

Web Semântica: Relatório do Projeto

Luís Duarte - 2017234316 e Rúben Alves - 2012142647

I. INTRODUÇÃO

Affective Computing consiste numa área da Inteligência Artificial respetiva ao reconhecimento, interpretação, processamento e simulação dos afetos humanos. Como tal, inúmeras experiências têm sido efetuadas nesta área sendo o reconhecimento de sentimento e de emoções as mais exploradas ao longo do tempo.

Uma das áreas com mais impacto nessa exploração é a Web Semântica que permite a ligação e fácil acesso ao inúmero conteúdo presente na Web. Desta forma conseguimos uma expansão significativa na pesquisa de conteúdo com carácter emotivo que se tornou especialmente relevante com o aparecimento de *Chatbots* e novas maneiras de Interação Humana com o Computador. Devido a estas duas áreas e a sua expansão decidiu-se realizar este projeto, englobando-as num domínio com grande conteúdo como o dos filmes que se encontra amplamente disponibilizado e espalhado pela Web. Um ponto importante a retirar é que as maiores fontes de dados não fornecem qualquer tipo de informação sobre o carácter emotivo e sentimental deste levando por vezes ao utilizador escolher conteúdos em que nada apela ao seu estado emotivo corrente.

Este projeto consiste numa aplicação web que tenta alterar este paradigma e tem como objetivo principal a aplicação de conhecimentos da área da Web Semântica e *Affective Computing* como base de construção de um motor de busca de filmes, onde um determinado utilizador pode pesquisar através das emoções associadas aos mesmos.

A. Âmbito e Esboço do Documento

Este documento descreve o desenvolvimento de todo o projeto realizado no âmbito da disciplina de Web Semântica da Universidade de Coimbra, encontrando-se dividido em diversos capítulos para melhor compreensão de todas as etapas percorridas.

O **primeiro capítulo** apresenta uma introdução ao projeto e todas as áreas de conhecimento envolvidas no mesmo.

No **segundo capítulo** é abordada, com detalhe, toda a literatura relevante estudada para o auxílio do desenvolvimento do projeto.

O **terceiro capítulo** descreve toda a arquitetura da aplicação web e recursos utilizados na construção da mesma.

No **quarto capítulo** são descritos todos os processos utilizados na implementação do projeto.

O **quinto capítulo** mostra todas as experiências realizadas após o desenvolvimento do projeto.

Por ultimo, no **sexto capítulo** é realizada a conclusão dos resultados obtidos.

II. TRABALHO RELACIONADO

Nesta secção é abordada a bibliografia analisada referente ao contexto do trabalho e dos temas de Semântica Web que ao longo do desenvolvimento deste trabalho serviu como guia, especialmente na fase inicial do trabalho e do seu planeamento.

A. *Ontology Based Opinion Mining for Movie Reviews*

Este artigo [2] é do ano de 2009 mas, apesar da sua idade, constituiu uma base sólida para o desenvolvimento do projeto devido a diversas semelhanças com o mesmo. Consiste na extração de opiniões (*opinion mining*) e análise de sentimentos (*sentiment analysis*) com base numa ontologia sobre filmes, no qual referem a utilização da *Movie Ontology*. O capítulo 2 deste artigo consiste numa explicação detalhada sobre *opinion mining*, contendo diversos processos sobre como este tem sido efetuado. Apesar de alguns pontos essenciais do capítulo se encontrarem desatualizados, devido a avanços feitos na área de *sentiment analysis* continuam a ser válidos na atualidade. O tema de *feature level opinion mining* também levou avanços significativos desde a escrita do artigo.

Depois desta explicação o autor aborda a construção do seu modelo de classificação de sentimento que numa nota pessoal consideramos ser a parte mais importante deste artigo. Este modelo é focado em quatro pontos essenciais, sendo eles:

- **Data Pre-processing:** são efetuados métodos de *word segmentation* e de *POS tagging* à frase ou crítica submetida ao modelo
- **Feature Identification:** como o nome indica é a identificação das *features*, normalmente **n-gramas**, sendo que a identificação tem como ajuda a ontologia
- **Polarity Identification:** Cálculo da polaridade da frase com recurso às *features* previamente selecionadas.
- **Sentiment Analysis:** Atribuição de um sentimento geral dentro das categorias inicialmente especificadas.

