LIMA

APLICAÇÃO DA ARQUITETURA TRANSFORMER PARA SUMARIZAÇÃO DE ARTIGOS CIENTÍFICOS

Dissertação de mestrado apresentada ao Centro Universitário da FEI para obtenção do título de Mestre em Engenharia Elétrica. Orientado pelo Prof. Dr. Paulo Sérgio Silva Rodrigues.

São Bernardo do Campo

2023

Maciel de Lima, Amanda.
Aplicação da Arquitetura Transformer para Sumarização de Artigos Científicos / Amanda Maciel de Lima. São Bernardo do Campo, 2023.
96 f. : il.

Dissertação - Centro Universitário FEI.
Orientador: Prof. Dr. Paulo Sérgio Silva Rodrigues.

1. Processamento de Linguagem Natural. 2. Sumarização. 3. Artigos Científicos. 4. Modelo de Atenção. 5. Arquitetura Transformer. I. Silva Rodrigues, Paulo Sérgio, orient. II. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os dados fornecidos pelo(a) autor(a).

centro universitário FEI

APRESENTAÇÃO DE DISSERTAÇÃO
ATA DA BANCA EXAMINADORA

Mestrado

Programa de Pós-Graduação Stricto Sensu em Engenharia Elétrica

PGE-10

Aluno(a): Amanda Maciel de Lima **Matrícula:** 120119-3

Título do Trabalho: Aplicação da Arquitetura Transformer para Sumarização de Artigos Científicos

Área de Concentração: Processamento de Sinais e Imagens

Orientador(a): Prof. Dr. Paulo Sérgio Silva Rodrigues

Data da realização da defesa: 06/03/2023 **ORIGINAL ASSINADA**

Avaliação da Banca Examinadora:

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO ☒ REPROVADO ☐

MEMBROS DA BANCA EXAMINADORA
Prof. Dr. Paulo Sérgio Silva Rodrigues
Prof. Dr. Roberto Baginski Batista dos Santos
Prof. Dr. Jefferson Magalhães de Morais

Aprovação do Coordenador do Programa de Pós-graduação

Prof. Dr. Carlos Eduardo Thomaz

À minha mãe, Vilmaci, e ao meu avô materno, José Dionatil.

AGRADECIMENTOS

Ao refletir sobre a minha jornada até aqui, tenho imensa gratidão por ter tido a contribuição de pessoas que acreditaram e confiaram na minha capacidade e nos meus objetivos.

Antes de tudo, gostaria de agradecer a minha mãe, Vilmaci dos Reis Maciel. Por tudo o que representa para mim. Por cada conselho e motivação nos momentos que precisei. Pelo sinônimo de força, fé e resiliência que você representa. É difícil colocar em palavras a grandeza da sua importância em tudo na minha vida.

Ao Pedro Domingues, agradeço imensamente por sua paciência, atenção, companheirismo, apoio e motivação incondicionais. Obrigada por me dar forças e acreditar em mim, mesmo nos momentos mais difíceis. Eu não poderia ter um apoio melhor de alguém cuja lealdade e dedicação são valores tão inspiradores.

A todos os meus amigos; suas qualidades sempre me inspiram. Obrigada por cada palavra animadora, reflexão e cumplicidade. Em especial, à Mariana Bastos, uma grande amiga, por todo suporte e parceria desde a graduação.

À orientação do professor Paulo Sérgio, sou grata por viabilizar e idealizar este trabalho. Obrigada por introduzir todo conhecimento acadêmico e científico que aprendi.

Agradeço também a contribuição e comentários dos membros avaliadores da qualificação e da banca final: Guilherme Wachs Lopes, Jefferson Magalhães de Morais e Roberto Baginski Batista dos Santos.

Por último, e não menos importante, agradeço ao Centro Universitário FEI e seus funcionários, corpo docente, infraestrutura, secretaria e inspetoria. Tenho grande orgulho de ter vivenciado e aprendido tantos ensinamentos nessa instituição.

“It is good to have an end to journey toward; but it is the journey that matters, in the end.”

Ursula K. Le Guin

RESUMO

O processo de pesquisa científica tem como sua fase inicial a exploração de artigos para o conhecimento do estado da arte do tema a ser investigado. Em virtude do crescimento de dados em artigos científicos e do curso constante da informatização, tornam-se necessários mecanismos que sejam capazes de resumir artigos científicos com a finalidade de melhorar o processo de aquisição de pesquisas e direcionar a pessoa pesquisadora a acessar conteúdos relevantes. Os trabalhos de sumarização de artigos científicos, de modo geral, apresentam métodos de relevância de sentenças e aprendizado de máquina. Nos últimos anos, mecanismos de atenção associados a redes neurais e processamento de linguagem natural vêm sendo propostos para interpretar e contextualizar atividades de processamento de linguagens, sendo uma delas a textual. Paralelamente, a arquitetura *Transformer* sugere uma modelagem de transdução com mecanismos de autoatenção - prescindindo de convoluções e recorrências - é aplicada a diversos campos da Inteligência Articial com resultados considerados promissores. Este trabalho propôs empregar o modelo pré-treinado Longformer para a atividade de sumarização de artigos científicos da base de dados *SciSummNet* através de etapas de pré-processamento, *fine-tuning* e geração dos resumos. Os resultados obtidos indicaram melhoria de 20,8% para ROUGE-2 *recall* e 22,69% para ROUGE-2 *F-Measure* em relação ao trabalho original da base *SciSummNet* através do modelo *ComAbstract*.

Palavras-chave: Processamento de Linguagem Natural. Sumarização. Artigos Científicos. Modelo de Atenção. Arquitetura *Transformer*. Modelos Pré-Treinados. Longformer.

ABSTRACT

The scientific research process has as its initial phase the exploration of articles for the knowledge of the state of the art of the theme to be investigated. Due to the growth of data in scientific articles and the constant course of computerization, mechanisms that are capable of summarizing scientific articles become necessary in order to improve the research acquisition process and direct the researcher to access relevant content. Scientific articles summarizing works, in general, present sentence relevance and machine learning methods. In recent years, attention mechanisms associated with neural networks and natural language processing have been proposed to interpret and contextualize language processing activities, one of which is textual. In recent years, attention mechanisms associated with neural networks and natural language processing have been proposed to interpret and contextualize language processing activities, one of which is textual. At the same time, the Transformer architecture suggests a transduction modeling with self-attention mechanisms - dispensing with convolutions and recurrences - is applied to several fields of Artificial Intelligence with results considered promising. This work proposes to use the Longformer pre-trained model for summarizing scientific articles from the *SciSummNet* database through pre-processing, fine-tuning and summary generation steps. The results obtained indicated an improvement of 20.8% for ROUGE-2 recall and 22.69% for ROUGE-2 F-Measure in relation to the original work of the base SciSummNet through the variation model called WithAbstract.

Keywords: Natural Language Processing. Summarization. Scientific Articles. Attention Model. Transformer Architecture. Pre-Trained Models. Longformer.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O processo de busca por artigos científicos é a fase inicial de um projeto de pesquisa para análise do estado da arte e para a avaliação da viabilidade da proposta de pesquisa. Habitualmente, o levantamento bibliográfico de um estudo consiste em um processo de aquisição de artigos científicos para transcrição da linha de pesquisa. Após essa análise, a pessoa pesquisadora seleciona os trabalhos aderentes à sua proposta para continuidade do projeto.

No decorrer dos anos, o processo de obtenção de artigos tem se tornado cada vez mais informatizado; diversos documentos científicos são disponibilizados de forma online com o apoio de mecanismos de buscas como por exemplo o *Google Scholar*[1], *Semanthic Scholar*[2] e *PubMed*[3]. Dado o volume de possibilidades de trabalhos a serem analisados por um pesquisador, o tempo de identificação e leitura tende a ser proporcional ao volume de documentos adquiridos. Do mesmo modo que a inspeção de artigos é sujeita ao tempo de análise, ela também é suscetível a falhas na extração de informação e restrição do conhecimento a outros trabalhos da literatura. Para isto, mecanismos de resumo do conteúdo de artigos científicos são estudados a fim de oferecer ferramentas práticas para esta atividade, além de explorar outros aspectos encontrados em artigos científicos.

O Processamento de Linguagem Natural, em inglês, *Natural Language Processing* (NLP) consiste em um conjunto de técnicas computacionais para análise e representação da linguagem humana de acordo com Young et al. (2018). Tem como finalidade interpretar e principalmente compreender expressões linguísticas do ser humano a fim de realizar tarefas e aplicações como tradução, reconhecimento de fala, extração de informação e sistemas de diálogo.

Segundo o levantamento de um período de 50 anos sobre pesquisas na área de NLP, o editorial de Mariani, Francopoulo e Paroubek (2019), publicado na plataforma aberta de publicação científica *Frontiers*[4], afirma que a pesquisa voltada à análise de textos científicos tem sido cada vez mais explorada nos últimos anos.

Conciliando o tema de NLP e métodos baseados em atenção, o primeiro uso de um modelo de atenção para tarefas *sequence-to-sequence* (no qual a entrada e saída do modelo são sequências de texto) foi proposto por Bahdanau, Cho e Bengio (2014) para a tarefa de tradução. Desde então é uma estratégia utilizada em Inteligência Artifical e nas áreas de NLP e

[1]Disponível em: `https://scholar.google.com/`
[2]Disponível em: `https://www.semanticscholar.org/`
[3]Disponível em: `https://pubmed.ncbi.nlm.nih.gov/`
[4]Disponível em: `https://www.frontiersin.org/`

Visão Computacional, de acordo com Chaudhari et al. (2021). Métodos de atenção têm como interesse obter as partes mais importantes de um contexto e trabalhar nos conteúdos de destaque. Segundo Xu et al. (2015), esse comportamento é inspirado no sistema biológico humano, no qual o ser humano tende a ignorar informações irrelevantes e focalizar seletivamente em certas partes de um contexto.

A arquitetura *Transformer* de Vaswani et al. (2017) surgiu com o intuito de propor um modelo *sequence-to-sequence* originalmente para a tarefa de tradução e tem sido amplamente utilizada para diversas tarefas de NLP e em áreas como Visão Computacional, Processamento de Áudio e Química. O diferencial proposto é uma abordagem baseada em mecanismos de autoatenção, sem a aplicação de recorrência, convoluções ou dependência sequencial em seu funcionamento, sendo possível utilizar mecanismos de processamento paralelo que até então não eram viáveis. A arquitetura *Transformer* é composta por blocos de codificação e decodificação no qual cada bloco possui subcamadas de atenção e redes neurais totalmente conectadas.

Correlacionado à temática da arquitetura *Transformer*, os modelos pré-treinados foram idealizados com o intuito de representar e modelar linguagens para tarefas *downstream*; entre os mais utilizados, estão o BERT, GPT-2 e XLNet, respectivamente dos trabalhos de Devlin et al. (2018), Radford et al. (2019) e Zhilin Yang et al. (2019). De acordo com Lin et al. (2021), os modelos pré-treinados que utilizam a arquitetura *Transformer* podem atingir desempenho do estado da arte em diversas tarefas.

Com o propósito de analisar uma estratégia para pesquisadores na etapa de resumir o conteúdo de trabalhos científicos, este trabalho propôs explorar e aplicar a arquitetura de autoatenção *Transformer* para a tarefa de sumarização híbrida de artigos científicos, operando com o modelo pré-treinados Longformer proposto por Beltagy, Peters e Cohan (2020), derivado dos modelos BERT e RoBERTa. O conjunto de artigos científicos com o resumo padrão-ouro *SciSummNet* sugerido por Yasunaga et al. (2019) foi utilizado para contemplar o desenvolvimento e conteúdo deste trabalho.

1.1 OBJETIVO

O objetivo deste trabalho consiste em aplicar a arquitetura *Transformer* para sumarização híbrida de artigos científicos por meio do modelo pré-treinado Longformer.

1.2 ESTRUTURA DO TRABALHO

A organização e estruturação dos capítulos desta pesquisa segue ordem apresentada nos parágrafos a seguir.

O conteúdo do Capítulo 2 mostrará os Trabalhos Relacionados para o desenvolvimento da pesquisa, finalizando-o com uma análise dos principais objetivos, métodos, métricas e bases de dados encontradas.

O Capítulo 3 apresentará os Conceitos Fundamentais, onde as principais definições relacionadas ao vigente trabalho serão apresentadas.

O Capítulo 4 exibirá a Metodologia Proposta com a descrição de sistematização do trabalho para contemplar o objetivo de pesquisa.

No Capítulo 5 de Resultados e Discussão, serão apresentados os apuramentos obtidos a partir da implementação da metodologia.

Por último, o Capítulo 6 expõe a Conclusão do desenvolvimento do trabalho.

2 TRABALHOS RELACIONADOS

Neste capítulo serão demonstrados os trabalhos correlatos assim como análises dos trabalhos pesquisados. Para cada artigo, foram extraídos os respectivos objetivos, metodologias, métricas e bases de dados a partir de buscas realizadas no *Semantic Scholar*[1] e *Google Scholar*[2] – considerando a seleção de publicações com fatores de relevância em destaque e citações. As principais chaves de busca foram: *scientific article summarization*, *scientific text generation*, *text summarization* e *Transformer architecture*.

Na Figura 1 é apresentada a contagem de artigos obtidos em relação ao ano de publicação. Os principais temas pesquisados foram relacionados às aplicações, abordagens e técnicas de processamento de texto em artigos científicos. As pesquisas dos últimos cinco anos foram priorizadas e trabalhos levantados anteriores a esse período foram escolhidos para este capítulo devido à sua relação e contribuição para o vigente trabalho.

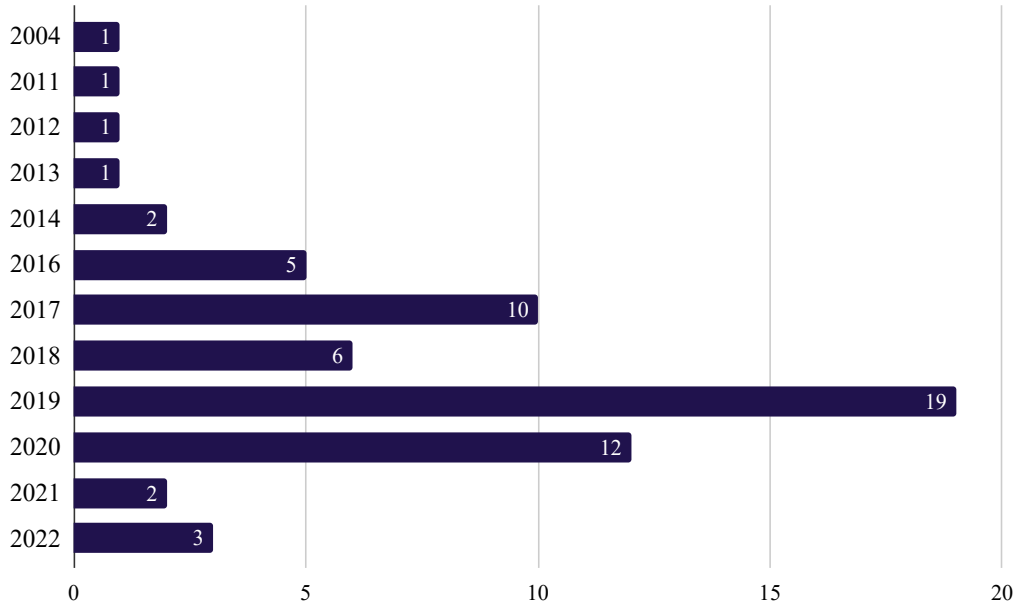

Figura 1 – Contagem de artigos por ano de publicação.

Fonte: Autora.

A Seção 2.1 mostra os trabalhos relacionados à sumarização de artigos científicos. A Seção 2.2 expõe trabalhos relacionados a aplicações e propostas em textos científicos. Na Seção 2.3 são apresentados artigos de sumarização de textos em geral. A Seção 2.4 apresenta trabalhos

[1] https://www.semanticscholar.org/
[2] https://scholar.google.com/

voltados à arquitetura *Transformer*. Na Seção 2.5 são exibidos artigos de revisão e, por fim, na Seção 2.6 é apresentada uma análise dos trabalhos relacionados.

2.1 ARTIGOS DE SUMARIZAÇÃO DE ARTIGOS CIENTÍFICOS

Abu-Jbara e Radev (2011) apresentam uma abordagem para gerar resumos de artigos de forma legível e compreensível através de citações. A metodologia envolve três etapas: o pré-processamento no conjunto de frases no texto (para considerar as sentenças ou partes de sentenças mais importantes), ordenação das sentenças selecionadas na primeira etapa e o refinamento para melhorar a legibilidade do resumo. A classificação de sentenças (entre adequada ou não adequada para o resumo) pela técnica do SVM (*Support Vector Machine*) aliado à clusterização melhoraram a qualidade de extração da abordagem, resultando em um intervalo de confiança ROUGE-L com 95% com base na sumarização de 30 textos.

Contractor, Guo e Korhonen (2012) propõem a utilização do método de Zoneamento Argumentativo para investigar a sua aplicação para a sumarização de textos científicos do ramo biomédico. O processo ocorre em duas fases principais: classificação e agrupamento de frases. O classificador obtém um conjunto inicial de frases e o agrupador identifica grupos de frases semelhantes para gerar o resumo final. Nos experimentos foram utilizadas dois tipos de fonte de dados: com os documentos completos e com resumos gerados por especialistas. Comparando o resumo gerado a partir dos tipos de zoneamentos argumentativo em relação a um trabalho que utiliza ao título das seções do artigo houve um aumento de 7% na métrica F1 na sumarização de documentos completos em sumarizações customizadas o aumento foi entre 54% e 57%.

Qazvinian et al. (2013) demonstram, a partir de sentenças que contenham citações, a elaboração de resumos a partir de técnicas de mineração bibliométrica e sumarização. Foram coletados 30 artigos derivados de 6 temas da base de dados relacionados a NLP. Quatro técnicas conhecidas de sumarização foram utilizadas: C-LexRank, C-RR, LexRank e MASCS. Para avaliar os resultados, as técnicas de ROUGE-L e *pyramid* foram medidas para cada técnica de sumarização. A partir da métrica de *pyramid*, o MASCS obteve o melhor resultado. Através da métrica de ROUGE-2, as técnicas C-LexRank e LexRank se destacaram obtendo valores entre 17% e 28%. Baseado nos resultados obtidos, os autores concluem que a técnica de MASCS se mostrou útil para identificar partes importantes do texto, e o C-LexRank e LexRank geram unigramas e bigramas esperados para a sumarização.

Chen e Zhuge (2014) projetam um sistema de resumo de múltiplos documentos, a partir da detecção de fatos em comum. A proposta consistiu em desenvolver um algoritmo de descoberta de associação de termos, para na próxima etapa passar por um módulo de processamento de citação. A terceira etapa é a detecção do fato em comum que consiste nas palavras mais importantes encontradas nos artigos científicos, e por fim a geração e saída do resumo. Comparando com os métodos de sumarização MEAD, SciSumm, e CSIBS, o projeto obteve os melhores resultados em ROUGE-1 e ROUGE-2 para *recall* e ROUGE-2 *F-Measure*, respectivamente 0,66, 0,45 e 0,43.

Shansong Yang et al. (2017) apresentam um mecanismo de sumarização nomeado KeyphraseDS, para organizar artigos científicos utilizando palavras-chave. A técnica proposta é composta por três etapas: construção do grafo de palavras-chave - com sua extração em CRF (*Conditional Random Field*), geração do aspecto semântico e seleção de conteúdo utilizando otimização com base em ILP (*Integer Linear Programming*). Foram gerados resumos de 250 palavras para a avaliação do mecanismo apresentado. O KeyphraseDS foi melhor avaliado pela métrica de *pyramid*, onde obteve maiores resultados em *recall* e F1.

Cohan e Goharian (2018) apresentam uma ferramenta para sumarização de textos científicos com base em citações e estrutura do documento. As seguintes etapas foram apresentadas como metodologia: contextualização da citação para extrair contextos relevantes, identificação do discurso do contexto e por fim o resumo. A proposta foi testada em nas bases de dados TAC e CL-SciSum. Segundo os autores a identificação de citações de acordo com o contexto do documento gerou um aumento entre 17% e 55% pela métrica ROUGE-2.

Nikolov, Pfeiffer e Hahnloser (2018) geraram dois conjuntos de base de dados, respectivamente para gerar a seção de resumo do artigo e títulos a fim de aplicar modelos de redes neurais extrativas e abstrativas. Os conjuntos de dados foram extraídos originalmente do sistema de busca MEDLINE com aplicação de etapas de pré-processamento. Através da medição das métricas a partir das métricas ROUGE-1, ROUGE-2 e ROUGE-L em ambas finalidades, para os resultados de geração de título, os modelos abstrativos alcançaram valores superiores em relação aos modelos extrativos, com destaque para o *c2c* e *fconv*. Na geração da seção de resumo, os pesquisadores notaram uma taxa de repetição de 44% e as técnicas extrativas tiveram resultados ligeiramente superiores.

Almugbel, El Haggar e Bugshan (2019) implementam técnicas de NLP e aprendizado de máquina para gerar um resumo de artigos científicos a partir da Introdução, Métodos, Resultados e Discussão. A base de dados é pré-processada para ser usada como entrada pelos

classificadores *Naive Bayes* e SVM que serão utilizados para determinar qual a seção a sentença pertence. O *Naive Bayes* atingiu resultados superiores para classificar as seções do artigo em relação ao SVM de acordo com as métricas *F-Score* e acurácia. A pesquisa indica também que a classificação obtém melhores resultados utilizando unigramas e bigramas ao invés de frequência de palavras.

Romanov, Lomotin e Kozlova (2019) estudam a aplicação de algoritmos de aprendizado de máquina classificação automática de artigos científicos e resumos russos. O conjunto de dados foi pré-processado e as características dos textos foram extraídas através do *word2vec*. As técnicas de classificação escolhidas foram: *Random Forest*, SVM, *Logistic Regression* e Redes Neurais Artificiais. A técnica de SVM baseada em uma função radial mostrou o melhor resultado no experimento do trabalho.

Xu, Wang e Weng (2019) introduzem um mecanismo de recurso de estrutura de documento e atenção hierárquica para sumarização da literatura científica. Um modelo de atenção hierárquica foi sugerido para aprender a estrutura do documento e a semântica a nível de discurso e sentença utilizando a rede LSTM. A partir do nível de atenção da frase, ela pode ser selecionada para a fase de treinamento por um valor binário. Para treinar o modelo é utilizada a função de *cross-entropy loss*. A proposta foi avaliada a partir das métricas ROUGE-1, ROUGE-2, ROUGE-L e ROUGE-SU4 e é comparada as técnicas como PageRank, SVM e LSTM. Apesar do modelo proposto, HASum, superar as técnicas selecionadas, os autores concluem o tempo computacional de treinamento do modelo é custoso.

Cagliero e La Quatra (2020) introduzem uma abordagem supervisionada de sumarização para oferecer uma visão rápida do artigo com base em métodos de regressão. A proposta consiste em verificar a similaridade com base em co-ocorrência em um n-grama entre frases e destaques de um artigo. As principais etapas são: extração de informações importantes do texto, medição de similaridade entre frases e partes importantes, treinamento e aplicação do modelo. Os resultados, em comparação com as sumarização baseadas em BERT, se mostrou superior, especificamente quando o número de sentenças em destaque estão entre 3 e 5.

Altmami e Menai (2020b) propõem uma teoria de estrutura transversal de artigos nomeada CAST (*Cross-Article Structure Theory*) para sumarização de múltiplos artigos. Essa teoria é baseada na teoria de estrutura retórica (*Rhetorical Structure Theory*) e de estrutura entre documentos (*Cross-Document Structure Theory*). As relações entre as frases são determinadas entre a própria seção e em relação as demais seções, para que a teoria seja aplicada a nível de múltiplos artigos. Uma base anotada foi construída, chamada *CAST Bank*, e nela foram aplicados

dois modelos baseados no algoritmo KNN. Os resultados obtidos pelo melhor modelo foram de 0,91 de acurácia, 0,69 de precisão, 0,71 de *recall* e 0,69 de *F-Measure*.

Debnath e Das (2021) apresentam um método para resumir artigos científicos de forma extrativa, baseada em citações, pela técnica de MODE (*Multi-Objective Differential Evolution*). O primeiro passo verifica a similaridade entre uma citação e os textos relacionados a citação. Depois, os textos extraídos são representados em vetor a nível de sentença considerando atributos como o seu tamanho, e pontuaçao da sentença pelo TF-IDF. Uma matriz de relação de sentenças é elaborada para verificar a similaridade, para que a técnica MODE seja aplicada para gerar o resumo. Os conjuntos de dados utilizados foram o CL-SciSumm-18 e o SciSummNet e segundo os autores, os resultados atingidos foram equivalentes ao estado da arte.

2.2 APLICAÇÕES GERAIS EM ARTIGOS CIENTÍFICOS

Essa seção apresenta trabalhos de aplicações gerais em artigos científicos - propondo novas análises e funcionalidades para diferentes aplicabilidades desse tema na literatura.

Afonso e Duque (2014) realizam um estudo empírico sobre agrupamento automático de textos científicos e revistas brasileiras. A produção do estudo teve os seguintes passos: seleção do *corpus*, seleção de classes de palavras (substantivos, adjetivos e verbos), algoritmos de filtragem e algoritmos de agrupamento - K-Means, SIB (*Sequential Information Bottleneck*) e EM (*Expectation Maximization*). O algoritmo de agrupamento SIB foi a melhor escolha para ambos os *corpus*, tendo 68,9% de acerto no corpus de revistas e 77,8% no corpus de artigos científicos.

Hamedani, Kim e Kim (2016) propõem um método chamado SimCC a fim de calcular a similaridade de artigos científicos. O método proposto considera o conteúdo de citações - diretas e indiretas - e consiste em duas etapas: extração de recursos e cálculo de similaridade. São levadas em consideração a recência do artigo e experiência do autor para o uso de uma ponderação RA (relevância e autoridade). Segundo os autores, a proposta pode ser utilizada em sistemas de recomendação de artigos e identificação de plágio.

Ronzano e Saggion (2016) propõem um sistema para extrair e caracterizar o conteúdo de artigos científicos, disponibilizando esse conteúdo por meio de um conjunto de dados RDF (*Resource Description Framework*). Os artigos foram processados através do *framework* Dr. Inventor, para a ferramenta classificar o conteúdo das sentenças e realizar o resumo extrativo do

texto. Os resumos possuíram em média 250 palavras. O valor mais alto da métrica ROUGE-2 teve a pontuação de 0,3617 utilizando o método *TextRank*.

