

Open-Domain Search Assistant

Gonalo Antunes - 53044
ga.antunes@campus.fct.unl.pt
NOVA School of Science and
Technology
Caparica, Setbal, Portugal

Joo Diogo - 50483
jp.diogo@campus.fct.unl.pt
NOVA School of Science and
Technology
Caparica, Setbal, Portugal

Samuel Viegas - 53600
sm.viegas@campus.fct.unl.pt
NOVA School of Science and
Technology
Caparica, Setbal, Portugal

CONVERSATION: 31 MODEL: BART METHOD: EG SUMMARY LENGTH: 70

Topic: Throat Cancer

Messages

#	Sender	Message	See Knowledge
0	user	What is throat cancer?	
1	assistant	Throat cancer refers to cancerous tumors that develop in your throat (pharynx), voice box (larynx) or tonsils. Your throat is a muscular tube that begins behind your nose and ends in your neck. Throat cancer most often begins in the flat cells that line the inside of your throat. It can also affect the piece of cartilage (epiglottis) that acts as a lid for your windpipe.	SHOWING
2	user	Is it treatable?	
3	assistant	Throat cancers may be cured when detected early. If the cancer has spread (metastasized) to surrounding tissues or lymph nodes in the neck, about half of patients can be cured. After treatment, therapy is needed to help with speech and swallowing. Treatment is aimed at prolonging and improving quality of life. The cancer is not curable if it has spread outside the head and neck.	KNOWLEDGE
4	user	Tell me about lung cancer.	
5	assistant	Lung cancer is a malignant lung tumor characterized by uncontrolled cell growth. If left untreated, this growth can spread beyond the lung by the process of metastasis. The two main types are small-cell lung carcinoma (SCLC) and non-small-cell Lung Carcinoma (NSCLC). The most common symptoms are coughing, weight loss, shortness of breath, and chest pains.	KNOWLEDGE

Knowledge Support for Message 1

Knowledge Used
Text
Knowledge Used What is throat cancer? Throat cancer is any cancer that forms in the throat. The throat, also called the pharynx, is a 5-inch-long tube that runs from your nose to your neck. The larynx (voice box) and pharynx are the two main places throat cancer forms. Throat cancer is a type of head and neck cancer, which includes cancer of the mouth, tonsils, nose, sinuses, salivary glands and neck lymph nodes. Learn about the types of throat cancer.
What is throat cancer? Throat cancer is any cancer that forms in the throat. The throat, also called the pharynx, is a 5-inch-long tube that runs from your nose to your neck. The larynx (voice box) and pharynx are the two main places throat cancer forms. Throat cancer is a type of head and neck cancer, which includes cancer of the mouth, tonsils, nose, sinuses, salivary glands and neck lymph nodes.
Throat or larynx cancer. Throat cancer is cancer of the vocal cords, larynx (voice box), or other areas of the throat. People who smoke or use tobacco are at risk of developing throat cancer. Drinking too much alcohol over a long time also increases risk. Smoking and drinking alcohol combined lead to an increased risk for throat cancer. Most throat cancers develop in adults older than 50.
Throat or larynx cancer. Throat cancer is cancer of the vocal cords, larynx (voice box), or other areas of the throat. People who smoke or use tobacco are at risk of developing throat cancer. Drinking too much alcohol over a long time also increases risk. Smoking and drinking alcohol combined lead to an increased risk for throat cancer.

Figure 1: Knowledge-Driven Answer Generation Demo

ABSTRACT

A utilizao de uma barra de pesquisa onde so inseridas keywords por parte de um utilizador  feita, na sua maioria, de forma passiva e tendo como resultado a listagem de documentos relacionados. Ao contrrio desta abordagem, o objetivo do desenvolvimento deste projeto tem por base o uso de conjunto de conversas sobre diferentes contedos onde procuramos estudar quais os fatores que influenciam a qualidade de respostas geradas no decorrer de uma dada conversa.

KEYWORDS

Computer Science, Web Search, Search Assistant, Entity Linker, Entity, Page Rank, Summarization, Chatbot, Open-Domain, Python

1 INTRODUO

O trabalho desenvolvido no mbito desta cadeira tem por base o tema “Open-Domain search assistant”. De facto, um dos objetivos principais  o estudo e desenvolvimento de um sistema (conversa com um chatbot) para gerao de respostas sucintas e com um elevado nvel de qualidade em detrimento da usual visualizao de vrios documentos (ex: pesquisa num search engine como o google).

