Verificação de Factualidade

João Carlos Cerqueira
(jc.cerqueira13@gmail.com)

Novembro 2022

Abstract

O objetivo do projeto é implementar um sistema capaz de automatizar a verificação da veracidade de afirmações. Mais especificamente, o sistema será composto de múltiplos modelos de aprendizado de máquina, onde cada um deles será responsável por uma sub-tarefa dentro do fluxo de verificação de veracidade. A tarefa não se limita em classificar as afirmações em verdadeiro ou falso, mas também fornecer evidências na forma de citações de sentenças de um corpus de literatura científica. O sistema será treinado e avaliado no contexto específico de medicina e biomedicina, fazendo uso da base de dados SciFact [1].

1. Introdução

Com a popularização de redes sociais e outras formas de disseminação de conteúdo, combinado com os algoritmos de viralização de conteúdo que amplificam o alcance em busca de aumentar o engajamento dos usuários, a proliferação da desinformação aumentou. Esse comportamento chamou a atenção da comunidade para a necessidade de construir sistemas de verificação de factualidade melhores [1].

Para incentivar o desenvolvimento de soluções que mitiguem esse problema, Wadden et al. [3] introduziram a tarefa de processamento de linguagem natural (PLN ou NLP em inglês) de *verificação de afirmações científicas*. Nessa tarefa, um sistema é responsável por verificar a veracidade de uma afirmação, tendo como base um corpus de artigos e textos científicos sobre a área do conhecimento. Para facilitar o desenvolvimento e compartilhamento de resultados, uma base de dados aberta foi criada, denominada SciFact [2].

A base de dados SciFact é um conjunto de afirmações científicas acompanhada de artigos que corroboram ou refutam essas afirmações. A base de dados também fornece anotações de quais sentenças foram necessárias e suficientes para tomada de decisão.

2. Data set

O dataset SciFact é naturalmente dividido em duas partes: um corpus de 5.183 documentos e um conjunto de 1.409 afirmações. Cada uma das afirmações é acompanhada de uma anotação denominada *golden label*, apresentada no seguinte formato:

- Quais dos documentos do corpus são importantes para racionalizar sobre a veracidade da afirmação;
- Para cada documento importante, quais as sentenças que foram úteis na identificação;
- Para cada conjunto de sentenças, uma classificação *c* indica se a afirmação e sentenças concordam ou não: $c \in$ *{"SUPPORT", "REFUTES"}*.

O dataset já vem dividido em treino, desenvolvimento e teste. A divisão é tal que temos 809 afirmações de treino, 300 afirmações de desenvolvimento e 300 afirmações de teste. As afirmações de teste são disponibilizadas sem a anotação de *golden label*.

Set	SUPPORTS	NOINFO	REFUTES	Total
Train	332	304	173	809
Dev	124	112	64	300
Test	100	100	100	300

Tabela 1: Distribuição das afirmações segmentado por classe e dataset.

3. Metodologia

A proposta de solução do presente trabalho é reproduzir o sistema proposto em Pradeep et al. [1]. Esse sistema, denominado pelos autores como VerT5erini, é composto por 3 modelos de NLP T5 [7], onde cada um deles é responsável por um dos estágios do fluxo de verificação de veracidade.Os modelos T5 são uma família de modelos de NLP do tipo *text-to-text*, que implementa a arquitetura Transformer completa, isto é, com encoder e decoder, e suas versões pré-treinadas foram disponibilizadas pelos autores em diversos tamanhos.

O fluxo de verificação de veracidade pode ser definido de diversas formas. No presente trabalho, decidimos separá-lo em três etapas distintas: **(1) Ranking de documentos relevantes, (2) Seleção de sentenças importantes** e **(3) Classificação**.

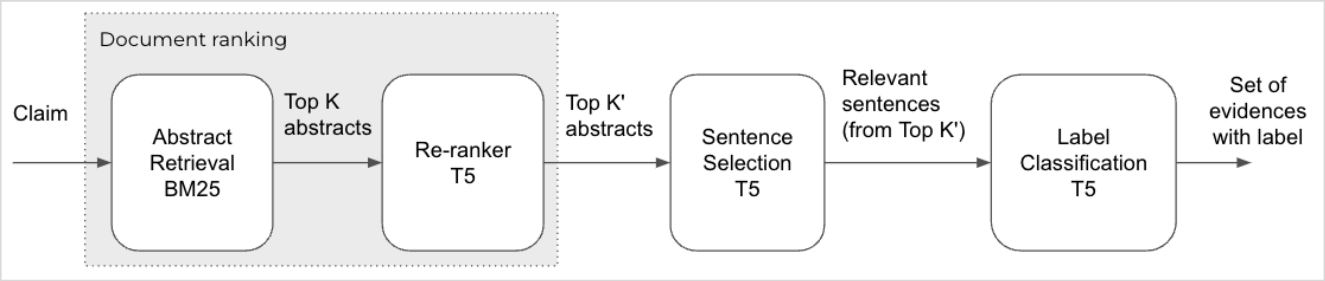

Figura 1: Visualização do fluxo de verificação de factualidade. A primeira etapa (Ranking de documentos relevantes) é implementada em 2 estágios, sistema de busca esparso (BM25) seguido de um modelo de ranking.