Putra e Khodra (2017) apresentam um estudo sobre a geração automática de títulos de artigos científicos a partir de frases consideradas retóricas. O método proposto consiste em extrair informações a partir do objetivo e metodologia para gerar um título. Após a extração dessas partes do texto, são elencados três possíveis títulos a partir de um método adaptativo do KNN (*K-Nearest Neighbors*) ou baseado em padrões de *tags* na base de dados. No experimento foram considerados artigos relacionados aos temas de Linguística Computacional e Química. A partir da métrica F1 foi obtido o resultado de 0,109 a 0,205, comparados a títulos originais. Segundo a avaliação de humanos o resultado foi de "relativamente aceitável"no domínio de Linguística Computacional e "não aceitável"para o domínio da Química.

Hashimoto et al. (2017) geram uma matriz de síntese para revisão de literatura científica. A proposta consiste em uma fase de classificação de sentenças utilizando uma adaptação do LexRank chamada Q-LexRank, além de uma técnica de extração de informação chamada expansão de consultas e o cálculo de relevância da consulta. Na fase de seleção de sentenças é utilizado um modelo baseado em ILP e uma sumarização comparativa. Concluem que método proposto com a utilização de ILP tende a classificar na mesma seção verbos similares entre si.

Baker et al. (2017) introduzem o CHAT (*Cancer Hallmarks Analytics Tool*) para classificar textos científicos a fim de auxiliar estudos de câncer. Utilizando técnicas de NLP, o modelo CHAT realiza a lematização do *corpus*, com a utilização de bigramas e trigramas. Posteriormente consiste em etapas de classificação de verbos, reconhecimento de entidades nomeadas, e identificação termos químicos e médicos. Para obter a similaridade semântica entre palavras, a técnica SVM foi utilizada. Foi treinada uma rede neural artificial para representar a similaridade de palavras e termos da área, para assim utilizar a similaridade de cossenos como medida de distância no espaço vetorial. Segundo os autores, a proposta teve aderência para a classificação de textos da área. Para cada tópico da taxonomia HoC, a classificação destes tópicos obteve uma média macro de 45,1% de precisão, 63% de *recall*, 52,3% de F1-score e 97,9% de acurácia.

Jha et al. (2017) propõem um conjunto de dados para análise de citações utilizando NLP. O artigo pontua dois problemas em análises de citações com NLP: extração do contexto da citação (identificação de citações implícitas e explícitas) e escopo de referência (quando uma frase pode conter mais de uma fonte de citação). A proposta sugere uma etapa inicial de extração de recursos informativos (verificar se a sentença tem similaridade com um artigo citado). Dessa forma, os autores determinaram duas formas de dependência modeladas por

MRFs (*Markov Random Fields*): entre as próprias frases do texto e entre as frases vizinhas. O método MRF é comparado com outras abordagens como o SVM. A proposta utilizando o MRF com 4 sentenças anteriores conectadas obteve a medida de F-Score superior a outras combinações na maioria dos artigos selecionados para o experimento.

Price e Arkin (2017) apresentam o PaperBLAST, uma ferramenta para encontrar artigos relacionados a proteínas na literatura e fornece ligações entre esses termos. O agente realiza a busca de proteínas em artigos de duas formas: pesquisando no conjunto de dados do *EuropePMC* por menções às proteínas e utilizando recursos manuais como Swiss-Prot, GeneRIF e EcoCyc. A metodologia seleciona as frases encontradas e as armazena em um banco de dados relacional. Durante o acompanhamento da execução do modelo, foram observados a porcentagem de artigos em que o PaperBLAST encontrou em relação aos artigos divulgados. O modelo não apresentou os artigos esperados em 28% dos casos.

Khan, Khattak e Afzal (2018) apresentam uma técnica baseada em co-citação para selecionar artigos científicos. Primeiramente os dados do *CiteSeer* são coletados, e os documentos na extensão PDF são convertidos para XML. Após a conversão, as citações são extraídas por *tags* com base na lista de referências. A proposta também verifica a frequência de citações de texto. Para cada documento é calculado a soma do peso de cada seção com o número de ocorrências de co-citações do documento. A proposta foi comparada com dois trabalhos, um com técnica CPA (*Citation Proximity Analysis*) e outro de co-citação, e respectivamente, houve um acréscimo de 39% e 68% de correlação do coeficiente de Spearman.

Govoni et al. (2019) apresentam a ferramenta e interface chamada *Qresp* para a organização de conteúdo a fim da reprodutibilidade de artigos a partir de metadados. A ferramenta *Qresp* realiza a organização do conteúdo do artigo científico em três etapas: a primeira etapa é chamada de fase organizadora, qual o usuário define sua preferência de disponibilização dos metadados; a segunda fase é a parte de curadoria onde os metadados são extraídos; e por fim, no final a fase de exploração exibe o resultado gerado pelo *Qresp*. A partir de uma modelagem distribuída, a pesquisa realizada no sistema é transformada em uma consulta para assim apresentar os dados da pesquisa realizada. Os autores finalizam o artigo mencionando a viabilidade da ferramenta, assim como a sua aplicação para o desenvolvimento científico por pesquisadores.

Chen e Zhuge (2019) geram automaticamente a seção de trabalhos relacionados de uma pesquisa, intitulado RWS-Cit. A abordagem coleta os documentos de referência e extrai a sentença de citação por meio das seguintes seções: título, resumo, introdução, conclusão e lista de referências. Depois, extrai as palavras-chave do documento de referência e do artigo que está

sendo escrito para criar um grafo de palavras-chave. A estrutura de Árvore de Steiner Mínima é criada através das palavras-chave em comum entre o documento de referência e o artigo; o resumo é gerado através das sentenças obtidas pela árvore. A partir da avaliação com 25 artigos de Computação Linguística com a métrica ROUGE-1 superou os métodos tradicionais de sumarização MEAD, LexRank e ReWoS. Concluem que o uso de citações para a abordagem proposta é mais promissor em relação a extração de partes dos próprios artigos.

Wang et al. (2019) apresentam um assistente de pesquisa automática de artigos no domínio da Biomedicina intitulado PaperRobot. O agente conduz uma pesquisa profunda em artigos escritos por humanos, prediz links e constrói KG (*knowledge-graphs*), e escreve de forma incremental elementos-chave baseados em redes de atenção de memória para assim gerar um título e conclusão do trabalho em questão. A partir do Teste de Turing feito com especialistas, os seguintes percentuais mostram o resultado da preferência do texto gerado pelo agente ao invés de textos gerados por humanos: geração de resumos (30%), seção de conclusão e trabalhos futuros (24%) e novos títulos (12%).

Habib e Afzal (2019) propõem um sistema de recomendação de busca de artigos através de análise de citações e mapeamento de seções lógicas. O primeiro módulo, *Data Acquisition* é responsável pela captura de dados. O *XML Conversion* converte o arquivo XML para o formato PDF e encontra as referências do documento. O módulo de extração de seções lógicas é chamado de *Section Extraction*, o qual usa a teoria proposta no artigo de Ding et al. (2014), em que a seção que tende a ter mais citações é a de Introdução, seguida de Trabalhos Relacionados, Metodologia, Resultados e Conclusão. Para avaliar o mapeamento de seções, na primeira base de dados, mapeada seguindo a teoria de número de citações; a correlação de Spearman resultou em 0,85 de 1. Na segunda base foram extraídos 100 artigos e segundo os experimentos o resultado de 90% de acurácia foi atingido. Em relação ao sistema de recomendação, o trabalho proposto superou métodos baseados em bibliografia e em conteúdo na média dos resultados de correlação por JSD (*Jensen Shannon Divergence*); o método baseado em seções lógicas obteve o maior resultado.

Bugnon et al. (2020) apresentam o método *DL4papers* que extrai relações entre palavras-chaves de documentos. Como entrada, o modelo recebe um par de palavras-chaves e um conjunto de artigos, e retorna a relação entre os documentos a partir das palavras-chave. O método *DL4papers* opera com uma rede neural convolucional profunda e com um *corpus* de oncologia chamado BRONCO. Segundo os autores, o *DL4papers* demonstrou um alto índice de confiabi-

lidade pois os documentos mais relevantes eram listados nas primeiras posições. A medida F1 atingiu cerca de 75%.

Marcos-Pablos e García-Peñalvo (2020) propõem um método para auxiliar no processo de revisão sistemática de literatura. O procedimento sugerido envolve técnicas de mineração de texto, recuperação automática de informação (para recomendar termos e melhorar a busca dentro da base de dados), aprendizado de máquina (para classificar artigos por relevância e pelo tópico de interesse) em um corpus de resumos de artigos científicos. Na escolha do classificador, foram realizados testes com o SVM, KNN e as variações de Bernoulli e Multinomial do *Naive Bayes*. O *F-score* mais alto foi obtido com o classificador SVM. Os autores concluem que o trabalho é capaz de sugerir termos relevantes em uma pesquisa, porém a avaliação de pesquisadores é necessária para ter uma visão imparcial.

2.3 ARTIGOS DE SUMARIZAÇÃO DE TEXTO

Nesta seção serão apresentados os trabalhos de geração de resumos de textos ou documentos em geral, que foram separados em dois grupos de sumarização: extrativo e abstrativo.

Um resumo feito de forma extrativa é elaborado a partir das frases mais relevantes do texto original, desse modo, estratégias para medir a importância das frases são propostas para essa finalidade. O resumo abstrativo consiste na interpretação das ideias e conceitos do texto para que a geração resumo compreenda sentenças novas, mantendo o sentido do texto original de acordo com Gambhir e Gupta (2017). A sumarização híbrida utiliza a combinação das estratégias da sumarização extrativa e abstrativa.

2.3.1 Sumarização Extrativa

Erkan e Radev (2004) apresentam um método fundamentado em grafos estocásticos para estimar a importância de partes em textos. A abordagem proposta consiste em calcular a importância com base na centralidade dos autovalores de um grafo de frases. A matriz de conectividade com a similaridade de cossenos é utilizada como matriz de adjacência. Concluem que submeter o método a um conjunto de dados ruidosos, afeta a qualidade da sumarização. Os autores destacam que o LexRank supera métodos baseados em grau.

Al-Sabahi, Zuping e Nadher (2018) propõem o modelo HSSAS para sumarização extrativa de documentos com foco na resolução do problema de memória e incorporação da estrutura

de documentos. A proposta consiste em um mecanismo de auto atenção com estrutura hierárquica. A primeira camada é a nível de palavra, e a segunda camada opera a nível das sentenças; após cada uma existe uma camada de atenção pela técnica de LSTM. Na camada logística é classificada se a frase está resumida ou não. O modelo foi comparado com propostas de sumarização abstrativa e extrativa através da métrica de ROUGE-1, ROUGE-2 e ROUGE-L onde o HSSAS se mostrou equivalente ou superior.

Al Saied, Dugué e Lamirel (2018) utilizam a técnica de *FeatureMaximization* para sumarização extrativa de textos estruturados. Para cada palavra é calculado o valor de F-Measure, para assim calcular o peso da sentença (média do *F-Measure* das palavras da frase), definir o tamanho do resumo de acordo com o peso ideal, diminuir a redundância e assim gerar o resumo. O modelo foi submetido a três testes: geração resumo de um documento, geração de resumo baseada em consulta e o último é uma combinação dos dois anteriores. O artigo é concluído dando ênfase ao fator de que o modelo feito por *FeatureMaximization* não necessita de treinamento.

Alguliyev et al. (2019) propõem o COSUM, um modelo de seleção de sentenças composto em duas partes: otimização e agrupamento. O conjunto de sentenças é agrupado de acordo com os tópicos do texto através do *K-Means*. Para selecionar as frases mais relevantes, os valores gerados por função objetivo através de uma média harmônica são otimizados. O COSUM foi comparado com modelos do estado da arte entre eles DPSO-EDASum, LexRank, CollabSum, UnifiedRank, SVM e *fuzzy*. Através da comparação do COSUM e outros métodos, pela métrica ROUGE-1 e ROUGE-2, respectivamente, obteve um aumento de 1,06% e 1,09% em comparação com o MA-SingleDocSum e 2.34% e 0.61% em comparação com o LexRank.

Goularte et al. (2019) encontram as informações mais importantes em um texto através de lógica *fuzzy*. O esquema proposto consiste em etapas de pré-processamento, medição de relevância de palavras e sentenças, análise *fuzzy* a partir de 27 regras e a avaliação do texto pela métrica ROUGE-1. Os experimentos foram comparados com cinco abordagens de resumo. A métrica ROUGE-1 obteve o maior valor em todas as porcentagens de sumarização. A base de dados foi construída com textos em português feitos por estudantes.

Diao et al. (2020) apresenta o modelo CRHASum, baseado em redes neurais, para sumarização extrativa de textos. A concepção do modelo é composta por um dois módulos de compreensão de atenção, a nível de palavra e de frase. A arquitetura modela e extrai recursos a nível de palavra, codifica a frase para posteriormente gerar recursos a nível de frase. Os autores constataram que a partir da arquitetura proposta, o CRHASum tem a capacidade de aprender recursos semânticos.

Mohd, Jan e Shah (2020) disponibilizam um recurso de sumarização preservando a semântica do texto. A abordagem fundamenta-se, inicialmente com o pré-processamento da base de dados, em registrar a semântica do texto usando *Distributional Semantic Models*, agrupar frases semelhantes, pontuar cada frase e por fim a normalização da pontuação. Cinco métodos de sumarização foram comparados em relação ao recurso proposto no trabalho: OPINOSIS, Genism, PKUSUMSUM e PyTextRank. O sistema proposto superou todos os métodos pela avaliação ROUGE. Os autores concluem o trabalho afirmando a importância do viés semântico para a qualidade na geração de resumos.

2.3.2 Sumarização Abstrativa

Nallapati et al. (2016) modelam a sumarização abstrativa de textos e comparam com dois métodos do estado da arte. A sumarização abstrativa foi implementada utilizando redes neurais recorrentes e a rede neural de atenção chamada *EncoderDecoder*. O artigo propõe modelos com abordagens que não são tratados na sumarização tradicional, como palavras-chave, hierarquia de sentenças e a consideração de palavras incomuns durante o treinamento. O modelo apresentou falhas em questões semânticas e performance, e os autores concluem que os resultados e trabalhos futuros podem contribuir para a melhoria da sumarização.

Tan, Wan e Xiao (2017) revisam as dificuldades sumarização de textos abstrativos e propõe um modelo de grafo baseado em atenção. Foi utilizado um *framework* codificador-decodificador que costuma ser usado em tarefas de tradução e diálogos. O diferencial proposto é o modelo de atenção e a decodificação hierárquica com referências para lidar com a geração de resumo abstrativa. A proposta do artigo, segundo os experimentos realizados em comparação com outras propostas do estado da arte se mostraram equivalentes através das métricas ROUGE-1, ROUGE-2 e ROUGE-L.

Gerani, Carenini e Ng (2019) geram sumarizações abstrativas por NLG (*Natural Language Generation*) sem necessidade dados de treinamento ou textos feitos manualmente. Os dados de entrada são processados por um AHT (*Aspect Hierarchy Tree*). A proposta consiste em receber um AHT como entrada, e a partir do processamento da árvore, gerar a sumarização. O artigo explora três vertentes: um modelo retórico (que explora a estrutura do discurso e entidades), um modelo conceitual (que explora a base de conhecimento) e um modelo híbrido com a combinação das duas vertentes anteriores. A análise foi feita de forma qualitativa e os autores

constataram que os três modelos tiveram saídas que diferem entre si em questões de estrutura e conteúdo.

Song, Huang e Ruan (2019) propõem um *framework* de sumarização abstrativa baseado em LSTM-CNN para produzir frases semânticas. Inicialmente é usado um modelo nomeado MOSP para extrair as frases-chave do texto original e em seguida são utilizadas as técnicas de aprendizado profundo para gerar o resumo. Após o treinamento o modelo gera o resumo de acordo com uma estrutura sintática. Informações de localização de frases também foram utilizadas para decifrar o problema de palavras raras. O modelo proposto obteve 34,9% e 17,8% respectivamente na medida ROUGE-1 e ROUGE-2.

2.4 ARQUITETURA TRANSFORMER

Vaswani et al. (2017) apresentam uma arquitetura de rede chamada *Transformer* baseada em atenção. A proposta utiliza mecanismos de atenção empilhados e totalmente conectados, junto a uma arquitetura codificador-decodificador. A arquitetura foi aplicada inicialmente para a tarefa de tradução do inglês para o francês e também do inglês para o alemão. Na tarefa de tradução em 2014 do WMT (*Workshop on Statistical Machine Translation*), o *Transformer* apresentou resultados superiores a todos os modelos. Os autores destacam que o *Transformer* pode ser treinado mais rápido em comparação a arquiteturas que utilizam redes recorrentes ou convolucionais.

Devlin et al. (2018) introduzem o BERT (*Bidirectional Encoder Representations from Transformers*) para a representação de linguagens pré-treinadas de forma bidirecional. A estrutura contém duas etapas: *pré-training* e *fine-tuning*. A primeira etapa consiste no treinamento em dados não rotulados. A segunda etapa inicializa os parâmetros de acordo com a primeira etapa realiza o ajuste. O BERT obteve a pontuação GLUE 7,7% superior a outros métodos estado da arte e acurácia no MultNLI com aumento de 4,6%. Para as tarefas de pergunta e resposta SQuAD v1.1 e SQuAD v1.2, respectivamente, obteve melhoria no resultado F1 em 1,5 e 5,1 pontos.

O trabalho de Liu et al. (2019) apresenta um estudo do pré-treinamento do modelo BERT proposto por Devlin et al. (2018) e nomeia-o como RoBERTa (*Robustly Optimized BERT Pre-training Approach*). Os pontos de estudo foram: avaliação dos efeitos dos ajustes os hiperparâmetros e conjunto de treinamento. As sugestões propostas de treinamento foram: aumentar o tempo com lotes maiores e maior quantidade de dados, remover o processo de NSP (*Next Sen-*

tence Prediction), sequências mais longas e alterar o padrão de mascaramento. Em relação as tarefas SQuAD, GLUE e RACE, com as adaptações sugeridas, alcançaram resultados melhores que o BERT.

Radford et al. (2019) demonstram que o treinamento de modelos linguísticos pode ser realizado de forma não-supervisionada pela base de dados WebText. O método de treinamento utiliza a arquitetura *Transformer*, com algumas modificações: a camada de normalização foi movida para a entrada de cada sub-bloco e na auto-atenção final uma camada extra de normalização foi inserida. O maior modelo que o artigo apresenta é o GPT-2, treinado com 1,5 bilhões de parâmetros. Experimentos foram realizados nas seguintes frentes de NLP: modelagem de linguagem, compreensibilidade, QA (*Question Answering*), tradução e sumarização. No aspecto da tarefa de configuração *zero-shot* (que mostra o quanto o modelo pode aprender sem ter sido orientado no treinamento), o GPT-2 superou 7 de 8 trabalhos de modelagem de linguagem.

Miller (2019) apresenta uma ferramenta de sumarização de palestras a fim de resumir o conteúdo à alunos. O modelo BERT é usado para *embedding* e o *K-Means* é utilizado para identificar as sentenças mais próximas do centróide. Como não há modelos de sumarização de palestras, os autores fizeram experimentos com colaboradores. No geral os resultados a partir da utilização do BERT foram promissores, porém em certas situações o modelo não gerou o resultado esperado, como para documentos grandes e lidar com uma linguagem conversacional presente em palestras.

Zhilin Yang et al. (2019) propõem um método generalizado XLNet de pré-treinamento com autoregressão e bidirecionalidade. O método propõe o PLM (*Permutation Language Modeling*) e busca maximizar a função objetivo do modelo de linguagem através da permutação da ordem de fatoração, assim, sendo possível obter o aspecto bidirecional. A proposta também conta com uma dupla camada de atenção, baseada em conteúdo e em consulta. O XLNet superou o modelo BERT em 20 tarefas, como em classificação de texto e compreensibilidade de leitura.

Guan, Smetannikov e Tianxing (2020) revisam o campo de sumarização de textos através da arquitetura *Transformer* e seus modelos pré-treinados e propor um modelo de sumarização abstrativa. O artigo revisa as técnicas de sumarização adotadas ao longo do tempo, revisando os métodos baseados em classificadores, redes neurais e por fim o Transformer. O modelo proposto de sumarização abstrativa, XLSum, utiliza o modelo XLNet na base CNN/Daily Mail. Os resultados obtidos pelo XLSum foram competitivos, obtendo 43,36 para ROUGE-1, 20,68 para ROUGE-2 e 40,52 para ROUGE-3.

Kieuvongngam, Tan e Niu (2020) utilizam o BERT e o GPT-2 para resumir artigos científicos sobre COVID-19. O projeto foi dividido para realizar sumarização extrativa e abstrativa. Na parte extrativa, o modelo BERT foi utilizado para incorporar as sentenças, e o método *K-medoid* foi utilizado para agrupar as sentenças; os centroides foram utilizados como as sentenças do resumo. Na parte abstrativa, o resumo foi gerado a partir de palavras-chave de 3 grupos: verbos, substantivos e um conjunto unindo verbos e substantivos. Depois da extração, as palavras-chaves são pareadas com o resumo do artigo para que os pares sejam inseridos no GPT-2. Os autores afirmam que o resumo abstrativo realizado pelo GPT-2 obteve boa compreensibilidade, porém os resultados ROUGE foram inferiores a proposta extrativa.

Kitaev, Kaiser e Levskaya (2020) denotam duas técnicas para melhorar a eficiência do Transformer. Uma das alterações consiste no cálculo de operação que utiliza produto escalar, para um método de hash localizado, alterando a complexidade para $\mathcal{O}(L \log L)$ sendo L o tamanho da sentença. A outra alteração proposta consiste em armazenar as ativações uma vez por meio de camadas residuais reversíveis. Os resultados obtidos foram similares ao Transformer original, no entanto apresentou mais eficiência em uso de memória e processamento de sequências longas.

Zaheer et al. (2020) propõem o modelo BigBird para solucionar a complexidade quadrática do Transformer. Para contemplar o objetivo, métodos de grafos esparsos foram inspirados para o cálculo de atenção para realizá-lo com complexidade linear. Durante a implementação, foram investigados o uso de tokens globais, como o CLS, auxiliam no mecanismo de atenção esparsa. Para diversas tarefas de NLP experimentadas com o BigBird, o mesmo atingiu resultados competitivos para as tarefas de QA, sumarização e classificação.

Beltagy, Peters e Cohan (2020) apresentam o Longformer, um modelo baseado em Transformer para lidar com sequências longas de texto. O Longformer utiliza o mecanismo de atenção chamado *dilated sliding window*, no qual apenas um número fixo de diagonais são calculados na multiplicação de matrizes, tornando a operação linear ao invés de quadrática; esse cálculo foi implementado através de um *kernel* customizado do CUDA por TVM. A abordagem foi experimentada em tarefas de classificação de documentos e QA nos quais os resultados obtidos superaram o RoBERTa . Na sumarização, a partir da base de dados do arXiv, os resultados foram competitivos com o BigBird, utilizando o modelo de tamanho 16384.

Luu, Le e Hoang (2021) propõem um sistema de sumarização extrativa. O sistema é composto por uma rede convolucional, arquitetura *encoder-decoder* e uma rede totalmente conectada. O modelo pré-treinado M-BERT (*Multilingual* BERT) é utilizado para a incorporação

de palavras, combinado aos valores do TF-IDF. Frases que apresentem redundância são identificadas através do método *Maximal Marginal Relevance*. A proposta apresentou resultados de ROUGE-1, ROUGE-2, e ROUGE-L superiores a métodos como LexRank, TextRank e LEAD, para textos em inglês e vietnamita.

Li et al. (2022) apresentam dois modelos de linguagem chamados ClinicalLongformer e ClinicalBigbird a partir de textos longos de temática clínica. Inspirado em modelos pré-treinados para textos longos como Longformer e Bigbird, os modelos propostos foram aplicados em tarefas como *Question Answering*, reconhecimento de entidades e classificação de documentos. Superando a proposta anterior, nomeada ClinicalBERT, os resultados indicaram que o modelo baseado em Longformer e Bigbird apresentaram os melhores resultados para sentenças curtas e longas, sendo o ClinicalLongformer o modelo que apresentou a maior acurária nos modelos pré-treinados OpenI, MIMIC-AKI e medNLI.

Mamakas et al. (2022) utilizaram o modelo Longformer para textos longos jurídicos. A proposta aplica a técnica de modificação do Longformer para *warm-stated* para lidar com textos com mais de 8192 palavras, e a modificação do modelo LegalBERT para utilizar representações TF-IDF. Através da média harmônica para a tarefa de classificação de documentos (ECtHR A e B, SCOTUS) as variações LegalLongformer-8192 e LegalLongformer-8192-PAR obteram os melhores resultados.

2.5 ARTIGOS DE REVISÃO

Gambhir e Gupta (2017) apresentam abordagens extrativas de sumarização de texto. As vantagens e desvantagens de cada abordagem são mostradas de forma comparativa, assim como abordagens abstrativas e multilíngues. As formas de avaliar um resultado de uma sumarização, de forma intrínseca e extrínseca são descritos com base em dados de conferências e *workshops* da área. Destaca que a avaliação de uma sumarização é uma dificuldade a ser resolvida. As diferentes métricas de ROUGE são comumente utilizadas para medir numericamente uma avaliação de acordo com os dados obtidos no artigo.

Sun, Luo e Chen (2017) apresentam técnicas de NLP para mineração de textos de opinião. Procedimentos de NLP para pré-processamento de textos são introduzidas. O artigo também revisa abordagens utilizadas em situações de mineração de opinião, entre elas a aprendizagem profunda. Métodos supervisionados e não supervisionados a níveis de sentença, documento e de idiomas são revisados. Finalizam o artigo citando problemas abertos e desafios da

área, como *corpus* anotados para outros idiomas além do inglês e falhas de pré-processamento de textos de opinião.

Yao, Wan e Xiao (2017) oferecem uma visão abrangente do assunto de sumarização de textos. A revisão do artigo destaca os trabalhos importantes da área, separando em abordagens, métricas utilizadas, vantagens, desvantagens e conclusões. O artigo finaliza apontando tendências, como a utilização de redes neurais, sumarização abstrativa e análise semântica. Entre os desafios, são citadas a qualidade da avaliação de sumarização, dados disponíveis e capacidade de respostas em consultas.