Efetivamente, foi necessrio abordarmos o problema com base em 4 etapas iniciais. Inicialmente comeamos pela preparao da groundtruth e do dataset, ou seja, extramos vrias perguntas de vrios tpicos e as melhores 10 respostas a essas perguntas. Depois de completar a base dos nossos dados, passmos para a extrao de entidades tanto das perguntas como das respostas (DBpedia Spotlight para entity linking), de seguida construo do grafo de entidades e aplicao do pageRank. Deste modo, -nos tmbm possvel escolher quais as melhores passagens com base no pageRank. Por ltimo, usamos o Bart para sumarizar e gerar a resposta nica usando para tal as passagens referidas. Procedemos tmbm ao estudo de diferentes formas para gerar o grafo de entidades, aplicando um threshold para eliminar entidades fracas. Explormos diferentes modos distintos do uso do grafo das entidades. Por fim, avalimos as respostas sumarizadas por entre os diferentes modos.

2 ALGORITMOS

2.1 Name Entity Linker

O Name Entity Linker   o algoritmo respons vel pela extra o das entidades das perguntas e das passagens. Para tal utilizamos o DBpedia Spotlight, uma solu o open source, desenvolvida para extrair as entidades de texto em linguagem natural. O algoritmo de desambigua o   baseado em semelhan as de cosseno e uma altera o das m tricas do TF-IDF [2]. O algoritmo   dividido em quatro fases. A fase inicial, *spotting*, reconhece nas frases as entidades que podem ter alguma liga o aos recursos do DBpedia. A fase de *Candidate selection*   respons vel por mapear as frases selecionadas com os recursos que s o candidatos das respectivas . A *disambiguation stage* utiliza o contexto em volta das frases para escolher os melhores candidatos e a fase de *annotation* que permite selecionar grupos restritos escolhidos pelo utilizador na configura o.

2.2 Pagerank

O Pagerank consiste num m todo de avalia o e c lculo da import ncia de uma entidade. Funciona tendo como base a p gina da wikip dia da entidade e com o numero de p ginas da Wikip dia que incidem nessa p gina e no n mero de p ginas a que esta p gina redireciona. Com o n mero de "In Comings" e "Out Goings" de uma p gina   poss vel calcular uma import ncia ponderada da entidade. Neste caso, a vers o do algoritmo utilizada   tamb m conhecido como google PageRank sendo que os par metros mais relevantes s o:

- (1) G, grafo n o direcionado onde ser o realizados os c lculos de modo a obter o Pagerank.
- (2) Alpha, correspondente ao damping factor, valor sobre o qual o algoritmo decide de forma probabil stica se   necess rio saltar para outra p gina ao acaso, evitando, por exemplo, ciclos.
- (3) Maximum Iterations, relativo ao n mero m ximo de itera es no m todo de pot ncias inerente ao c lculo do Pagerank.
- (4) Tolerance, toler ncia de erro utilizado de modo a verificar a converg ncia no m todo de pot ncias.

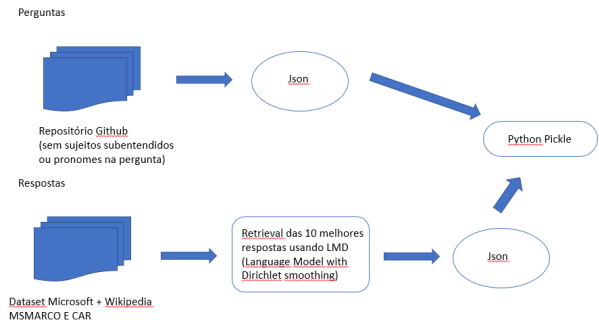
3 IMPLEMENTA O

3.1 Extra o dos t picos e das perguntas e respostas

Para formar o nosso dataset, constitu do por perguntas e respostas de v rios t picos, us mos uma biblioteca de uma framework estandardizada para t picos de NLP, Treccastweb, constitu da por respostas da Wikip dia (CAR) e da microsoft (MSMARCO). Us mos o seguinte reposit rio Github - <https://github.com/daltonj/treccastweb/tree/master/2019/data/evaluation> - para as perguntas, visto que no dataset estavam perguntas com sujeitos subentendidos e pronomes. Para retrieval dos documentos us mos um Language Model com Dirichlet smoothing - LMD no dataset. Guard mos os 10 melhores documentos que obtivemos com a aplica o desse algoritmo. Fic mos assim com um dataset constitu do por 20 t picos, cada um com 6 a 8 perguntas e com as 10 melhores respostas a cada pergunta. Transform mos esses dados num pickle de python para

guardar os dados em mem ria e conseguir aceder rapidamente aos dados.

