Matching Patient Cases to Clinical Trials

Ana Beatriz Breia, João Funenga, and Mário Miranda

NOVA School of Science and Technology - Universidade NOVA de Lisboa

Abstract. No mundo da medicina e da saúde é fundamental fazer uma boa correspondência entre os casos dos pacientes e os ensaios clínicos, porém estima-se que não se consegue encontrar um bom ensaio clínico em 80% dos pacientes, no tempo previsto. Assim, este projeto tem como principal objetivo utilizar técnidas de recuperação e análise de informação por forma a conseguir resolver problemas de correspondência e relevância entre documentos.

Keywords: Ensaios Clínicos · Recuperação de Informação· Relevância entre Dcocumentos.

1 Introdução

Por forma a encontrar ensaios clínicos onde os pacientes possam participar consideramos duas grandes fontes de dados: as *queries*, que representam os registos dos pacientes, e os ensaios clínicos, que contêm uma panóplia de informações relativas a quem pode ou não ser recrutado para esse determinado ensaio. Uma vez que estas informações são bastante extensas, apenas consideremos uma secção destes documentos, os *brief_summaries*, que constituirão o nosso *corpus.*
As principais estratégias aplicadas e analisadas neste projeto, para realizar boas correspondências entre estes os dados, são modelos como o *Vector Space Model* ou modelos de linguagem.

2 Métodos Implementados

Este capítulo é dedicado a um breve resumo teórico dos métodos implementados no projeto. Para mais detalhes veja [1].

2.1 Vector Space Model e LMJM smoothing

Neste projecto foram testados dois *retrieval models*. O primeiro modelo utilizado foi um modelo de *"geometric/linear spaces"*, o *Vector Space Model*.
O *Vector Space Model* (VSM) é um modelo algébrico para representar documentos de texto. Uma forma de analisar a importância de um documento dado uma certa *query* é usar o método TF-IDF, isto é, vamos dar "pesos" diferentes aos termos desse documento. Estes pesos são determinados segundo a frequência desse termo no documento ($tf_i(d)$), e no grau de raridade desse termo em todo o *corpus* (idf_i). Temos então,

$$w_{i,j} = tf_i(d_j) \cdot idf_i$$

Também é possível comparar o quão similares são documentos diferentes, calculando o cosseno do ângulo x entre ambos (função *pairwise_distance*).

O segundo modelo utilizado foi o *"Language Model with Jelineck-Mercer smoothing"*. A ideia básica deste método é estimar um modelo de linguagem, por forma a determinar a probabilidade de determinada *query* pertencer ao nosso *corpus*.

2.2 Avaliação

Para avaliar os nossos resultados vamos utilizar as seguintes métricas:
Precision@10: que determina quantos documentos no top 10 são relevantes para uma dada *query;*
nDCG@5: que mede o ganho cumulativo dos vários níveis de relevância nos documentos.
Recall@100: que dá o nº de documentos relevantes que não foram disponibilizados ao paciente;
Mean Reciprocal Rank: que calcula a fração $\frac{1}{rank}$, onde *rank* designa a posição do primeiro documento relevante;
Mean Average Precision: para avaliar a precisão média dos documentos;
Precision-recall curves: que mostra graficamente a relação entre a precisão e o recall.

3 Configuração Experimental

3.1 Ler os Ensaios Clínicos e os Casos dos Pacientes

A primeira fase do nosso projeto é ler e extrair a informação pertinente dos documentos dos ensaios clínicos. Para tal, apenas analisamos os ensaios presentes no ficheiro *qrels-clinical_trials.txt*. Este ficheiro é composto por várias linhas em que cada uma contém 3 informações que nos interessam, o "Patient Id" (primeira coluna), "Clinical Trial Id" começado por NCT (terceira coluna) e o valor dado à qualidade da correspondência entre o respetivo paciente e ensaio, podendo este ser 0 (mau *match*), 1 (bom *match*), 2 (excelente *match*). Os documentos que não estão presentes neste ficheiro não foram avaliados por profissionais, ou seja, não conseguimos saber se são relevantes ou não. Assim, começamos por percorrer o ficheiro *qrels-clinical_trials.txt* e guardar todos os ID's nele presente para depois podermos filtrá-los do ficheiro ".zip", que contém todas as entradas de ensaios clínicos de dezembro de 2015. Este ficheiro de texto corresponde à nossa *ground-truth*, isto é, temos de facto a qualidade do *match* entre os casos dos pacientes e os ensaios clínicos.
Os ensaios clínicos são o *corpus* do nosso projeto. As *queries* são os casos dos pacientes.