Bai et al. (2019) oferecem uma análise do campo de sistemas de recomendação para artigos científicos. Os artigos de revisão foram classificados em quatro grupos, cada um deles representando um tipo de sistema de recomendação: filtragem baseada em conteúdo e colaborativa, métodos baseados em grafo e híbrido. Para cada tipo, foi abordado suas vantagens e desvantagens. A técnica mais utilizada é a híbrida e as métricas utilizadas para avalia esses sistemas são: precisão, *recall*, F-Measure, NDCG, MAP, MRR, MAE e UCOV. O trabalho conclui pontuando desafios da área, como: escalabilidade, serendipidade, privacidade, esparsidade e unificação de padrões de dados.

Gupta e Gupta (2019) revisam os trabalhos no campo de sumarização abstrativa. Artigos recentes do diretórios dos sites acadêmicos Elsevier, ACM, IEEE, Springer, *ACL Anthology*, Universidade de Cornell e Google Scholar foram eleitos para o estudo e depois agrupados a partir da técnica de sumarização abstrativa utilizada. A análise semântica e de discurso com técnicas recentes como modelos de redes neurais ajudam a solucionar as dificuldades da sumarização abstrativa.

Čebirić et al. (2019) fornecem uma pesquisa sobre sumarização de grafos semânticos baseados em estruturas de grafos RDF. Os principais métodos foram classificados através das seguintes categorias: reconhecimento de padrão, estrutural, estatístico e híbrido. O artigo é concluído enfatizando a abrangência do uso de grafos RDF, como a compreensão e consulta de dados e que a sumarização é sujeita ao modelo proposto nos trabalhos levantados. Como desafios futuros, pontuam: a qualidade do resumo e a padronização de fonte de dados.

Altmami e Menai (2020a) mostram o estado da arte do assunto de sumarização de textos científicos. O artigo realiza um estudo das técnicas utilizadas, conteúdo utilizado para a sumarização, métodos de avaliação, além de mostrar os problemas a serem resolvidos na área. O estudo também envolve as vantagens e limitações encontradas em trabalhos selecionados. Foi possível concluir que técnicas como TF-IDF, classificação por técnicas de aprendizado de má-

quina e clusterização são técnicas comuns para essa tarefa. A métrica ROUGE é amplamente utilizada para mensurar a qualidade da sumarização. Os autores finalizam que a utilização de redes neurais profundas é uma potencial técnica dado os bons resultados em tarefas relacionadas a NLP.

El-Kassas et al. (2020) fornecem uma pesquisa abrangente sobre a sumarização automática de textos, apresentando tópicos como abordagens, técnicas e conjuntos de dados. O trabalho de revisão mostra como os sistemas de sumarização automática de textos são classificados e suas aplicações em análises de textos. Entre as abordagens revisadas - extrativa, abstrativa e híbrida - são destacadas suas vantagens e desvantagens, assim como as técnicas e métodos são separados. Entre melhorias levantadas no artigo, estão: sumarização de múltiplos documentos, para outros idiomas além do inglês e qualidade da saída dos modelos de sumarização.

Iqbal e Qureshi (2020) revisam modelos de aprendizado profundo aplicados a geração de textos. A revisão aborda a aplicação de diferentes tipos de redes neurais relacionadas a tarefa de geração de texto. Entre os algoritmos mais populares para a geração de texto, estão o uso de RNNs, LSTMs, CNNs e GRUs. As técnicas citadas como mais recentes são: VAE (*Variational Auto-Encoders*) e GAN (*Generative Adversarial Network*). O artigo conclui que são necessárias melhorias na aplicação da área de NLP no contexto da pesquisa, além de novas métricas para avaliação do texto gerado por um sistema, pois os sistemas atuais não são totalmente eficazes para interpretar as nuances da linguagem de textos mais longos.

Merrouni, Frikh e Ouhbi (2020) fornecem uma revisão de trabalhos relacionados a extração de palavras-chave para identificar conteúdos em destaque. O artigo lista as principais aplicações de extração de palavras-chave, entre elas: sistemas de extração de informação, sumarização, agrupamento de documentos, mineração de texto, ontologia e sistemas de recomendação. As técnicas levantadas na revisão foram de aprendizagem supervisionada, não-supervisionada e profunda. Em geral, as práticas do assunto consistem nas fases de pré-processamento, identificação de frases em potencial, seleção e classificação das palavras-chave. Entre os desafios encontrados pelo artigo, podem ser destacados a redundância entre as palavras-chaves geradas, a concentração dos métodos em utilizar palavras frequentes e os métodos de avaliação.

2.6 ANÁLISE E VISUALIZAÇÃO DOS TRABALHOS RELACIONADOS

Os trabalhos avaliados nessa revisão possuem diversos atributos, dos quais selecionamos como fundamentais os cinco seguintes: objetivos, metodologias, ferramentas, métricas e bases

de dados. Para cada artigo, foi verificado quais desses atributos ocorrem. Cada um deles possui então um número de instâncias diferentes. Por exemplo, para os 63 artigos analisados, foram encontrados 43 objetivos, 69 metodologias, 90 ferramentas, 32 métricas e 47 bases de dados. Assim, foi construída uma tabela *Instância de Atributos* $\times$ *Artigos*. Caso uma instância i, específica de um dos cinco atributos acima, ocorra em um artigo j, então a célula i,j da tabela recebe o valor binário um (1); e zero (0) caso contrário. A Tabela 1 mostra um exemplo dessa ideia.

A Tabela 1 é chamada aqui de Tabela Binária de Atributos de Artigos (TBAA). Essa tabela quantifica o número de instâncias de atributos que aparecem em toda a revisão bibliográfica realizada, dando um parecer geral sobre as ocorrências mais abrangentes dos atributos que nortearão o trabalho aqui proposto.

	Objetivos			Metodologias		Ferramentas			Métricas		Bases de Dados	
	Identificar flores	Identificar árvores	Identificar frutas	Rotacionar Imagem	Filtro de Média	Random Forest	Rede Neural	SVM	*F-Score*	Acurácia	A	B
Artigo 1	1	0	0	1	1	1	0	1	1	1	1	0
Artigo 2	0	1	1	1	0	0	1	0	1	0	0	1
Artigo 3	0	0	1	0	1	0	1	1	0	1	1	0

Tabela 1 – Exemplo de uma tabela binária.

Fonte: Autora.

Assim sendo, os gráficos apresentados ao longo dessa seção representam a contagem dos atributos em destaque de cada elemento (observados na Tabela 1). Uma vez que atributos diferentes podem existir em uma mesma pesquisa - a contagem de cada atributo não representa exclusivamente um artigo.

A Figura 2 mostra os principais objetivos de pesquisa encontrados no levantamento bibliográfico em relação aos anos de publicação.

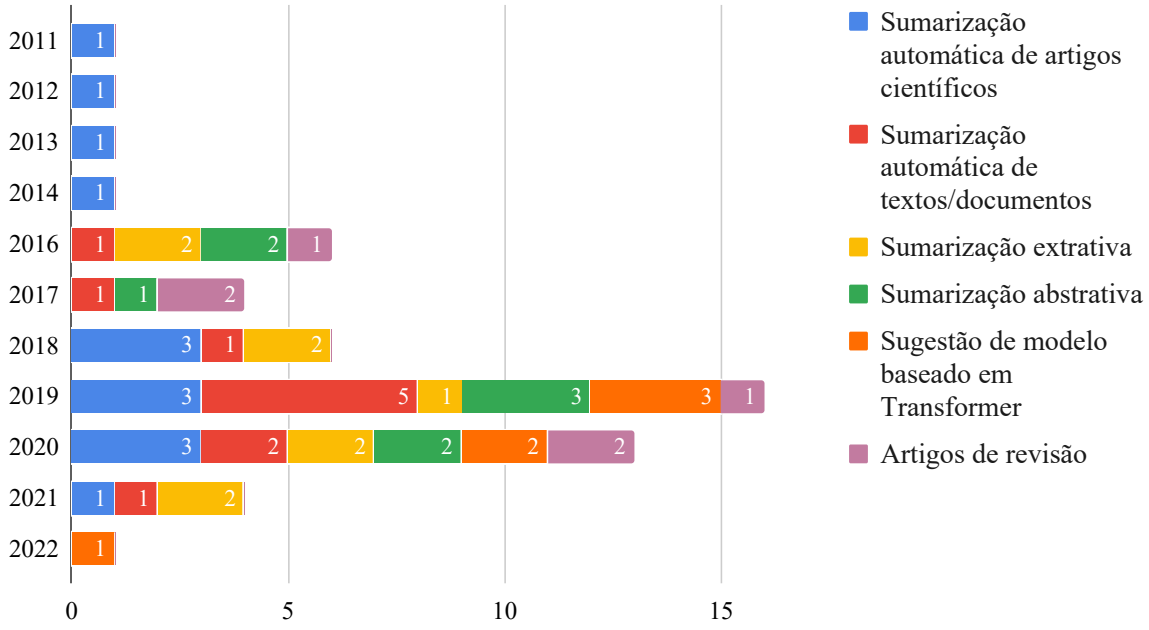

Figura 2 – Principais objetivos encontrados nos trabalhos relacionados. Como exemplo, no ano de 2018, foram levantados 5 trabalhos, dos quais 3 foram de sumarização automática de artigos científicos, 1 de sumarização automática de textos/documentos e 2 de sumarização extrativa.

Fonte: Autora.

Por sua vez, na Figura 3 são exibidos as principais metodologias realizadas de um total de 69 diferentes atributos. Os atributos de metodologia são as etapas, estratégias ou processos utilizados nas pesquisas. Métodos como técnicas de NLP, pré-processamento, classificação, cálculo de saliência de sentenças ou palavras são pontos comumente encontrados em trabalhos de sumarização de artigos científicos ou textos e documentos em geral.

Analogamente, ferramentas de classificação como SVM e aplicação do modelo BERT foram as mais encontradas nas pesquisas. De um total de 90, as principais são mostradas na Figura 4.

Em relação às métricas, as avaliações predominantes são as variações do ROUGE, amplamente utilizadas pela comunidade científica para sumarização, assim como métricas utilizadas em classificadores como *F-Measure*, precisão e acurácia, como mostra a Figura 5.

Na pesquisa foram encontradas 47 bases de dados, cujas principais são mostradas na Figura 6. O DUC (*Document Understanding Conferences*) e AAN (*ACL Anthology Network*) foram as mais utilizadas.

De acordo com a quantificação do levantamento bibliográfico mostrado aqui, nota-se que há um grande interesse em aplicações que visam a sumarização de textos científicos. Fo-

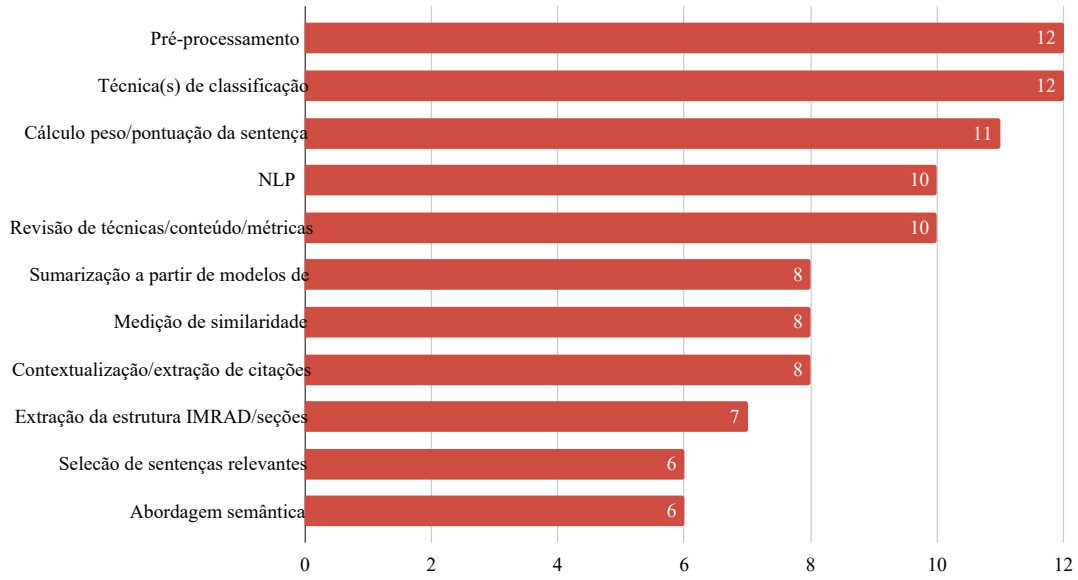

Figura 3 – Principais metodologias adotadas nos trabalhos relacionados.

Fonte: Autora.

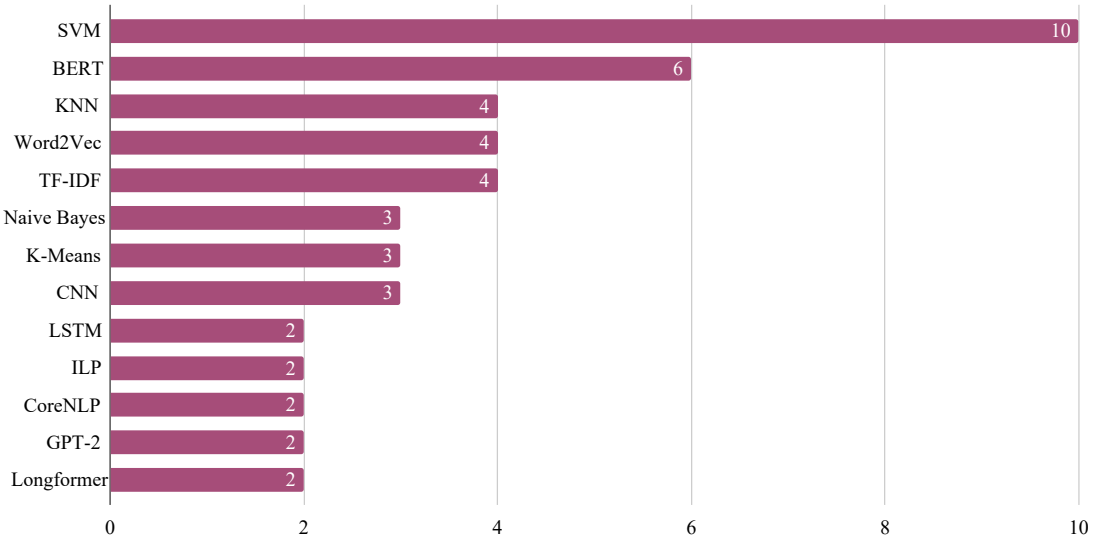

Figura 4 – Principais técnicas utilizadas nos trabalhos relacionados.

Fonte: Autora.

ram levantados cinco elementos principais que compõem a escrita de um texto acadêmico-científico: objetivos, metodologia, ferramentas, métricas e base de dados. Especificamente, os principais objetivos foram a sumarização automática de artigos científicos e documentos. Da mesma forma, as principais metodologias foram etapas de pré-processamento, uso de classificadores e cálculo de peso de sentenças. Para as ferramentas, nota-se que o SVM, BERT também

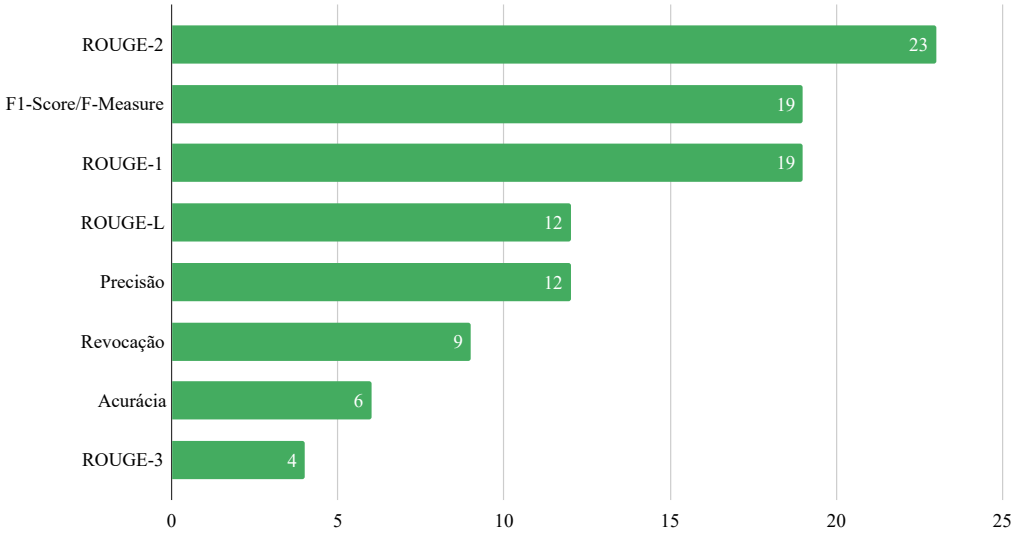

Figura 5 – Principais métricas e avaliações utilizadas nos trabalhos relacionados.

Fonte: Autora.

foram aquelas mais utilizadas. No caso específico de métricas, *F-Measure*, precisão e ROUGE foram as mais aplicadas para avaliação de textos. Finalmente, as bases de dados DUC e AAN apresentaram-se como aquelas de maior frequência.

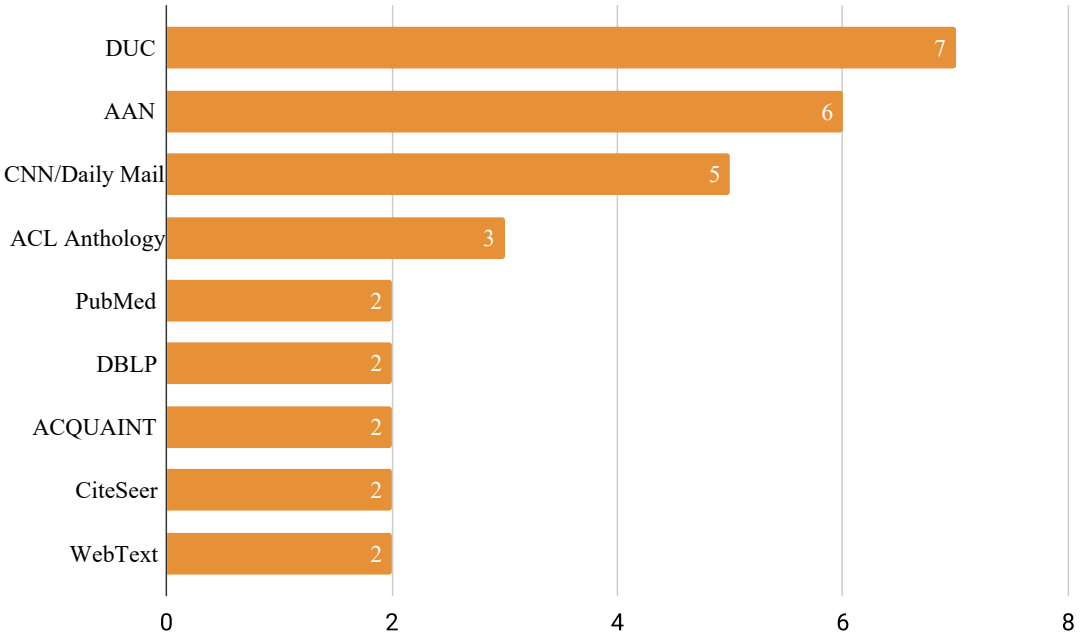

Figura 6 – Principais bases de dados utilizadas nos trabalhos relacionados.

Fonte: Autora.

3 CONCEITOS FUNDAMENTAIS

Neste capítulo serão abordados os conceitos fundamentais da pesquisa, em que serão apresentados os componentes da arquitetura *Transformer* e *Longformer* bem como os conceitos relacionados a atenção, processamento de texto e métricas.

3.1 ABORDAGENS DE SUMARIZAÇÃO DE ARTIGOS CIENTÍFICOS

No artigo de resumo apresentado por Altmami e Menai (2020a) são exemplificados duas abordagens para a sumarização de artigos científicos que serão apresentadas nas próximas subseções.

3.1.1 Sumarização Baseada em Resumo

Esse tipo de sumarização equivale a geração da seção de resumo do artigo científico. O processo consiste em determinar quais as seções do artigo científico serão utilizadas para posteriormente identificar seus principais tópicos e descobertas. As sentenças são escolhidas de acordo com sua relevância e as mais importantes são selecionadas para o resumo. A Figura 7 apresenta essa abordagem.

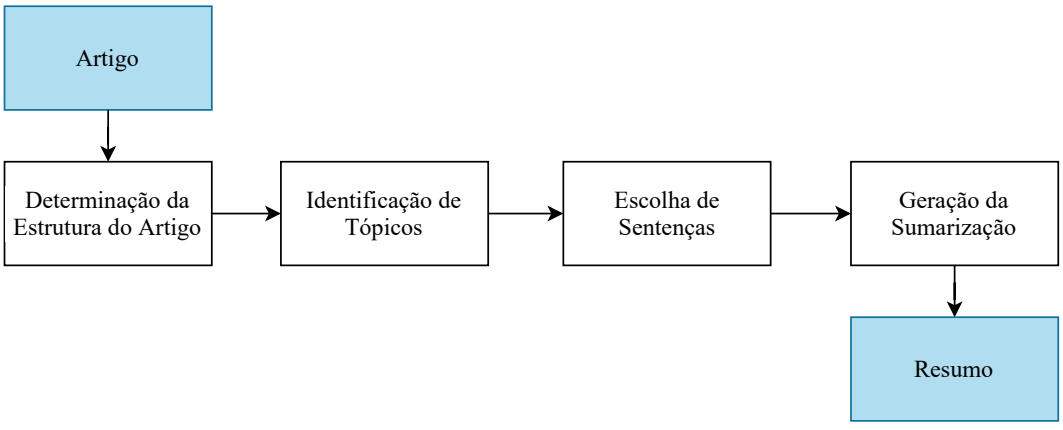

Figura 7 – Etapas para a sumarização de um artigo científico baseada em resumo.

Fonte: Adaptado de Altmami e Menai (2020a).

3.1.2 Sumarização Baseada em Citações

Na abordagem baseada em citação, os documentos dos artigos citados são recuperados e as sentenças com citação desses documentos são extraídas. Em seguida, o contexto da citação é obtido através de modelos de linguagem e construção de uma rede de citações. As etapas finais representam o agrupamento dos contextos obtidos pelas etapas anteriores e seleção das sentenças de diferentes grupos de contexto para a geração do resumo, conforme mostra a Figura 8.

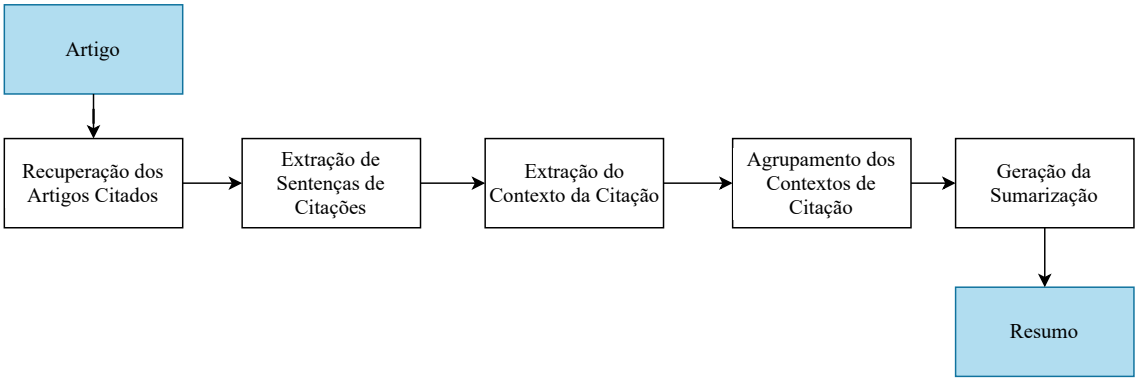

Figura 8 – Etapas para a sumarização de um artigo científico baseado em citações.

Fonte: Adaptado de Altmami e Menai (2020a).

3.2 SCISUMMNET

A base de dados padrão-ouro proposta por Yasunaga et al. (2019) para sumarização de artigos científicos, *SciSummNet*[1], é formada por 1009 resumos dos artigos de Linguística Computacional mais citados do AAN (*ACL Anthology Network*) e anotados por especialistas. A construção do *SciSummNet* levou 600 horas, com uma média de 151 palavras por resumo. O processo de anotação pode ser visto na Figura 9.

