

Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Engenharia de Software

# Utilizando Inteligência Artificial para Melhorar a Busca por Informação em Textos Científicos

Autor: Henrique Martins de Messias  
Orientador: Dr. Fabricio Ataides Braz

Brasília, DF  
2022



Henrique Martins de Messias

# **Utilizando Inteligência Artificial para Melhorar a Busca por Informação em Textos Científicos**

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dr. Fabricio Ataides Braz

Brasília, DF

2022

---

Henrique Martins de Messias

Utilizando Inteligência Artificial para Melhorar a Busca por Informação em  
Textos Científicos/ Henrique Martins de Messias. – Brasília, DF, 2022-  
47 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Fabricio Ataides Braz

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA , 2022.

1. Inteligência Artificial. 2. Científico. I. Dr. Fabricio Ataides Braz. II.  
Universidade de Brasília. III. Faculdade UnB Gama. IV. Utilizando Inteligência  
Artificial para Melhorar a Busca por Informação em Textos Científicos

CDU 02:141:005.6

---

Henrique Martins de Messias

## **Utilizando Inteligência Artificial para Melhorar a Busca por Informação em Textos Científicos**

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

---

**Dr. Fabricio Ataides Braz**  
Orientador

---

**Dr. Nilton Correia da Silva**  
Convidado 1

---

**Dr. Bruno Cesar Ribas**  
Convidado 2

Brasília, DF  
2022

Este trabalho é dedicado a Deus e aos meus pais, pois sempre estiveram presentes em meus sonhos e conquistas.

# Agradecimentos

Gostaria de agradecer e dedicar este trabalho, primeiramente, a Deus, por me ajudar a superar todos os desafios, obstáculos e dificuldades da vida e a conquistar meus objetivos, não apenas acadêmicos. Se estou concluindo mais uma importante etapa da minha vida, é porque Ele sempre esteve comigo e sei que estará nas próximas etapas também.

Também sou muito grato aos meus pais, que me deram suporte e encorajamento durante toda a minha vida. Graças aos seus esforços e ensinamentos, pude finalizar essa etapa da minha vida com responsabilidade e da melhor maneira possível. Seus esforços para que eu obtivesse sucesso em tudo o que eu tentasse não serão esquecidos e espero, um dia, poder retribuir de maneira adequada tudo o que fizeram por mim.

Gostaria de agradecer aos meus amigos, tanto aos que conheci durante a faculdade quanto aos que conheço antes desse período, que me ajudaram não só com os trabalhos, mas comemoram minhas conquistas junto comigo.

Gostaria de agradecer também a todos os professores que me auxiliaram em meus estudos desde o início da faculdade até o fim, e agradecer também por sua paciência, dedicação e excelente trabalho.

Por fim, gostaria de agradecer a todos que, direta ou indiretamente, me auxiliaram em minha formação.

# Resumo

Com o surgimento da pandemia de COVID-19, muitos artigos foram publicados sobre o novo coronavírus e seus impactos. Com um grande volume de artigos sendo publicado, aumenta a dificuldade que um pesquisador tem de encontrar a informação que está procurando, pois pode se deparar com artigos que tratam de temas semelhantes, mas não sobre o assunto desejado. O objetivo central do trabalho é usar inteligência artificial para melhorar a busca por informação em textos científicos ao extrair dados como entidades e relações. Dessa forma, a busca será feita não com a simples menção de um ou mais termos no artigo, mas será feita com base na relação entre os termos, reduzindo o tempo necessário para que o pesquisador encontre o que deseja. Para alcançar esse objetivo, tomou-se como base um trabalho já realizado no assunto para usar como referência, e, em seguida, foram realizadas tentativas para melhorar o resultado alcançado pelo trabalho ao trocar o modelo utilizado originalmente e ao usar de supervisão fraca para adquirir mais dados rotulados. Verificou-se que substituir o modelo por outro foi capaz de melhorar os resultados, enquanto adquirir mais dados não foi. A partir dos resultados encontrados podemos concluir que é possível melhorar os resultados e, por consequência, aperfeiçoar a busca por informação em textos científicos.

**Palavras-chave:** busca; informação; científico; inteligência artificial.

# Abstract

With the emergence of the COVID-19 pandemic, many articles were published about the new coronavirus and its impacts. With a large volume of articles being published, the difficulty that a researcher has to find the information he or she is looking for increases, as he or she may come across articles that deal with similar topics, but not on the desired subject. The main objective of the work is to use artificial intelligence to improve the search for information in scientific texts by extracting data such as entities and relations. In this way, the search will not be done with the simple mention of one or more terms in the article, but will be done based on the relation between the terms, reducing the time necessary for the researcher to find what he or she wants. To achieve this objective, work already done on the subject was taken as a basis to be used as a reference, and then attempts were made to improve the result achieved by the work by changing the model used originally and by using weak supervision to acquire more labeled data. It was found that replacing the model with another one was able to improve the results, while acquiring more data was not. From the results found, we can conclude that it is possible to improve the results and, consequently, improve the search for information in scientific texts.

**Key-words:** search; information; scientific; artificial intelligence.



# Lista de ilustrações

Figura 1 – Tipos de supervisão fraca . . . . .	17
Figura 2 – Arquitetura do modelo Transformer . . . . .	21
Figura 3 – BERT - Procedimentos . . . . .	22
Figura 4 – Exemplo de amostra do MECHANIC . . . . .	29
Figura 5 – Arquitetura do DyGIE++ . . . . .	30
Figura 6 – Distribuição das relações no conjunto de dados de teste do MECHANIC	32
Figura 7 – Tamanho das entidades no conjunto de dados de treino do MECHANIC	32
Figura 8 – Nova distribuição das relações no conjunto de dados de treino do ME- CHANIC . . . . .	33
Figura 9 – Nova distribuição das relações no conjunto de dados de validação do MECHANIC . . . . .	33
Figura 10 – Nova distribuição das relações no conjunto de dados de teste do ME- CHANIC . . . . .	34
Figura 11 – Arquitetura do SpERT . . . . .	36
Figura 12 – Exemplo de amostra do MECHANIC após adaptá-lo para o SpERT . .	38

# Lista de tabelas

Tabela 1 – Melhores resultados no DyGIE++ . . . . .	35
Tabela 2 – Hiperparâmetros do DyGIE++ ajustados e seus intervalos . . . . .	35
Tabela 3 – Pontuação F1 do DyGIE++ e SpERT com o SciERC . . . . .	38
Tabela 4 – Melhores resultados no SpERT . . . . .	39
Tabela 5 – Hiperparâmetros do SpERT ajustados e seus intervalos . . . . .	40
Tabela 6 – Melhores resultados com mais dados no DyGIE++ . . . . .	42
Tabela 7 – Melhores resultados com mais dados no SpERT . . . . .	43

# Lista de abreviaturas e siglas

DL	Aprendizado Profundo, do inglês <i>Deep Learning</i>
IA	Inteligência Artificial
ML	Aprendizado de Máquina, do inglês <i>Machine Learning</i>
NLP	Processamento de Linguagem Natural, do inglês <i>Natural Language Processing</i>

# Sumário

	<b>Introdução</b>	<b>13</b>
<b>1</b>	<b>REFERENCIAL TEÓRICO</b>	<b>15</b>
<b>1.1</b>	<b>Aprendizado profundo</b>	<b>15</b>
1.1.1	Redes neurais artificiais	15
1.1.2	Treino, validação e teste	16
1.1.3	Otimização de hiperparâmetros	16
1.1.4	Supervisão fraca	16
1.1.5	Métricas	17
1.1.5.1	Precisão	18
1.1.5.2	Revocação	18
1.1.5.3	Pontuação F1	19
<b>1.2</b>	<b>Processamento de Linguagem Natural</b>	<b>20</b>
1.2.1	Tokenização	20
1.2.2	Incorporação	20
1.2.3	Extração de Informações	20
<b>1.3</b>	<b>Mecanismo de atenção</b>	<b>21</b>
<b>1.4</b>	<b>Transformer</b>	<b>21</b>
1.4.1	Arquitetura	21
1.4.2	BERT	22
1.4.2.1	Pré-treino	22
1.4.2.2	Ajuste fino	23
1.4.3	SciBERT	23
<b>2</b>	<b>MÉTODOS</b>	<b>24</b>
<b>2.1</b>	<b>Encontrar referência</b>	<b>24</b>
<b>2.2</b>	<b>Melhorar resultados da referência</b>	<b>24</b>
2.2.1	Técnicas	24
2.2.1.1	Trocar modelo	24
2.2.1.2	Incrementar amostragem	25
<b>3</b>	<b>RESULTADOS</b>	<b>26</b>
<b>3.1</b>	<b>Base de conhecimento de mecanismos</b>	<b>26</b>
3.1.1	O que foi realizado no trabalho	26
3.1.1.1	Relação de mecanismos	26
3.1.1.2	MECHANIC	27

3.1.1.3	COMB . . . . .	29
3.1.2	DyGIE++ . . . . .	29
3.1.2.1	Arquitetura . . . . .	30
3.1.3	Resultados . . . . .	31
3.1.4	Problemas encontrados . . . . .	31
3.1.4.1	Trabalho irreprodutível . . . . .	31
3.1.4.2	Conjunto de dados mal distribuído . . . . .	31
3.1.4.3	Hiperparâmetro mal configurado . . . . .	32
3.1.5	Soluções . . . . .	33
3.1.5.1	Conjunto de dados redistribuído . . . . .	33
3.1.5.2	Ajuste nos hiperparâmetros . . . . .	34
3.1.5.3	Novos resultados . . . . .	34
<b>3.2</b>	<b>Novo modelo - SpERT . . . . .</b>	<b>35</b>
3.2.1	Arquitetura . . . . .	35
3.2.2	Resultados com dados científicos . . . . .	37
3.2.3	Treino do SpERT com o MECHANIC . . . . .	38
3.2.3.1	Adaptação do conjunto de dados . . . . .	38
3.2.3.2	Resultados . . . . .	39
<b>3.3</b>	<b>Incremento de amostragem . . . . .</b>	<b>40</b>
3.3.1	Origem do dado . . . . .	40
3.3.2	Classificando sentenças com SpERT . . . . .	40
3.3.2.1	Tratamento do dado . . . . .	41
3.3.2.2	Resultado . . . . .	42
3.3.3	Classificando sentenças com DyGIE++ . . . . .	42
3.3.3.1	Tratamento do dado . . . . .	43
3.3.3.2	Resultado . . . . .	43
<b>4</b>	<b>CONCLUSÃO . . . . .</b>	<b>44</b>
<b>4.1</b>	<b>Trabalhos futuros . . . . .</b>	<b>44</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>46</b>

# Introdução

No dia 31 de dezembro de 2019, a Organização Mundial da Saúde (OMS) foi informada de que várias pessoas na cidade de Wuhan, na China, estavam sendo hospitalizadas por causa de pneumonia. Na semana seguinte, autoridades chinesas confirmaram a identificação de um novo tipo de coronavírus (SARS-CoV-2). A doença causada pelo novo coronavírus foi mais tarde chamada de COVID-19.