3.2 Utilização do Vector Space Model

Vamos agora analisar a importância de um documento dada uma certa *query*, isto é, vamos utilizar o método TF-IDF (*"TfidfVectorizer"* da biblioteca sklearn), bem como a função *pairwise_distance*. De modo a testarmos várias abordagens, iremos experimentar este modelo tanto com unigramas como com bigramas, bem como com e sem "stop words", por forma a comparar a performance de todos estes cenários.

3.3 Utilização do LMJM com smoothing

Com o intuito de descobrir o melhor lambda que integra a fórmula do LMJM, fizemos a separação das *queries* num grupo de treino (80% das *queries*) e num grupo de teste. A razão pela qual são separadas as *queries* e não os ensaios é simples: do ponto de vista de um paciente é desejável saber quais os ensaios mais relevantes segundo o seu caso, por isso a análise para cada paciente (*query*) tem de ser feita com todos os ensaios disponíveis. Caso fossem os ensaios separados num grupo de treino e outro de teste, estaríamos a comparar para cada ensaio os pacientes que seriam atribuídos, o que faria sentido da perspetiva de um médico responsável por um *trial*, porém não é esse o objetivo do trabalho.

Relativamente à escolha do melhor lambda, este foi escolhido de acordo com o P@10 e, de facto, vamos poder confirmar no próximo capítulo que o lambda escolhido correponde à curva com uma área superior.

Para isto, em cada iteração sobre os dados de treino, calculamos o valor do P@10 e caso este valor seja melhor que o anterior, guardamos o lambda correspondente para usarmos posteriormente sobre o grupo de teste.
Também utilizaremos este modelo tanto com unigramas como com bigramas.

4 Análise dos Resultados

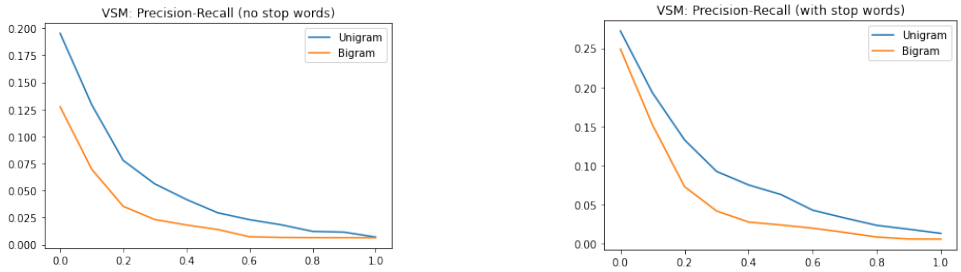
4.1 Resultados com Vector Space Model

Ao aplicarmos este modelo sem "stop words" (veja-se a **figura 1a** da curva *precision-recall*), podemos observar que inicialmente a precisão é elevada, tanto para unigramas como para bigramas, e vai decrescendo à medida que vamos classificando os documentos, isto significa que quanto mais documentos classificarmos mais documentos não relevantes vamos descobrir, mas por outro lado, vamos conseguir aproximarmo-nos do número total de documentos relevantes (o recall aumenta). Assim, o melhor cenário seria a curva ser o mais próxima de uma reta na forma $y = 1$, isto significaria que o nosso modelo conseguiu classificar corretamente todos os documentos relevantes como relevantes, de todos os que existem.