[1] `https://cs.stanford.edu/~myasu/projects/scisumm_net/`

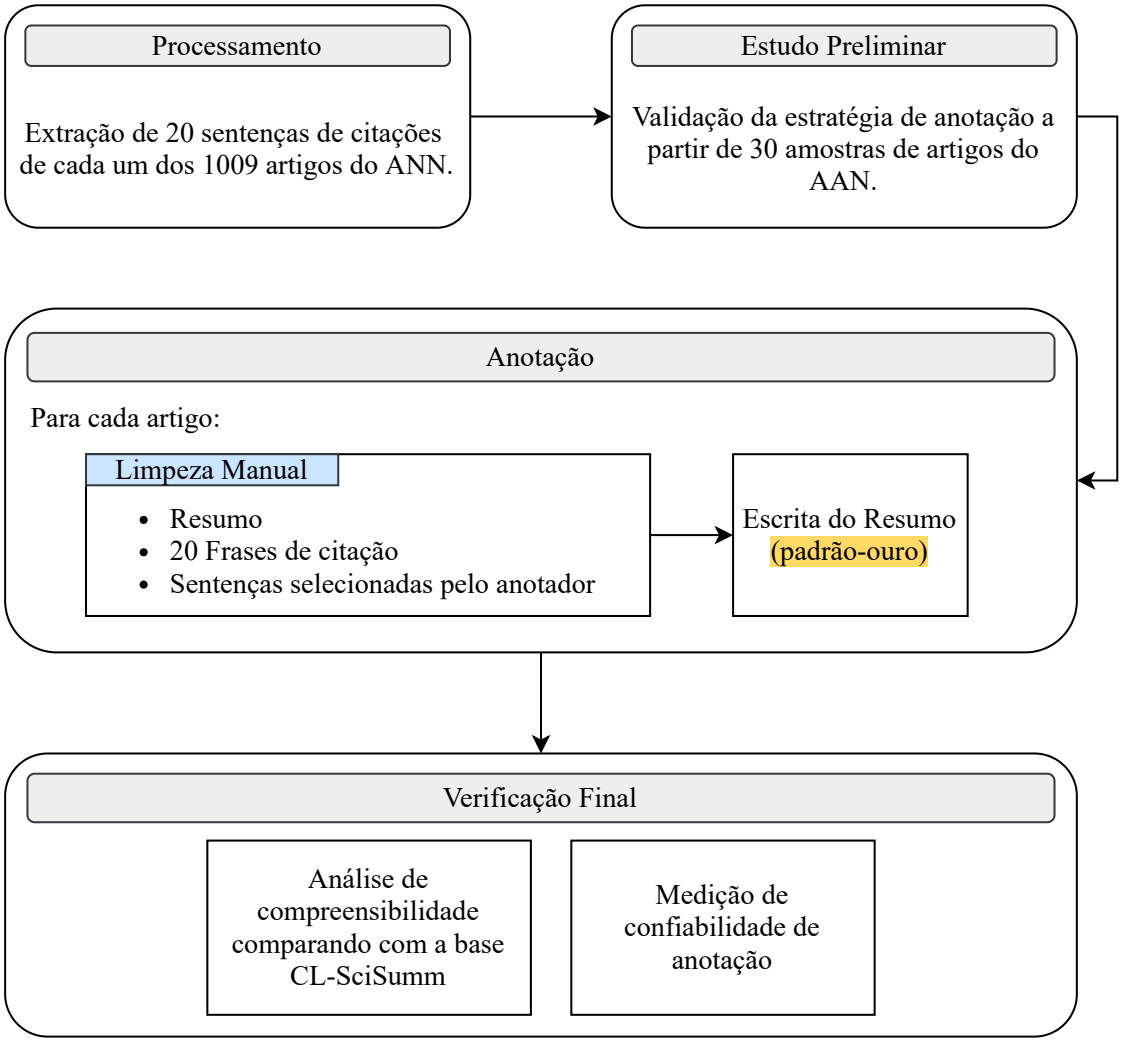

Figura 9 – Fluxo de trabalho para anotação da base de dados *SciSummNet*.

Fonte: Adaptado de Yasunaga et al. (2019).

A Etapa 1 (Processamento) consistiu em extrair 20 citações de cada um dos 1009 artigos científicos selecionados; as citações mais antigas, mais recentes e uma amostragem aleatória foram escolhidas para compor esse número. Após essa etapa, essas citações foram avaliadas e aquelas com defeito, referência a listas ou tabelas foram removidas. Deste modo, em média, 15 citações foram selecionadas por artigo científico.

Na Etapa 2 (Estudo Preliminar), 30 dos 1009 artigos científicos do conjunto de dados AAN foram selecionados e designados aos anotadores para a definição do método de anotação. Duas estratégias foram avaliadas: leitura do resumo original junto com as frases de citações selecionadas na Etapa 1 e leitura do artigo completo. A primeira estratégia foi escolhida por cobrir 90% dos principais pontos do artigo científico além de reduzir em 30% o tempo de anotação.

Na Etapa 3 (Anotação), os 1009 artigos foram divididos entre 5 especialistas na área de NLP, e para cada artigo, receberam o texto do resumo original e as frases de citações selecionadas na Etapa 1. Em seguida, os anotadores identificaram frases de citações relevantes que não foram contempladas no resumo original e complementam o resumo com informações consideradas relevantes do artigo. A escrita do resumo padrão-ouro foi realizada com base no resumo original do artigo, nas frases de citações selecionadas na Etapa 1 e nas frases escolhidas pelo anotador. Ao final, o processo foi revisado para prevenção de falhas.

A Etapa 4 (Verificação Final) foram extraídos 15 artigos do *SciSummNet* em comum com a base *CL-SciSumm* - construída lendo o artigo completo - e enviados a estudantes. Na perspectiva dos estudantes em relação a compreensibilidade dos resumos, 54% consideraram que as bases são semelhantes nesse aspecto, 22% consideraram o *SciSummNet* mais compreensível, e 20% o *CL-SciSumm*. Além disso, 40 artigos resumidos pelos especialistas foram enviados a um outro anotador para calcular o coeficiente Kappa a fim de medir de confiabilidade de anotação. O resultado do coeficiente Kappa atingiu 0,75 na escala Landis e Koch (1977) que representa uma confiabilidade forte de acordo com a Tabela 2.

Coeficiente Kappa	**Confiabilidade**
<0.00	Pobre
0.00 - 0.20	Fraca
0.21 - 0.40	Razoável
0.41 - 0.60	Moderada
0.61 - 0.80	Forte
0.81 - 1.00	Quase Perfeita

Tabela 2 – Classificação de confiabilidade do coeficiente Kappa.

Fonte: Traduzido de Landis e Koch (1977).

3.3 TOKENIZAÇÃO

A *tokenização* é uma ação realizada em um conjunto de caracteres a fim de separar seu conteúdo em pedaços de acordo com a definição de Christopher, Prabhakar e Hinrich (2008); cada pedaço é considerado um *token* podendo este ser à nível de sentenças, letras, palavras ou sub-palavras.

Existem diversas técnicas para a separação do texto em *tokens*, desde formas simples utilizando expressões regulares e separadores até técnicas mais complexas utilizando probabilidade e baseados em métodos neurais. Um dos desafios das abordagens de tokenização é

obter a menor incidência possível de *tokens* atribuídos como [OOV] (*Out Of Vocabulary*) ou [UNK] (*Unknown*) na etapa de incorporação a fim de preparar o modelo para reconhecimento do contexto a ser trabalhado.

3.3.1 WordPiece

A técnica de tokenização WordPiece, baseada em sub-palavras, foi proposta pelo Google no artigo de Schuster e Nakajima (2012). O WordPiece utiliza o vocabulário para definir a separação de cada palavra a partir da quantidade de ocorrências no vocabulário. Tendo como exemplo a palavra em inglês *books*, a sub-palavra mais comprida encontrada no vocabulário será *book*. Assim, a palavra é dividida em [$book$, ##s] sendo ##s o segundo *token*. A Tabela 3 apresenta exemplos da aplicação do WordPiece.

Palavras	***Token(s)***
surf	['surf']
surfing	['surf', '##ing']
surfboarding	['surf', '##board', '##ing']
surfboard	['surf', '##board']
snowboard	['snow', '##board']
snowboarding	['snow', '##board', '##ing']
snow	['snow']
snowing	['snow', '##ing']

Tabela 3 – Exemplo de *tokens* gerados pelo WordPiece.

Fonte: Briggs (2021).

3.3.2 BPE

O BPE (*Byte Pair Enconding*) introduzido por Gage (1994) é uma abordagem híbrida de tokenização abrangente a nível de palavra e caractere. Assim como o WordPiece (Seção 3.3.1), utiliza análise estatística do vocabulário para a definição dos *tokens*.

Primeiramente, o *token* </w> é inserido no final de cada palavra para calcular as ocorrências no texto. Em seguida, é calculado a ocorrência de cada caractere das palavras do texto. Após esses dois passos, a frequência de pares consecutivos é contabilizada até o fim da iteração limite.

Por exemplo, considerando que as palavras em inglês e suas ocorrências no texto: [(car</w>, 4), (far</w>, 3), ($cars$</w>, 2)], a sub-palavra ar é a mais comum nesse vocabulário de tama-

nho 2, então esses caracteres seriam separados da seguinte forma: [(c_ar</w>, 4), (f_ar</w>, 3), (c_ar_s</w>, 2)]. A próxima subsequência mais frequente é a palavra car com tamanho 3, seguindo da forma: [(car</w>, 4), (f_ar</w>, 3), (car_s</w>, 2)]. Esse procedimento é realizado até o fim da iteração, que consiste na definição de *tokens* desejados para o vocabulário.

3.4 MODELO SEQUENCE-TO-SEQUENCE

O modelo neural *sequence-to-sequence*, abreviado para *seq2seq*, também é conhecido como arquitetura *Encoder-Decoder*. É caracterizado para a modelagem e treinamento de tarefas em que a sua entrada e saída são itens, tais como áudio, imagem ou texto. No contexto de processamento de texto, são aplicados aos segmentos como QA, tradução e sumarização. Proposto inicialmente por Sutskever, Vinyals e Le (2014) para a tarefa de tradução, opera com dois componentes principais: o codificador e o decodificador. Respectivamente, o codificador atua no processamento da entrada e constrói sua representação vetorial para o decodificador produzir a saída. A Figura 10 demonstra sua forma tradicional em alto nível.

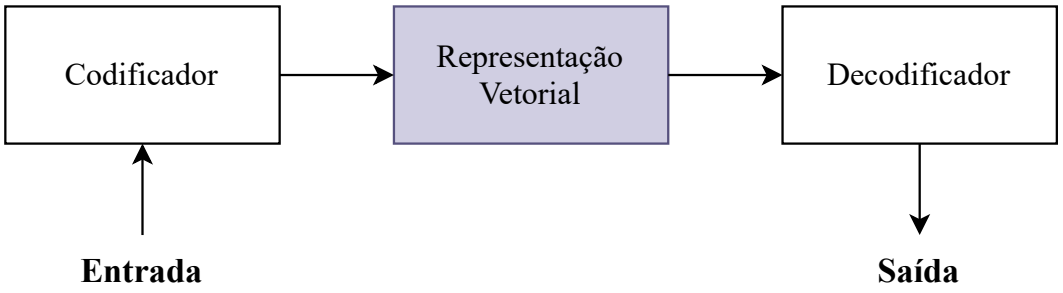

Figura 10 – Exemplo do modelo *seq2seq* a alto nível.

Fonte: Autora.

Na proposta orignal de Sutskever, Vinyals e Le (2014), a partir da Figura 11, dada a representação vetorial dos elementos de entrada (ABC), o decodificador inicia o processamento pelo *token* <BOS> (*Beggining of Sentence*) para indicar o início da saída. O primeiro elemento de saída, W, é obtido a partir da representação com maior probabilidade de início da sentença dado o contexto do codificador. O elemento de saída W é utilizado como referência para obter o próximo, e assim sucessivamete, até que o *token* <EOS> (*End of Sentence*) seja obtido, indicando a finalização do processamento de saída. Posto isso, a entrada ABC produz a saída WXYZ.

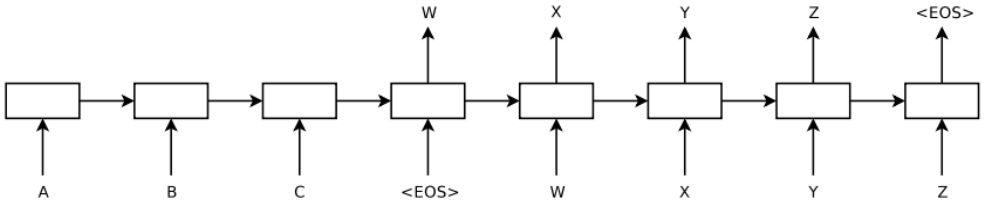

Figura 11 – Exemplo do funcionamento do modelo *seq2seq*.

Fonte: Sutskever, Vinyals e Le (2014).

3.5 REDE NEURAL FEED-FORWARD

A rede *feed-forward* é caracterizada por seus neurônios serem conectados com os próximos, em apenas uma direção, como mostra a Figura 12. Os neurônios são organizados em camadas, como as de entrada (*Input Layer*), escondida (*Hidden Layer*) e saída (*Output Layer*).

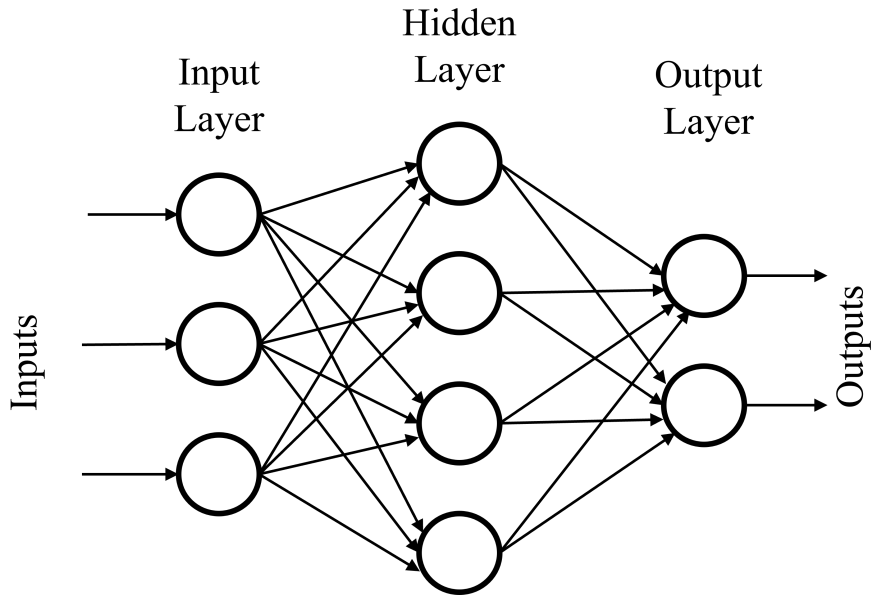

Figura 12 – Rede neural *Feed-Forward*.

Fonte: Adaptado de Quiza e Davim (2011).

De acordo com a Figura 13, sendo x_1, x_2 e x_3 as entradas do neurônio, e w_1, w_2 e w_3 os pesos conectados ao neurônio, a função de ativação linear (*Linear Activator*) recebe o somatório do produto entre as entradas x_n e os pesos w_n, podendo um valor de bias b ser adicionado. A função de inibição não-linear (ou função de ativação, podendo ser também linear) é aplicada para determinar o valor de saída y.

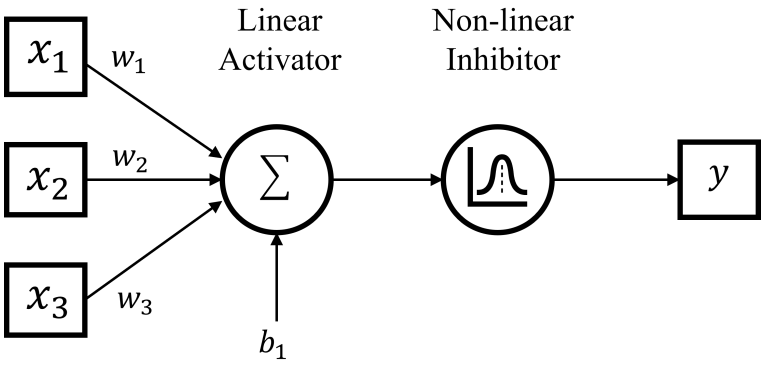

Figura 13 – Esquema lógico de um neurônio.

Fonte: Adaptado de Quiza e Davim (2011).

3.6 MODELO DE ATENÇÃO

As definições de Xu et al. (2015) e Chaudhari et al. (2021) retratam que atenção é fundada através do sistema biológico humano que tende a selecionar partes relevantes da percepção para a interpretação de um contexto. Nesse âmbito, os contextos relacionados a linguagem, visão, diálogos e texto, fragmentos dessas informações são mais importantes do que outras em sua conjuntura.

O primeiro trabalho conhecido na comunidade científica que utiliza os princípios de atenção foi de Nadaraya (1964) no qual realizava métodos para medir a relevância de pesos durante o treinamento.

Desta forma, o modelo matemático genérico é composto pelos vetores Q (*query*) de consulta, K (*key*) de chave e V (*value*) de valor para o cálculo de atenção. De modo geral, o vetor Q armazena as consultas de cada chave em relação a todas as chaves do modelo, atribuindo o valor do vetor V para indicar a importância de uma chave em relação as demais. Essa operação é apresentada na Equação 1.

$$Attention(q, K, V) = \sum_i p(a(q_i), k) \times v \tag{1}$$

3.7 ARQUITETURA TRANSFORMER

Proposta por Vaswani et al. (2017) e dispensando recorrências e convoluções, o modelo de arquitetura *Transformer* é a primeira proposta que aplica mecanismo de auto-atenção ba-

seado em transdução associado ao modelo *seq2seq* (Seção 3.4). O esquema geral arquitetura *Transformer* pode ser visto na Figura 14.

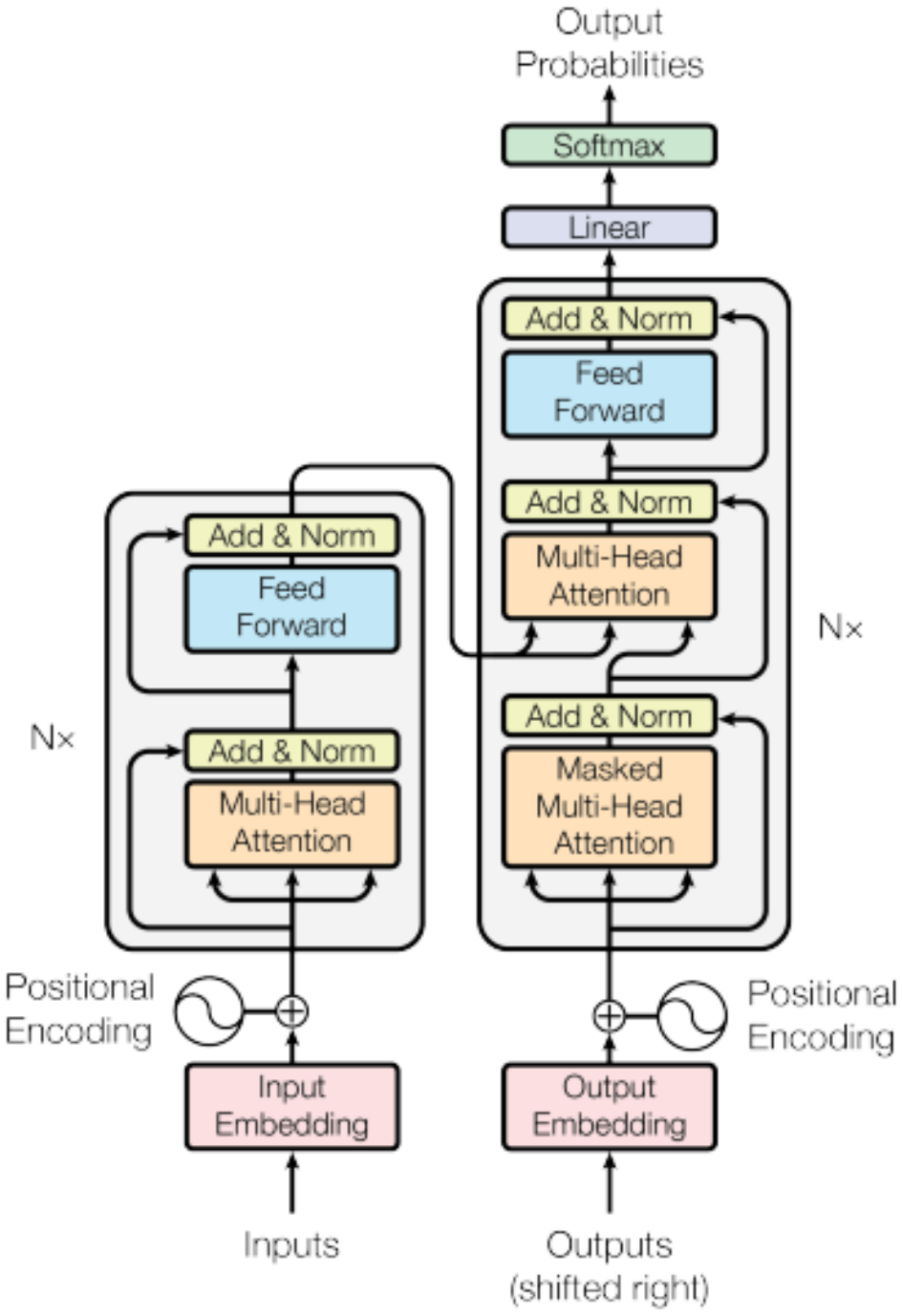

Figura 14 – Arquitetura *Transformer*.

Fonte: Vaswani et al. (2017).

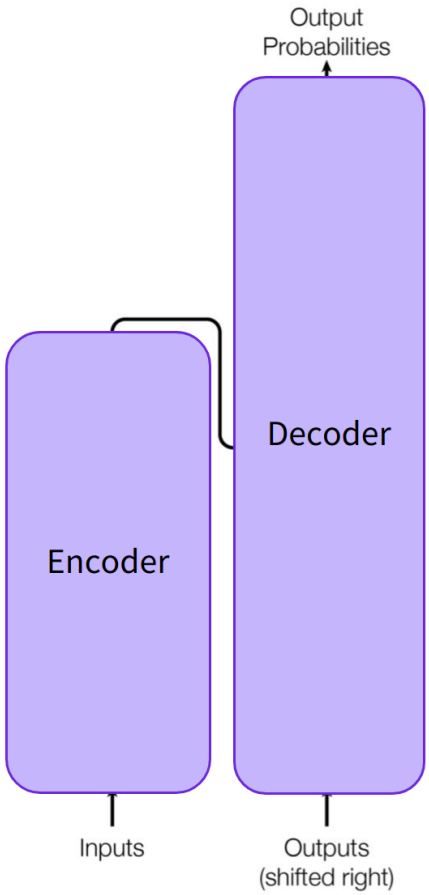

Figura 15 – Blocos de codificação (*encoder*) e decodificação (*decoder*) da arquitetura *Transformer*.

Fonte: Hugging Face (s.d.).

3.7.1 Arquitetura Encoder-Decoder no Transformer

A Figura 15 mostra o blocos que correspondem ao *encoder* e *decoder* pela aplicação da arquitetura *Encoder-Decoder* (Seção 3.4) no *Transformer*. A quantidade de camadas do *encoder* e *decoder* no modelo da arquitetura *Transformer* é representada por Nx na Figura 14, e o valor padrão utilizado no artigo original é de 6 camadas.

Para cada bloco de *encoder* e *decoder* da Figura 15, existem 6 camadas empilhadas e conectadas, como mostra a Figura 16. As camadas funcionam de forma autoregressiva; a saída da camada anterior é a entrada da próxima camada.

Cada camada *encoder* da Figura 16 contém duas subcamadas: *Multi-Head Attention* (Seção 3.7.3) e *Feed-Forward* (Seção 3.7.4). A etapa de *Add & Norm* tem como resultado a saída normalizada de acordo com a Equação 2 onde $Sublayer(x)$ é a função implementada em cada subcamada.

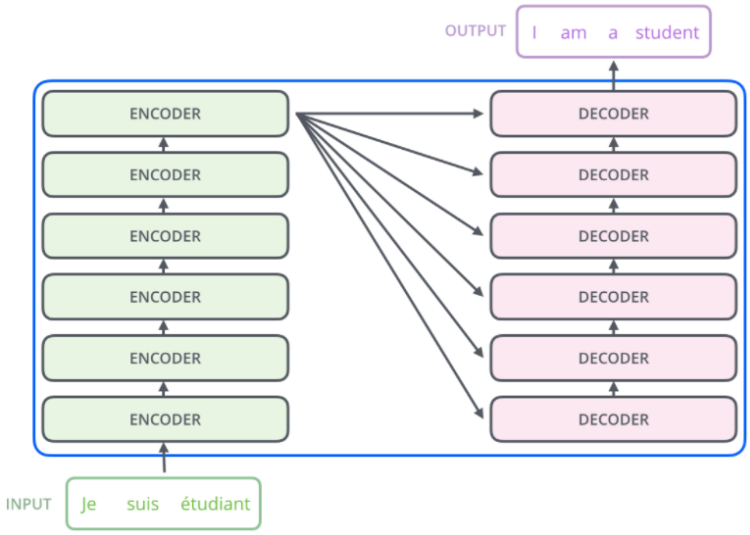

Figura 16 – Representação do empilhamento e conexão entre as camadas de *encoder* e *decoder* no *Transformer*. A saída do último *encoder* é aplicada para todos os *decoders*. O exemplo de entrada e saída corresponde a tarefa de tradução do francês para o inglês.

Fonte: Alammar (2018).

$$LayerNorm(x + Sublayer(x)) \tag{2}$$

Continuando na Figura 16, o *decoder* é composto pelas duas subcamadas do *encoder* porém adicionando uma subcamada extra chamada *Masked Multi-Head Attention*.

3.7.2 Scale Dot-Product Attention

O cálculo de atenção, proposto na arquitetura *Transformer*, também chamado *Scale Dot-Product Attention*, é calculado através da matriz de consulta Q e vetores de chave-valor, respectivamente K e V. O cálculo final de atenção de uma sentença é mostrado na Equação 3.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

O cálculo de atenção por palavra pode ser visto na Equação 4, na qual o valor de atenção da palavra, a_i, é o somatório da atenção da palavra atual em relação a cada palavra da sentença. Os componentes da fórmula são:

a) i corresponde a palavra atual;

b) j corresponde as demais palavras da sentença;

c) q_i é o valor da palavra atual do vetor Q;

d) k_j é o valor correspondente a cada palavra da sentença do vetor K;

e) n é o número de palavras na sentença;

f) a_i é a pontuação de atenção da palavra i.

g) V_i é o valor do vetor V da palavra atual.

h) d_{k_j} é a dimensão do vetor K.

$$a_i = \sum_{j=0}^{n} \left[softmax \left(\frac{q_i k_j}{\sqrt{d_{k_j}}} \right) V_i \right] \tag{4}$$

O produto escalar $q_i k_j$ é dividido por $\sqrt{d_k}$ para que evitar que a função $softmax$ resulte em regiões de gradientes pequenos como consequência de um valor $\sqrt{d_k}$ alto.

A Figura 17 apresenta o procedimento descrito nessa subseção. O produto escalar é calculado entre K e Q, em seguida é realizada a divisão por $\sqrt{d_k}$, aplicação do *softmax* e por fim a multiplicação pelo valor do vetor V.

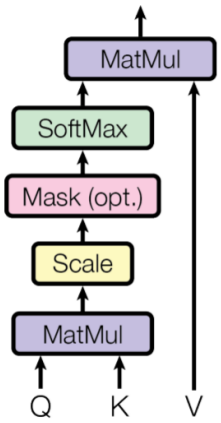

Figura 17 – Cálculo de atenção realizado pelo *Scaled Dot-Product Attention*.

Fonte: Vaswani et al. (2017).

3.7.3 Multi-Head Attention

A proposta do *Multi-Head Attention* é projetar h (*head*) conjuntos de vetores Q, K e V, respectivamente com dimensões d_q, d_k e d_v; o valor de h proposto no artigo original de Vaswani et al. (2017) é 8. Posterior a projeção, a atenção é calculada pelo *Scaled Dot-Product Attention* (Seção 3.7.2, Figura 17) paralelamente resultando em d_v saídas. Ao final, os resultados projetados são concatenados, gerando os valores finais. A Figura 18 exibe o fluxo realizado pela subcamada de *Multi-Head Attention*.