O importante a retirar deste modelo é como o autor atribui a importância a **n-gramas** como *features* relevantes, devido a estas normalmente não existirem em dicio-

nários formais, onde com recurso à informação presente na ontologia é possível fornecer informação adicional e melhorar a *performance* final. O resto do artigo aborda a construção deste modelo e dos resultados obtidos, no qual obtiveram-se níveis de cerca de 70% no cálculo de *accuracy* do sentimento e as conclusões retiradas de todo o processo.

Em suma, a importância deste artigo deveu-se à maneira de como o autor obtém informação adicional para o seu sistema, com conteúdo da ontologia, de modo a obter melhores classificadores.

B. The Emotion Ontology: Enabling Interdisciplinary Research in the Affective Sciences

O artigo [4] aborda a construção de uma ontologia para as ciências afetivas, nomeadamente para a designação de emoções ou mais concretamente em estados de emoções, a *Emotion Ontology (EMO)*. Começa com uma introdução ao tema onde são apresentadas diversas aplicações relacionadas no domínio da saúde. De seguida, o artigo introduz um modelo de ontologia semelhante a usar no projeto e uma breve descrição de como pode ser aplicada em casos práticos.

A parte crucial deste artigo é mesmo a disponibilização da ontologia e das suas relações entre entidades, de uma forma gráfica que apesar não ser descrita no trabalho as imagens permitem facilmente perceber como a ontologia está disposta e verificar as várias relações entre estímulos e estado emotivo. Apesar do artigo ser relativamente curto, tem relevância, pois a ontologia serve como base e alternativa para a utilização para o trabalho a desenvolver.

C. Combining and relating ontologies: an analysis of problems and solutions

Outro artigo abordado [3], serve de ponte entre os dois artigos previamente falados [2] [4]. Este começa com uma breve introdução do tema da Semântica Web e das suas aplicações, bem como alguns desafios e problemas que precisam de ser resolvidos. Após este ponto o autor começa por descrever diversos termos importantes das ontologias da Semântica Web como *combining* e *aligning* de maneira a definir que cada um destes corresponde a uma operação distinta e não uma operação única visto que normalmente são confundidos. O mais importante a retirar é o termo *combining* que é a combinação de duas ou mais ontologias para uma tarefa em que a sua relação é relevante.

Por fim, este artigo é relevante para o nosso trabalho a realizar pois introduz os termos de junção de ontologias de uma maneira simplificada e apresenta os diversos problemas e soluções possíveis que podemos vir a enfrentar fornecendo assim alternativas para os resolver.

D. Notas finais

A bibliografia analisada serviu como base ao projeto desenvolvido pois o culminar deste três artigos nomeadamente a utilização de uma ontologia de um domínio

de filmes para busca de informação, efetuar processos de *sentiment analysis* e por fim a população de uma ontologia de um domínio afetivo com toda a informação, foram as guias seguidas.

III. ARQUITETURA DO SISTEMA

O Sistema desenhado inicialmente sofreu algumas alterações devido a complicações que surgiram no decorrer do desenvolvimento do projeto. O ponto principal destas alterações foi o abandono da ontologia *Movie Ontology* e a preferência pela utilização do conteúdo disponível na **DBPedia**. No seu estado final a arquitetura do projeto ficou no estado descrito na figura 1