Conforme relatado no artigo (GIROLAMO; REYNDERS, 2020), nos primeiros 3 meses da pandemia de COVID-19, 1165 artigos foram publicados e indexados no *PubMed*, motor de busca de livre acesso que permite pesquisar e obter literaturas sobre ciência e biomedicina, enquanto que nos primeiros 3 meses da pandemia de H1N1, que começou em 2009, foram indexados 223 artigos. Ou seja, avaliando o mesmo período de tempo, 3 meses, para as duas pandemias, foram publicados 5 vezes mais artigos sobre a COVID-19 do que sobre H1N1, demonstrando o avanço na velocidade de produção e publicação de artigos científicos.

Além disso, segundo o artigo (RICCABONI; VERGINER, 2022), a pandemia de COVID-19 mobilizou a comunidade científica mundial. Isso fica comprovado ao verificar que, desde 2019, mais de 240.000 artigos relacionados à COVID-19 foram publicados e estão disponíveis para consulta no *PubMed*. Em muitos deles pode ser necessário ter algum conhecimento prévio em diversas áreas, como biomedicina, física, química, engenharia, ciências sociais, entre outros. Com tantos artigos publicados abordando termos e contextos novos e diversos, pesquisadores podem enfrentar dificuldades em encontrar as informações que procuram.

Devido à urgência em acabar com a pandemia de COVID-19, aprimorar a busca pela informação desejada pode otimizar o tempo de pesquisa, ainda mais em um assunto recente e interdisciplinar. Além disso, melhorar a busca por informação não irá auxiliar apenas durante a pandemia, mas pode ser um benefício utilizado a qualquer momento.

Considerando o contexto e os problemas mencionados, este trabalho tem como objetivo apresentar conceitos e definições de inteligência artificial e, em seguida, demonstrar como utilizar processamento de linguagem natural e outras técnicas para melhorar a busca por informação em artigos científicos.

Para atingir esse objetivo, primeiro será feita uma busca por algum trabalho já realizado sobre extração de informação em textos científicos. Ao selecionar um trabalho no assunto, que será usado como referência, o próximo passo será a busca pela melhoria dos resultados por meio da substituição do modelo utilizado originalmente por outro modelo e pela técnica de supervisão fraca para adquirir mais dados rotulados.

Os próximos capítulos abordarão, primeiramente, tópicos essenciais para entender o desenvolvimento e os resultados do trabalho. Em seguida, serão apresentados os métodos utilizados para alcançar o objetivo proposto, ou seja, será explicado sobre a escolha do trabalho de referência e sobre as técnicas utilizadas para tentar melhorar seus resultados. Após isso, serão apresentados os detalhes da execução dos métodos e os seus resultados. E, concluindo, um resumo da proposta do trabalho, seus resultados, sua importância e sugestões para trabalhos futuros.

# 1 Referencial Teórico

Antes de explicar sobre os métodos utilizados para alcançar o objetivo do trabalho e os resultados atingidos, é necessário apresentar alguns conceitos e técnicas de inteligência artificial, como o que é aprendizado profundo, processamento de linguagem natural e *BERT*.

## 1.1 Aprendizado profundo

Como explicado em (JANIESCH; ZSCHECH; HEINRICH, 2021), Inteligência Artificial (IA) envolve qualquer técnica que permite que computadores resolvam tarefas complexas de forma independente ou com intervenção mínima de um ser humano. As primeiras pesquisas na área de IA se concentraram em programar explicitamente como o computador deveria agir em determinadas situações. Entretanto, esse paradigma possui diversas limitações, já que se tornou cada vez mais difícil para o ser humano programar todo o conhecimento necessário para realizar tarefas complexas.

*Machine Learning*, ou aprendizado de máquina (ML), é uma área da IA que estuda como fazer com que computadores aprendam automaticamente por meio da experiência. Muitos desenvolvedores consideram muito mais fácil treinar um sistema mostrando exemplos do comportamento desejado de entrada-saída do que programá-lo manualmente, antecipando a resposta desejada para todas as entradas possíveis. (JORDAN; MITCHELL, 2015). Porém, técnicas convencionais de ML eram limitadas em sua capacidade de processar dados naturais em sua forma bruta (LECUN; BENGIO; HINTON, 2015).

*Deep Learning*, ou aprendizado profundo (DL), é um conceito de ML baseado em redes neurais artificiais, que consistem em uma camada ou mais de neurônios artificiais (JANIESCH; ZSCHECH; HEINRICH, 2021). Técnicas de DL permite que uma máquina seja alimentada com dados brutos e descubra automaticamente as representações necessárias para detecção ou classificação (LECUN; BENGIO; HINTON, 2015).

### 1.1.1 Redes neurais artificiais

As redes neurais artificiais são modelos computacionais paralelos massivos que imitam a função do cérebro humano e consistem em um grande número de nós (chamados de neurônios) ligados por conexões ponderadas (os pesos amplificam ou diminuem o valor da informação). A saída de cada nó depende apenas das informações disponíveis localmente no nó, armazenadas internamente ou de informações que chegam por meio de conexões ponderadas. Cada unidade recebe entradas de muitos outros nós e transmite sua saída



para outros nós. Por si só, um único nó não é muito poderoso, mas a combinação de muitos neurônios de forma adequada é capaz de produzir um sistema poderoso ([DONGARE et al., 2012](#)).

### 1.1.2 Treino, validação e teste

Como explicado em ([HASTIE et al., 2009](#)), caso seja necessário estimar o desempenho de diferentes modelos para escolher o melhor, ou, tendo escolhido um modelo final, estimar seu erro de previsão (erro de generalização) em novos dados, a melhor abordagem em ambos os casos é dividir o conjunto de dados em três partes:

- Conjunto de treino: Usado para treinar o modelo.
- Conjunto de validação: Usado para estimar o erro de previsão para a seleção do modelo.
- Conjunto de teste: Usado para avaliação do erro de generalização do modelo final escolhido. Idealmente, o conjunto de teste não deve ser visto pelo modelo e deve ser usado apenas no final da análise dos dados.

### 1.1.3 Otimização de hiperparâmetros

Existem dois tipos de parâmetros em modelos de ML e DL, como explicado em ([YANG; SHAMI, 2020](#)): os parâmetros do modelo, que podem ser atualizados conforme o modelo aprende com o dado de treino; e os hiperparâmetros, que não aprendem com o dado e precisam ser configurados antes do treino para definir a arquitetura do modelo. Por essa razão, os hiperparâmetros devem ser configurados da melhor forma possível, isto é, devem ser otimizados. É possível utilizar técnicas para automatizar a otimização e algumas das vantagens de automatizar esse processo são:

- Reduzir o esforço humano necessário.
- Melhorar a performance do modelo.
- Tornar o modelo mais reproduzível.

### 1.1.4 Supervisão fraca

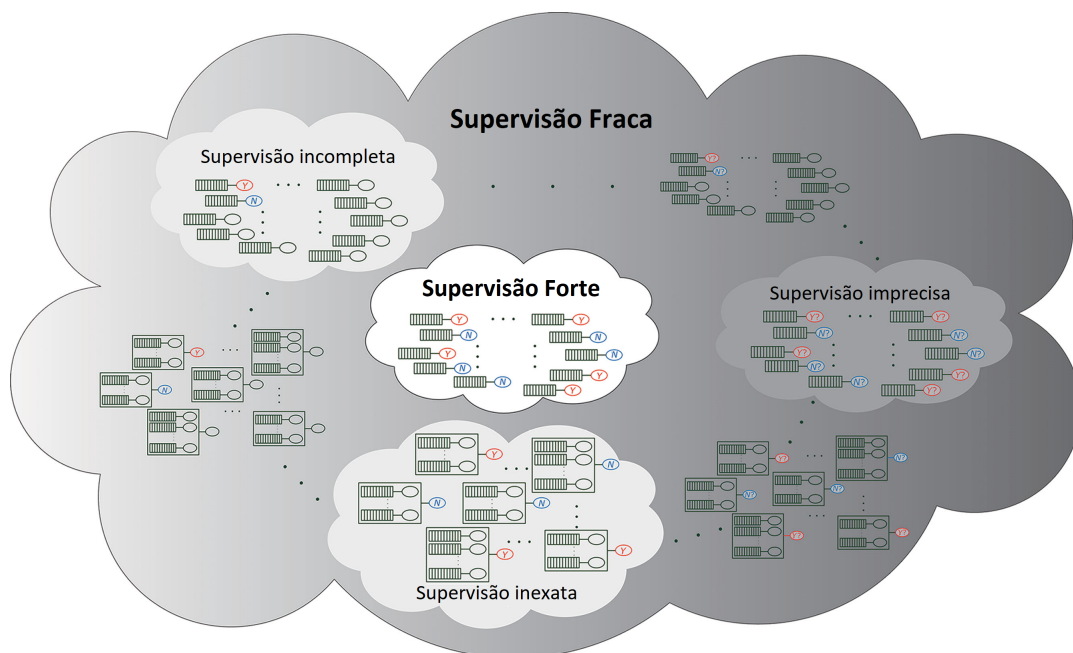
Segundo ([ZHOU, 2018](#)), muitos modelos de ML aprendem de um conjunto de dados de treino que contém muitas amostras, sendo que cada amostra consiste em duas partes: um vetor de características, que descreve a amostra, e um rótulo, que representa a saída verdadeira (o rótulo pode ser uma classe se a tarefa for de classificação ou um

número se a tarefa for de regressão). Entretanto, existe, em muitas tarefas, a dificuldade de conseguir grandes volumes de dado rotulado. Uma possível forma de contornar esse problema é utilizando a supervisão fraca.

A supervisão fraca é uma abordagem de ML em que fontes imprecisas ou limitadas de dados, como funções de rotulação e modelos de IA antigos, são usadas para rotular grandes quantidades de dados. Como mostrado na Figura 1, a supervisão fraca pode ser dividida em três tipos:

- Supervisão incompleta: poucos dados rotulados estão disponíveis, sendo insuficiente para treinar um modelo de maneira satisfatória, mas uma quantidade considerável de dado não rotulado está disponível.
- Supervisão inexata: existe informação supervisionada, mas não tão exato quanto desejado.
- Supervisão imprecisa: os rótulos nem sempre estão corretos.