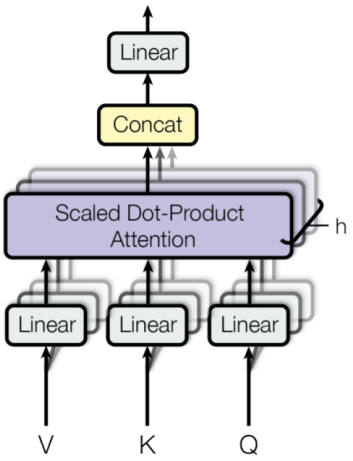

Figura 18 – Funcionamento da subcamada *Multi-Head Attention*.

Fonte: Vaswani et al. (2017).

A subcamada extra existente no *decoder*, chamada *Masked Multi-Head Attention*, utiliza uma máscara na operação de *Scaled Dot-Product Attention* da Figura 17 nomeada como $Mask(opt)$. Esse procedimento consiste em definir como $-\infty$ os valores de entrada da função *softmax* para as posições futuras, para que o modelo apenas conheça as posições de saída precedentes, preservando sua característica autoregressiva.

3.7.4 Rede Neural Feed-Forward na Arquitetura Transformer

A rede neural *Feed-Forward* (Seção 3.5) é aplicada para cada posição do *Transformer*, com duas transformações lineares e função de ativação *ReLU*, conforme a Equação 5.

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{5}$$

3.7.5 Positional Encoding

A arquitetura *Transformer* não armazena as posições das palavras que são processadas no modelo. Para este fim, a ordem das posições são calculadas pelo componente *Positional Encoding* (Figura 14) na entrada do *encoder* e na saída do *decoder* através das Equações 6 e 7; onde pos é a posição e i a dimensão da incorporação. Dessa forma, a posição é representada por um comprimento de onda que varia entre 2π até $10000 \times 2\pi$, permitindo que o modelo observe a ordem das palavras.

$$PE(pos, 2i) = \sin\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right) \tag{6}$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right) \tag{7}$$

3.8 BERT

O BERT (*Bidirectional Encoder Representations from Transformers*) exposto no artigo de Devlin et al. (2018), é um modelo pré-treinado de *Transformer* (Seção 3.7) utilizando apenas o componente *encoder* da arquitetura (Figura 15). O diferencial proposto é sua representação bidirecional do lado direito e esquerdo profunda em todas as camadas, que permite a arquitetura processar o contexto da palavra de acordo com as palavras ao seu redor.

O modelo BERT é processado em duas etapas: pré-treinamento (Seção 3.8.1) e *fine-tuning* conforme a Figura 19 apresenta.

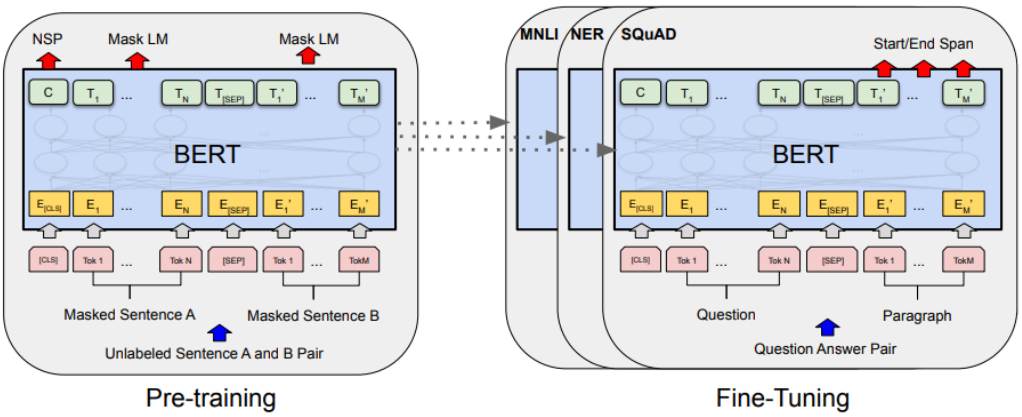

Figura 19 – Etapas do modelo BERT.

Fonte: Devlin et al. (2018)

3.8.1 Pré-Treinamento

O pré-treinamento do BERT é realizado a partir de duas tarefas não supervisionadas, conforme as próximas seções apresentam, correspondendo ao lado esquerdo (*Pre-Training*) da Figura 19.

3.8.1.1 Masked LM (MLM)

A fim de adaptar o pré-treinamento a característica de bidirecionalidade do BERT e evitar que o modelo visualize a própria palavra, é aplicada uma técnica chamada *Masked LM* conhecida na literatura por Taylor (1953).

A tarefa representa a inclusão de uma máscara nomeada [MASK] que é inserida em até 15% dos *tokens* de forma aleatória. Desse modo, o modelo poderá prever quais serão os *tokens* que estão com a máscara [MASK].

3.8.1.2 Next Sentence Prediction (NSP)

O *Next Sentence Prediction* tem a finalidade de introduzir ao modelo BERT a relação entre sentenças, especificamente para tarefas onde serão necessárias a predição da próxima sentença, como QA (*Question Answering*) e NLI (*Natural Language Inference*). A estratégia sugerida é que para sentenças relacionadas A e B, a sentença B será classificada como a próxima de A em 50% das vezes ou uma sentença aleatória será classificada como a próxima de A.

3.8.2 Representação de Entrada

A representação de entrada no modelo pode ser uma sentença ou um par de sentenças. É utilizada a incorporação WordPiece proposto por Schuster e Nakajima (2012) com um vocabulário de 30 mil *tokens*. A forma de *tokenização* (Seção 3.3) aplicada a técnica WordPiece.

A Figura 20 apresenta a representação de duas sentenças: *my dog is cute* e *he likes playing*. O primeiro *token* de cada sentença é representado por uma sigla de classificação especial denominada [CLS]. Na situação de um par de sentenças, as sentenças são separadas pelo *token* [SEP]. A representação de entrada, *Input*, é a soma das incorporações *Token Embeddings*, *Segment Embeddings* e *Position Embeddings*, respectivamente: o *embedding* de cada palavra, o *embedding* de qual sentença pertence e o *embedding* de sua posição na entrada.

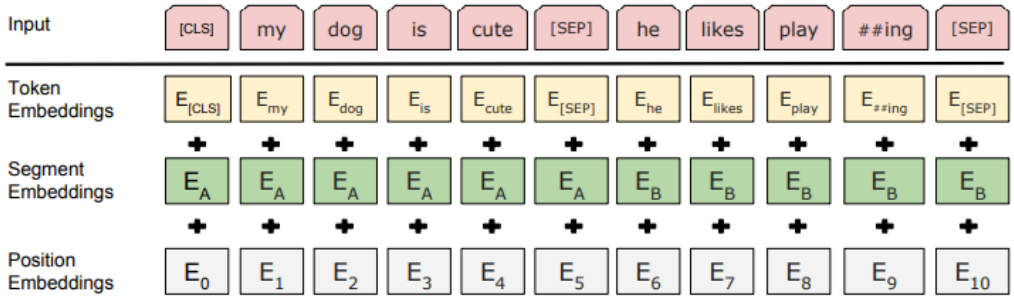

Figura 20 – Representação de entrada do BERT.

Fonte: Devlin et al. (2018).

3.9 ROBERTA

A fim de estudar o método de pré-treinamento do modelo BERT (Seção 3.8), Liu et al. (2019) proporam o RoBERTa. As principais modificações propostas implementadas no RoBERTa foram: MLM (Seção 3.8.1.1) dinâmico, a remoção do processo de NSP, aumento dos lotes de treinamento e da incorporação. As mudanças propostas são exemplificadas abaixo:

a) O MLM dinâmico do modelo RoBERTa consiste em mudar o *token* mascarado gerando um novo padrão de mascaramento a cada sentença incluída no modelo.

b) Na ação de remoção do NSP, um *token* separador de documentos é implementado. A cada entrada do documento, as sentenças são amostradas com a próxima sentença, limitado ao tamanho de 512 *tokens*. As sentenças podem cruzar um documento e outro, e a cada documento novas frases são selecionadas.

c) O aumento dos *batches* - ou lotes de treinamento - melhora o índice de perplexidade do modelo além de viabilizar a paralelização e permitir o treinamento com sequências mais longas.

d) A tokenização a nível de *bytes* utilizando o BPE (Seção 3.3.2) que no modelo BERT suportaria 30K sub-palavras a nível de caractere pelo WordPiece (Seção 3.3.1), e com a alteração proposta poderia chegar a 50K sub-palavras.

3.10 LONGFORMER

Considerando a arquitetura padrão do Transformer (Seção 3.7), limitado ao valor padrão de 512 *tokens*, o mecanismo de atenção é calculado para todas as possibilidades de *tokens*

de entrada. Devido ao alto custo em termos de processamento e memória, resulta em uma complexidade $\mathcal{O}(n^2)$ - sendo n a sequência de entrada - conforme a Figura 21 demonstra.

Um tratamento comum para endereçar a limitação *tokens* do Transformer é empilhar a entrada do modelo em segmentos de 512 *tokens*. Como consequência, essa abordagem possibilita a ocorrência de perda de informação e erros de processamento no procedimento de combinação.

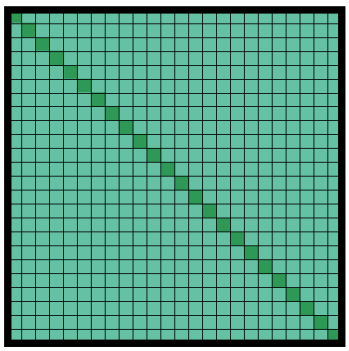

Figura 21 – Demonstração do cálculo de atenção total da arquitetura *Transformer* no qual os blocos em verde escuro são selecionados para leitura da atenção e os blocos em tom mais claro representam os locais no qual o cálculo de atenção são executados.

Fonte: Beltagy, Peters e Cohan (2020).

A Figura 22 demontra as variações implementadas do modelo *Longformer* de Beltagy, Peters e Cohan (2020), nos quais:

a) *Full self-attention*: implementa a auto-atenção total típica da arquitetura *Transformer* (Seção 3.7.2);

b) *Longformer-loop*: implementa via PyTorch, suporta a operação de dilatação e a atenção é calculada em *loop* nas diagonais.

c) *Longformer-chunks*: não suporta a dilatação e foi utilizado para o procedimento de *fine-tuning* devido a otimização de tempo de processamento em uma única operação mesmo com o consumo de memória sendo 2 vezes maior que a implementação mais otimizada.

d) *Longformer-cuda*: variação completa utilizando um *kernel* customizado do CUDA por meio de TVM (Tensor Virtual Machine) proposto por Chen et al. (2018).

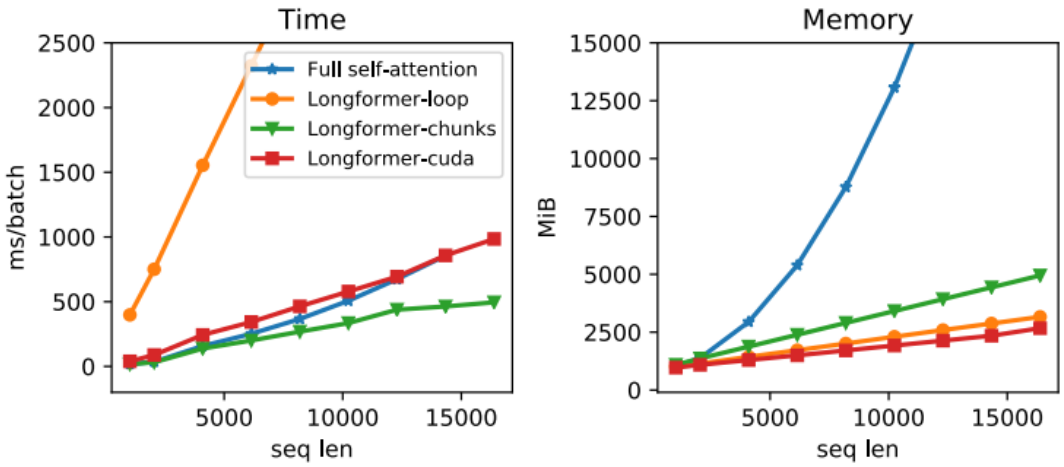

Figura 22 – Utilização de tempo e memória para as variações implementadas do modelo *Longformer*.

Fonte: Beltagy, Peters e Cohan (2020).

3.10.1 Cálculo de Atenção

O modelo *Longformer* propõe modificações no mecanismo de auto-atenção a fim de viabilizar o processamento de longas sequências de texto. Essa modificação consiste em calcular esparsamente a multiplicação da matriz de atenção, alterando a complexidade do cálculo de atenção para $\mathcal{O}(n)$. Os métodos de atenção *Sliding Window* (Seção 3.10.1.1), *Dilated Sliding Window* (Seção 3.10.1.2) e *Global Attention* (Seção 3.10.1.3) foram abordados em sua implementação.

Referente ao cálculo de atenção do *Transformer* da Equação 3, o *Longformer* utiliza os vetores Q_s, K_s e V_s para computar o cálculo de atenção do método *Sliding Window* e Q_g, K_g e V_g para o método *Global Attention*. Os vetores do método *Sliding Window* são utilizados para inicializar o cálculo de atenção do método *Global Attention*.

3.10.1.1 Sliding Window

A atenção é calculada ao redor de cada *token*. Dado um tamanho w de *tokens* ao redor de um valor da sequência n, metade do valor de w é aplicado para cada lado do *token* em questão, conforme ilustra a Figura 23. A complexidade se torna linear, $\mathcal{O}(n \times w)$.

Quando esse método é aplicado de forma empilhada, as camadas superiores acessam as camadas inferiores, de modo similar à arquitetura de uma Rede Neural Convolucional. Nesse caso, a complexidade é $\mathcal{O}(\ell \times w)$ sendo ℓ o número de camadas.

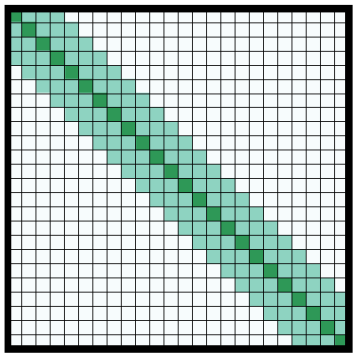

Figura 23 – Demonstração do cálculo de atenção aplicando o método de *Sliding Window*. Cada bloco em verde escuro opera com a atenção calculada em $w = 3$ blocos ao redor do *token* de referência.

Fonte: Beltagy, Peters e Cohan (2020).

3.10.1.2 Dilated Sliding Window

Com o intuito de ampliar a extensão do foco do cálculo de atenção sem afetar a complexidade, o método *Dilated Sliding Window* seleciona os *tokens* para o cálculo de atenção a partir de um valor de dilatação de tamanho d e de amplitude w. A complexidade desse método é de $\mathcal{O}(\ell \times d \times w)$. A Figura 23 demonstra esse comportamento.

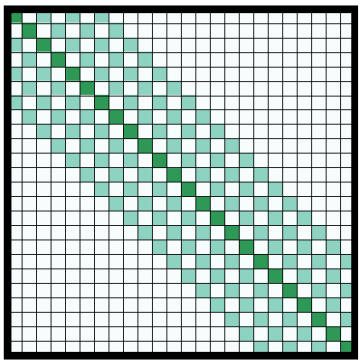

Figura 24 – Demonstração do cálculo de atenção aplicando o método de *Dilated Sliding Window*. Neste exemplo, $w = 6$ e a dilatação $d = 1$.

Fonte: Beltagy, Peters e Cohan (2020).

3.10.1.3 Global Attention

Segundo os autores do *Longformer*, as estratégias de *Sliding Window* e *Dilated Sliding Window* não são flexíveis para representações específicas de tarefas.

Seguindo o exemplo de modelos inspirados no BERT (Seção 3.8), a técnica de MLM (Seção 3.8.1.1) é aplicada para predição de palavras, o *token* especial CLS para classificação e auto-atenção para tarefas QA. Dessa forma, a complexidade do cálculo de atenção continua $\mathcal{O}(n)$ e atende ao $bias$, ou viés, da representação da tarefa.

Os *tokens* de referência escolhidos baseiam-se na tarefa; no caso da classificação, o *token* CLS é utilizado de referência. A Figura 25 apresenta a proposta desse método.

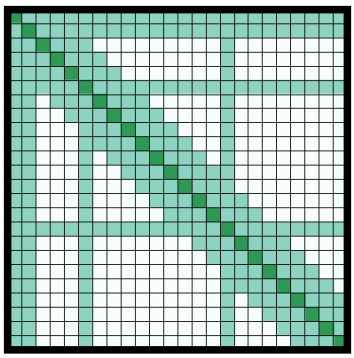

Figura 25 – Demonstração do cálculo de atenção aplicando o método de *Global Attention*. Tendo como exemplo o *token* localizado na primeira posição da linha e coluna selecionado para o método de *Global Attention*, todos os *tokens* simétricos a ele atendem o *token* de referência.

Fonte: Beltagy, Peters e Cohan (2020).

3.10.2 Modelagem Autoregressiva

A modelagem autoregressiva de linguagem consiste na aplicação da distribuição de probabilidade de uma saída de acordo com os dados anteriores; essa abordagem também é conhecida por modelagem *left-to-right* (esquerda para direita). Segundo o levantamento de Beltagy, Peters e Cohan (2020), é um elemento fundamento para processamento de linguagem e utilizada especificamente para processamento de longas sequências de texto usando *Transformers* por Sukhbaatar et al. (2019), Rae et al. (2019) e Dai et al. (2019).

Na implementação do *Longformer*, a autoregressão operou por meio de uma janela deslizante dilatada de valor j, seguindo o direcionamento de Sukhbaatar et al. (2019). Conforme o processamento da aprendizagem, o valor j aumenta; para que no início o contexto inicial

imediato seja instruído ao modelo, e no fim para obter a capacidade de compreender o contexto local.

3.10.3 Pré-Treinamento e Fine-Tuning

O pré-treinamento do *Longformer* utilizou a técnica de MLM (Seção 3.8.1.1) do modelo BERT (Seção 3.8) que objetiva aprender os *tokens* mascarados aleatoriamente. O treinamento foi realizado a partir do *checkpoint* do modelo RoBERTa (Seção 3.9) com os ajustes necessários para operar os cálculos de atenção propostos na Seção 3.10.1. O pré-treinamento foi aplicado no *corpus* de documentos longos fairseq referente ao trabalho de Ott et al. (2019).

A etapa de *Position Embeddings*, ou incorporação de posição, utilizando a incorporação de tamanho 512 do RoBERTa para inicialização, com o limite de tamanho 4096; essa ação, segundo os idealizadores do *Longformer*, provou um forte viés de contexto local em função do armazenamento da estrutura local de forma geral no modelo, permitindo a convergência rápida do *Longformer* com poucas atualizações de gradiente. Os hiperparâmetros e pesos do modelo RoBERTa foram reaproveitados a fim de garantir os resultados do modelo em documentos curtos, e a abordagem incrementou somente as novas incorporações.

3.11 MÉTRICAS

As métricas empregadas neste trabalho, ROUGE (Seção 3.11.1) e perplexidade (Seção 3.11.2), terão seus conceitos e métodos abordados nas próximas subseções.

3.11.1 ROUGE

O ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) é uma métrica utilizada para avaliar resumos ou traduções, sugerido inicialmente por Lin (2004).

3.11.1.1 N-Gram na Métrica ROUGE

Segundo Damashek (1995), um n-*gram* é uma sequência de n caracteres consecutivos. Na métrica ROUGE, o n-*gram* é separado em pares ou sequência de palavras. A Tabela 4 demonstra a formação de n-*grams* a nível de palavra.

Tipo de N-Gram	***n***	**Resultado**	**Tamanho**
Unigrama	1	'I', 'like', 'classic', 'books'	4
Bigrama	2	['I', 'like'], ['like', 'classic'], ['classic', 'books']	3
Trigrama	3	['I', 'like', 'classic'], ['like', 'classic', 'books']	2

Tabela 4 – Exemplo de um unigrama, bigrama e trigrama para a frase *I like classic books*.

Fonte: Autora.

3.11.1.2 ROUGE-N

Seguindo no mesmo trabalho de Lin (2004), sendo N o número de n-*gram* escolhido para a avaliação, o ROUGE-N calcula a ocorrência das combinações dos n-*grams* gerados pelo sistema e compara com a ocorrência dos n-*grams* do texto de referência ou do modelo.

A precisão no ROUGE é calculada pela Equação 8, que obtém os *n-grams* iguais entre o resumo gerado e o gabarito e divide pela quantidade de *n-grams* do resumo gerado pelo modelo. O *recall* é calculado através da divisão entre os *n-grams* em comum, e divide pela contagem de *n-grams* gerados pelo resumo gabarito, ou de referência. A Equação 9 mostra a operação. Por último, a variação de *F-Measure*, é calculada pela Equação 10.

$$ROUGE_{N_{precision}} = \frac{Count_{match}(gram_n)}{CountModel(gram_n)} \tag{8}$$

$$ROUGE_{N_{recall}} = \frac{Count_{match}(gram_n)}{CountReference(gram_n)} \tag{9}$$

$$ROUGE_{N_{F1}} = 2 \times \frac{precision \times recall}{precision + recall} \tag{10}$$

Dado o resumo de referência *I like classic books* e supondo que o resumo do sistema gere a frase *I like books*, a Tabela 5 mostra quais seriam os valores de *n-grams*, para $N = 1$, $N = 2$ e $N = 3$.

3.11.1.3 ROUGE-L

A variação L do ROUGE deriva da sigla LCS (*Longest Commom Subsequence*), no qual o LCS é o cálculo da maior subsequência comum entre duas cadeias de texto. Inspirado

N	**Referência**	**Tamanho do *N-Gram* de Referência**	**Gerado pelo Sistema**	**Número de *N-Gram* Igual(is)**
1	'I', 'like', 'classic', 'books'	4	'I', 'like', 'books'	3
2	['I', 'like'], ['like', 'classic'], ['classic', 'books']	3	['I', 'like'], ['like', 'books']	1
3	['I', 'like', 'classic'], ['like', 'classic', 'books']	2	['I', 'like', 'books']	0

Tabela 5 – Os n-*grams* em que há correspondência entre a referência e o sistema estão coloridos de azul. O resultado para o ROUGE-1 seria $3/4$, para ROUGE-2 resultaria em $1/3$ e ROUGE-3 seria zero pois não houve correspondência entre os trigramas de refêrencia e os gerados pelo sistema.

Fonte: Autora.

nesse conceito, o ROUGE-L proposto por Lin (2004) indica que quanto maior o valor de LCS entre dois resumos, mais similares eles são. Uma das características dessa abordagem é que a definição de n-*gram* não é necessária.

Assumindo que X e Y são duas sentenças, e que $LCS(X,Y)$ é o tamanho da maior subsequência em comum, obtemos a Equação 11 e 12 - sendo $Recall_{LCS}$ o cálculo de *recall* e $Precision_{LCS}$ de precisão.

$$Recall_{LCS} = \frac{LCS(X,Y)}{CountReference(gram_n)} \tag{11}$$

$$Precision_{LCS} = \frac{LCS(X,Y)}{CountModel(gram_n)} \tag{12}$$

O cálculo final do ROUGE-L é a versão da métrica *F-Measure* aplicada ao LCS, mostrada previamente na Equação 10.

Para computar o LCS a nível de resumo, a Equação 13 e 14 mostram os cálculos para $Recall_{LCS}$ e $Precision_{LCS}$ onde r_i é a frase de um resumo de referência, e C é o conjunto de frases de resumo gerados pelo sistema. Para cada frase em C, é realizada uma operação de união com r_i. Onde n representa o tamanho do resumo de referência e m do modelo.

$$Recall_{LCS} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i,C)}{n} \tag{13}$$

$$Precision_{LCS} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i,C)}{m} \tag{14}$$

3.11.2 Perplexidade

A definição de Jurafsky e Martin (2019) sobre perplexidade em um modelo de linguagem conceitua que essa métrica equivale a probabilidade inversa do conjunto de teste, normalizada pelo número de palavras. A Equação 15 mostra a operação realizada, sendo W o conjunto de teste.

$$PP(W) = P(w_1, w_2,..., w_n)^{-\frac{1}{N}} = \sqrt[n]{\frac{1}{P(w_1, w_2,..., w_n)}} \tag{15}$$

Aplicando a regra da cadeia, obtemos a Equação 16. Onde w_i é a probabilidade da palavra ocorrer dadas as ocorrências anteriores entre w_1 e w_{i-1}.

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1...w_{i-1})}} \tag{16}$$

Para modelos de linguagem de unigrama ou bigrama, a probabilidade de ocorrência, respectivamente, é dado pela Equação 17 e Equação 18.

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i)}} \tag{17}$$

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}} \tag{18}$$

Na prática, a perplexidade define o quão surpreso - perplexo - o modelo está durante sua validação em relação ao conjunto de teste. Supondo que o modelo, em certo momento, possui 10 possibilidades com valores de probabilidades iguais, o valor da perplexidade é 10. Ou seja, para avaliar a qualidade de um modelo de linguagem, ele deve atribuir a maior probabilidade corretamente para o próximo elemento. Outra maneira de entender a perplexidade é pelo chamado fator de ramificação ponderado (traduzido do inglês, *weighted average branching factor*) que consiste no número de palavras posteriores possíveis.

4 METODOLOGIA PROPOSTA

A metodologia proposta nesse trabalho foi um modelo de sumarização híbrido de artigos científicos com a aplicação da arquitetura *Transformer* (Seção 3.7) por meio do modelo pré-treinado Longformer (Seção 3.10).

A visão geral do modelo pode ser vista na Figura 26, na qual cada parte será detalhada nas seções subjacentes deste capítulo.

De maneira geral, o modelo passou pelo Pré-Processamento Manual (Módulo I - Seção 4.1) para uniformização da base. Na fase de *Fine-Tuning* (Módulo II - Seção 4.2) foi executado o modelo Longformer com o objetivo de sumarizar os artigos da base *SciSummNet* e por último, a Saída (Módulo III - Seção 4.3), gerou e retornou os artigos científicos resumidos.