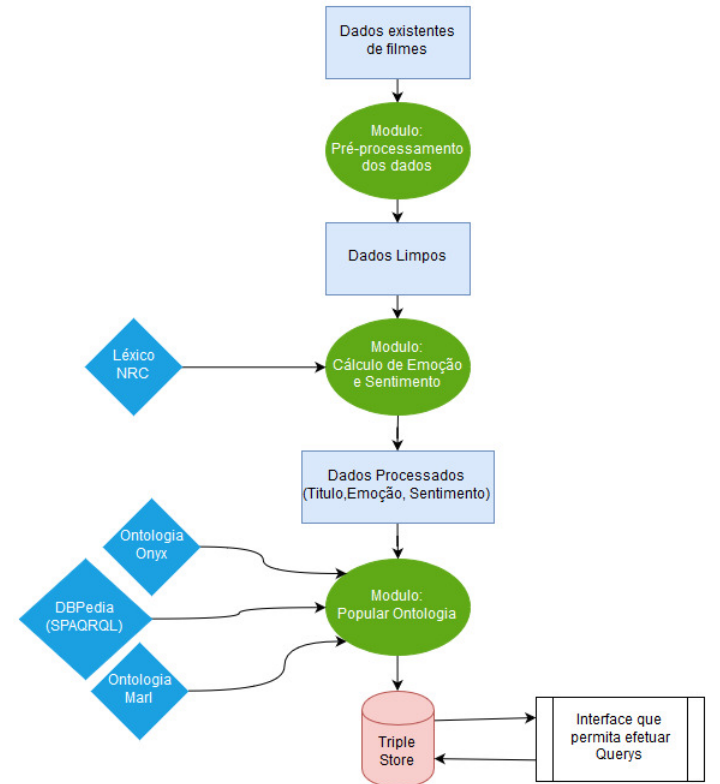


Fig. 1. Diagrama da arquitetura do sistema desenvolvido

O processo do sistema começa por um processamento dos dados retirados do **Kaggle**¹ nomeadamente na remoção e formatação de colunas não relevantes para os objetivos pretendidos visto que alguma ainda vinham no formato **JSON** e ainda a eliminação de linhas que contenham parâmetros relevantes a **Null**.

De seguida estes dados limpos serão aplicados ao módulo de cálculo de emoção e sentimento com recurso ao léxico **NRC**² sendo mais conhecido por **EmoLex** onde resulta um conjunto de dados que contem o **título**

¹<https://www.kaggle.com/rounakbanik/the-movies-dataset/home>

²<https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm>

do filme, sinopse, sentimento associado à sinopse, emoção associada à sinopse. O Processo de cálculo de Sentimento e da Emoção é depois abordada na secção IV.

Estes dados são posteriormente utilizados no módulo de população de ontologia onde através da biblioteca **RDFLib** são efetuadas *queries* à **DBPedia** onde é procurado o **URI** correspondente ao recurso que contenha o nome equivalente ao título do filme. Este módulo também utiliza as ontologias **Marl** e **Onyx** para efetuar a população das ontologias sendo estas guardadas num *triple store* para depois serem efetuadas *queries* agora com a informação gerada com todo este processo. Por fim, a *interface* foi desenvolvida com recurso à *framework* **Django**.

IV. IMPLEMENTAÇÃO

Os módulos descritos anteriormente na secção III aplicam vários métodos de **Processamento de Linguagem Natural** e de **Semântica Web** para que sejam capazes de atingir os seus propósitos. O módulo correspondente ao pré-processamento de dados não tem muita relevância, pois não aplica nenhum destes métodos e apenas faz uma formatação e limpeza dos dados iniciais.

O módulo de cálculo de emoção e de sentimento é de todos o mais relevante pois é aqui que é gerada a informação, que é aplicada ao longo de todo o projeto e que aplica várias técnicas importantes. Antes de abordarmos o módulo em si, precisamos de compreender como o léxico **NRC** funciona. O léxico consiste num ficheiro composto por **14180** palavras de língua Inglesa, onde para cada palavra estão os valores binários de sentimento e de emoção, sendo esta dividida em colunas onde se a palavra corresponde à categoria o valor **1** indica essa relação, caso contrário será **0**. As categorias são **Positiva**, **Negativa** relativas ao sentimento, e as oito emoções presentes na roda de Plutchik [1] sendo estas **Raiva**, **Antecipação**, **Nojo**, **Medo**, **Felicidade**, **Tristeza**, **Surpresa** e **Confiança**. É também importante realçar que uma palavra pode conter várias emoções associadas, ou mesmo nenhuma e no caso do sentimento se uma palavra não é assinalada nem como positiva ou negativa, decidimos tomar a abordagem de declarar esta como tendo um sentimento associado como neutro.