Para testar o modelo com "stop words" utilizou-se:

common_stop_words = *"is", "the", "an", "a", "to", "and", "be", "been", "that", "this", "i", "than", "patient", "am", "health", "sick", "clinical",*

tendo-se obtido o gráfico da **figura 1b**:



(a) VSM sem Stop Words

(b) VSM com Stop Words

Fig. 1: Comparação da curva Precision-Recall método sem e com Stop Words

Conclui-se, pelos gráficos, que o método com "stop words" apresenta um pequeno aumento na melhoria de performance, tanto em unigramas como em bigramas. Já relacionando os n-gramas, conclui-se que os unigramas, em ambos os casos, apresentam resultados mais satisfatórios, uma vez que a área abaixo das curvas é superior à dos bigramas, traduzindo-se num valor mais alto de precisão e de recall.

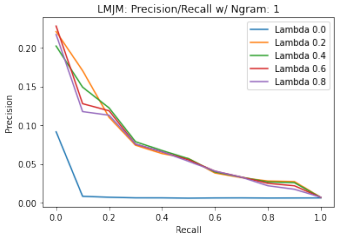
4.2 Resultados com LMJM

O mesmo raciocínio utilizado na curva precision-recall no caso do *Vector Space Model*, também é válido na curva no caso do LMJM. Veja-se a **figura 2** (a precisão decresce e o recall aumenta).

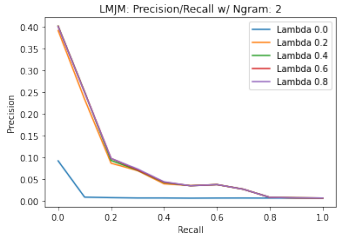
Calculando o melhor lambda para unigramas e bigramas, a partir da métrica P@10, e atendendo a figura 2, o melhor lambda para ambos os casos, como já foi referido, é 0.4, o que na teoria faz sentido, uma vez que baixos valores de lambda são adequados para *queries* longas.

Para além disto, neste caso é nítido que a utilização de bigramas aumenta bastante a perfomance do método em termos de precisão, porém em termos de recall os unigramas são superiores.

Tendo calculado o melhor lambda, analisemos agora a curva de *precision-recall* relativa ao conjunto de dados reservados para o teste do modelo.



(a) Unigramas



(b) Bigramas

Fig. 2: Comparação dos diferentes valores para lambda na curva Precision-Recall

Através do gráfico da **figura 3** percebemos que a performance neste *set* de dados desconhecido não corresponde ao esperado, o que nos leva a crer que possivelmente o modelo sofreu *overfitting*, por exemplo, e portanto não consegue generalizar corretamente as ocorrências dos dados.

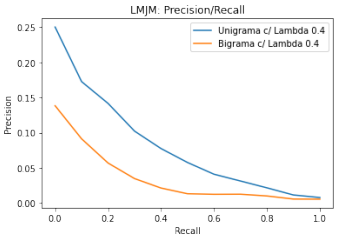
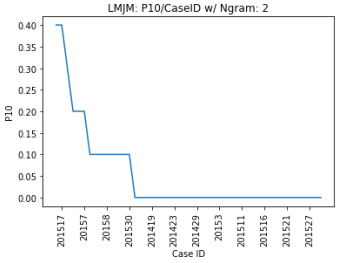
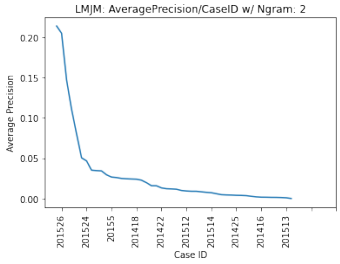


Fig. 3: Comparação da curva Precision-Recall para modelo LMJM



(a) P@10 por paciente



(b) AP por paciente

Fig. 4: Comparação de P@10 e AP por paciente.

O gráfico da **figura 4a**, relaciona o valor do P@10 com os 60 pacientes que compõem as nossas *queries*. Fazendo a sua análise, podemos verificar que existem pacientes onde nenhum documento relevante apareceu nos 10 primeiros resultados. Caso não nos queiramos limitar aos 10 primeiros

documentos, podemos fazer a mesma análise no gráfico com a precisão média (AP) pelos pacientes (gráfico da **figura 4b**).