**Figura 1** – Tipos de supervisão fraca



Fonte: (ZHOU, 2018)

### 1.1.5 Métricas

Para avaliar a qualidade de um modelo, é necessário algum tipo de métrica para quantificar quão bem um modelo executa determinada tarefa. Segundo (TAKAHASHI et al., 2022), a melhor métrica a ser utilizada varia dependendo da tarefa e do tipo de classificação, que pode ser binária (quando há duas possibilidades de classes) ou multiclasse

(quando há mais de duas possibilidades de classes). Para as métricas a seguir, considere que:

- $N$  = número de classes
- $TP$  = verdadeiros positivos, isto é, o valor correto é positivo e o modelo previu um valor positivo
- $FP$  = falsos positivos, isto é, o valor correto é negativo e o modelo previu um valor positivo
- $FN$  = falsos negativos, isto é, o valor correto é positivo e o modelo previu um valor negativo

#### 1.1.5.1 Precisão

A métrica *precision* (precisão), segundo (POWERS, 2020), denota a proporção de casos previstos como positivos que são, de fato, positivos. Essa métrica é útil quando falsos positivos são muito prejudiciais, como no caso de classificar um e-mail como *spam* ou não *spam* (classificar de maneira errada um e-mail como *spam* pode fazer com que o usuário não verifique e-mails importantes). A fórmula para calcular a precisão de um modelo para uma classe específica está descrita em 1.1.

$$P = \frac{TP}{TP + FP} \quad (1.1)$$

Caso seja necessário calcular a precisão para mais de uma classe, é possível usar uma métrica chamada *micro precision* (micro precisão), que leva em conta cada uma das classes em sua fórmula, descrita em 1.2.

$$miP = \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FP_i)} \quad (1.2)$$

#### 1.1.5.2 Revocação

A métrica *recall* (revocação), segundo (POWERS, 2020), é a proporção de casos positivos que foram corretamente previstos como positivos. Essa métrica é útil quando falsos negativos são muito prejudiciais, como no caso de detecção de fraudes bancárias (classificar uma transação como não fraudulenta, quando na verdade é uma fraude, pode trazer muitos problemas para o banco). A fórmula para calcular a revocação de um modelo para uma classe específica está descrita em 1.3.

$$R = \frac{TP}{TP + FN} \quad (1.3)$$

Caso seja necessário calcular a revocação para mais de uma classe, é possível usar uma métrica chamada *micro recall* (micro revocação), que leva em conta cada uma das classes em sua fórmula, descrita em 1.4.

$$miR = \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FN_i)} \quad (1.4)$$

### 1.1.5.3 Pontuação F1

Essa métrica é amplamente utilizada para medir a performance na classificação de texto na área de extração de informação em ML, segundo (TAKAHASHI et al., 2022). A métrica F1-score (pontuação F1) é a média harmônica da precisão e da revocação e é útil quando o equilíbrio entre a precisão e a revocação é desejado. A fórmula dessa métrica, para uma classe específica, é descrita em 1.5.

$$F_1 = 2 \times \frac{P \times R}{P + R} \quad (1.5)$$

Caso seja necessário fazer o cálculo para mais de uma classe, existem duas métricas que podem ser usadas: micro-média pontuação F1 e macro-média pontuação F1.

A métrica micro-média pontuação F1 é baseada na micro precisão e na micro revocação, ou seja, nos somatórios de verdadeiro positivo, falso positivo e falso negativo de todas as classes, como mostrado na fórmula em 1.6. Dessa forma, se o dado não for balanceado (isto é, se houver muito mais amostras de uma classe do que de outra), essa métrica será mais afetada pelas classes com mais amostras, enquanto que as classes que possuem poucas amostras não irão afetar muito o resultado.

$$miF_1 = 2 \times \frac{miP \times miR}{miP + miR} \quad (1.6)$$

Já a macro-média pontuação F1 é calculada usando a média da pontuação F1 de cada classe, ou seja, classes com muitas amostras podem influenciar o resultado da métrica da mesma forma que classes com poucas amostras. 1.7.

$$maF_1 = \frac{1}{N} \times \sum_{i=1}^N F1_i \quad (1.7)$$

## 1.2 Processamento de Linguagem Natural

*Natural Language Processing*, ou Processamento de Linguagem Natural (NLP), como descrito em (CHOPRA; PRASHAR; SAIN, 2013), é um subcampo da IA e da linguística dedicado a fazer os computadores entenderem as declarações ou palavras escritas em linguagem natural, isto é, a linguagem falada ou escrita por seres humanos para se comunicarem.

### 1.2.1 Tokenização

O primeiro passo em tarefas de NLP é a tokenização, que é o processo de identificar os *tokens* de uma sentença, sendo que cada *token* é uma unidade básica que não precisa ser decomposta em um processamento subsequente (WEBSTER; KIT, 1992).

### 1.2.2 Incorporação

*Network embedding*, ou incorporação de rede, refere-se à abordagem de aprender representações de características latentes de baixa dimensão para os nós ou conexões em uma rede. O princípio básico é aprender codificações para os nós na rede de forma que a similaridade no espaço de incorporação reflita a similaridade na rede (ARSOV; MIRCEVA, 2019).

*Word embeddings*, ou incorporações de palavras, é um exemplo de uso dessa abordagem na área de NLP. As incorporações de palavras têm sido muito utilizadas como entrada de recursos para modelos de ML devido a sua capacidade de capturar propriedades semânticas úteis e relações linguísticas entre palavras (ou seja, palavras que possuem significados semelhantes são representados de forma semelhante) (WANG et al., 2018).

### 1.2.3 Extração de Informações

Extração de informação, como explicado em (MOONEY; BUNESCU, 2005), é a tarefa de localizar pedaços de dados específicos em documentos escritos em linguagem natural, dessa forma transformando texto não estruturado em informação estruturada.

Ainda segundo (MOONEY; BUNESCU, 2005), um tipo de extração de informação é conhecido como reconhecimento de entidade nomeada (ou extração de entidade) e consiste em identificar referências a certos tipos de objetos como pessoas, empresas e locais.

(MOONEY; BUNESCU, 2005) também fala sobre a tarefa de extrair relações, que consiste em identificar quais entidades possuem uma relação entre si e que tipo de relação possuem.

### 1.3 Mecanismo de atenção

Mecanismo de atenção foi introduzido pela primeira vez em NLP por (BAHDANAU; CHO; BENGIO, 2014), ao fazer com que um modelo que realizava a tarefa de tradução fosse capaz de buscar, em uma sentença, partes que fossem relevantes para prever uma palavra alvo.

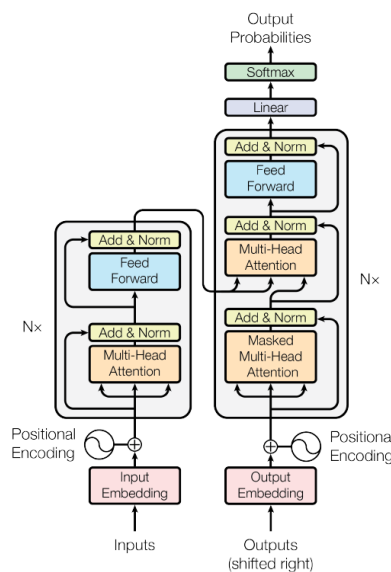
Em muitas tarefas da área de NLP, os elementos que compõem o texto possuem relevâncias diferentes para a tarefa em questão. Por exemplo, em análise de sentimentos, palavras como **bom** ou **ruim** podem ser mais relevantes que outras palavras e, em outros contextos, podem não ter relevância. Já na tarefa de traduzir uma sentença de um idioma para o outro, certas palavras podem não ter relevância na tradução da palavra seguinte. Portanto, é interessante que as soluções sejam capazes de identificar partes relevantes em uma sentença. Por essa e outras razões, atenção vem se tornando uma característica cada vez mais comum em arquiteturas de NLP (GALASSI; LIPPI; TORRONI, 2020).

### 1.4 Transformer

O modelo Transformer é uma arquitetura de modelo que usa mecanismo de atenção para extrair dependências entre as entradas e saídas (VASWANI et al., 2017). O Transformer foi projetado para ser um modelo de transdução, ou seja, resolver tarefas de modelagem de linguagem e tradução automática, por exemplo.

#### 1.4.1 Arquitetura

**Figura 2** – Arquitetura do modelo Transformer



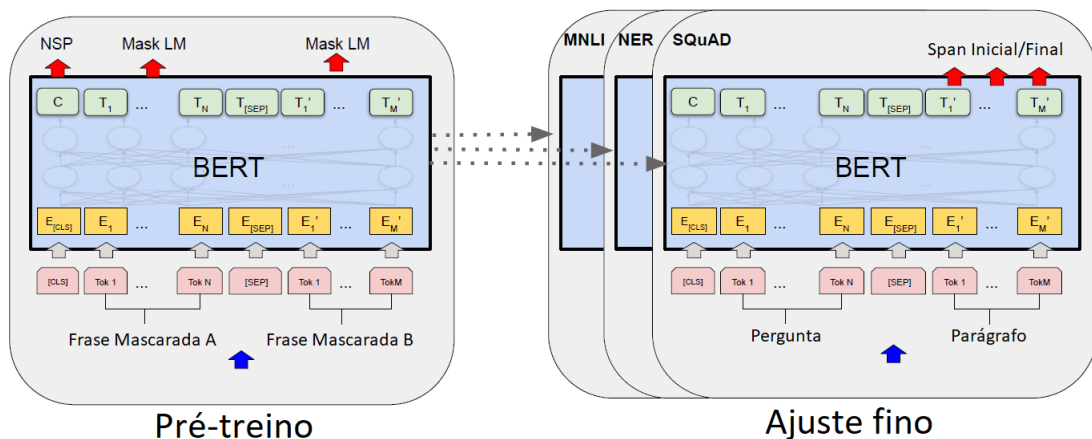
Fonte: (VASWANI et al., 2017)

O *encoder* (codificador), representado pela metade da esquerda na Figura 2, mapeia uma sequência de entrada de representações de símbolos  $(x_1, \dots, x_n)$  para uma sequência de representações contínuas  $\mathbf{z} = (z_1, \dots, z_n)$ . Já o *decoder* (decodificador), representado pela metade da direita na Figura 2, recebe  $\mathbf{z}$  e gera uma sequência de saída  $(y_1, \dots, y_m)$  de símbolos um elemento de cada vez. A cada passo o modelo consome os símbolos gerados anteriormente como entrada adicional ao gerar o próximo.