3.1 Ranking de documentos relevantes

O ranking de documentos relevantes foi implementado em duas partes. A primeira parte é chamada de rankeamento, e optou-se por implementar essa etapa utilizando uma função de rankeamento de relevância chamada BM25 [5].

A segunda parte do ranking de documentos relevantes é denominado re-ranking. Trata-se de um ajuste sobre a pré-seleção de documentos feita pelo BM25, de maneira que o número de documentos seja reduzido. Essa parte foi implementada utilizando-se um T5.

Mais especificamente, o modelo utilizado foi o monoT5 [9], que é um modelo T5 pré-treinado na tarefa de rankeamento de relevância de documentos. Esse treinamento foi feito utilizando-se o dataset MS MARCO [11], e o modelo está disponível na Hugging Face para download [10] para diversos tamanhos de modelos T5. O tamanho do modelo escolhido foi o T5-Base, que possui 220 milhões de parâmetros.

O BM25 foi configurado para recuperar 20 documentos (*Top K = 20*, figura 1), enquanto que o monoT5 foi configurado para reduzir esse número para apenas 3 documentos finais (*Top K' = 3*, figura 1).

A sequência de entrada utilizada durante o pré-treinamento do monoT5 no dataset MS MARCO foi:

Query: q Document: a Relevant:

onde *q (query)* é a afirmação e *a (abstract)* é o documento a ser analisado. O modelo foi treinado para produzir as palavras *true* ou *false*, dependendo se o texto em questão é relevante ou não. Não foi feito mais nenhum fine-tuning sobre este modelo.

2.2 Seleção de sentenças importantes

A seleção de **sentenças importantes** é a etapa na qual deve-se extrair dos Top K' **documentos** as sentenças que servem de **racional**. Em outras palavras, são sentenças-chave do **documento** que permitem argumentar a favor ou contra a **afirmação** inicial.

Essa etapa também foi implementada utilizando o modelo monoT5 [9]. Porém, identificamos que seria necessário utilizar um modelo maior. Utilizamos o modelo T5-Large, que conta com 770 milhões de parâmetros.

Além do pré-treino padrão do T5 e do pré-treino sobre o dataset MS MARCO, também fizemos um treinamento específico no nosso dataset. A partir das anotações *golden label*, criamos exemplos de sentenças importantes (tanto as sentenças que suportam (SUPPORTS) ou que refutam (REFUTES) foram consideradas importantes. Exemplos não-importantes foram selecionados dos mesmos documentos, excluindo-se as sentenças do *golden label*, e a seleção foi feita aleatoriamente, mas com pesos maiores para sentenças semelhantes à afirmação (utilizou-se TF-IDF para calcular a métrica de semelhança).

A sequência de entrada é semelhante ao da etapa anterior, com a diferença de que agora temos sentenças no lugar dos documentos.

2.3 Classificação

A etapa de classificação é a última. E a tarefa a ser executada é a predição de se a afirmação de entrada é verdadeira (SUPPORTS), falsa (REFUTES), ou se não há informação suficiente para concluir (NOINFO), tendo como base as sentenças selecionadas como importantes na tarefa anterior.

Para essa etapa, um modelo T5-Large (770 milhões de parâmetros) foi treinado para a tarefa de classificação multi-classe. Dessa vez, o T5-Large não foi pré-treinado no dataset MS MARCO.

A sequência de entrada para o modelo foi a seguinte:

hyphothesis: q sentence1: s1 … sentenceZ: sZ

E a sequência-alvo usada no treinamento foi uma dentre *true*, *weak* ou *false*, correspondendo a SUPPORTS, NOINFO e REFUTES, respectivamente.

Nos exemplos em que não haviam *golden label*, foram criadas sentenças NOINFO artificialmente. As sentenças foram extraídas aleatoriamente dos textos que citam o assunto da afirmação, com pesos maiores para sentenças mais semelhantes, usando também o TF-IDF como métrica de semelhança.

4. Experimentos

A tarefa de ranking de documentos relevantes foi avaliada em relação à métrica de recall, limitada a recuperar até 3 ou 5 documentos. A tabela 2 apresenta os resultados obtidos por nosso set-up experimental em conjunto com os resultados do artigo de referência [1]. Os resultados ficaram todos muito semelhantes.