Para os processos de **Processamento de linguagem natural** e devido a termos optado no final da utilização do *Python* utilizamos a biblioteca **NLTK**, e ainda de uma biblioteca que permite a utilização do **Stanford-corenlp** para *Python*. Desta forma optamos por uma abordagem em relação ao tratamento do texto das sinopses onde inicialmente dividimos o texto em frases sendo aqui aplicado o *tokenizer* fornecido pelo **NLTK**. Para cada frase é feita um tratamento de todas as letras para minúsculas devido a ser este o formato das palavras presentes no **NRC** e é feita ainda uma *tokenização* para cada palavra presente na frase e um **PoS tagging** para cada um destes *tokens*. Aqui cada

palavra, ou para ser mais exato cada *token* é feita a verificação se é uma *stopword* sendo aqui utilizado a lista fornecida pelo **NLTK** para a língua inglesa, e caso seja é ignorada da frase.

Após a verificação se é *stopword* é feita a análise se nos encontramos perante um *token* na sua forma negada, e é verificado se os **uni-gramas**, **bi-gramas** ou **tri-gramas** anteriores contêm a existência de um **not**, **no** ou **'n**. Por fim é efetuado a verificação se o *token* corresponde a uma palavra ou verbo (visto que já foi efetuado o processo de **PoS tagging**) e caso seja é feita uma lematização do *token*. Estando as frases todas da sinopse processadas é então aplicado cada *token* a uma comparação com as palavras existentes no léxico. Caso o *token* se encontre no léxico é apontado os valores de sentimento e de emoção associada. Após este processo ser efetuado para todos os *tokens* é então feito o cálculo do sentimento e da emoção através das categorias com mais ocorrência, sendo que o sentimento neutro é um caso em que a quantidade de valores totais de palavras positivas é igual ao de palavras negativas. Por fim pode acontecer o caso de haver sinopses cujas palavras não se encontram no léxico no qual atribuímos o sentimento e de emoção como neutro.

Para o módulo de popular ontologia processos de **Semântica Web** foram usados, mais concretamente pesquisas **SPARQL** à **DBPedia**. Para esta parte do trabalho foi usada a biblioteca **RDFLib** que através do formato *Graph* permite armazenar a informação em *triple stores*. Com os dados obtidos do processos de atribuição de emoção e sentimento temos um conjunto de dados onde está incluído o título do filme, e verificando as ontologias utilizadas, a **Marl** e a **Onyx** verificamos que estas já têm uma classe para indicar o recurso a que a análise efetuada é dedicada. Devido à existência desta classe decidimos como já mencionado anteriormente por abandonar a utilização de uma terceira ontologia, a *Movie Ontology* pois indicando o recurso da **DBPedia** era possível obter a informação existente sobre o filme. De maneira a obter o recurso sobre o respetivo filme é efetuada uma *query* ao *endpoint* da **DBPedia** que ficou definida como a apresentada na representação 1

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?film WHERE {
    ?film rdf:type <http://dbpedia.org/ontology/Film>.
    ?film rdfs:label ?film_name.
    FILTER (LANG(?film_name)="en"
    && ?film_name="Título do Filme"@en}
```

Listing 1: Query SPARQL efetuada

Com a *query* efetuada caso exista algum filme com aquele título presente na **DBPedia** irá devolver algo semelhante ao apresentado em 2

```
{'head': {'link': [], 'vars': ['film']}, 'results':
  {'distinct': False, 'ordered': True,
   'bindings': [{'film': {'type': 'uri', 'value':
     'http://dbpedia.org/resource/Toy_Story'}}]}}
http://dbpedia.org/resource/Toy_Story
```

Listing 2: Output obtido

Extraído o **URI** do filme conseguimos então proceder à população do *triple store* com a informação gerada no processo anterior. O **RDFLib** tem vários métodos de criação de um *triple store* onde o *SleepyCat* é a designação dada para um armazenamento no disco. Para cada filme são criados grafos e são adicionados ao *triple store* no formato (**URI do Filme, Predicado, Valor**) sendo que aqui o **Predicado** corresponde à ação dentro da ontologia da informação adquirida sendo que aqui as mais importantes são as de **hasEmotion** e **hasSentiment** que como o nome indica é a atribuição da emoção e do sentimento do filme do qual obtivemos o recurso. Por fim o estado resultante do processo de população do *triple store* resultará em algo semelhante ao apresentado em 3