### 1.4.2 BERT

O BERT, como mostrado em (DEVLIN et al., 2018), é um modelo projetado para pré-treinar a partir de texto não rotulado e ser ajustado com uma camada de saída adicional para criar novos modelos que resolvam tarefas no campo de NLP, como responder perguntas e extrair informações de textos. A arquitetura do BERT é praticamente idêntica ao modelo Transformer original.

Figura 3 – BERT - Procedimentos



Fonte: (DEVLIN et al., 2018)

A Figura 3 mostra os procedimentos de pré-treino e de ajuste fino. A arquitetura dos dois procedimentos é a mesma, exceto pelas camadas de saída, que é adaptada à tarefa que se deseja realizar.

#### 1.4.2.1 Pré-treino

Durante a etapa de pré-treino, o modelo é treinado usando dados não rotulados em duas tarefas diferentes: mascaramento e previsão da próxima sentença.

Na parte de mascaramento, 15% dos tokens de cada sentença são mascarados (ou seja, se tornam desconhecidos) aleatoriamente. A tarefa é prever qual era a palavra antes de ser mascarada.

Na parte de prever a próxima sentença, ao escolher duas sentenças A e B para cada exemplo do pré-treino, 50% das vezes B é a sentença que vem depois de A e 50% das

vezes é uma sentença aleatória do texto. A tarefa é prever se a sentença B vem depois de A ou não e, dessa forma, o modelo entenderá os relacionamentos entre sentenças.

#### 1.4.2.2 Ajuste fino

A etapa de ajuste fino é dependente da tarefa que se quer realizar e consiste em passar as entradas e saídas dessa tarefa para o BERT e ajustar todos os parâmetros. O processo de ajuste fino leva muito menos tempo para ser realizado se comparado com o processo de pré-treino.

#### 1.4.3 SciBERT

O SciBERT, como explicado em (BELTAGY; LO; COHAN, 2019), é um modelo de linguagem pré-treinado baseado no BERT, com a diferença que o SciBERT tem como objetivo resolver tarefas de NLP no domínio científico. Para alcançar esse objetivo, o SciBERT foi treinado com um grande volume de texto científico.

Com exceção do domínio científico, os procedimentos de pré-treino e ajuste fino do SciBERT utilizam as mesmas configurações, arquitetura e escolhas de hiperparâmetros que o BERT.



## 2 Métodos

Para alcançar o objetivo de melhorar a busca por informação em textos científicos, primeiro foi necessário encontrar algum estudo sobre o assunto. Assim que a referência foi analisada e escolhida, o próximo passo foi aplicar técnicas de inteligência artificial que não foram usadas no trabalho escolhido, na tentativa de melhorar os resultados do trabalho.

### 2.1 Encontrar referência

A busca por um estudo sobre NLP no meio científico foi feito no *Papers With Code*, que possui um grande acervo de artigos na área de inteligência artificial, além de conjuntos de dados, códigos e tabelas comparando resultados entre diferentes trabalhos. O trabalho encontrado foi usado como referência para as próximas etapas, ou seja, os resultados informados se tornaram as metas a serem superadas.

### 2.2 Melhorar resultados da referência

Após escolher um trabalho para usar como referência, o próximo passo foi encontrar possíveis formas de melhorar seus resultados. Durante essa etapa, foi necessário considerar os recursos disponíveis, como tempo e dinheiro, antes de escolher uma técnica como possível forma de tornar os resultados melhores que aqueles informados no trabalho de referência.

#### 2.2.1 Técnicas

Duas técnicas foram utilizadas na tentativa de melhorar os resultados do trabalho. Uma das técnicas foi a substituição do modelo usado originalmente no trabalho por outro modelo que apresente resultados melhores no meio científico. A outra técnica foi a aquisição de mais dados, para que os modelos sejam capazes de aprender melhor.

##### 2.2.1.1 Trocar modelo

Para que o objetivo possa ser alcançado, o trabalho escolhido como referência deverá usar algum modelo de ML ou DL. Ao trocar o modelo do trabalho por outro e treiná-lo nos mesmos dados, busca-se melhorar os resultados.

Essa hipótese foi pensada considerando que, na época da pesquisa, desenvolvimento e escrita do artigo usado como referência, o modelo com os melhores resultados disponível para realizar a tarefa de extração de informação em textos científicos tenha

sido usado, mas que um modelo com resultados melhores possa ter sido desenvolvido no intervalo entre a publicação da referência e o desenvolvimento deste trabalho.

#### 2.2.1.2 Incrementar amostragem

Considerando que modelos de aprendizagem profunda são capazes de conseguir resultados melhores baseado na quantidade de dado disponível para treino (quanto mais dado, melhor o resultado), uma possível forma de melhorar o resultado de um trabalho já realizado é adquirindo e rotulando mais dado.

O *Papers With Code* permite buscar por conjuntos de dados, inclusive textos científicos para serem usadas na tarefa de extração de informação. Para incrementar a amostragem foi escolhida um conjunto de dados e, em seguida, esse conjunto foi rotulado.

Rotular dados do meio científico requer domínio na área em questão e muito tempo para realizar a rotulação, o que não é possível considerando o tempo disponível e que não há pessoas especialistas na área da ciência disponíveis para realizar essa tarefa. Portanto, para rotular o dado foi usada a técnica de supervisão fraca para que essa tarefa fosse realizada rapidamente.

## 3 Resultados

Este capítulo mostra os resultados do trabalho após os métodos descritos no capítulo anterior serem seguidos, o que inclui a escolha do trabalho de referência, a escolha do modelo substituto e o incremento de amostragem, assim como os resultados de cada etapa.

### 3.1 Base de conhecimento de mecanismos

O artigo escolhido para ser a referência do trabalho se chama *Extracting a Knowledge Base of Mechanisms from COVID-19 Papers* (Extraíndo uma Base de Conhecimento de Mecanismos de documentos sobre COVID-19) (HOPE et al., 2021). Neste artigo, os pesquisadores falam do objetivo de criar uma base de conhecimento de mecanismos para auxiliar na busca por informação científica em artigos sobre COVID-19.

#### 3.1.1 O que foi realizado no trabalho

Ao analisar os artigos sobre a pandemia de COVID-19, é possível notar que realizar pesquisas sobre o assunto é desafiador devido à diversidade de informações e novos termos e conceitos (HOPE et al., 2021).

O grupo responsável pelo estudo tinha como objetivo unificar essas informações tão diversas e, para alcançar esse objetivo, criaram uma base de conhecimento de mecanismos, que chamaram de COMB, sigla para *COVID-19 Open Mechanism Knowledge Base* (Base de Conhecimento de Mecanismo Aberto de COVID-19), ao extrair informação da linguagem natural de textos científicos. Para isso, eles definiram o que é uma relação de mecanismos, criaram um conjunto de dados de mecanismos e treinaram um modelo para extrair essa informação e formar a base de conhecimento.

##### 3.1.1.1 Relação de mecanismos

A definição de mecanismo é motivo de debate na filosofia da biologia (RÖHL, 2012), com algumas definições sendo:

- Um sistema complexo que produz um comportamento pela interação de várias partes.
- Entidades e atividades organizadas de tal forma que produzam mudanças regulares.

- Uma estrutura que desempenha uma função em virtude de suas partes, operações e sua organização.

Existem muitos estudos realizados sobre extrair informações de documentos científicos, principalmente no campo da biomedicina. Essas informações normalmente correspondem a tipos específicos de mecanismo, que é um conceito fundamental no meio científico (HOPE et al., 2021). Dessa forma, após generalizar o conceito de mecanismo para que fosse possível cobrir diversos conceitos que aparecem em estudos relacionados ao COVID-19, cada mecanismo ficou definido no trabalho como a relação entre duas entidades, sendo que a relação pode ser do tipo direto ou indireto, como mostrado em 3.1.

$$\text{Mecanismo} = (E_1, E_2, \text{tipo}) \quad (3.1)$$

Cada entidade, nesse caso, é um *span* de texto, ou seja, é formado por uma parte do texto, possuindo começo e fim. O tipo das entidades não é importante para o estudo, já que o interesse é identificar corretamente o conteúdo de cada entidade (tudo entre o começo e o fim), se há alguma relação entre elas e, se houver, de qual tipo.

Mecanismos diretos são ações explicitamente realizadas por uma entidade, como atividades de natureza mecanicista. Descrições de um vírus se ligando a uma célula e uso de um remédio para tratamento são exemplos de mecanismos diretos. Um exemplo de mecanismo direto usado no artigo de referência pode ser visto em 3.2.

$$\underbrace{\text{Rapid diagnosis of respiratory virus infections}}_{E_1} \text{ contributes to } \underbrace{\text{patient care}}_{E_2} \quad (3.2)$$

Já mecanismos indiretos são influências ou associações sem informações mecanicistas explícitas ou menção de uma função. Enquanto mecanismos diretos informam os estados intermediários de como a pandemia (estado inicial) pode levar a impactos econômicos (estado final), por exemplo, mecanismos indiretos não informam esses estados, apenas mencionam a influência de um sobre o outro. Um exemplo de mecanismo indireto usado no artigo de referência pode ser visto em 3.3.

$$\underbrace{\text{Urinary tract infection}}_{E_1} \text{ is expressed as } \underbrace{\text{bladder infection}}_{E_2} \quad (3.3)$$

### 3.1.1.2 MECHANIC

O MECHANIC, sigla para *Mechanisms Annotated in COVID-19 papers* (Mecanismos anotados em documentos COVID-19), é um conjunto de dados rotulado manualmente

por especialistas em NLP, biomedicina e engenharia. Ele cobre diversas áreas da ciência, como medicina, biomedicina, virologia, microbiologia, farmácia, ciência da computação, engenharia, matemática, entre outros.

A equipe coletou sentenças retiradas de resumos de artigos científicos e rotulou as entidades, definindo o começo, fim e tipo de cada uma (todas as entidades possuem o mesmo tipo, *ENTITY*, já que o importante é definir o começo e o fim da entidade no texto) e rotulou as relações, definindo as duas entidades envolvidas e o tipo de cada uma, que pode ser *MECHANISM* (significando que é uma relação direta) ou *EFFECT* (significando que é uma relação indireta).