Modelo	R@3	R@5
Oracle	97,61%	100,00%
TF-IDF	69,38%	75,60%
BM25	79,90%	84,69%
T5 (MS MARCO)	86,12%	**89,95%**
T5 (MS MARCO MED)	85,65%	89,00%
T5 (SciFact)	**86,60%**	89,40%
T5 (nossa solução)	85,65%	88,52%

Tabela 2: Comparação do recall para a tarefa de ranking de documentos, limitados a até 3 (R@3) e 5 (R@5) documentos selecionados, dataset de desenvolvimento.

A tarefa de seleção de sentenças importantes foi avaliada em relação a precisão, recall e F1-Score para a predição da classe positiva (isto é, sentença importante). Os resultados estão consolidados na tabela 3, juntamente com os resultados do artigo de referência. Os resultados que obtivemos estão em linha com o artigo de referência.

Modelo	Precisão	Recall	F1-Score
RoBERTa-large	73,71%	70,49%	72,07%
T5 (3-B)	**79,29%**	73,22%	76,14%
T5 (nossa solução)	76,86%	**76,23%**	**76,54%**

Tabela 3: Desempenho da tarefa de seleção de sentenças importantes, dataset de desenvolvimento.

Como os exemplos negativos do dataset de treino e desenvolvimento foram criados artificialmente, é possível que exista divergências na metodologia e os resultados não sejam diretamente comparáveis. De todo modo, os resultados são semelhantes aos observados no artigo de referência.

A tarefa de classificação foi avaliada também para as métricas de precisão, recall e F1-Score, mas agora trata-se de um problema de classificação multi-classes. Os resultados estão consolidados na tabela 4.

Podemos observar um resultado muito interessante para essa tarefa: essa é a primeira tarefa na qual o desempenho do modelo T5-3B de 3 bilhões de

parâmetros, extraído do artigo de referência [1], apresenta um desempenho significativamente superior ao desempenho da solução construída nesse projeto, que utilizou um modelo T5-Large. O desempenho da solução desenvolvida ficou semelhante ao do modelo RoBERTa-large.

Modelo	Classe	Precisão	Recall	F1-Score
RoBERTa-large	SUPPORTS	92,56%	81,16%	86,49%
	NOINFO	74,82%	**92,86%**	82,87%
	REFUTES	77,05%	66,20%	71,21%
T5 (3-B)	SUPPORTS	**93,13%**	**88,41%**	**90,71%**
	NOINFO	85,25%	**92,86%**	**88,89%**
	REFUTES	**86,76%**	**83,10%**	**84,89%**
T5 (nossa solução)	SUPPORTS	88,62%	87,90%	88,26%
	NOINFO	**86,54%**	80,36%	83,33%
	REFUTES	69,86%	79,69%	74,45%

Tabela 4: Desempenho para a tarefa de classificação, dataset de desenvolvimento.

Por fim, a última avaliação de desempenho foi feita sobre o desempenho do fluxo completo de verificação de factualidade. Para essa avaliação, duas configurações foram implementadas. Na primeira configuração, o sistema foi testado no modo “oráculo”, que é a situação onde a primeira etapa do fluxo (ranking de documentos) é substituída por um oráculo que fornece exatamente os documentos relevantes para o sistema. A tabela 5 compara o desempenho entre o modelo implementado e a referência [1] para essa configuração.

Modelo	Precision	Recall	F1-Score
Label Only			
Oracle (referência, 3B)	**92,70%**	**78,95%**	**85,27%**
Oracle (nosso)	59,09%	37,32%	45,75%
Label + Rationale			
Oracle (referência, 3B)	**88,76%**	**75,60%**	**81,65%**
Oracle (nosso)	49,24%	31,10%	38,12%
Selection Only			
Oracle (referência, 3B)	**83,54%**	**72,13%**	**77,42%**
Oracle (nosso)	64,82%	44,81%	52,99%
Selection + Label			
Oracle (referência, 3B)	**78,16%**	**67,49%**	**72,43%**
Oracle (nosso)	37,15%	25,68%	30,37%

Tabela 5: Desempenho do fluxo com oráculo para todas as métricas de desempenho, dataset de desenvolvimento.

Para o desempenho do fluxo completo, quatro tarefas diferentes foram avaliadas. As duas primeiras tarefas (Label Only e Label + Rationale) avaliam a performance na tarefa de selecionar documentos importantes, levando em conta

se a seleção foi corretamente racionalizada (+Rationale) ou não (Label Only). As duas últimas tarefas avaliam o desempenho na seleção de sentenças importantes, contabilizando todas as citações (Selection Only) ou então apenas quando a classificação da afirmação foi correta (Selecion+Label) além da seleção. Em todas as tarefas, as métricas de desempenho avaliadas foram a precisão, o recall e o F1-Score. Mais detalhes podem ser encontrados nas referências [2, 3, 6].