4.3 Resultados com Métricas de Avaliação

	VSM (w/o Stop Words)					VSM (w/ Stop Words)				
	P@10	*Recall@100*	*AP*	*NDCG@5*	*MRR*	*P@10*	*Recall@100*	*AP*	*NDCG@5*	*MRR*
Unigramas	0.095	0.2580	0.0718	0.0968	0.0051	0.0967	0.2630	0.0715	0.0940	0.0051
Bigramas	0.0583	0.1200	0.0377	0.0675	0.0051	0.0566	0.1289	0.0408	0.0723	0.0051

	LMJM				
	P@10	*Recall@100*	*AP*	*NDCG@5*	*MRR*
Unigramas	0.0958	0.2272	0.0658	0.0880	0.0050
Bigramas	0.0999	0.2310	0.0662	0.1049	0.0058

Analisemos primeiro o método VSM. Conseguimos verificar que, na generalidade, os valores das métricas entre o método com ou sem "stop words" são muito semelhantes, porém ligeiramente mais elevados em algumas métricas com a presença destas. Isto porque, quando usadas no método, estas palavras são ignoradas, não enviesando a probabilidade do *match*. Em relação ao uso de unigramas e bigramas, podemos verificar que com unigramas a perfomance é significativamente melhor, tanto no *P@10* como no *recall@100*.

Note-se que neste domínio o recall, comparativamente à precisão, é uma métrica mais importante, visto ter em conta falsos negativos (documentos considerados não relevantes, mas relevantes), enquanto que na precisão, os falsos negativos não entram no cálculo desta. Não ter em conta os falsos negativos resulta, neste caso, em ignorarmos ensaios clínicos (analisando-os como um mau *match*) e estes serem possívelmente positivos (relevantes). Visto ser do interesse do paciente saber todos os ensaios em que este pode participar, faz sentido analisá-los a todos.

Em jeito de conclusão da análise do VSM, se tivéssemos de utilizar este método , aplicá-lo-íamos com "stop words" e recorrendo a unigramas.

Relativamente ao LMJM acontece o contrário, sendo a performance com bigramas melhor que com unigramas. Por estarmos a lidar com termos médicos que por vezes são constituídos por mais que uma palavra, faz sentido que o uso de bigramas seja mais relevante, embora não tenha sido o concluído com o VSM. Isto pode ser devido ao modelo VSM ser mais antigo e não se comportar tão bem para o tipo de dados que estamos a analisar.

Relativamente ao MRR, este não é muito relevante no nosso caso de estudo uma vez que apenas tem em conta a posição do primeiro documento relevante. Ora, um paciente não tem apenas em conta o primeiro ensaio relevante que lhe é apresentado, este pode, e deve, analisar outros ensaios em posições "maiores" das ordenadas.

Posto isto, entre ambos os modelos, o que seria mais indicado escolhermos seria o método LMJM com bigramas, visto que apresenta os melhores resultados.

4.4 Conclusão

Face ao exposto anteriormente, é-nos possível concluir que em termos de resultados ambos os modelos apresentam valores semelhantes, mesmo sabendo que o VSM é um modelo mais antigo

e sendo o LMJM, segundo a literatura, mais eficaz para *queries* longas. Isto acontece porque o modelo VSM utiliza classe TF-IDF, que dá "pesos" às palavras consoante a sua raridade. Com o nosso dataset de índole médica, por ter um vocabulário especifico, esta vantagem do VSM face ao modelo LMJM é evidente. (Esta "vantagem" poderia ser reduzida e talvez eliminada com a utilização de N-gramas mais altos no modelo LMJM).
Contudo, ao analisar a tabela, observamos que os valores são bastante baixos em ambos os métodos. Prevemos, para um futuro próximo, implementar métodos mais eficazes por forma a conseguir resultados mais satisfatórios.

References

1. Magalhães, J. (2021). *Information Retrieval, slides a03,a04,a05.*, NOVA-School of Science and Technology.
2. Chaudhary, A. (2020, 8 de outubro). *Evaluation Metrics For Information Retrieval.* Acedido a 02/11/2021, em https://amitness.com/2020/08/information-retrieval-evaluation/