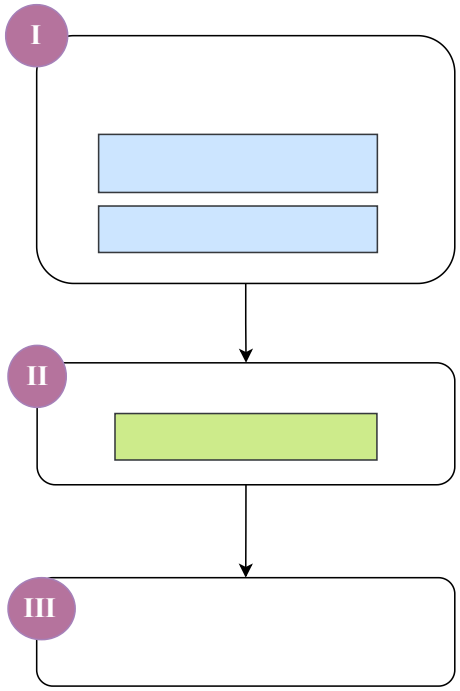

Figura 26 – Esquema geral da metodologia proposta.

Fonte: Autora.

4.1 ETAPA I - PRÉ-PROCESSAMENTO MANUAL

A primeira etapa do modelo proposto consistiu em realizar o pré-processamento de todos os artigos científicos brutos do *SciSummNet*. O formato dos arquivos do conjunto de dados será apresentado na Seção 4.1.1. Esta etapa também tem a finalidade de padronizar o *corpus* que contém formatos diferentes de citações (Seção 4.1.2) além de selecionar sentenças elegíveis

para a sumarização (Seção 4.1.3). Ao final dessa seção, a saída esperada por essa etapa é apresentada (Seção 4.1.4).

4.1.1 Formato da Base *SciSummNet*

De acordo com o conteúdo da base de artigos sumarizados *SciSummNet*, cada artigo contém uma pasta com:

a) Sub-pasta com arquivo XML do conteúdo do artigo científico;
b) Sub-pasta com arquivo texto do resumo padrão-ouro;
c) Arquivo JSON com informações dos artigos que citaram o artigo em questão na base do *SciSummNet*.

Um exemplo genérico do arquivo XML do conteúdo de um artigo científico da base *SciSummNet* pode ser visto na Figura 27. A *tag* <PAPER> engloba o conteúdo do arquivo, além de *tags* específicas para o resumo do artigo <ABSTRACT>, seções <SECTION> e as sentenças <S>. Dentro da *tag* <S>, o *sid* é a identificação numérica ordenada da sentença no artigo e *ssid* é a identificação da sentença dentro do resumo ou da seção.

Formato Genérico

```
<PAPER>

<S sid="0">Título do Artigo</S>

<ABSTRACT>
<S sid="1"ssid="1">Sentença 1 Abstract</S>
<S sid="2"ssid="2">Sentença 2 Abstract</S>
</ABSTRACT>

<SECTION title="Nome Seção 1"number="1">
<S sid="3"ssid="1">Sentença 1 Seção 1</S>
<S sid="4"ssid="2">Sentença 2 Seção 1</S>
</SECTION>

<SECTION title="Nome Seção 2"number="2">
<S sid="5"ssid="1">Sentença 1 Seção 2</S>
<S sid="6"ssid="2">Sentença 2 Seção 2</S>
</SECTION>

</PAPER>
```

Figura 27 – Exemplo de um arquivo XML do *SciSummNet*.

Fonte: Autora.

4.1.2 Padronização de Citações

Esse estágio representa o pré-processamento das sentenças de citação. Essa etapa foi realizada nas sentenças dos artigos originais do *SciSummNet* - vinculadas às *tags* <ABSTRACT> e <SECTION> - e para o resumo padrão-ouro, com a finalidade de uniformizar a entrada e saída do modelo.

As sentenças de citação foram padronizadas para o formato $(Autor, Ano)$ para citações simples e para citações que possuem mais de uma citação seguida da outra, chamadas aqui de compostas, o seguinte formato foi adotado: $(Autor, Ano; Autor, Ano)$. As citações que estavam no formato adotado não foram alteradas. A identificação e substituição das citações foram realizadas por meio de expressões regulares.

Os possíveis tipos de formatos de citações e exemplos da aplicação do tratamento proposto podem ser vistos na Tabela 6 para citações simples e na Tabela 7 para citações compostas.

Formato	Antes do Pré-Processamento	Depois do Pré-Processamento
Autor (Ano)	The interest data was first studied by Bruce and Wiebe (1994).	The interest data was first studied by (Bruce and Wiebe, 1994).
[Autor, Ano]	The interest data was first studied by [Bruce and Wiebe, 1994].	

Tabela 6 – Exemplos formatos e pré-processamento de sentenças de citações simples. Frase de exemplo extraída do artigo de Poelmans et al. (2012).

Fonte: Autora.

Formato	Antes do Pré-Processamento	Depois do Pré-Processamento
Autor (Ano) e Autor (Ano)	Paradigmatic relations in WordNet have been used by many to determine similarity, including Li et al. (1995) and Agirre and Rigau (1996).	Paradigmatic relations in WordNet have been used by many to determine similarity, including (Li et al.,1995; Agirre and Rigau,1996).
[Autor, Ano; Autor, Ano]	Paradigmatic relations in WordNet have been used by many to determine similarity, including [Li et al., 1995; Agirre and Rigau, 1996]	

Tabela 7 – Exemplos formatos e pré-processamento de sentenças de citações compostas. Frase de exemplo extraída do artigo de Agirre e Soroa (2007).

Fonte: Autora.

4.1.3 Seleção de Sentenças

Essa etapa de tratamento e pré-processamento classificou as sentenças como elegíveis ou não elegíveis. Sentenças não elegíveis serão aquelas que contenham menções a objetos de figura, equação, tabela, anexos, capítulos e seções, assim como as sentenças da seção de agradecimentos ou reconhecimentos (em inglês, *Acknowledgment*). As sentenças que contêm referências a esses objetos não agregam sentido ao resumo final e não é esperado que o modelo linguístico interprete-os.

A Tabela 8 apresenta exemplos de sentenças elegíveis e não elegíveis.

Sentença	Elegível?
The most accurate classifier in each of the nine range categories is selected for inclusion in the ensemble.	Sim
Search engines are increasingly being used by amongst others web users who have an information need.	
FCA has been used as the basis for many web-based knowledge browsing systemsdeveloped during the past years.	
These senses and their frequency distribution are shown in Table 1.	Não
In section 3 we describe the dataset used.	
Jonas Poelmans is aspirant of the “Fonds voor Wetenschappelijk Onderzoek – Vlaanderen” or “Research Foundation Flanders”.	

Tabela 8 – Exemplos de sentenças e sua elegibilidade para o modelo proposto. Frases de exemplo extraídas do artigo de Poelmans et al. (2012).

Fonte: Autora.

4.1.4 Saída do Pré-Processamento

A Etapa I recebe como entrada todos os arquivos XML originais do *SciSummNet* (exemplo da Figura 27) e teve como saída a base de todos os artigos selecionados para treinamento em formato JSON para destinar a biblioteca do Python intitulada Datasets, com as sentenças formatadas e selecionadas de acordo com os critérios da Seção 4.1.2 e 4.1.3. A Tabela 9 mostra um exemplo de entrada e saída esperado pelo pré-processamento em relação ao texto bruto.

Para cada item do arquivo JSON, contém a referência do ID do artigo da base *SciSummNet* pela chave $article\_id$, o texto pré-processado com a chave $content$ e o padrão ouro com a chave nomeada $gold\_summary$, todos estes englobados na chave $data$.

Antes do Pré-Processamento	Depois do Pré-Processamento
<PAPER> <S sid="0"> Forest-based Translation Rule Extraction </S> <ABSTRACT> <S sid="1"ssid="1"> We propose a novel approach which extracts rules a forest compactly encodes exponentially many parses. </S> </ABSTRACT> <SECTION title="Introduction"number="1"> <S sid="2"ssid="1"> To alleviate this problem, an obvious idea is to extract rules from k-best parses instead. </S> <S sid="3"ssid="2"> However, a k-best list, with its limited scope, has too few variations and too many redundancies Huang (2008). </S> </SECTION> <SECTION title="Related Work"number="2"> <S sid="4"ssid="1"> Our experiments are on Chinese-to-English translation based on a tree-to-string system similar to Huang et al. (2006) and Liu et al. (2006). </S> <S sid="5"ssid="2"> The final BLEU score results are shown in Table 4. </S> </SECTION> </PAPER>	We propose a novel approach which extracts rules a forest compactly encodes exponentially many parses. To alleviate this problem, an obvious idea is to extract rules from k-best parses instead. However, a k-best list, with its limited scope, has too few variations and too many redundancies (Huang, 2008). Our experiments are on Chinese-to-English translation based on a tree-to-string system similar to (Huang et al. 2006; Liu et al., 2006).

Tabela 9 – Exemplos de entrada (lado esquerdo) e saída (lado direito) do pré-processamento. No exemplo, a sentença com *ssid* 6, destacada em vermelho, foi removida devido a referência a uma tabela. As sentenças com *ssid* 4 e 5 tiveram suas citações padronizadas e as alterações podem ser vistas na cor azul. Frases extraídas do artigo de Mi e Huang (2008).

Fonte: Autora.

4.2 ETAPA II - FINE-TUNING

Essa etapa realizou o ajuste fino - aprendizado e adaptação do modelo para a tarefa deste trabalho - do modelo pré-treinado Longformer (Seção 3.10) para a geração dos resumos. A Figura 28 demonstra o exemplo da entrada dos artigos científicos do *SciSummNet*.

Trigrams'n'Tags (TnT) is an efficient statistical part-of-speech tagger.
Contrary to claims found elsewhere in the literature, we argue that a tagger based on Markov models performs at least as well as other current approaches, including the Maximum Entropy framework.
A recent comparison has even shown that TnT performs significantly better for the tested corpora.
We describe the basic model of TnT, the techniques used for smoothing and for handling unknown words.
Furthermore, we present evaluations on two corpora.
A large number of current language processing systems use a part-of-speech tagger for pre-processing.
The tagger assigns a (unique or ambiguous) part-ofspeech tag to each token in the input and passes its output to the next processing level, usually a parser.
Furthermore, there is a large interest in part-ofspeech tagging for corpus annotation projects, who create valuable linguistic resources by a combination of automatic processing and human correction.
For both applications, a tagger with the highest possible accuracy is required.
The debate about which paradigm solves the part-of-speech tagging problem best is not finished.
Recent comparisons of approaches that can be trained on corpora (van Halteren et al., 1998; Volk and Schneider, 1998) have shown that in most cases statistical aproaches (Cutting et al., 1992; Schmid, 1995; Ratnaparkhi, 1996) yield better results than finite-state, rule-based, or memory-based taggers (Brill, 1993; Daelemans et al., 1996).
Among the statistical approaches, the Maximum Entropy framework has a very strong position.
Nevertheless, a recent independent comparison of 7 taggers (Zavrel and Daelemans, 1999) has shown that another approach even works better: Markov models combined with a good smoothing technique and with handling of unknown words.
This tagger, TnT, not only yielded the highest accuracy, it also was the fastest both in training and tagging.
The tagger comparison was organized as a "blackbox test": set the same task to every tagger and compare the outcomes.
This paper describes the models and techniques used by TnT together with the implementation.
The reader will be surprised how simple the underlying model is.
However, in this paper we clarify a number of details that are omitted in major previous publications concerning tagging with Markov models.
As two examples, (Rabiner, 1989) and (Charniak et al., 1993) give good overviews of the techniques and equations used for Markov models and part-ofspeech tagging, but they are not very explicit in the details that are needed for their application.
We argue that it is not only the choice of the general model that determines the result of the tagger but also the various small decisions on alternatives.
The aim of this paper is to give a detailed account of the techniques used in TnT.
Additionally, we present results of the tagger on the NEGRA corpus (Brants et al., 1999) and the Penn Treebank (Marcus et al., 1993).
The Penn Treebank results reported here for the Markov model approach are at least equivalent to those reported for the Maximum Entropy approach in (Ratnaparkhi, 1996).
For a comparison to other taggers, the reader is referred to (Zavrel and Daelemans, 1999).

Figura 28 – Exemplo do formato entrada dos artigos científicos para o processo de *fine-tuning*. Extraído do artigo de Brants (2000).

Fonte: Autora.

Foram inseridos 8192 *tokens* para cada artigo científico do modelo. A estratégia adotada no *fine-tuning* para o cálculo de atenção foi a aplicação do *Global Attention* (Seção 3.10.1.3) conforme recomendação do artigo original do Longformer de Beltagy, Peters e Cohan (2020).

4.3 ETAPA III - GERAÇÃO DOS RESUMOS

Por último, a saída do modelo proposto correspondeu a um conjunto de textos, onde cada texto é o resumo gerado pelo modelo aqui proposto de um artigo científico do *SciSummNet* que são identificados pelo atributo de $article\_id$. Um exemplo do conteúdo textual de saída esperado do resumo de um artigo científico é apresentada na Figura 29. O limite de *tokens* para sumarização foi de 512.

Trigrams'n'Tags (TnT) is an efficient statistical part-of-speech tagger.
Contrary to claims found elsewhere in the literature, we argue that a tagger based on Markov models performs at least as well as other current approaches, including the Maximum Entropy framework.
A recent comparison has even shown that TnT performs significantly better for the tested corpora.
We describe the basic model of TnT, the techniques used for smoothing and for handling unknown words.
Furthermore, we present evaluations on two corpora.
We achieve the automated tagging of a syntactic-structure-based set of grammatical function tags including phrase-chunk and syntactic-role modifiers trained in supervised mode from a tree bank of German.

Figura 29 – Exemplo de uma saída padrão-ouro esperada no resumo. Extraído do artigo de Brants (2000).

Fonte: Autora.

4.4 FERRAMENTAS DA IMPLEMENTAÇÃO

A metodologia proposta foi implementada em um computador com as seguintes especificações: CPU: AMD Ryzen 5 5600x (6 cores), memória RAM 32GB (4x8GB 3200MHz DDR4) e uma GPU NVIDIA GeForce RTX 3070 (8Gb VRAM).

A codificação foi realizada através da linguagem de programação Python. A base de dados *SciSummNet* foi tratada para a biblioteca Datasets[1]. A biblioteca Transformers[2] foi utilizada para o uso do modelo pré-treinado LED do tipo $base16384$ e o PyTorch[3] foi utilizado no processo de *fine-tuning*.

[1] https://huggingface.co/docs/datasets/
[2] https://huggingface.co/docs/transformers/
[3] https://pytorch.org/

5 RESULTADOS E DISCUSSÃO

Nesse capítulo serão apresentados os resultados obtidos na aplicação da metodologia sugerida no Capítulo 4 através do modelo pré-treinado LED (*Longformer Encoder-Decoder*) (Seção 3.10) utilizando a base de artigos científicos *SciSummNet* (Seção 3.2). As principais métricas que serão abordadas neste capítulo, ROUGE e perplexidade, foram introduzidas respectivamente na Seção 3.11.1 e Seção 3.11.2.

Em todos os experimentos realizados, o modelo adotado do LED foi o de referência $base16384$. Foram 641 artigos científicos para o processo de *fine-tuning*, e 302 exemplos para validação.

5.1 RESULTADOS SEM O PROCESSO DE FINE-TUNING

Com o propósito de verificar o comportamento do modelo pré-treinado LED sem o *fine-tuning* e também de ter uma base de comparação da efetividade do processo, esse experimento executou o modelo LED para gerar um texto de saída a partir dos artigos científicos pré-processados da base *SciSummNet*. Os resultados relativos a esse experimento serão chamados de *SemFineTuning*.

Os resultados do modelo *SemFineTuning* são apresentados a Tabela 10. Dado o valor de precisão alto e *recall* baixo no ROUGE-1, ROUGE-2 e ROUGE-L, é possível denotar que o modelo *SemFineTuning* acerta palavras do gabarito como indica a precisão, porém o *recall* baixo indica pouca efetividade dos *n-grams* encontrados no resumo gerado pelo sistema. Ao aumentar o valor de *n-grams* da métrica ROUGE-1 para ROUGE-2, a medida apresentou resultados menores pois o modelo sem o *fine-tuning* não se aproximou do gabarito no cenário de duas palavras consecutivas combinadas ao longo do texto, o que siginifica que o modelo *SemFineTuning* tende a gerar resumos mais desconexos. Por último, o resultado do *F-Measure* equilibra os resultados da precisão e *recall*.

	ROUGE-1	**ROUGE-2**	**ROUGE-L**
Precisão	0.54	0.21	0.46
Recall	0.06	0.02	0.05
F-Measure	0.11	0.04	0.10

Tabela 10 – Resultados ROUGE da sumarização sem o processo de *fine-tuning*.

Fonte: Autora.

5.2 PARÂMETROS E EXECUÇÃO DO PROCESSO DE FINE-TUNING

Essa seção apresentará os parâmetros adotados no processo de *fine-tuning* e detalhes de treinamento, abordando o otimizador, valores da taxa de aprendizado e perda. Primordialmente, parâmetros empregados ao modelo para o processo de *fine-tuning* são listados a seguir:

a) Tamanho do lote (*batch size* = 4): quantidade de amostras processadas a cada acúmulo de gradiente;
b) Etapas de acumulação de gradiente (*gradient accumulation steps* = 4): quantidade de etapas em que o gradiente será acumulado e transportado antes da atualização em forma de pesos no modelo;
c) Tamanho total do lote (*total train batch size* = 16): *batch size* multiplicado por *gradient accumulation steps*;
d) Etapas de otimização (*optimization steps* = 40): quantidade de exemplos (641) dividido pelo *total train batch size*;
e) Épocas (*epochs* = 2): quantidade de execuções do treinamento no conjunto de exemplos.

O modelo foi treinado utilizando o otimizador padrão do PyTorch, chamado AdamW proposto por Loshchilov e Hutter (2017) que consiste na implementação do Adam com adaptação de decaimento de peso. A função de perda adotada foi a entropida cruzada - em inglês, *Cross Entropy Loss*.

Para modelo de linguagem proposto, foram criados dois modelos análogos: um com o ajuste-fino realizado incluindo o *Abstract* do artigo científico e outro excluindo esse texto. Esses modelos serão chamados aqui de *ComAbstract* e *SemAbstract*. Ambos modelos foram explorados por 2 *epochs*. Esse teste pretendeu avaliar a execução do modelo para definir o número de *epochs*.

A Figura 30 e 31, nessa ordem, apresentam a função de perda do processo de ajuste fino do modelo *ComAbstract* e *SemAbstract*. Nota-se que em ambos os casos, os valores de *loss* convergem entre o final da primeira e segunda *epoch*.

Considerando a métrica ROUGE-1 pela *F-Measure*, a validação no final das *epochs* 1 e 2 para a o modelo *ComAbstract* foram respectivamente de 0,60 e 0,63. No cenário da sumarização gerada pelo *SemAbstract*, foi obtido para as *epochs* 1 e 2 o resultado 0,36; indicando estabilidade.

Pelas motivações acima citadas, o parâmetro de *epoch* adotado será 1 para os modelos conferidos ao longo deste capítulo - *ComAbstract* e *SemAbstract*.

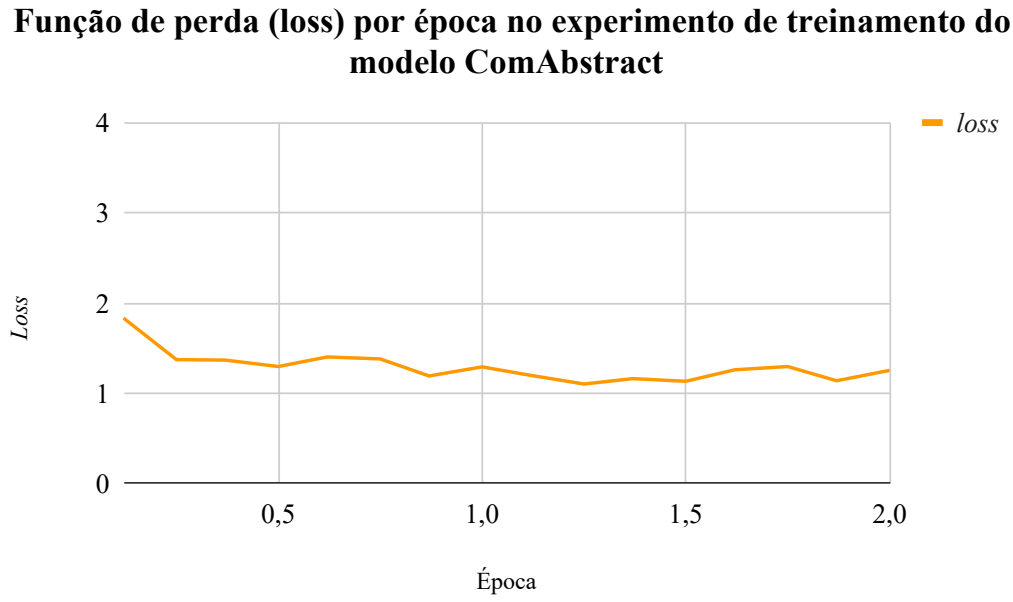

Figura 30 – Gráfico de *loss* em função da *epoch* do com a base de treinamento contendo o texto de resumo do artigo científico - modelo *ComAbstract*.

Fonte: Autora.

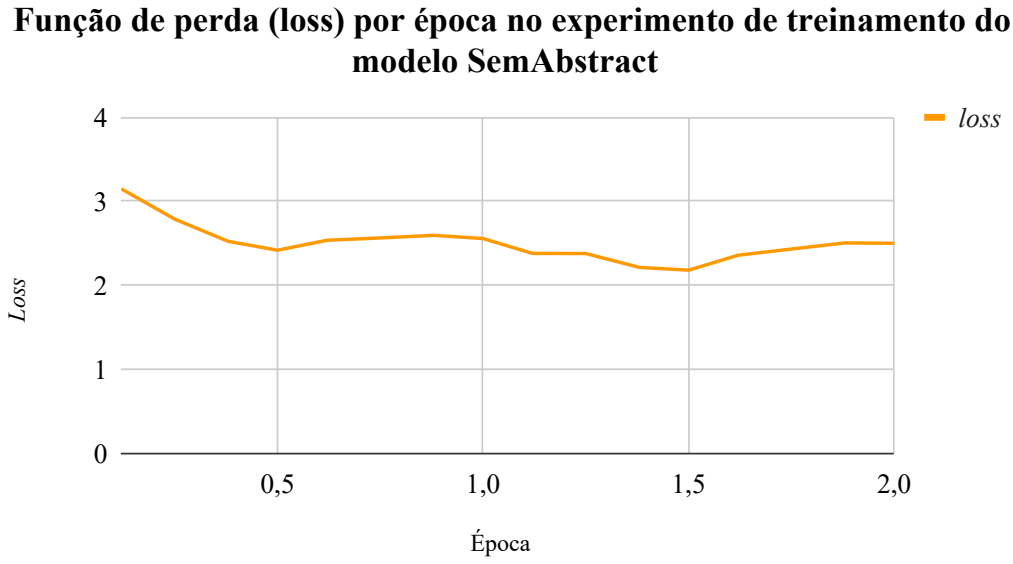

Figura 31 – Gráfico de *loss* em função da *epoch* do com a base de treinamento sem o texto de resumo do artigo científico - modelo *SemAbstract*.

Fonte: Autora.

5.3 VALIDAÇÃO DA MÉTRICA ROUGE COM E SEM TEXTO DE RESUMO DOS ARTIGOS CIENTÍFICOS

Essa validação executou o ajuste-fino da base de artigos científicos para treinamento com o texto incluído na *tag* <ABSTRACT> do XML da base *SciSummNet* (modelo *ComAbstract*), e em outra base de exemplos de treinamento, excluindo o texto dentro da *tag* <ABSTRACT> (modelo *SemAbstract*). Os resultados foram capturados após a execução da primeira *epoch*.

A Tabela 11 apresenta os resultados da métrica ROUGE do modelo *SemAbstract* e a Tabela 12 o resultado do modelo *ComAbstract*. Em todos os cenários, de ambas as tabelas, o valor de *recall* foi maior que o valor de precisão. Isto indica que os modelos geraram palavras relevantes na geração do resumo em comparação com o padrão-ouro, e provoca a situação inversa dos resultados *SemFineTuning*, no qual a precisão foi mais alta que o *recall*.

Os resultados ROUGE da Tabela 12 apresentaram resultados mais significativos em relação à Tabela 11. O que mostra que os resumos padrão-ouro da base *SciSummNet* têm uma alta correspondência com a seção de *Abstract* do artigo científico, e o modelo de linguagem *ComAbstract* aprendeu essa correlação durante o *fine-tuning*. Além disso, o modelo *ComAbstract* alcançou resultados significativamente maiores de ROUGE-2 e ROUGE-L, o que diz que o modelo *ComAbstract* captura o efeito de 2 ou mais palavras consecutivas na geração do resumo comparado ao padrão-ouro. Os resultados do treinamento do modelo *ComAbstract*, em média, foram 33% maiores.

	ROUGE-1	**ROUGE-2**	**ROUGE-L**
Precisão	0.32	0.10	0.17
Recall	0.50	0.16	0.27
F-Measure	0.36	0.11	0.19

Tabela 11 – Resultados pela métrica ROUGE-1, ROUGE-2 e ROUGE-L da sumarização excluindo o texto de resumo do artigo científico (*SemAbstract*).

Fonte: Autora.

	ROUGE-1	**ROUGE-2**	**ROUGE-L**
Precisão	0.58	0.50	0.54
Recall	0.74	0.62	0.67
F-Measure	0.60	0.51	0.56

Tabela 12 – Resultados pela métrica ROUGE-1, ROUGE-2 e ROUGE-L da sumarização incluindo o texto de resumo do artigo científico (*ComAbstract*).

Fonte: Autora.

Em números gerais, do processo de *fine-tuning* pelo modelo *ComAbstract* aumentou o valor ROUGE por volta de 0,47, e para o modelo *SemAbstract*, 0,14 em relação a versão *SemFineTuning*.

5.4 COMPARAÇÃO DA PERPLEXIDADE

Com o propósito de conhecer o modelo de linguagem, o valor de perplexidade indica quanto o modelo pode estar seguro e assertivo em atribuir a saída de palavras perante às possibilidades.

A Figura 32 apresenta a perplexidade nos modelos *SemFineTuning*, *ComAbstract* e *SemAbstract*. Analisando pelo contexto, o modelo *ComAbstract* obteve em média 1,3 palavras possíveis durante a geração da sumarização, e o modelo *SemAbstract*, 2,8 palavras em média.