Figure 2: Dataset Perguntas e Respostas



3.2 Extra o das Entidades

Para a extra o das entidades, us mos um Name Entity Linker. Um Name entity linker permite extrair entidades importantes de um conjunto de texto e assim reduzir significativamente a dimensionalidade e complexidade dos dados de NLP. Para esse efeito, us mos o DBpedia Spotlight. Iter mos todo o dataset (perguntas e respostas) dos v rios t picos e guard mos num vetor de 2 dimens es constitu do por vetores [entidade, vetor de posi o da entidade]. Esse vetor de posi o de entidade   um vetor tendo primeiro indice a pergunta, e a seguir as seguintes 10 respostas, respectivamente.

Figure 3: Extra o das entidades

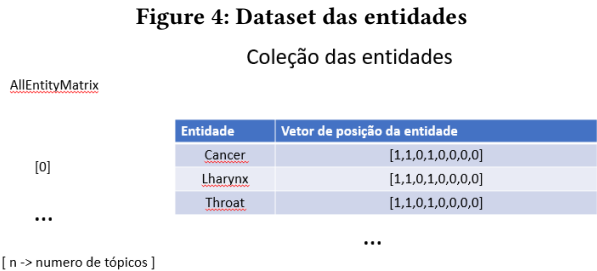


3.3 Vetor Coleção das Entidades

Assim, obtivemos o nosso novo conjunto de dados.

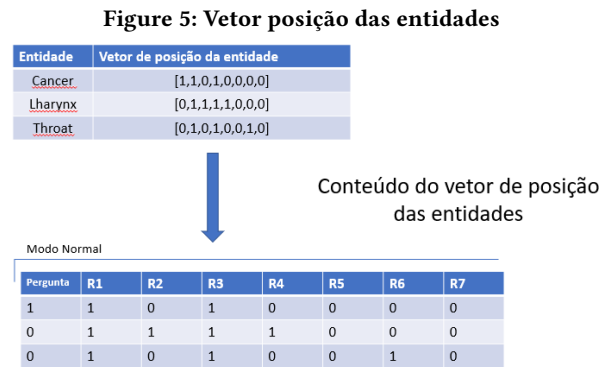
3.3.1 Coleção das entidades.

A coleção de entidades consiste na seguinte imagem:



3.3.2 Conteúdo do vetor de posição das entidades.

O conteúdo do vetor de posições das entidades consiste na seguinte imagem:



3.4 Construção da Matriz de entidades

Para a construção da matriz de entidades foi preciso construir primeiro duas matrizes:

- Matriz de ocorrência das entidades
- Transposta da Matriz de ocorrência das entidades

Para a construção da primeira, iterou-se o dataset das entidades e para cada entidade retirou-se o vetor de posição da entidade. Num vetor auxiliar, foi-se adicionando esses vetores e por fim criada uma matriz com esse vetor final com apenas as posições. Como foi feito por um for, a ordem mantém-se e a primeira posição coincide com a primeira entidade e assim sucessivamente... Tendo já a matriz de ocorrências, realiza-se a transposta com um método do Numpy e obtêm-se a segunda matriz - transposta das ocorrências.

Para finalmente obter a matriz de entidades, que irá relacionar cada entidade consigo mesma e com as outras, multiplicamos a matriz de ocorrência de entidades pela sua transposta.

Em especificação prática, criamos essa matriz como uma Matriz CSR (Compressed sparse row), que é um formato de matriz compacto que só guarda os resultados superior a 0 em memória, pois em

Figure 6: Matriz de ocorrências de entidades e Transposta

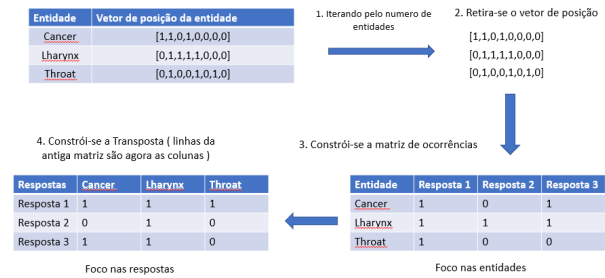
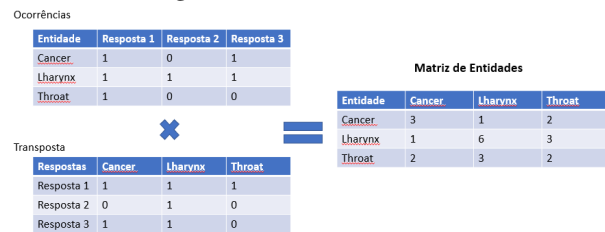


Figure 7: Matriz de Entidades



termos práticos melhora o desempenho e velocidade e é o formato necessário para aplicar o pagerank.