```
xmlns:
rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
<rdf:Description rdf:about=
"http://dbpedia.org/resource/Muppet_Treasure_Island">
  <ns1:hasSentimentm rdf:resource=
    "http://purl.org/marl/ns#Neutral"/>
  <ns2:hasEmotion> Anger </ns2:hasEmotion>
  <ns1:opinionText> After telling the story of
    Flint\'s last journey to young Jim Hawkins,
    Billy Bones has a heart attack and dies
    just as Jim and his friends are attacked by
    pirates. The gang escapes into the town
    where they hire out a boat and crew to find
    the hidden treasure, which was revealed by
    Bones before he died. On their voyage
    across the seas, they soon find out that
    not everyone on board can be trusted.
  </ns1:opinionText>
  <ns2:emotionText> After telling the story of
    Flint\'s last journey to young Jim Hawkins,
    Billy Bones has a heart attack and dies
    just as Jim and his friends are attacked by
    pirates. The gang escapes into the town
    where they hire out a boat and crew to find
    the hidden treasure, which was revealed by
    Bones before he died. On their voyage
    across the seas, they soon find out that
    not everyone on board can be
    trusted.</ns2:emotionText>
</rdf:Description>
```

Listing 3: Secção do triple store resultante

Com este *triple store* criado podemos assim efetuar *queries* à procura de informação sobre filmes e o seu estado emotivo associado. Devido a alguma instabilidade da biblioteca utilizada **RDFLib** a *query SPARQL* à *triple store* criada não devolve conteúdo nenhum, onde para

tentar resolver este problema optou-se pelo carregamento da *store* para o formato de *Graph* e de seguida percorrer este à procura da informação desejada pelo utilizador. Apesar desta solução ser menos eficiente do que um *query SPARQL* direta, não deixa de ser uma solução temporária que permite à mesma devolver a informação desejada pelo utilizador.

Para as pesquisas à *triple store* implementou-se algumas funções de maneira a facilitar a pesquisa. Considerando que o utilizador irá efetuar a pesquisa com recurso a texto através de uma pequena barra de pesquisas optamos por fazer um pequeno processamento do texto.

A *query* é transformada em *tokens* e para cada um destes é feita uma pesquisa pelos sinónimos da palavra. Para cada um dos sinónimos do *token* (importante referir que o próprio *token* é contabilizado) é feito um cálculo de similaridade (com recurso à biblioteca **NLTK**) **Wu Palmer**, para os sinónimos das emoções e dos sentimentos existentes no *triple store*. Desta forma o utilizador poderá pesquisar por filmes que provoquem comédia e alegres que desta forma a *query* será interpretada e efetuada como uma pesquisa por filmes pertencentes à categoria de "**Feliz**" e "**Positivos**".

Por fim foi desenvolvida uma interface *web* para facilitar a procura de informação dentro da *triple store* criada, no qual aplicou-se a *framework Django* devido à sua facilidade e de já estarmos familiarizados.

Devido ao **Django** criar muitos ficheiros, a explicação destes será muito extensa, no qual iremos mencionar aqueles mais importantes. O ficheiro **views.py** consiste na lógica existente na interface *web*, e é aqui que é utilizado os nossos métodos de pesquisa e consequente pesquisa no *triple store*. O **urls.py** é para a especificação do **URL** da interface bem como a definição das páginas **HTML** e **CSS** a aplicar. Por fim o ficheiro **manage.py** é responsável por iniciar ou terminar a aplicação, no qual ficará disponível através de qualquer *browser* no endereço: **127.0.0.1/WebSemantica/**.

De seguida segue-se casos de utilização da aplicação e a disponibilização dos resultados efetuados.

V. EXPERIÊNCIAS REALIZADAS

Esta secção será dedicada aos testes efetuados ao projeto desenvolvido. Os processos iniciais efetuados correspondem à leitura, processamento, e armazenamento dos dados correspondentes às sinopses dos filmes. O ficheiro **newData.csv** corresponde aos dados retirados do **Kaggle** que sofreram uma limpeza porévia, nomeadamente a remoção de colunas não relevantes e remoção de campos a *null*, entre outros. Estes dados são então aplicados ao módulo de cálculo de sentimento e de emoção como demonstra a figura 2