O modelo *SemFineTuning* obteve 1,86 de perplexidade mesmo com valores ROUGE da Tabela 10 (*SemFineTuning*) menores em relação aos valores ROUGE da Tabela 11 (modelo *SemAbstract*). Nesse cenário, constatou-se que apesar do modelo *SemFineTuning* estar mais seguro comparado ao *SemAbstract*, os resultados ROUGE do modelo *SemAbstract* foram superiores ao *SemFineTuning*. Comprova-se que a comparação da métrica de perplexidade - relativa ao modelo - combinada com a métrica para conferir a performance da tarefa de sumarização - ROUGE - proporcionam a visão do desempenho e avaliação destes modelos de linguagem.

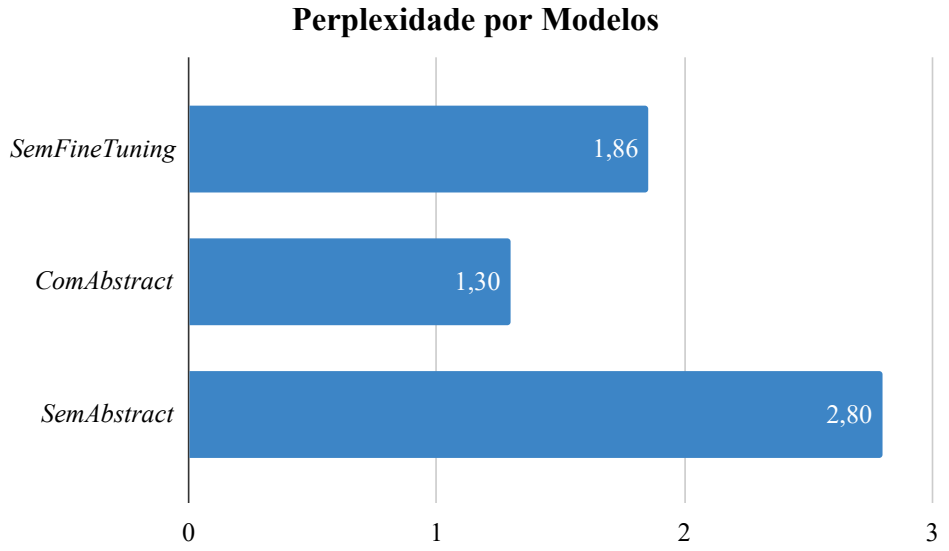

Figura 32 – Gráfico da métrica de perplexidade dos modelos *SemFineTuning*, *ComAbstract* e *SemAbstract* com $epoch = 1$.

Fonte: Autora.

5.5 ANÁLISE DOS RESUMOS GERADOS E PADRÕES OURO

A partir dos resumos gerados pelos modelos *SemFineTuning*, *ComAbstract* e *SemAbstract*, essa parte irá conferir a quantidade de palavras e sentenças entre o resumo gerado pela metodologia proposta e o padrão ouro. De modo geral, o resumo padrão-ouro tem em média 8 sentenças e 141 palavras. Os gráficos de comparação nessa seção foram obtidos a partir de 10 artigos científicos fixos da base de validação gerada a partir do *SciSummNet* para relacionar a mesma amostra em cada modelo.

A Figura 33 apresenta a comparação do modelo *SemFineTuning* e o padrão ouro a nível de palavras, e a Figura 34 a nível de sentença. Em média, foram 1,6 sentenças e 16 palavras nos resumos gerados na base de validação.

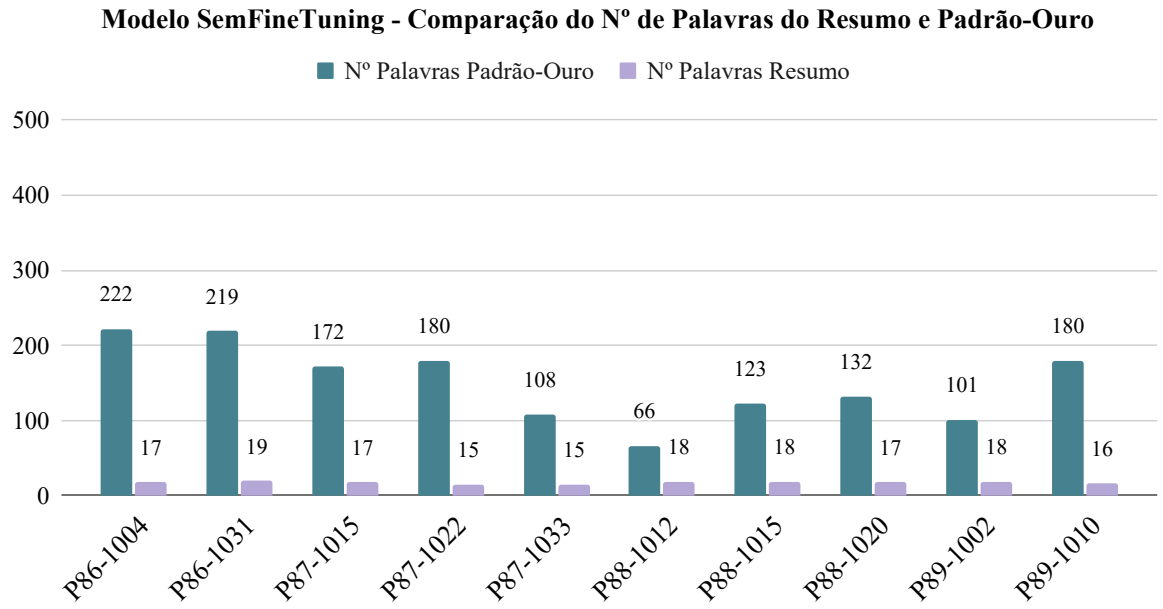

Figura 33 – Gráfico de comparação do número de palavras do resumo e padrão-ouro do modelo *SemFineTuning* em relação ao ID do artigo científico.

Fonte: Autora.

Em relação ao modelo *ComAbstract*, os valores de média foram de 222 palavras e 9 sentenças. A Figura 35 mostra a comparação de palavras e a Figura 36 a comparação de sentenças. O número de sentenças aproximou-se do padrão-ouro, porém o modelo *ComAbstract* gerou mais palavras.

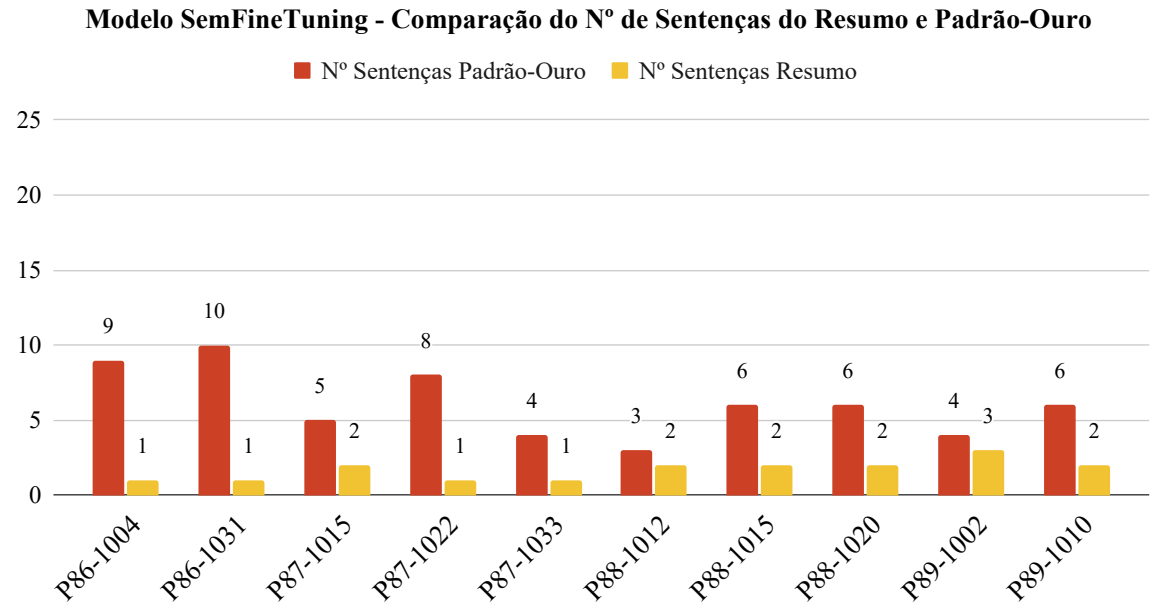

Figura 34 – Gráfico de comparação do número de sentenças do resumo e padrão-ouro do modelo *SemFineTuning* em relação ao ID do artigo científico.

Fonte: Autora.

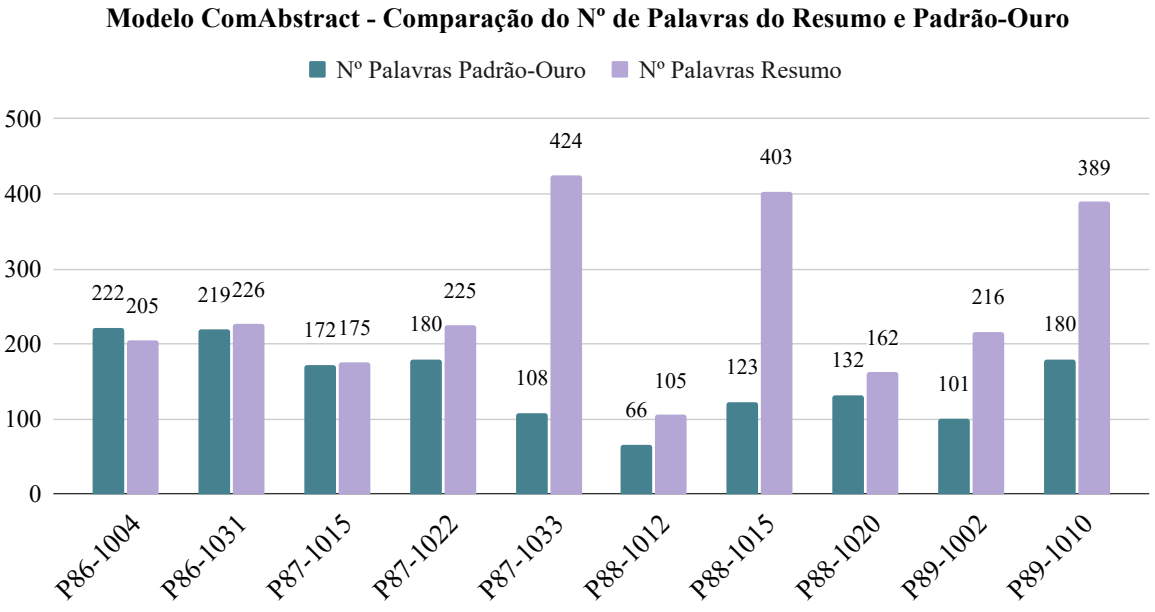

Figura 35 – Gráfico de comparação do número de palavras do resumo e padrão-ouro do modelo *ComAbstract* em relação ao ID do artigo científico.

Fonte: Autora.

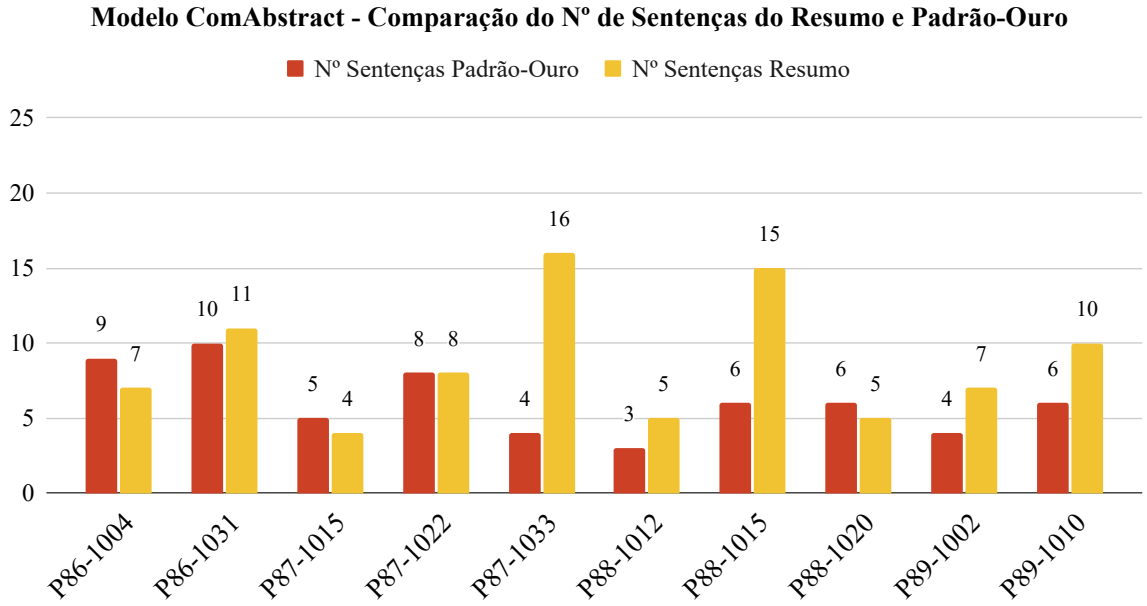

Figura 36 – Gráfico de comparação do número de sentenças do resumo e padrão-ouro do modelo *ComAbstract* em relação ao ID do artigo científico.

Fonte: Autora.

Por último, o modelo *SemAbstract* gerou em média 10 sentenças e 256 palavras. A nível de sentença, em relação ao padrão-ouro, o modelo *SemAbstract* teve em média 2 sentenças a mais, porém o número de palavras geradas pelo modelo em média foi maior que o padrão-ouro. A amostragem da comparação das palavras pode ser vista na Figura 37 e à nível de sentenças, na Figura 38.

Em suma, o modelo *SemFineTuning* gera resumos 89% menores em relação ao padrão-ouro. Nos modelos em que foram realizados o *fine-tuning* - *SemAbstract* e *ComAbstract* - o número de sentenças dos resumos gerados foram similares, porém os dois modelos geraram mais palavras comparado ao padrão-ouro.

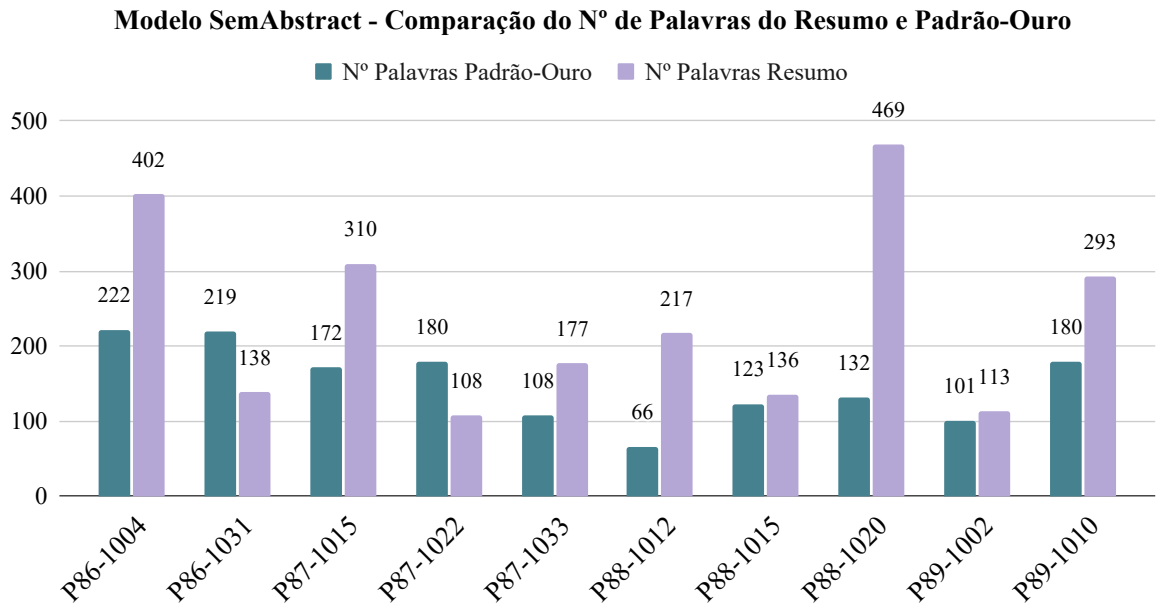

Figura 37 – Gráfico de comparação do número de palavras do resumo e padrão-ouro do modelo *SemAbstract*.

Fonte: Autora.

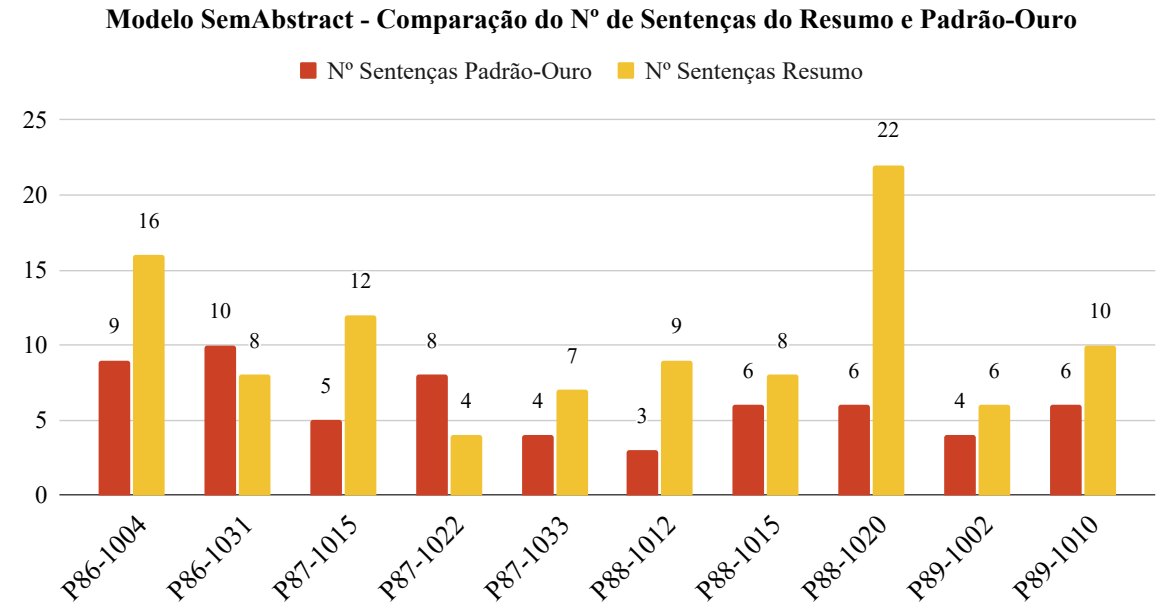

Figura 38 – Gráfico de comparação do número de sentenças do resumo e padrão-ouro do modelo *SemFineTuning*.

Fonte: Autora.

5.6 COMPARAÇÃO AMOSTRAL DA MÉTRICA ROUGE

Com a mesma amostra da Seção 5.5, essa análise irá comparar a distribuição dos valores ROUGE-1, ROUGE-2 e ROUGE-L para os modelos *SemFineTuning*, *SemAbstract* e *ComAbstract* pelo *F-Measure*. Respectivamente, os resultados finais de cada modelo estão na Tabela 10, Tabela 11, Tabela 12.

A Figura 39 apresenta os resultados da amostra de artigos para o modelo *SemFineTuning* para as variações da métrica ROUGE. O ROUGE-2 apresentou resultados próximos de zero, e o ROUGE-1 e ROUGE-L obtiveram resultados similares.

No caso do modelo *ComAbstract*, da Figura 40 e do modelo *SemAbstract* da Figura 41, ambos alcançaram resultados superiores ao *SemFineTuning*, o modelo *ComAbstract* atingiu em algumas exemplos picos na faixa de 0,75 ou mais, e os resumos gerados pelo modelo *SemAbstract* ficaram abaixo da faixa de 0,75.

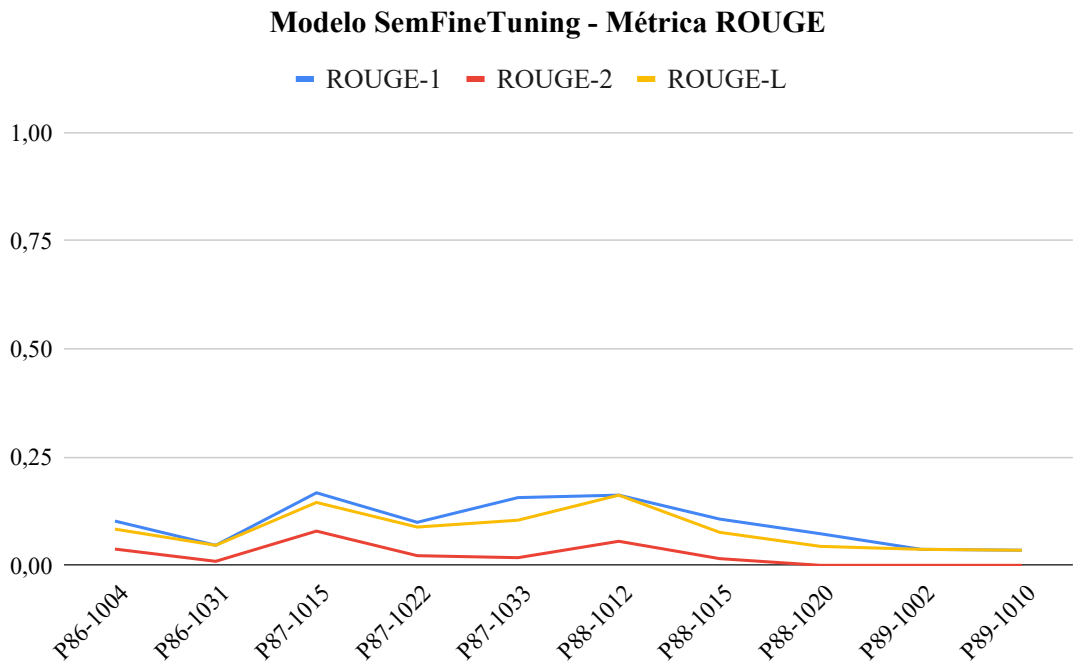

Figura 39 – Gráfico de comparação da métrica ROUGE-1, ROUGE-2 e ROUGE-L do modelo *SemFineTuning* em relação ao ID do artigo científico.

Fonte: Autora.

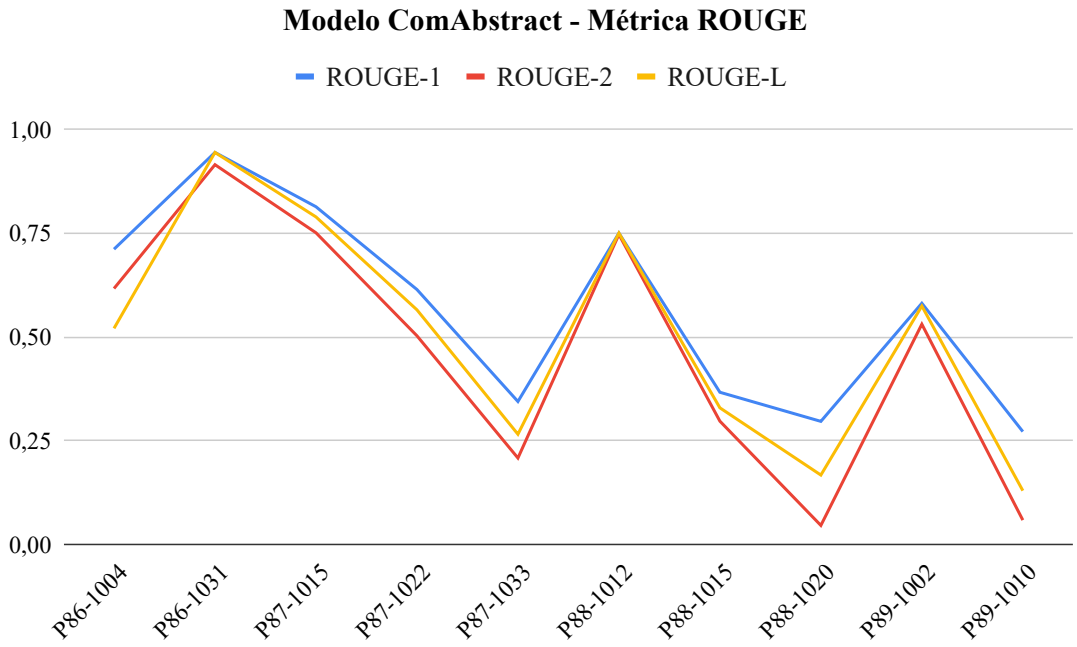

Figura 40 – Gráfico de comparação da métrica ROUGE-1, ROUGE-2 e ROUGE-L do modelo *ComAbstract* em relação ao ID do artigo científico.

Fonte: Autora.

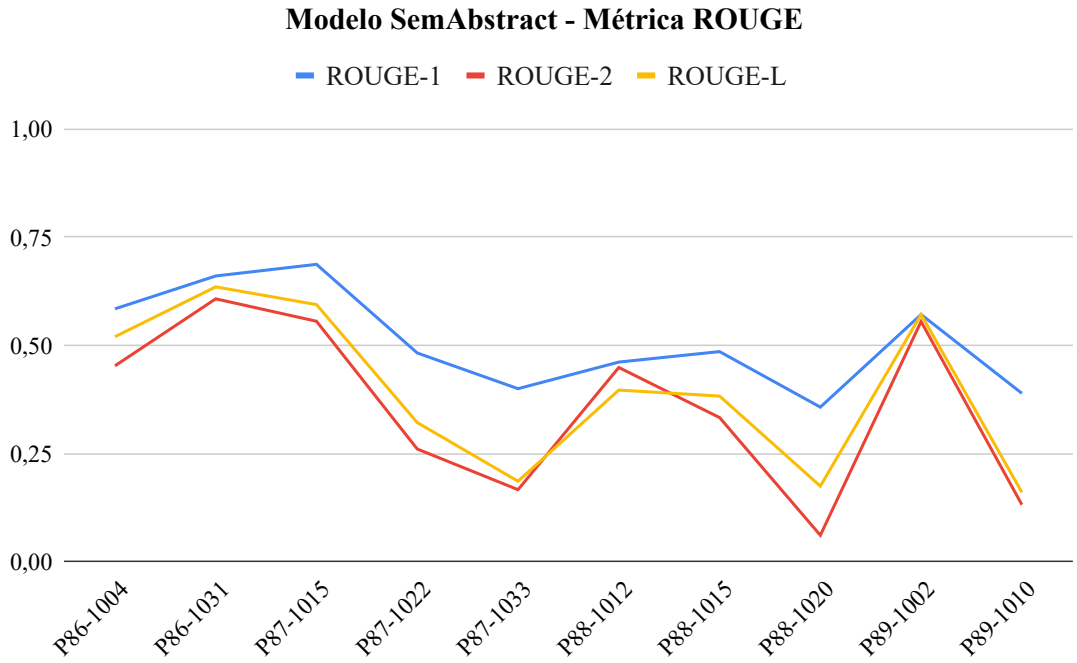

Figura 41 – Gráfico de comparação da métrica ROUGE-1, ROUGE-2 e ROUGE-L do modelo *SemAbstract* em relação ao ID do artigo científico.