3.5 Aplicação do Pagerank

Com efeito, o Pagerank permite obter a importância ponderada das diferentes entidades.

Assim, utilizando como base o código deste algoritmo em: https://scikit-learn.org/stable/auto_examples/applications/wikipedia_principal_eigenvector.html.

Usamos como parâmetros os valores default do algoritmo: alpha=0.85 max iter=100 tol=1e-10

No final obtemos um vetor de scores das entidades. Como a ordem é sempre preservada, os resultados continuam consistentes, sendo o primeiro score da primeira entidade e assim sucessivamente.

Figure 8: Exemplo do Pagerank



3.6 Re-rank das Respostas

Com um score ponderado e calculado anteriormente para cada entidade é agora possível fazer um novo e melhor ranking. Para cada resposta, são somadas todos os valores das entidades que apareçam nessa resposta. Para isso usamos a transposta anteriormente calculada. Essa matriz tem como linhas as respostas. Assim, com um ciclo for, são percorridas todas as colunas da matriz (entidades) e caso o seu valor seja maior que 0, é somado ao score final da resposta.

Figure 9: Cálculo do novo rank das respostas

Sendo que temos um peso ponderado de cada entidade, podemos usar essa "importância" calculada e reordenar as respostas.

Usamos a transposta por ter as respostas como foco e somamos a importância das entidades que estão presentes nessa resposta. (onde existe um 1)

Transposta				Scores		
Respostas	Cancer	Lharynx	Throat	[0,18384	0,04506	0,10345]
Resposta 1	1	1	1	[Cancer	Lharynx	Throat]
Resposta 2	0	1	0			
Resposta 3	1	1	0			

$$\text{Resposta 1: } 1 * 0,18384 + 1 * 0,04506 + 1 * 0,10345 = 0,33235$$

$$\text{Resposta 2: } 1 * 0,04506 = 0,04506$$

$$\text{Resposta 3: } 1 * 0,18384 + 1 * 0,04506 = 0,2289$$

Nova Ordem:
 Resposta 1
 Resposta 3
 Resposta 2

Cada resposta tem agora um score associado, que é o total da soma dos scores de todas as entidades que contém. Assim, as novas respostas são reordenadas por ordem decrescente. Isso é feito por reindex de um dataframe.

Figure 10: Reordenamento das Respostas

Original		
Respostas	Conteúdo	Scores
Resposta 1	"Throat cancer is originated in the Lharynx."	0,33235
Resposta 2	"The Lharynx is near the pharynx and can suffer infection."	0,04506
Resposta 3	"Lharynx cancer is one type of cancer."	0,2289

↓ Dataframe ordenado por "Scores"

Reordered		
Respostas	Conteúdo	Scores
Resposta 1	"Throat cancer is originated in the Lharynx."	0,33235
Resposta 3	"Lharynx cancer is one type of cancer."	0,2289
Resposta 2	"The Lharynx is near the pharynx and can suffer infection."	0,04506

3.7 Summarization

Finalmente, usamos um modelo de NLP já treinado com ênfase em "Text Sumarization" para gerar uma resposta final. Essa respostas final irá ser um sumário das N melhores repostas. O modelo usado foi o BART, mais concretamente o módulo de Text Sumarization do BART, e escolhemos para o sumário as 3 melhores respostas a cada pergunta.

Figure 11: Sumarization - Resposta Final

Usando um modelo de NLP já treinado, com a especificação de Text Sumarization, conseguimos obter um texto final concentrado das 3 melhores respostas. Usamos o modelo BART.

Reordered		
Respostas	Conteúdo	Scores
Resposta 1	"Throat cancer is originated in the Lharynx."	0,33235
Resposta 3	"Lharynx cancer is one type of cancer."	0,2289
Resposta 2	"The Lharynx is near the pharynx and can suffer infection."	0,04506

↓ BART Text Sumarization Model

"Throat cancer is one type of cancer that is originated in the Lharynx and easily spread and infect all body."