O MECHANIC, segundo o artigo, contém 1000 sentenças retiradas de 250 resumos de artigos, resultando em 2370 relações, sendo 1645 delas relações diretas e 725 relações indiretas. Entretanto, após analisar os dados disponibilizados pelos autores, verificou-se que o conjunto de dados possui 994 sentenças e 2370 relações, mas 1801 delas são relações diretas e 569 são relações indiretas. Um exemplo de amostra do conjunto de dados pode ser vista na Figura 4 e cada amostra possui os seguintes atributos:

- *doc\_key*: Identificador do documento que contém a sentença.
- *sentences*: Lista de sentenças (apesar de ser uma lista, contém apenas uma sentença). Cada sentença é uma lista de *tokens*.
- *relations*: Lista das relações existentes na sentença. Cada relação é uma lista com os seguintes valores:
  - Número que representa a posição do primeiro *token* da primeira entidade na sentença.
  - Número que representa a posição do último *token* da primeira entidade na sentença.
  - Número que representa a posição do primeiro *token* da segunda entidade na sentença.
  - Número que representa a posição do último *token* da segunda entidade na sentença.
  - Tipo da relação, podendo ser *MECHANISM* ou *EFFECT*.
- *ner*: Lista das entidades existentes na sentença. Cada entidade é uma lista com os seguintes valores:
  - Número que representa a posição do primeiro *token* da entidade na sentença.
  - Número que representa a posição do último *token* da entidade na sentença.

- Tipo da entidade, que sempre será *ENTITY*, já que o tipo da entidade não importa.

**Figura 4** – Exemplo de amostra do MECHANIC

```
{
  "doc_key": "i6nzh3vs_abstract_abstract",
  "sentences": [[
    "Rotavirus",
    "is",
    "a",
    "major",
    "cause",
    "of",
    "infantile",
    "viral",
    "gastroenteritis",
    "."
  ]],
  "relations": [
    [[0, 0, 6, 8, "EFFECT"]]
  ],
  "ner": [[
    [0, 0, "ENTITY"],
    [6, 8, "ENTITY"]
  ]]
}
```

Fonte: (HOPE et al., 2021)

### 3.1.1.3 COMB

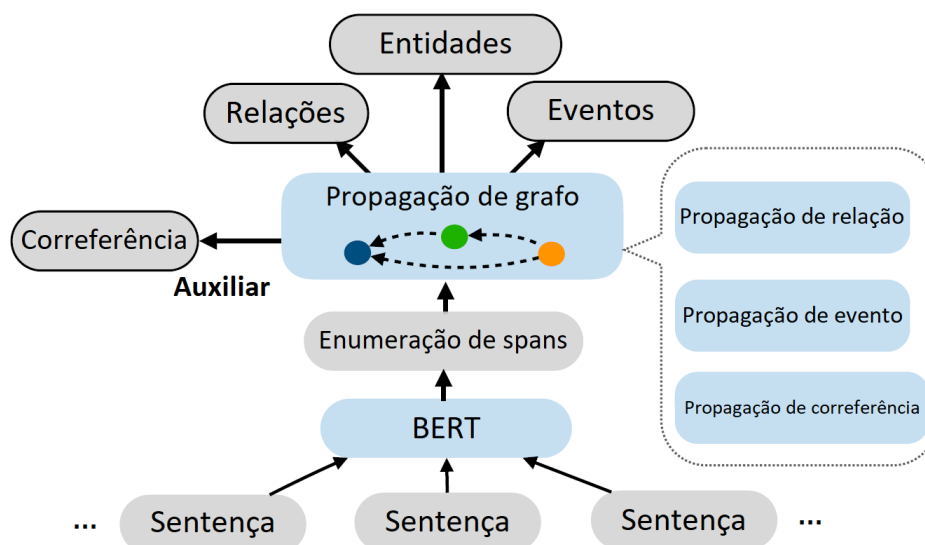
Após criar o MECHANIC, a equipe treinou um modelo de extração de entidades e relações chamado DyGIE++ para extrair relações de mecanismos de textos científicos. Em seguida, para criar o COMB, 160 mil resumos de artigos científicos foram passados para o modelo para que ele prevesse as entidades e relações de cada sentença. Após filtrar pelas previsões com alta pontuação de confiança ( $\geq 90\%$ ), a base de conhecimento consistiu em 1.5 milhão de relações no formato apresentado em 3.1.

### 3.1.2 DyGIE++

O DyGIE++ foi utilizado originalmente para criar o COMB ao ser treinado com o MECHANIC para extrair entidades e relações de mecanismos de artigos científicos. O DyGIE++, como explicado em (WADDEN et al., 2019), é um *framework* com o propósito de extrair 3 informações de textos: entidade, relação e evento.

## 3.1.2.1 Arquitetura

Figura 5 – Arquitetura do DyGIE++



Fonte: (WADDEN et al., 2019)

Como mostrado na Figura 5, a primeira etapa é a codificação dos *tokens*. Nessa etapa o modelo usa o BERT (no caso do MECHANIC, os desenvolvedores o substituíram pelo SciBERT) para a tokenização e codificação das sentenças. Cada sentença é passada como entrada junto com uma quantidade  $L$  de sentenças vizinhas.

Em seguida ocorre a enumeração de *spans*, ou seja, *spans* de texto são enumerados e construídos ao concatenar os *tokens* que representam seus extremos da esquerda e direita.

A próxima etapa é a de propagação de grafo de *spans*. Durante essa etapa, uma estrutura de grafo é gerada dinamicamente com base no melhor palpite do modelo no momento sobre as relações presentes entre os *spans* do documento. Cada representação de *span* é atualizado ao integrar representações de seus vizinhos no grafo de acordo com três variantes da propagação de grafo: propagação de relação, propagação de evento e propagação de correferência. Na propagação de relação, única utilizada no treino com o MECHANIC, os vizinhos são entidades relacionadas dentro de uma sentença. Dessa forma, o modelo aprende simultaneamente como identificar importantes conexões entre os *spans* e como compartilhar informação entre eles.

Durante a última etapa, que é classificação multitarefa, as representações recontextualizadas são entradas para funções de pontuação que fazem previsões para cada uma das tarefas finais (no caso do trabalho realizado, classificação de entidades e relações).

### 3.1.3 Resultados

A equipe responsável pela criação do MECHANIC e do COMB divulgou resultados para três métricas diferentes, sendo que para o desenvolvimento deste trabalho serão usadas duas delas:

- Detecção de entidade: Uma entidade será corretamente identificada se o começo e o fim da entidade forem corretamente identificados. Para essa métrica, a pontuação F1 foi de 50.2.
- Classificação de relação: Uma relação será corretamente classificada se as duas entidades que a compõe forem corretamente identificadas e se o tipo da relação (direto ou indireto) for corretamente identificado. Para essa métrica, a pontuação F1 foi de 42.8.

### 3.1.4 Problemas encontrados

Ao analisar o artigo, foi feita a tentativa de replicar os resultados alcançados ao seguir o que é descrito no trabalho. No entanto, não foi possível alcançar os resultados informados e, ao buscar as razões que levaram a esse problema, outros erros foram encontrados, como a falha na distribuição das amostras entre os conjunto de dados de treino, validação e teste e a falha em uma das configurações do modelo utilizado.

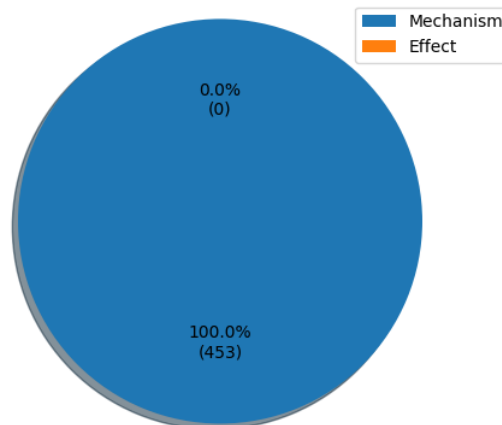
#### 3.1.4.1 Trabalho irreprodutível

Ao tentar reproduzir os resultados, os valores dos hiperparâmetros foram ajustados para ficarem iguais aos informados no artigo. Além disso, todas as bibliotecas de software necessárias, informadas no repositório de código que contém o trabalho realizado, foram instaladas, estando inclusive nas mesmas versões usadas pelos desenvolvedores do artigo. Entretanto, os resultados encontrados ao treinar o modelo com o conjunto de dados fornecido não foi semelhante aos descritos. Foram feitas duas tentativas de contato com os autores do artigo, porém nenhuma resposta foi enviada por eles até a data de conclusão deste trabalho.

#### 3.1.4.2 Conjunto de dados mal distribuído

Para que o resultado seja confiável, é necessário que a distribuição das classes entre os conjuntos de dados de treino, validação e teste seja feita respeitando a sua proporção. No entanto, apesar de existirem duas classificações para as relações de mecanismo (direta e indireta), o conjunto de dados de teste, usado para verificar os resultados, não possui nenhuma amostra de relação indireta, como mostrado na Figura 6.



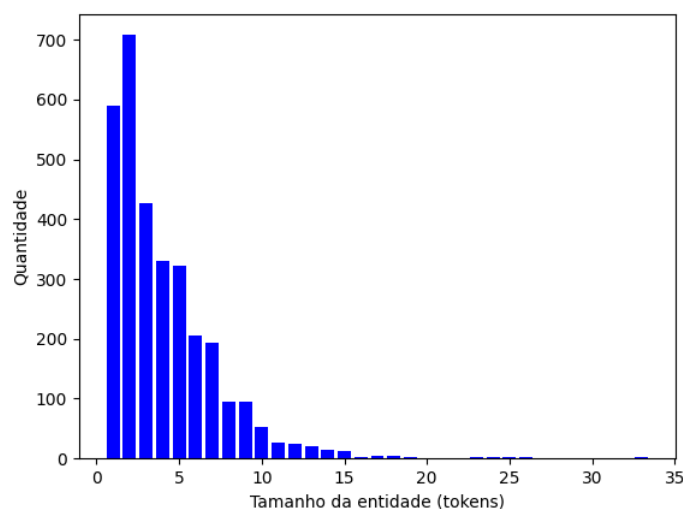
**Figura 6** – Distribuição das relações no conjunto de dados de teste do MECHANIC

Fonte: (HOPE et al., 2021)

Dessa forma, não é possível saber quão bem o modelo é capaz de classificar corretamente uma relação como indireta, já que existem apenas relações diretas no conjunto de dados usado para testar o modelo.