Fonte: Autora.

Na maioria dos casos, o ROUGE-2 obteve os menores resultados ao longo da validação. Também, os *SemAbstract* e *ComAbstract* apresentaram os resultados mais baixos nos mesmos exemplos da amostragem: P87-1013 e P88-1020. Um detalhe dessa amostra é que no modelo *ComAbstract* os valores de ROUGE tiveram comportamento similar entre si; no modelo *SemAbstract* ocorreu alternância nesse comportamento, no qual principalmente nas métricas ROUGE-1 e ROUGE-2 eram próximas ou afastadas.

5.7 COMPARAÇÃO COM PROPOSTA DE SUMARIZAÇÃO DA BASE SCISUMMNET

O trabalho de Yasunaga et al. (2019) utiliza a base *SciSummNet* para propor métodos de sumarização de artigos científicos. Essa seção irá realizar a comparação dos resultados obtidos por Yasunaga et al. (2019) em relação ao vigente trabalho a fim de analisar os resultados da metodologia proposta em relação à esta pesquisa que objetivou sumarizar artigos científicos pelo mesmo conjunto de dados utilizado este trabalho.

A pesquisa de Yasunaga et al. (2019) corresponde ao trabalho original da formação da base padrão-ouro, no qual também apresentam dois modelos de sumarização híbrida. A estratégia adotada no Modelo Híbrido 1 escolhe as sentenças mais relevantes do artigo científico, e na iteração dessa escolha, é verificado se a sentença é semelhante às que foram escolhidas anteriormente. Esse processo é realizado até atingir o tamanho limite especificado para o resumo. O Modelo Híbrido 2 utiliza a mesma estratégia de redução de redundância, porém considera a seção de *Abstract* como o resumo inicial e incrementa o resumo gerado com as sentenças mais relevantes que contém citações, até atingir o tamanho limite do resumo.

A Tabela 13 apresenta os resultados obtidos pelo trabalho de Yasunaga et al. (2019) através do ROUGE-2 em *recall* e *F-Measure* comparados ao trabalho aqui proposto. Constata-se que o modelo *ComAbstract* supera todos os resultados para as variações do ROUGE-2 e comparado ao melhor resultado do trabalho de Yasunaga et al. (2019) (Modelo Híbrido 1) apresenta melhoria de 20,8% para ROUGE-2 *recall* e 22,69% para ROUGE-2 *F-Measure*.

	ROUGE-2 ***Recall***	**ROUGE-2** ***F-Measure***
Modelo Híbrido 1	41.69	29.30
Modelo Híbrido 2	36.47	26.31
Modelo *ComAbstract*	62.49*	51.99*
Modelo *SemAbstract*	16.80	11.71

Tabela 13 – Resultados ROUGE-2 de *recall* e *F-Measure* do trabalho de Yasunaga et al. (2019) (Modelo Híbrido 1 e Modelo Híbrido 2) e do trabalho proposto (Modelo *ComAbstract* e Modelo *SemAbstract*). O símbolo de asterisco indica o melhor resultado em cada métrica.

Fonte: Autora.

6 CONCLUSÃO

O trabalho aqui apresentado visou utilizar a arquitetura Transformer (Seção 3.7) para a tarefa de sumarização de artigos científicos. Através da revisão bibliográfica do Capítulo 2, os trabalhos da área de NLP, modelos de atenção, e aplicações para processamento de textos científicos foram apresentados e analisados para o entendimento das principais metodologias, métricas e ferramentas do estado da arte. No Capítulo 3, as definições dos conceitos fundamentais foram salientadas para que a estratégia do trabalho fosse apresentada no Capítulo 4.

O modelo Longformer (Seção 3.10) foi designado ao objetivo do trabalho devido à capacidade de processar textos longos, ao mesmo tempo que otimiza o cálculo de atenção da arquitetura original *Transformer* diminuindo sua complexidade quadrática para linear. As métricas ROUGE (Seção 3.11.1) e perplexidade (Seção 3.11.2) foram escolhidas para avaliar a metodologia proposta do trabalho.

Por meio dos experimentos realizados, tendo como principal base de dados *SciSummNet* (Seção 3.2), foram abordados dois tipos de modelo: *ComAbstract* e *SemAbstract*.

O modelo *ComAbstract* apresentou resultados relevantes comparados ao padrão-ouro e obteve resultados ROUGE-2 em mais de 20% superiores o trabalho de Yasunaga et al. (2019) no qual foi realizado a anotação e proposta de sumarização de textos do *SciSummNet*. A perplexidade para o modelo resultou um valor baixo, indicando que o modelo de linguagem tem uma distribuição de probabilidades que o permite atribuir as palavras do texto gerado com maior assertividade.

A proposta *SemAbstract* obteve resultados inferiores a de *ComAbstract*. Assim é possível inferir que a seção de *Abstract* do artigo científico é relevante para a sumarização. Para compreender o efeito do processo de *fine-tuning*, foi realizado um experimento do modelo LED para gerar o texto final em relação à base *SciSummNet*, no qual apresentou resultados baixos de ROUGE em relação aos modelos *ComAbstract* e *SemAbstract*. Os resultados obtidos pelos modelos *ComAbstract* e *SemAbstract* mostraram que o processo de *fine-tuning* se adequou a tarefa de sumarização de artigos científicos.

Em vista dos resultados e valor de *epoch* adotado, em uma *epoch* foi possível notar o desempenho do processo de *fine-tuning*, o que destaca a utilidade de modelos pré-treinados para o aprendizado de tarefas a fim de reduzir o custo computacional, permitindo assim a possibilidade de implementar modelos específicos sem o retrabalho de treinar parâmetros pré-estabelecidos comuns.

Como contribuição, o presente trabalho demonstrou a aplicação da arquitetura Transformer direcionada à sumarização de artigos científicos, mostrando que os mecanismos de atenção e interpretação do modelo são úteis e que podem ser explorados pela comunidade acadêmica para tarefas específicas de processamento de texto através do *fine-tuning*.

Para trabalhos futuros, a estratégia proposta do modelo *SemAbstract* tem potencial para ser utilizada e melhorada para gerar um texto de *Abstract* dado o conteúdo do artigo científico, visto que funcionalidade pode ser aproveitada e também pelo fator que os resultados do modelo *SemAbstract* foram inferiores ao modelo *ComAbstract*. Outra aplicação para futuros trabalhos desta pesquisa é a implementação de uma interface sistematizada para usuários para a utilização do modelo aqui proposto. Sendo assim, esse tipo de ferramenta pode ser aprimorada com o respaldo e avaliação de especialistas. Adicionalmente, por intermédio de modelos de atenção aplicada a textos de artigos científicos, propostas de conexão entre trabalhos citados e relevância entre os artigos acadêmicos podem auxiliar pesquisadores relacionar propostas e contribuições da comunidade científica.

REFERÊNCIAS

ABU-JBARA, Amjad; RADEV, Dragomir. Coherent citation-based summarization of scientific papers. In: PROCEEDINGS of the 49th Annual Meeting of the Association for Computational Linguistics: Human language Technologies. 2011. P. 500–509.

AFONSO, Alexandre Ribeiro; DUQUE, Cláudio Gottschalg. Automated Text Clustering of Newspaper and Scientific Texts in Brazilian Portuguese: Analysis and Comparison of Methods. **JISTEM-Journal of Information Systems and Technology Management**, SciELO Brasil, v. 11, n. 2, p. 415–436, 2014.

AGIRRE, Eneko; SOROA, Aitor. SEMEVAL-2007 Task 02: Evaluating Word Sense Induction and Discrimination Systems. In: PROCEEDINGS of the 4h International Workshop on Semantic Evaluations (SEMEVAL-2007). 2007. P. 7–12.

AL SAIED, Hazem; DUGUÉ, Nicolas; LAMIREL, Jean-Charles. Automatic Summarization of Scientific Publications using a Feature Selection Approach. **International Journal on Digital Libraries**, Springer, v. 19, n. 2-3, p. 203–215, 2018.

ALAMMAR, Jay. **The Illustrated Transformer**. 2018. http://jalammar.github.io/illustrated-transformer/. Acesso em 13/12/2021.

ALGULIYEV, Rasim M et al. COSUM: Text Summarization based on Clustering and Optimization. **Expert Systems**, Wiley Online Library, v. 36, n. 1, e12340, 2019.

ALMUGBEL, Zainab; EL HAGGAR, Nahla; BUGSHAN, Neda. Automatic Structured Abstract for Research Papers Supported by Tabular Format Using NLP. **International Journal of Advanced Computer Science and Applications (IJACSA)**, v. 10, n. 2, 2019.

ALTMAMI, Nouf Ibrahim; MENAI, Mohamed El Bachir. Automatic Summarization of Scientific Articles: A Survey. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, 2020.

_____. CAST: A Cross-Article Structure Theory for Multi-Article Summarization. **IEEE Access**, IEEE, v. 8, p. 100194–100211, 2020.

BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua. Neural Machine Translation by Jointly Learning to Align and Translate. **arXiv preprint arXiv:1409.0473**, 2014.

BAI, Xiaomei et al. Scientific Paper Recommendation: A Survey. **IEEE Access**, IEEE, v. 7, p. 9324–9339, 2019.

BAKER, Simon et al. Cancer Hallmarks Analytics Tool (CHAT): A Text Mining Approach to Organize and Evaluate Scientific Literature on Cancer. **Bioinformatics**, Oxford University Press, v. 33, n. 24, p. 3973–3981, 2017.

BELTAGY, Iz; PETERS, Matthew E; COHAN, Arman. Longformer: The Long-Document Transformer. **arXiv preprint arXiv:2004.05150**, 2020.

BRANTS, Thorsten. TnT-a Statistical Part-of-speech Tagger. **arXiv preprint cs/0003055**, 2000.

BRIGGS, James. **How to Build a WordPiece Tokenizer For BERT**. 2021. https://towardsdatascience.com/how-to-build-a-wordpiece-tokenizer-for-bert-f505d97dddbb. Acesso em 08/12/2021.

BUGNON, Leandro Ariel et al. DL4papers: A Deep Learning Approach for the Automatic Interpretation of Scientific Articles. **Bioinformatics**, Oxford University Press, v. 36, n. 11, p. 3499–3506, 2020.

CAGLIERO, Luca; LA QUATRA, Moreno. Extracting Highlights of Scientific Articles: A Supervised Summarization Approach. **Expert Systems with Applications**, Elsevier, v. 160, p. 113659, 2020.

ČEBIRIĆ, Šejla et al. Summarizing Semantic Graphs: A Survey. **The VLDB Journal**, Springer, v. 28, n. 3, p. 295–327, 2019.

CHAUDHARI, Sneha et al. An Attentive Survey of Attention Models. **ACM Transactions on Intelligent Systems and Technology (TIST)**, ACM New York, NY, v. 12, n. 5, p. 1–32, 2021.

CHEN, Jingqiang; ZHUGE, Hai. Automatic Generation of Related Work through Summarizing Citations. **Concurrency and Computation: Practice and Experience**, Wiley Online Library, v. 31, n. 3, e4261, 2019.

CHEN, Jingqiang; ZHUGE, Hai. Summarization of Scientific Documents by Detecting Common Facts in Citations. **Future Generation Computer Systems**, Elsevier, v. 32, p. 246–252, 2014.

CHEN, Tianqi et al. {TVM}: An Automated {End-to-End} Optimizing Compiler for Deep Learning. In: 13TH USENIX Symposium on Operating Systems Design and Implementation (OSDI 18). 2018. P. 578–594.

CHRISTOPHER, D Manning; PRABHAKAR, Raghavan; HINRICH, SCHUTZE. **Introduction to Information Retrieval**. Cambridge University Press, 2008.

COHAN, Arman; GOHARIAN, Nazli. Scientific Document Summarization via Citation Contextualization and Scientific Discourse. **International Journal on Digital Libraries**, Springer, v. 19, n. 2-3, p. 287–303, 2018.

CONTRACTOR, Danish; GUO, Yufan; KORHONEN, Anna. Using Argumentative Zones for Extractive Summarization of Scientific Articles. In: PROCEEDINGS of COLING 2012. 2012. P. 663–678.

DAI, Zihang et al. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. **arXiv preprint arXiv:1901.02860**, 2019.

DAMASHEK, Marc. Gauging Similarity With N-Grams: Language-Independent Categorization of Text. **Science**, American Association for the Advancement of Science, v. 267, n. 5199, p. 843–848, 1995.

DEBNATH, Dipanwita; DAS, Ranjita. Automatic Citation Contextualization Based Scientific Document Summarization Using Multi-objective Differential Evolution. In: SPRINGER. INTERNATIONAL Conference on Emerging Applications of Information Technology. 2021. P. 289–301.

DEVLIN, Jacob et al. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DIAO, Yufeng et al. CRHASum: Extractive Text Summarization with Contextualized-Representation Hierarchical-Attention Summarization Network. **Neural Computing and Applications**, Springer, p. 1–13, 2020.

DING, Ying et al. Content-based Citation Analysis: The Next Generation of Citation Analysis. **Journal of the Association for Information Science and Technology**, Wiley Online Library, v. 65, n. 9, p. 1820–1833, 2014.

ERKAN, Günes; RADEV, Dragomir R. Lexrank: Graph-based Lexical Centrality as Salience in Text Summarization. **Journal of Artificial Intelligence Research**, v. 22, p. 457–479, 2004.

GAGE, Philip. A New Algorithm for Data Compression. **C Users Journal**, McPherson, KS: R & D Publications, c1987-1994., v. 12, n. 2, p. 23–38, 1994.

GAMBHIR, Mahak; GUPTA, Vishal. Recent Automatic Text Summarization Techniques: A Survey. **Artificial Intelligence Review**, Springer, v. 47, n. 1, p. 1–66, 2017.

GERANI, Shima; CARENINI, Giuseppe; NG, Raymond T. Modeling Content and Structure for Abstractive Review Summarization. **Computer Speech & Language**, Elsevier, v. 53, p. 302–331, 2019.

GOULARTE, Fábio Bif et al. A Text Summarization Method based on Fuzzy Rules and Applicable to Automated Assessment. **Expert Systems with Applications**, Elsevier, v. 115, p. 264–275, 2019.

GOVONI, Marco et al. Qresp, A Tool for Curating, Discovering and Exploring Reproducible Scientific Papers. **Scientific Data**, Nature Publishing Group, v. 6, p. 190002, 2019.

GUAN, Wang; SMETANNIKOV, Ivan; TIANXING, Man. Survey on Automatic Text Summarization and Transformer Models Applicability. In: 2020 International Conference on Control, Robotics and Intelligent System. 2020. P. 176–184.

GUPTA, Som; GUPTA, SK. Abstractive summarization: An Overview of the State of the Art. **Expert Systems with Applications**, Elsevier, v. 121, p. 49–65, 2019.

HABIB, Raja; AFZAL, Muhammad Tanvir. Sections-based Bibliographic Coupling For Research Paper Recommendation. **Scientometrics**, Springer, v. 119, n. 2, p. 643–656, 2019.

HAMEDANI, Masoud Reyhani; KIM, Sang-Wook; KIM, Dong-Jin. SimCC: A Novel Method to Consider both Content and Citations for Computing Similarity of Scientific Papers. **Information Sciences**, Elsevier, v. 334, p. 273–292, 2016.

HASHIMOTO, Hayato et al. Automatic Generation of Review Matrices as Multi-document Summarization of Scientific Papers. In: BIRNDL@ SIGIR (1). 2017. P. 69–82.

HUGGING FACE. **How do Transformers work?** https://huggingface.co/course/chapter1/4. Acesso em 31/01/2022.

IQBAL, Touseef; QURESHI, Shaima. The Survey: Text Generation Models in Deep Learning. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, 2020.

JHA, Rahul et al. NLP-Driven Citation Analysis for Scientometrics. **Natural Language Engineering**, Cambridge University Press, v. 23, n. 1, p. 93–130, 2017.

JURAFSKY, Dan; MARTIN, James H. **Speech and Language Processing (3rd (draft) ed.)** Stanford University, 2019.

EL-KASSAS, Wafaa S et al. Automatic Text Summarization: A Comprehensive Survey. **Expert Systems with Applications**, Elsevier, p. 113679, 2020.

KHAN, Arjumand Yar; KHATTAK, ABDUL SHAHID; AFZAL, Muhammad Tanvir. Extending Co-citation Using Sections Of Research Articles. **Turkish Journal of Electrical Engineering & Computer Sciences**, The Scientific e Technological Research Council of Turkey, v. 26, n. 6, p. 3345–3355, 2018.

KIEUVONGNGAM, Virapat; TAN, Bowen; NIU, Yiming. Automatic Text Summarization of COVID-19 Medical Research Articles Using BERT and GPT-2. **arXiv preprint arXiv:2006.01997**, 2020.

KITAEV, Nikita; KAISER, Łukasz; LEVSKAYA, Anselm. Reformer: The Efficient Transformer. **arXiv preprint arXiv:2001.04451**, 2020.

LANDIS, J Richard; KOCH, Gary G. The Measurement of Observer Agreement for Categorical Data. **Biometrics**, JSTOR, p. 159–174, 1977.

LI, Yikuan et al. Clinical-Longformer and Clinical-BigBird: Transformers for Long Clinical Sequences. **arXiv preprint arXiv:2201.11838**, 2022.

LIN, Chin-Yew. ROUGE: A Package for Automatic Evaluation of Summaries. In: TEXT summarization branches out. 2004. P. 74–81.

LIN, Tianyang et al. A Survey of Transformers. **arXiv preprint arXiv:2106.04554**, 2021.

LIU, Yinhan et al. Roberta: A Robustly Optimized BERT Pre-training Approach. **arXiv preprint arXiv:1907.11692**, 2019.

LOSHCHILOV, Ilya; HUTTER, Frank. Decoupled Weight Decay Regularization. **arXiv preprint arXiv:1711.05101**, 2017.

LUU, Tuan Minh; LE, Huong Thanh; HOANG, Tan Minh. A Hybrid Model Using The Pretrained BERT And Deep Neural Networks With Rich Feature For Extractive Text Summarization. **Journal of Computer Science and Cybernetics**, v. 37, n. 2, p. 123–143, 2021.

MAMAKAS, Dimitris et al. Processing Long Legal Documents with Pre-trained Transformers: Modding LegalBERT and Longformer. **arXiv preprint arXiv:2211.00974**, 2022.

MARCOS-PABLOS, Samuel; GARCÍA-PEÑALVO, Francisco J. Information Retrieval Methodology for Aiding Scientific Database Search. **Soft Computing**, Springer, v. 24, n. 8, p. 5551–5560, 2020.

MARIANI, Joseph; FRANCOPOULO, Gil; PAROUBEK, Patrick. The nlp4nlp Corpus (i): 50 Years of Publication, Collaboration and Citation in Speech and Language Processing. **Frontiers in Research Metrics and Analytics**, Frontiers, p. 36, 2019.

MERROUNI, Zakariae Alami; FRIKH, Bouchra; OUHBI, Brahim. Automatic Keyphrase Extraction: A Survey and Trends. **Journal of Intelligent Information Systems**, Springer, v. 54, n. 2, p. 391–424, 2020.

MI, Haitao; HUANG, Liang. Forest-based Translation Rule Extraction. In: PROCEEDINGS of the 2008 Conference on Empirical Methods in Natural Language Processing. 2008. P. 206–214.

MILLER, Derek. Leveraging BERT for Extractive Text Summarization on Lectures. **arXiv preprint arXiv:1906.04165**, 2019.

MOHD, Mudasir; JAN, Rafiya; SHAH, Muzaffar. Text Document Summarization using Word Embedding. **Expert Systems with Applications**, Elsevier, v. 143, p. 112958, 2020.

NADARAYA, Elizbar A. On Estimating Regression. **Theory of Probability & Its Applications**, SIAM, v. 9, n. 1, p. 141–142, 1964.

NALLAPATI, Ramesh et al. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. **arXiv preprint arXiv:1602.06023**, 2016.

NIKOLOV, Nikola I; PFEIFFER, Michael; HAHNLOSER, Richard HR. Data-driven Summarization of Scientific Articles. **arXiv preprint arXiv:1804.08875**, 2018.

OTT, Myle et al. fairseq: A Fast, Extensible Toolkit for Sequence Modelling. **arXiv preprint arXiv:1904.01038**, 2019.

POELMANS, Jonas et al. Text Mining Scientific Papers: A Survey on FCA-based Information Retrieval Research. In: SPRINGER. INDUSTRIAL Conference on Data Mining. 2012. P. 273–287.

PRICE, Morgan N; ARKIN, Adam P. PaperBLAST: Text Mining Papers for Information about Homologs. **MSystems**, Am Soc Microbiol, v. 2, n. 4, e00039–17, 2017.

PUTRA, Jan Wira Gotama; KHODRA, Masayu Leylia. Automatic Title Generation in Scientific Articles for Authorship Assistance: A Summarization Approach. **Journal of ICT Research and Applications**, v. 11, n. 3, p. 253–267, 2017.

QAZVINIAN, Vahed et al. Generating Extractive Summaries of Scientific Paradigms. **Journal of Artificial Intelligence Research**, v. 46, p. 165–201, 2013.

QUIZA, Ramón; DAVIM, J Paulo. Computational Methods and Optimization. In: MACHINING of hard materials. Springer, 2011. P. 177–208.

RADFORD, Alec et al. Language Models Are Unsupervised Multitask Learners. **OpenAI blog**, v. 1, n. 8, p. 9, 2019.

RAE, Jack W et al. Compressive Transformers for Long-Range Sequence Modelling. **arXiv preprint arXiv:1911.05507**, 2019.

ROMANOV, Aleksandr; LOMOTIN, Konstantin; KOZLOVA, Ekaterina. Application of Natural Language Processing Algorithms to the Task of Automatic Classification of Russian Scientific Texts. **Data Science Journal**, Ubiquity Press, v. 18, n. 1, 2019.

RONZANO, Francesco; SAGGION, Horacio. Knowledge Extraction and Modeling from Scientific Publications. In: SPRINGER. INTERNATIONAL workshop on semantic, analytics, visualization. 2016. P. 11–25.

AL-SABAHI, Kamal; ZUPING, Zhang; NADHER, Mohammed. A Hierarchical Structured Self-Attentive Model for Extractive Document Summarization (HSSAS). **IEEE Access**, IEEE, v. 6, p. 24205–24212, 2018.

SCHUSTER, Mike; NAKAJIMA, Kaisuke. Japanese and Korean Voice Search. In: IEEE. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2012. P. 5149–5152.

SONG, Shengli; HUANG, Haitao; RUAN, Tongxiao. Abstractive Text Summarization using LSTM-CNN based Deep Learning. **Multimedia Tools and Applications**, Springer, v. 78, n. 1, p. 857–875, 2019.

SUKHBAATAR, Sainbayar et al. Adaptive Attention Span in Transformers. **arXiv preprint arXiv:1905.07799**, 2019.

SUN, Shiliang; LUO, Chen; CHEN, Junyu. A Review of Natural Language Processing Techniques for Opinion Mining Systems. **Information fusion**, Elsevier, v. 36, p. 10–25, 2017.

SUTSKEVER, Ilya; VINYALS, Oriol; LE, Quoc V. Sequence to Sequence Learning with Neural Networks. In: ADVANCES in Neural Information Processing Systems. 2014. P. 3104–3112.

TAN, Jiwei; WAN, Xiaojun; XIAO, Jianguo. Abstractive Document Summarization with a Graph-based Attentional Neural Model. In: PROCEEDINGS of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017. P. 1171–1181.

TAYLOR, Wilson L. “Cloze procedure”: A New Tool for Measuring Readability. **Journalism Quarterly**, SAGE Publications Sage CA: Los Angeles, CA, v. 30, n. 4, p. 415–433, 1953.

VASWANI, Ashish et al. Attention Is All You Need. In: ADVANCES in Neural Information Processing Systems. 2017. P. 5998–6008.

WANG, Qingyun et al. PaperRobot: Incremental Draft Generation of Scientific Ideas. **arXiv preprint arXiv:1905.07870**, 2019.

XU, Huiyan; WANG, Zhijian; WENG, Xiaolan. Scientific Literature Summarization using Document Structure and Hierarchical Attention Model. **IEEE Access**, IEEE, v. 7, p. 185290–185300, 2019.

XU, Kelvin et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In: PMLR. INTERNATIONAL Conference on Machine Learning. 2015. P. 2048–2057.

YANG, Shansong et al. KeyphraseDS: Automatic Generation of Survey by Exploiting Keyphrase Information. **Neurocomputing**, Elsevier, v. 224, p. 58–70, 2017.

YANG, Zhilin et al. XLNet: Generalized Autoregressive Pretraining for Language Understanding. **Advances in Neural Information Processing Systems**, v. 32, 2019.

YAO, Jin-ge; WAN, Xiaojun; XIAO, Jianguo. Recent Advances in Document Summarization. **Knowledge and Information Systems**, Springer, v. 53, n. 2, p. 297–336, 2017.

YASUNAGA, Michihiro et al. ScisummNet: A Large Annotated Corpus and Content-Impact Models for Scientific Paper Summarization with Citation Networks. In: PROCEEDINGS of AAAI 2019. 2019.

YOUNG, Tom et al. Recent Trends in Deep Learning Based Natural Language Processing. **IEEE Computational Intelligence Magazine**, IEEE, v. 13, n. 3, p. 55–75, 2018.

ZAHEER, Manzil et al. Big Bird: Transformers for Longer Sequences. **Advances in Neural Information Processing Systems**, v. 33, p. 17283–17297, 2020.