Resposta final do nosso agente a uma pergunta

4 AVALIAÇÃO

4.1 Descrição do Dataset

Com efeito, o dataset sobre o qual operamos tem várias dimensões: existem 20 conversas sendo que cada uma destas é relativa a tópicos diferentes (ex: cancer, netflix, sports, etc.). Cada conversa é composta por 8 perguntas que têm associadas às mesmas 10 respostas de topo (adquiridas através de documentos da Wikipédia). Usamos a função gaussiana como decay (valor que subtraímos à ocorrências das entidades) - consideramos o primeiro valor e vai diminuindo e volta a aumentar no final até ao valor inicial.

4.2 Baselines

Abordamos três métodos diferentes para melhorar a nossa implementação. O primeiro método baseia-se em destacar as entidades extraídas das perguntas anteriores à pergunta actual, p.e. ao avaliar a 4ª pergunta da conversa 31, "What are lung cancer's symptoms?", iríamos considerar as entidades extraídas das perguntas anteriores, ou seja, das perguntas 1, 2 e 3. Esta abordagem foi desenvolvida em dois procedimentos diferentes. O primeiro em que apenas é utilizado as entidades extraídas da primeira pergunta, que é a pergunta que introduz o tema da conversa. O segundo em que utiliza todas as entidades extraídas de todas as perguntas anteriores à pergunta actual. Este método tem a vantagem de realizar um melhor acompanhamento da conversa, p.e. na conversa 61 em que a primeira pergunta é sobre os *Avengers* mas que na sexta pergunta questiona sobre o *DC universe* e o Batman, alterando o ponto fulcral da conversa.

O segundo método desenvolvido foca-se nos vectores de posição das entidades. Alteramos a forma como cada entidade era classificada, passando de quando uma entidade ocorre num documento ser classificada com 1, para uma escala de 1 a 0.5, diminuindo 0.05 a cada documento, em outros termos, as entidades da pergunta seriam classificadas com 1, as entidades do primeiro documento com 0.95, as entidades do segundo documento com 0.9, etc. Esta abordagem dá um maior destaque ao trabalho realizado pelo classificador dos documentos utilizado para seleccionar os dez melhores documentos de cada pergunta.

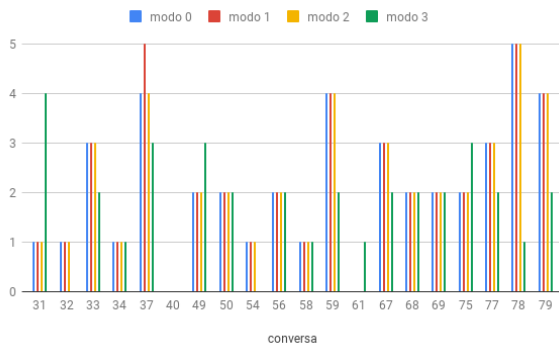
O terceiro método abordado focou-se nas matrizes de entidade em que aplicamos um *threshold* de valor três, por defeito. Deste modo podemos remover as entidades mais irrelevantes de entrarem no algoritmo de Page Ranking.

4.3 Análise de Resultados

Para comparar as diferentes métodos foram implementados quatro modos. O primeiro modo que classifica as entidades com 0 ou 1 e que não aborda nenhuma das perguntas anteriores. O segundo modo que classifica as entidades com 0 ou 1 e que tem em conta apenas a primeira pergunta de cada conversa. O terceiro modo que classifica as entidades com 0 ou 1 e que tem em conta as entidades de todas as perguntas da conversa, anteriores à pergunta actual. E o quarto modo que classifica as entidades de 1 a 0.5 e que não aborda nenhuma das perguntas anteriores. Em todos os modos utilizamos um *threshold* de 3.

Para analisarmos os resultados das diferentes abordagens descritas anteriormente utilizamos dois métodos distintos. O primeiro método é mais matemático e utiliza a classificação existente dos documentos tendo em conta a conversa e a pergunta utilizada no Text Retrieval Conference[1]. Os documentos estão classificados de 0 a 4, em que 0 significa não relacionado com a pergunta e 4 muito relacionado com a pergunta. Ao calcularmos a soma das classificações dos três documentos escolhidos para serem sumariados, obtemos uma classificação de 0 a 12 em que 0 é não relacionado e 12 é extremamente relacionado. Ao calcular estes resultados chegamos à conclusão que os modos 1, 2 e 3 apresentam resultados quase idênticos, em que os documentos seleccionados são os mesmos, logo o seu cálculo final apresenta quase sempre os mesmos resultados. Em relação ao modo 4 apresenta uma maior variação dos resultados, no entanto não se pode concluir que estes resultados sejam melhores que os apresentados pelos restantes métodos, pois não existe um ganho concreto comparado com os restantes métodos. Para além disso, nem todos os documentos estão classificados, logo também não podemos concluir com total confiança que algum modo é melhor que os seus concorrentes. No entanto para podermos criar uma classificação que seja justa e deste modo calculamos o somatório das classificações onde todos os documentos avaliados possuíam classificação. Após este somatório contamos o número de vezes que cada modo é o vencedor, sendo que pode existir mais do que um vencedor. Esta abordagem é importante pois é pretendido destacar qual o modo que é mais consistente em detrimento daquele que obtém apenas uma excelente classificação numa única pergunta. Assim sendo podemos observar na figura 12 o número de vitórias obtidas por conversa.