#### 3.1.4.3 Hiperparâmetro mal configurado

O DyGIE++ possui um hiperparâmetro para configurar o tamanho máximo das entidades, no qual foi configurado com o valor 12 pela equipe que desenvolveu o artigo. Entretanto, como é possível ver pela Figura 7, existem entidades com tamanhos maiores do que 12. Caso um modelo com essa configuração tente classificar uma sentença que possua uma entidade com mais de 12 *tokens* (como existe nos conjunto de dados de validação e teste), o modelo nunca irá detectar a entidade corretamente.

**Figura 7** – Tamanho das entidades no conjunto de dados de treino do MECHANIC

Fonte: (HOPE et al., 2021)

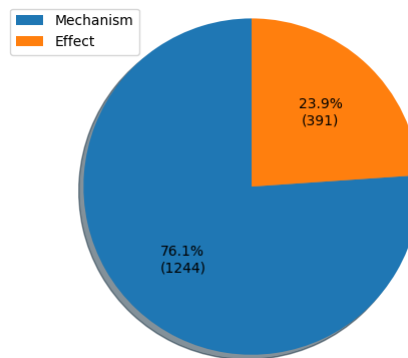
### 3.1.5 Soluções

Antes de tentar melhorar os resultados do artigo, foi necessário corrigir esses erros, o que inclui redistribuir as amostras entre os arquivos de treino, validação e teste de maneira que as proporções sejam respeitadas e ajustar as configurações dos hiperparâmetros para refletirem o que foi encontrado ao analisar o dado.

#### 3.1.5.1 Conjunto de dados redistribuído

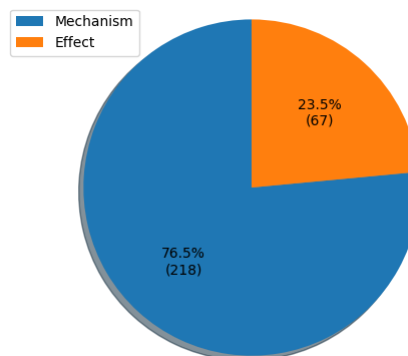
Para resolver o problema da distribuição que não respeitava as proporções das classes das relações, o conjunto de dados inteiro foi redistribuído para que ficasse com as proporções adequadas, como mostram as Figuras 8, 9, 10.

**Figura 8** – Nova distribuição das relações no conjunto de dados de treino do MECHANIC

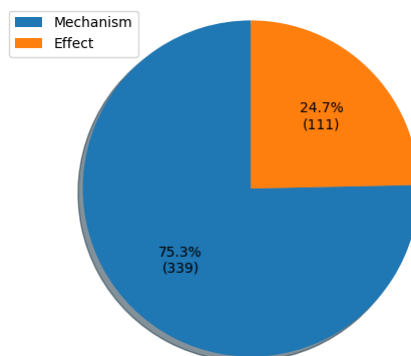


Fonte: Compilação do autor

**Figura 9** – Nova distribuição das relações no conjunto de dados de validação do MECHANIC



Fonte: Compilação do autor

**Figura 10** – Nova distribuição das relações no conjunto de dados de teste do MECHANIC

Fonte: Compilação do autor

### 3.1.5.2 Ajuste nos hiperparâmetros

Em seguida, os hiperparâmetros foram analisados em busca de outros erros que poderiam afetar os resultados, tornando-os não confiáveis. Entretanto, o único erro encontrado foi no hiperparâmetro mencionado anteriormente (tamanho máximo de uma entidade), que foi ajustado para, de fato, ser o tamanho máximo, considerando os dados disponíveis.

### 3.1.5.3 Novos resultados

Ao realizar as mudanças descritas acima, foi executado novamente o processo de treino, validação e teste do modelo, dessa vez com os dados redistribuídos e com as configurações ajustadas. As métricas que se tornarão referência e que deverão ser melhoradas pelas técnicas serão a pontuação F1 da detecção de entidade e a micro-média pontuação F1 da classificação de relação (a micro-média foi escolhida pois há um desequilíbrio entre as classes, já que a quantidade de relações diretas é o triplo das indiretas).

Os melhores resultados estão informados na Tabela 1, que está em ordem decrescente com base na pontuação F1 da classificação de relação, pois essa métrica ficou muito abaixo se comparada aos valores informados no artigo. Dessa forma, ao pegar o melhor resultado para essa métrica, a diferença entre os dois valores será o menor possível.

Os hiperparâmetros que foram ajustados na otimização, seus intervalos mínimos e máximos (escolhidos aleatoriamente a cada treino) e os valores que levaram ao melhor resultado estão informados na Tabela 2.

Ao invés de considerar os resultados do artigo como sendo a referência para as próximas etapas, foi considerado o melhor resultado obtido após realizar a busca pelos melhores hiperparâmetros. Os valores usados como referência serão os melhores valores informados na Tabela 1 ou seja, 50.19 para a métrica de detecção de entidade e 24.28 para a métrica de classificação de relação.

**Tabela 1** – Melhores resultados no DyGIE++

Posição	Detecção de entidade			Classificação de relação		
	Precisão	Revocação	F1	Precisão	Revocação	F1
1	53.11	47.57	50.19	32.16	19.50	24.28
2	55.80	49.02	52.19	28.50	20.92	24.13
3	53.01	49.02	50.94	25.00	23.04	23.98
4	51.82	44.90	48.11	25.92	19.85	22.48
5	52.10	45.14	48.37	31.41	17.37	22.37
6	53.35	48.30	50.70	23.98	20.92	22.34
7	47.85	51.45	49.59	25.22	19.50	22.00
8	54.41	44.90	49.20	20.63	23.04	21.77
9	55.48	44.17	49.18	27.62	17.73	21.59
10	53.47	42.96	47.64	30.32	16.66	21.51

Fonte: Compilação do autor

**Tabela 2** – Hiperparâmetros do DyGIE++ ajustados e seus intervalos

Hiperparâmetro	Mínimo	Máximo	Melhor
Camadas	1	10	2
Dropout	0.1	0.4	0.18
Hidden size	100	300	268
Learning rate	5e-4	5e-3	3.4e-3
Loss weight (Relação)	1	10	3
Weight decay	0.01	0.1	0.05

Fonte: Compilação do autor

## 3.2 Novo modelo - SpERT

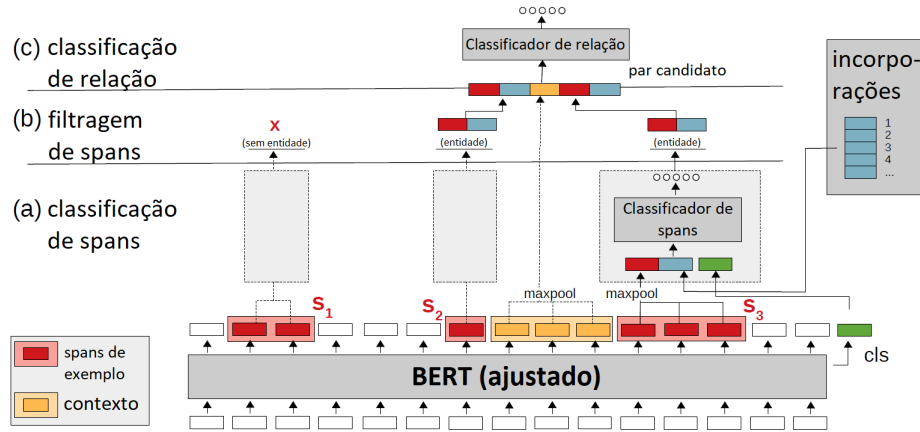
Após corrigir os erros e obter novos resultados, o próximo passo foi melhorá-los. A primeira técnica que foi utilizada foi a substituição do modelo usado originalmente, o DyGIE++, por um novo modelo, capaz de conseguir resultados melhores. O modelo escolhido para ser o substituto foi o SpERT.

O SpERT, segundo o artigo (EBERTS, 2020), é um modelo de atenção para extrair entidades e relações. Segundo a equipe que desenvolveu o modelo, o uso de amostras negativas, filtragem de spans e o uso de contexto localizado são as características mais importantes do modelo e que são responsáveis pelos seus bons resultados.

### 3.2.1 Arquitetura

Como mostrado na Figura 11, o SpERT utiliza o BERT (ou SciBERT, caso os dados sejam do meio científico) em sua arquitetura. O BERT recebe uma sentença *tokenizada* de tamanho  $n$  como entrada e retorna uma sequência incorporada de tamanho  $n + 1$  (o último *token*, em verde, é o *token* de classificação, que representa a frase geral ou

Figura 11 – Arquitetura do SpERT



Fonte: (EBERTS, 2020)

contexto)

A próxima etapa é a de classificação de *spans*. Um *span* candidato, no formato demonstrado em 3.4, é selecionado. Os elementos do *span* são unidos usando uma função de *max-pooling*  $f$ .

$$s = (e_i, e_{i+1}, \dots, e_{i+k}) \quad (3.4)$$

Em seguida, uma incorporação  $\mathbf{w}_{k+1}$  (em azul), baseada no tamanho do *span* (retirada de uma matriz dedicada, que contém uma incorporação para cada tamanho de *span*), é concatenada ao *span*, formando a representação do *span*, mostrada em 3.5 (o símbolo  $\circ$  significa concatenação).

$$R(s) = f(e_i, e_{i+1}, \dots, e_{i+k}) \circ \mathbf{w}_{k+1} \quad (3.5)$$

Finalmente, é concatenado o *token* de classificação  $\mathbf{cls}$ . O contexto é uma fonte importante de desambiguação, pois as palavras-chave são indicadores fortes para classes de entidades. A entrada resultante está representada em 3.6. Essa entrada é passada para o classificador de *spans*, que mapeia o *span* candidato em uma classe, podendo a classe ser *none* (nesse caso, o *span* não é uma entidade).

$$x^s = R(s) \circ \mathbf{cls} \quad (3.6)$$

A próxima etapa consiste na filtragem de *spans*. A saída do classificador de *spans* estima se um *span* é uma entidade ou não e, se for, a qual classe pertence (a classe

escolhida é aquela com a pontuação mais alta na estimativa). Os *spans* com classe *none* são removidos.

