Esse artigo faz parte de uma série de estudos onde tentaremos construir um NER (Named Entity Recognition) para identificação de entidades nas Políticas Operacionais (PO) do BNDES.

O BNDES promove rotineiramente ajustes nas suas Politicas Operacionais, cujo objetivo principal é nortear as regras aplicadas aos seus financiamentos.

A mudança da política operacional impacta equipes de negócio, sistemas e seus respectivos gestores, gerências de testes e homologadores. A quantidade de homem-hora (hh) gasta nessas mudanças é enorme e a maior parte do trabalho realizado é manual, sujeita a diversos tipos de falha e recebe rotineiramente apontamentos da área de auditoria do BNDES.

O projeto PO-NER (Políticas Operacionais Named Entity Reconigtion) tem como objetivo a construção de um ou mais modelos de inteligência artificial aplicados a NLP (Natural Language Processing) para realizar os seguintes objetivos:

* Reconhecimento de entidades nas Politicas Operacionais do BNDES (NER).
* Identificação de diferentes entidades cujo formato é muito similar, por exemplo: Um valor percentual pode estar relacionado a um nível de participação ou uma taxa de remuneração, ambos são valores percentuais, mas representam diferentes entidades. Esse tipo desambiguação é conhecido como NED (Named Entity Disambiguation) e deverá ser tratado pelo modelo.
* Conectar as entidades reconhecidas (NEL).
* Criação de uma base de conhecimento (Knowledge Base) para que outros modelos possam realizar tarefas sobre o conhecimento extraído.

Caso o(s) modelo(s) construídos alcancem um nível de acurácia satisfatório então será possível utilizá-lo para apoiar os gestores e técnicos envolvidos nas mudanças das PO’s do BNDES.

Como estudante do curso de Pós Graduação BI-Master da PUC-RJ e orientado pelo professor Leonardo Mendoza mergulharei na jornada de aprender sobre reconhecimento de entidades e conexão de entidades, compartilhando a cada fase do projeto o conhecimento adquirido através de artigos técnicos. Agradeço ao Mestre Leonardo pela ajuda e apoio no aprendizado necessário para iniciar o projeto.

As fases de o projeto planejadas (cada fase é finalizada com um artigo) são as seguintes:

1. NER Básico – Aplicação de Modelos BERT para NER. Estudo exploratório para compreensão do uso de modelos BERT em NER.
   1. Utilização de modelos em inglês – maior disponibilidade de datasets e modelos para teste.
   2. Preparação de tokens.
   3. Tratamento dos labels - alinhamento de *wordpiece*, importação de labels por *span* e *token/tag.*
   4. Estudo das métricas para avaliação do modelo.
2. NER Português – Aplicação de Modelos BERT para NER em Português. Estudo exploratório para compreensão do uso de modelos BERT em NER, focado em datasets e modelos da língua portuguesa.
3. Anotações de Politicas Operacionais – Criação de algoritmos para anotar os dados existentes para construção de um dataset anotado que servirá para treinar, validar e testar o modelo construído.
4. PO-NER BERT – Construção de um modelo BERT para Extração de Entidades em Politicas Operacionais no BNDES.
5. PO-NEL – Construção de um modelo para Conectar Entidades em Politicas Operacionais no BNDES
6. KB PO – Construção da Base de Conhecimento de Politicas Operacionais.

Não sabemos até onde o projeto chegará, mas isso não importa, o caminho e seu aprendizado é o maior objetivo, se chegarmos ao cume da montanha será ótimo, mas isso não guiará o sucesso ou fracasso do trabalho realizado.

Sem mais delongas, após contextualizar nosso trabalho e objetivo, iniciemos o primeiro artigo!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Antes de treinarmos o modelo precisaremos fazer a preparação dos tokens. O treinamento dos modelo BERT gera um vetor de *embeddings* que definirá o sentido de cada palavra conhecida e seu relacionamento com outras palavras. Para que uma sentença (conjunto de tokens) entre em um modelo BERT cada token precisa ser transformado em seu respectivo *tensor (vetor)* de embedding. A dimensão desse *embedding* vai depender do tipo de modelo BERT, quanto maior a dimensão, mais complexo o modelo, maior a memória GPU utilizada e mais informações de contexto.

O responsável por transformar um token (palavra) em seu respectivo *embedding* é o *Tokenizer*, sendo que cada modelo BERT possui o seu respectivo *Tokenizer*.

Um problema que será tratado nesse artigo é o alinhamento entre os *labels* que são informados por *token* ou por *span* (inicio e fim de uma entidade) e sua representação nos *tokens* gerados pelo *Tokenizer BERT*... Mas como assim??

O modelo BERT não contém todas as palavras de uma língua, é um subconjunto, geralmente 30.000 (aproximadamente), por isso o tokenizador gera *wordpieces* quando não encontra um token em seu vocabulário. Vamos dar um exemplo.