Figure 12: Classificação segundo os Documentos Classificados



No entanto este dilema não pode ser somente avaliado por base nos resultados obtidos, por isso, para o segundo método de classificação analisamos a qualidade das respostas com base na sumarização final comparando os modos 2 e 4, omitindo comparações relativas aos modos 1 e 3 devido à elevada semelhança entre estes e o modo 2. Foram comparadas as respostas finais nos casos em que o modo 2 obtém uma melhor classificação que o modo 4 e vice versa. A partir desta avaliação concluímos que as respostas geradas pelo modo 3 apresentavam qualidade superior ou igual para a maioria das perguntas. Mesmo quando a sua avaliação inicial é inferior como acontece na conversa 79, na pergunta 4 "What is Herbert Spencer known for in sociology?" onde a resposta apresentada pelo modo 2 é "British philosopher Herbert Spencer was a major figure in the intellectual life of the Victorian era . He was one of the principal proponents of evolutionary theory in the mid nineteenth century . In the late 1800s, many Americans embraced Spencer's Social Darwinism to justify laissez-faire capitalism .". Em comparação, a resposta do modo 4 é "Herbert Spencer was an English philosopher, biologist, anthropologist, sociologist, and prominent classical liberal political theorist of the Victorian era . He is popularly known as the British Aristotle and often called the second founding father of sociology . Spencer's ideas have left an indelible impression on succeeding writers .". Ambas as respostas apresentam um elevado grau de qualidade, não nos sendo possível classificar uma resposta como superior à outra, sendo este apenas um exemplo dos vários que encontramos.

5 DISCUSSÃO

Com efeito, podemos concluir com base nos resultados apresentados no ponto anterior e dando ênfase à avaliação realizada com base na sumarização final podemos concluir que o modo 4 foi o que consistentemente apresentou respostas com maior qualidade. Assim, é ainda importante referir que os seguintes aspetos relativos ao dataset em causa influenciaram os resultados finais:

- (1) Ao termos substituído os pronomes pelas palavras correspondentes (i.e. "its" refere-se a "throat cancer" por exemplo) houve um claro um impacto no facto de os modos 1, 2 e 3 terem resultados semelhantes. Isto deve-se ao facto de apenas os modos 2 e 3 utilizarem entidades de passagens anteriores ao contrário do modo 1, logo caso essa substituição tivesse sido omitida o modo 1 dificilmente teria a mesma performance relativamente a estes.
- (2) No modo 4, a ordem dos documentos utilizada para fazer a atribuição do peso das entidades com base no método LMD consoante a passagem em causa (quando mais longe das passagens iniciais menor o peso). Deste modo, existe um impacto inerente na escolha das melhores respostas ao atribuir maior peso aqueles que estão inicialmente no topo.
- (3) Por último, é também relevante realçar alguns dos pontos que após realizadas as análises finais consideramos como sendo possibilidades de alcançar resultados mais variados. A utilização de um Entity Linker capaz de ser ainda mais sensível à utilização de verbos em diferentes contextos é umas destas possibilidades. De facto, um exemplo simples onde podemos verificar esta necessidade é o seguinte "How was

the Neverending Story film received?" cujas respostas esto claramente relacionadas com o filme Neverending Story contudo escapam ao intuito da pergunta. Uma outra possibilidade  a utilizao de um dataset obtido por outros mtodos como por exemplo o Bert (com T-5 Canard) ou ainda recorrer a um modelo de Text Sumarization que tenha tambm em conta a pergunta de modo a que a sumarizao final possa mais facilmente estar relacionada com a mesma.

REFERENCES

- [1] Text Retrieval Conference. 2019. . <https://trec.nist.gov/data/cast/2019qrels.txt>
- [2] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*.