A última etapa é a de classificação de relação. Nessa etapa o classificador de relação processa cada par de entidades (cada entidade é formado pelo *span*, em vermelho, e pela incorporação baseado no tamanho do *span*, em azul) e estima se há alguma relação entre elas. É utilizado um contexto mais localizado, coletado das proximidades das entidades (entre o final da primeira entidade e o começo da segunda entidade, em amarelo), ao invés do *token* de classificação mencionado anteriormente. A entrada do classificador de relação é a concatenação da primeira entidade, do contexto localizado  $c$  e da segunda entidade, como representado em 3.7. Já que relações podem ser assimétricas, a ordem inversa também é verificada.

$$\begin{aligned} x^r_1 &= R(s_1) \circ c(s_1, s_2) \circ R(s_2) \\ x^r_2 &= R(s_2) \circ c(s_1, s_2) \circ R(s_1) \end{aligned} \tag{3.7}$$

Durante o treino, é passado como entrada do modelo não apenas os dados do conjunto de dados contendo as entidades e as relações classificadas corretamente, mas também é passado amostragens negativas, feito da seguinte forma:

- Entidades: Para o classificador de *span*, além da amostragem positiva (todas as entidades classificadas), é utilizado um número fixo de *spans* aleatórios de “não entidades”, ou seja, se um *span* não tiver sido classificado como entidade, ele pode fazer parte do conjunto de amostragem negativa.
- Relações: Para o classificador de relações, além da amostragem positiva (todas as relações classificadas), é utilizado um número fixo de relações entre duas entidades que não possuem relação entre si, ou seja, se duas entidades da amostragem positiva não possuem relação entre si, essa relação negativa pode fazer parte do conjunto de amostragem negativa. Os desenvolvedores consideram vital selecionar entidades positivas para fazer a amostragem de relações negativas (por isso o nome amostragens negativas fortes), ao invés de selecionar *spans* aleatoriamente.

### 3.2.2 Resultados com dados científicos

Tanto o DyGIE++ quanto o SpERT foram avaliados no SciERC, que inclui entidades científicas e as relações entre elas, retirados de 500 resumos de artigos científicos (LUAN et al., 2018).

Como é possível ver na Tabela 3, o modelo SpERT alcançou resultados superiores ao DyGIE++ tanto na tarefa de extrair entidades quanto na de extrair relações de textos científicos.

**Tabela 3** – Pontuação F1 do DyGIE++ e SpERT com o SciERC

Modelo	Extração de entidade	Extração de relação
DyGIE++	67.50	48.40
SpERT	70.33	50.84

Fonte: (EBERTS, 2020)

Considerando que o SpERT conseguiu resultados superiores no conjunto de dados *SciERC*, foi levantada a hipótese de que o modelo seria capaz de conseguir resultados melhores também no conjunto de dados MECHANIC se comparado ao DyGIE++.

### 3.2.3 Treino do SpERT com o MECHANIC

Após analisar e escolher o SpERT como modelo substituto, o próximo passo é treiná-lo com o conjunto de dados MECHANIC. Antes, entretanto, é necessário adaptar o conjunto de dados para que possa ser lido pelo modelo.

#### 3.2.3.1 Adaptação do conjunto de dados

O MECHANIC possui o formato necessário para ser lido pelo DyGIE++ e precisou ser adaptado para que pudesse ser lido pelo SpERT. Foi criado um *script* na linguagem *Javascript* que leu cada amostra do conjunto de dados e adaptou as entidades e relações para que pudessem ser lidas pelo modelo. Um exemplo da amostra do conjunto de dados adaptado pode ser vista na Figura 12 e cada uma possui os seguintes atributos:

**Figura 12** – Exemplo de amostra do MECHANIC após adaptá-lo para o SpERT

```
{
  "tokens": [
    "Rotavirus",
    "is",
    "a",
    "major",
    "cause",
    "of",
    "infantile",
    "viral",
    "gastroenteritis",
    "."
  ],
  "entities": [
    { "type": "Entity", "start": 0, "end": 1 },
    { "type": "Entity", "start": 6, "end": 9 }
  ],
  "relations": [
    { "type": "Effect", "head": 0, "tail": 1 }
  ],
  "orig_id": "i6nzh3vs_abstract_abstract"
}
```

Fonte: Compilação do autor

- *tokens*: Lista com os *tokens* da sentença.
- *entities*: Lista com as entidades da sentença. Cada entidade é um objeto com os seguintes parâmetros:
  - *type*: Tipo da entidade, que sempre será *Entity*.
  - *start*: Número que representa a posição do primeiro *token* da entidade na sentença
  - *end*: Número que representa a posição do primeiro *token* que não faz parte da entidade
- *relations*: Lista com as relações da sentença. Cada relação é um objeto com os seguintes parâmetros:
  - *type*: Tipo da relação, que pode ser *Mechanism* ou *Effect*.
  - *head*: Número que representa a posição da primeira entidade na lista de entidades
  - *tail*: Número que representa a posição da segunda entidade na lista de entidades
- *orig\_id*: Identificador do documento que possui a sentença.

### 3.2.3.2 Resultados

Os melhores resultados estão informados na Tabela 4, que, novamente, está em ordem decrescente com base na pontuação F1 da classificação de relação, enquanto que os hiperparâmetros que foram ajustados na otimização, seus intervalos mínimos e máximos (escolhidos aleatoriamente a cada treino) e os valores que levaram ao melhor resultado estão informados na Tabela 5.

**Tabela 4** – Melhores resultados no SpERT

Posição	Detecção de entidade			Classificação de relação		
	Precisão	Revocação	F1	Precisão	Revocação	F1
1	62.99	53.63	57.94	31.90	22.45	26.35
2	57.23	58.15	57.69	25.68	25.92	25.80
3	55.87	59.12	57.45	23.67	28.00	25.66
4	55.77	59.28	57.47	23.80	26.62	25.13
5	58.98	56.21	57.56	26.47	23.84	25.09
6	59.25	56.38	57.78	25.92	24.30	25.08
7	59.61	55.08	57.26	26.41	23.84	25.06
8	59.34	55.89	57.57	26.64	23.37	24.90
9	57.62	54.92	56.24	26.50	23.37	24.84
10	55.30	58.96	57.07	23.18	25.92	24.48

Fonte: Compilação do autor



**Tabela 5** – Hiperparâmetros do SpERT ajustados e seus intervalos

Hiperparâmetro	Mínimo	Máximo	Melhor
Camadas (entidade)	1	20	6
Camadas (relação)	1	20	2
Dropout	0.1	0.5	0.14
Learning rate	1e-6	1e-4	6.7e-5
Nº Entidades Negativas	10	300	32
Nº Relações Negativas	10	300	189
Weight decay	1e-4	0.1	1e-2

Fonte: Compilação do autor

Os resultados atingidos pelo SpERT foram superiores tanto na tarefa de detecção de entidade (57.94 do SpERT contra 50.19 do DyGIE++) quanto na de classificação de relação (26.35 do SpERT contra 24.28 do DyGIE++), portanto, o objetivo de melhorar os resultados foi alcançado.

### 3.3 Incremento de amostragem

Como dito anteriormente, o conjunto de dados MECHANIC possui 673 sentenças no arquivo de treino, 130 sentenças no arquivo de validação e 191 sentenças no arquivo de teste. Ao todo são 994 sentenças disponíveis. Ao longo do trabalho realizado, levantou-se a hipótese de que a quantidade de dados disponível era muito pequena e que, se houvessem mais sentenças para treinar os modelos, os resultados seriam melhores.

Para verificar essa hipótese, o primeiro passo foi adquirir mais dados (ou seja, sentenças do meio científico) e, em seguida, classificá-los e então treinar um modelo no novo dado. O processo de classificação e treino foi feito duas vezes, na primeira vez o SpERT foi usado para classificar e o DyGIE++ foi treinado no novo dado, em seguida o processo foi feito novamente, mas invertendo os papéis dos modelos.

#### 3.3.1 Origem do dado

Para adquirir mais sentenças, foi usado a base de dados do CORD19, que na época do experimento possuía 260 mil artigos científicos sobre a COVID-19. Foram coletadas 1.8 milhão de sentenças a partir dos resumos dos artigos, da mesma forma que foi feito na criação do COMB.

#### 3.3.2 Classificando sentenças com SpERT

Para a etapa de classificação, foi usado o modelo SpERT com as melhores configurações encontradas (as mesmas informadas na Tabela 5) para que a classificação fosse

a mais correta possível. O modelo foi adaptado para que a saída retornasse as seguintes informações:

- *tokens*: Lista com os *tokens* da sentença.
- *entities*: Lista com as entidades previstas pelo modelo. Cada entidade é um objeto contendo:
  - *type*: Tipo da entidade previsto pelo modelo (sempre será *Entity*)
  - *start*: Número que representa a previsão do modelo quanto a posição do primeiro *token* da entidade na sentença
  - *end*: Número que representa a previsão do modelo quanto a posição do primeiro *token* que não faz parte da entidade
  - *score*: Número entre 0 e 100 que representa a confiança do modelo. Quanto maior o valor, maior a certeza que o modelo tem que ele foi capaz de retornar o tipo, começo e fim da entidade corretamente.
- *relations*: Lista com as relações previstas pelo modelo. Cada relação é um objeto contendo:
  - *type*: Tipo da relação previsto pelo modelo
  - *head*: Número que representa a previsão do modelo quanto a posição da primeira entidade na lista de entidades
  - *tail*: Número que representa a previsão do modelo quanto a posição da segunda entidade na lista de entidades
  - *score*: Número entre 0 e 100 que representa a confiança do modelo. Quanto maior o valor, maior a certeza que o modelo tem que ele foi capaz de retornar o tipo da relação e as entidades que a compõe corretamente.

### 3.3.2.1 Tratamento do dado

As sentenças classificadas foram filtradas com base na confiança que o modelo tinha de ter classificado corretamente (o valor escolhido foi acima de 99% de confiança). Ao todo, 15000 sentenças formaram o conjunto de dados de treino, o que resultou em 38108 entidades e 19054 relações. Os conjunto de dados de validação e teste são os mesmos usados anteriormente. Antes de realizar o treino, no entanto, foi necessário adaptar o conjunto de dados para que pudesse ser lido pelo DyGIE++.

**Tabela 6** – Melhores resultados com mais dados no DyGIE++

Posição	Detecção de entidade			Classificação de relação		
	Precisão	Revocação	F1	Precisão	Revocação	F1
1	57.20	32.76	41.66	38.23	13.82	20.31
2	54.73	37.86	44.76	31.25	14.18	19.51
3	54.71	35.19	42.83	34.73	11.70	17.50
4	56.37	35.43	43.51	21.16	14.18	16.98
5	59.09	28.39	38.36	42.64	10.28	16.57

Fonte: Compilação do autor

### 3.3.2.2 Resultado

Após treinar o modelo DyGIE++ com os dados classificados pelo SpERT e validá-lo com o mesmo conjunto de dados de teste usado antes, os resultados foram analisados e concluiu-se que não foram capazes de superar os resultados SpERT treinado com o conjunto de dados original. Os resultados atingidos estão descritos na Tabela 6.