O desempenho do modelo desenvolvido foi bastante inferior quando comparado à referência. Em média, o F1-Score da referência foi o dobro do obtido nos experimentos.

A tabela 6 apresenta o desempenho para o sistema completo de verificação de factualidade do modelo desenvolvido e da referência. Em ambos os modelos, o sistema de ranking de documentos importantes utilizado foi o de dois estágios, composto por BM25+T5.

Modelo	Precision	Recall	F1-Score
Label Only			
T5-3B (referência)	**65,07%**	**65,07%**	**65,07%**
T5-Large (nosso)	37,00%	35,41%	36,19%
Label + Rationale			
T5-3B (referência)	**61,72%**	**61,72%**	**61,72%**
T5-Large (nosso)	30,50%	29,19%	29,83%
Selection Only			
T5-3B (referência)	**64,81%**	**57,37%**	**60,87%**
T5-Large (nosso)	42,18%	41,26%	41,71%
Selection + Label			
T5-3B (referência)	**60,80%**	**53,83%**	**57,10%**
T5-Large (nosso)	25,14%	24,59%	24,86%

Tabela 6: Desempenho do fluxo com BM25+T5 para todas as métricas de desempenho, dataset de desenvolvimento.

Os resultados foram semelhantes ao obtido na configuração “oráculo”. Em média, o F1-Score do modelo de referência foi o dobro.

5. Conclusões

Neste trabalho, concluímos que é possível construir um sistema de verificação de factualidade utilizando-se uma combinação de sistemas de busca e modelos de linguagem natural. Ao segmentar o sistema em etapas distintas, é possível criar modelos especializados para cada uma das tarefas de Ranking de documentos relevantes, Seleção de sentenças importantes e Classificação a partir de modelos pré-treinados de multipropósitos.

Também foi possível observar o impacto do tamanho dos modelos de linguagem natural (mensurado em número de parâmetros) versus o desempenho em cada uma das etapas do fluxo de verificação de factualidade. Nos resultados apresentados, observou-se que o tamanho do modelo é relevante para a tarefa de Classificação. Nas tarefas de Ranking de documentos importantes e de Seleção de Sentenças Importantes, o desempenho foi semelhante para todos os tamanhos de modelos.

6. Trabalhos futuros

É interessante investigar maneiras de melhorar o desempenho do modelo desenvolvido frente aos resultados obtidos no artigo de referência.

Durante os experimentos, observamos que o desempenho da etapa de Classificação é fortemente influenciado pelo desbalanceamento de classes. É possível corrigir o desempenho utilizando-se técnicas de oversampling, ou então fazendo o rebalanceamento dos logits dos tokens de cada classe obtidos na saída do modelo.

Outro parâmetro sensível do sistema é a escolha do valor de corte dos logits da etapa de Seleção de Sentenças Importantes. Valores alternativos deste parâmetro podem levar a aumentos na performance.

Por fim, devido à correlação encontrada entre desempenho e o número de parâmetros dos modelos, é interessante testar o desempenho do sistema caso substituíssemos o modelo T5-Large e T5-3B por modelos extremamente grandes como o PaLM [12], que possui 540 bilhões de parâmetros.

References

[1] Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. Scientific Claim Verification with VERT5ERINI. arXiv:2010.11930, October 2020. arXiv: 2010.11930. http://arxiv.org/abs/2010.11930

[2] Homepage about the dataset SciFact, provided by Allen Institute for AI: https://leaderboard.allenai.org/scifact/submissions/get-started

[3] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or Fiction: Verifying Scientific Claims. arXiv:2004.14974, October 2020, arXiv: 2004.14974. http://arxiv.org/abs/2004.14974

[4] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang, MS MARCO: A Human Generated MAchine Reading

COmprehension Dataset. https://microsoft.github.io/msmarco/

[5] Wikipedia: Okapi BM25 (https://en.wikipedia.org/wiki/Okapi_BM25)

[6] SciFact scoring function implementation details (with examples) https://github.com/allenai/scifact/blob/master/doc/evaluation.md

[7] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv:1910.10683.

[8] Pyserini, a Python toolkit for reproducible information retrieval research with sparse and dense representations. https://github.com/castorini/pyserini

[9] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In Findings of EMNLP.

[10] Hugging Face, MonoT5, a T5-base reranker fine-tuned on the MS MARCO passage dataset for 100k steps (or 10 epochs). https://huggingface.co/castorini/monot5-base-msmarco.

[11] MS MARCO, a collection of datasets focused on deep learning in search. https://microsoft.github.io/msmarco

[12] PaLM: Scaling Language Modeling with Pathways https://arxiv.org/abs/2204.02311