### 3.3.3 Classificando sentenças com DyGIE++

Após usar o DyGIE++ para treinar nos novos dados, ele foi usado para classificá-los. Da mesma forma que antes, foi usado o modelo com as melhores configurações encontradas (as mesmas informadas na Tabela 2) para que a classificação fosse a mais correta possível. A saída do modelo retorna informações como:

- *sentences*: Lista com os *tokens* da sentença
- *predicted\_ner*: Lista com as entidades previstas pelo modelo. Cada entidade é uma lista contendo:
  - Número que representa a previsão do modelo quanto a posição do primeiro *token* da entidade na sentença.
  - Número que representa a previsão do modelo quanto a posição do último *token* da entidade na sentença.
  - Tipo da entidade previsto pelo modelo, que sempre será *ENTITY*.
  - Número entre 0 e 1 que representa a confiança do modelo. Quanto mais próximo de 1, maior a certeza que o modelo tem que ele foi capaz de retornar o começo, fim e tipo da entidade corretamente.
- *predicted\_relations*: Lista com as relações previstas pelo modelo. Cada relação é uma lista contendo:
  - Número que representa a previsão do modelo quanto a posição do primeiro *token* da primeira entidade na sentença.

- Número que representa a previsão do modelo quanto a posição do último *token* da primeira entidade na sentença.
- Número que representa a previsão do modelo quanto a posição do primeiro *token* da segunda entidade na sentença.
- Número que representa a previsão do modelo quanto a posição do último *token* da segunda entidade na sentença.
- Tipo da relação previsto pelo modelo
- Número entre 0 e 1 que representa a confiança do modelo. Quanto mais próximo de 1, maior a certeza que o modelo tem que ele foi capaz de retornar o tipo da relação e as entidades que a compõe corretamente.

### 3.3.3.1 Tratamento do dado

As sentenças classificadas foram filtradas usando os mesmos critérios (99% de confiança do modelo), o que resultou em um conjunto de dados de treino com 15000 sentenças, 54579 entidades e 25480 relações. Os conjunto de dados de validação e teste também são os mesmos usados anteriormente. Antes de realizar o treino, no entanto, foi necessário adaptar o conjunto de dados para que pudesse ser lido pelo SpERT.

### 3.3.3.2 Resultado

**Tabela 7** – Melhores resultados com mais dados no SpERT

Posição	Detecção de entidade			Classificação de relação		
	Precisão	Revocação	F1	Precisão	Revocação	F1
1	44.61	61.55	51.73	22.14	21.99	22.06
2	47.30	59.61	52.75	25.87	18.75	21.74
3	40.14	63.48	49.18	21.31	21.06	21.18
4	44.56	59.61	51.00	22.25	19.21	20.62
5	47.04	55.25	50.81	25.44	16.66	20.13

Fonte: Compilação do autor

Após treinar o modelo SpERT com os dados classificados pelo DyGIE++ e validá-lo com o mesmo conjunto de dados de teste usado antes, os resultados foram analisados e concluiu-se que, novamente, não foram capazes de superar os resultados SpERT treinado com o conjunto de dados original. Os resultados atingidos estão descritos na Tabela 7.

## 4 Conclusão

Artigos científicos são escritos e informação é compartilhada com uma velocidade cada vez maior e encontrar a informação que deseja se torna cada vez mais desafiador para os pesquisadores. Ao usar inteligência artificial para buscar pelos assuntos desejados não pela suas menções no texto, mas pela relação entre os termos, é possível encontrar o que procura com mais precisão e, por consequência, diminuindo o tempo de busca.

Pelo fato de não ter sido possível reproduzir os resultados do artigo de referência e considerando que alguns erros foram encontrados no trabalho, foi necessário primeiramente arrumar esses erros e, em seguida, obter novos resultados para usá-los como referência para as próximas etapas ao invés de considerar os informados no artigo.

Como demonstrado anteriormente no trabalho, trocar o modelo utilizado originalmente por outro modelo com resultados melhores no meio científico melhorou os resultados se comparados com a referência, com um aumento considerável. Entretanto, rotular sentenças para aumentar a quantidade de dado disponível para treinar os modelos não foi capaz de melhorar os resultados.

A equipe que desenvolveu o COMB, a base de conhecimento de mecanismos, demonstrou como eles o utilizaram para melhorar as buscas por informação ao fazer com que as buscas retornem resultados mais relevantes. Considerando que o objetivo de melhorar os resultados do trabalho original foram alcançados, as buscas podem se tornar ainda mais corretas e relevantes.

O trabalho foi desenvolvido com o objetivo de melhorar a busca por informação em textos científicos. Considerando os resultados informados pelo artigo de referência como os valores que precisavam ser melhorados para tornar a busca por informação ainda mais relevantes, alterar o modelo usado, o *DyGIE++*, por outro modelo, o *SpERT*, mostrou ser uma estratégia bem-sucedida, já que os resultados foram de fato melhorados. O objetivo foi, portanto, alcançado, já que as pesquisas podem retornar resultados ainda mais corretos.

### 4.1 Trabalhos futuros

Uma possível forma de melhorar ainda mais o resultado seria rotular sentenças utilizando alguma função de rotulação (como uma inteligência artificial) e pedir para especialistas em biomedicina e NLP validarem se as sentenças foram rotuladas corretamente, ou então pedir para especialistas realizarem a rotulação manualmente. Dessa forma, com mais dados, os modelos poderão ser capazes de classificar melhor as entidades

e as relações.

Outra possível forma de melhorar o resultado é trocar o modelo *SpERT* por outro que realiza a mesma tarefa (extrair entidades e relações), mas que possui resultados melhores. Novos modelos de inteligência artificial são desenvolvidos constantemente e atingem resultados cada vez melhores. Da mesma forma que substituir o *DyGIE++* pelo *SpERT* melhorou os resultados, substituir novamente por um modelo mais novo pode melhorar ainda mais.

# Referências

- ARSOV, N.; MIRCEVA, G. Network embedding: An overview. *arXiv preprint arXiv:1911.11726*, 2019. Citado na página 20.
- BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. Citado na página 21.
- BELTAGY, I.; LO, K.; COHAN, A. Scibert: Pretrained language model for scientific text. In: *EMNLP*. [S.l.: s.n.], 2019. Citado na página 23.
- CHOPRA, A.; PRASHAR, A.; SAIN, C. Natural language processing. *International journal of technology enhancements and emerging engineering research*, Citeseer, v. 1, n. 4, p. 131–134, 2013. Citado na página 20.
- DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Citado na página 22.
- DONGARE, A. et al. Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, Citeseer, v. 2, n. 1, p. 189–194, 2012. Citado na página 16.
- EBERTS, A. U. M. Span-based joint entity and relation extraction with transformer pre-training. 2020. Citado 3 vezes nas páginas 35, 36 e 38.
- GALASSI, A.; LIPPI, M.; TORRONI, P. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, v. 32, n. 10, p. 4291–4308, 2020. Citado na página 21.
- GIROLAMO, N. D.; REYNDERS, R. M. Characteristics of scientific articles on covid-19 published during the initial 3 months of the pandemic. *Scientometrics*, Springer, v. 125, n. 1, p. 795–812, 2020. Disponível em: <<https://doi.org/10.1007/s11192-020-03632-0>>. Citado na página 13.
- HASTIE, T. et al. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: Springer, 2009. v. 2. Citado na página 16.
- HOPE, T. et al. Extracting a Knowledge Base of Mechanisms from COVID-19 Papers . In: . [S.l.: s.n.], 2021. Citado 4 vezes nas páginas 26, 27, 29 e 32.
- JANIESCH, C.; ZSCHECH, P.; HEINRICH, K. Machine learning and deep learning. *Electronic Markets*, Springer, v. 31, n. 3, p. 685–695, 2021. Citado na página 15.
- JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, American Association for the Advancement of Science, v. 349, n. 6245, p. 255–260, 2015. Citado na página 15.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Disponível em: <<https://doi.org/10.1038/nature14539>>. Citado na página 15.

- LUAN, Y. et al. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *arXiv preprint arXiv:1808.09602*, 2018. Citado na página 37.
- MOONEY, R. J.; BUNESCU, R. Mining knowledge from text using information extraction. *ACM SIGKDD explorations newsletter*, ACM New York, NY, USA, v. 7, n. 1, p. 3–10, 2005. Citado na página 20.
- POWERS, D. M. W. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020. Disponível em: <<https://doi.org/10.48550/arXiv.2010.16061>>. Citado na página 18.
- RICCABONI, M.; VERGINER, L. The impact of the covid-19 pandemic on scientific research in the life sciences. *Plos one*, Public Library of Science San Francisco, CA USA, v. 17, n. 2, p. e0263001, 2022. Disponível em: <<https://doi.org/10.1371/journal.pone.0263001>>. Citado na página 13.
- RÖHL, J. Mechanisms in biomedical ontology. In: SPRINGER. *Journal of Biomedical Semantics*. [S.l.], 2012. v. 3, n. 2, p. 1–14. Citado na página 26.
- TAKAHASHI, K. et al. Confidence interval for micro-averaged f1 and macro-averaged f1 scores. *Applied Intelligence*, Springer, v. 52, n. 5, p. 4961–4972, 2022. Disponível em: <<https://doi.org/10.1007/s10489-021-02635-5>>. Citado 2 vezes nas páginas 17 e 19.
- VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017. Citado na página 21.
- WADDEN, D. et al. Entity, relation, and event extraction with contextualized span representations. 2019. Citado 2 vezes nas páginas 29 e 30.
- WANG, Y. et al. A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics*, Elsevier, v. 87, p. 12–20, 2018. Citado na página 20.
- WEBSTER, J. J.; KIT, C. Tokenization as the initial phase in nlp. In: *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*. [S.l.: s.n.], 1992. Citado na página 20.
- YANG, L.; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, Elsevier, v. 415, p. 295–316, 2020. Disponível em: <<https://doi.org/10.1016/j.neucom.2020.07.061>>. Citado na página 16.
- ZHOU, Z.-H. A brief introduction to weakly supervised learning. *National science review*, Oxford University Press, v. 5, n. 1, p. 44–53, 2018. Disponível em: <<https://doi.org/10.1093/nsr/nwx106>>. Citado 2 vezes nas páginas 16 e 17.