```

1
2
3 When Tessa's father dies, Lewis contacts her late mother's wealthy family so they'll take care of Tessa and her sisters.
4
5
6 Lewis becomes taken with Tessa's haughty cousin Florence and the two soon marry and head off for Florence's estate in England.

```

Fig. 2. Processo de Cálculo de Sentimento e de Emoção

Por fim os dados resultantes deste processo são armazenados num ficheiro **data_emotion.csv**, onde ficam com o formato apresentado na figura 3

	A	B	C	D
1	Toy Story	Led by Woody, Andy's toys live happily in his room until Andy's birthday brings Buzz Lightyear onto the scene.	Neutral	Anticipation,
2				
3	Jumanji	When siblings Judy and Peter discover an enchanted board game that opens the door to a magical world, they	Negative	Fear,
4				
5	Grumpier	A family wedding reignites the ancient feud between next-door neighbors and fishing buddies John and Max.	Neutral	Disgust,
6				
7	Waiting to Be Cheated on,	mistreated and stepped on, the women are holding their breath, waiting for the elusive "good m	Positive	Joy,
8				
9	Father of	Just when George Banks has recovered from his daughter's wedding, he receives the news that she's pregnant	Neutral	Anger,
10				
11	Heat	Obsessive master thief, Neil McCauley leads a top-notch crew on various insane heists throughout Los Angeles	Negative	Anger,
12				
13	Sabrina	An ugly duckling having undergone a remarkable change, still harbors feelings for her crush: a carefree playbo	Negative	Trust,
14				
15	Tom and I	A mischievous young boy, Tom Sawyer, witnesses a murder by the deadly Injun Joe. Tom becomes friends wit	Neutral	Anticipation,
16				
17	Sudden D	International action superstar Jean Claude Van Damme teams with Powers Boothe in a Tension-packed, suspe	Negative	Fear,
18				

Fig. 3. Resultado final do ficheiro data_emotion.csv

Passando para a *interface web* criada, o resultado é o apresentado na figura 4 onde utilizador selecciona as emoções que pretende encontrar na sua pesquisa.

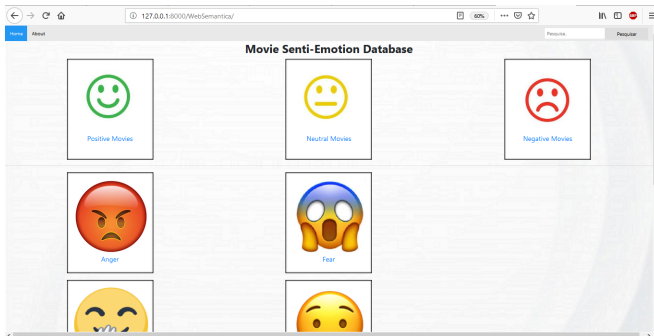


Fig. 4. Homepage da interface web

A *homepage* é constituída por ícones respetivos às categorias existentes de sentimento (Positivo, Neutro ou Negativo) e de emoção com os pares de opostos definidos na roda de emoções de Plutchik (Raiva/Medo, Antecipação/Surpresa, Felicidade/Tristeza, Confiança/Nojo). Clicando num dos ícones a aplicação redireciona o utilizador para a listagem dos filmes, que contenham aquela emoção ou sentimento como demonstrado na figura 5 que apresenta os resultados dos filmes com sentimento negativo.

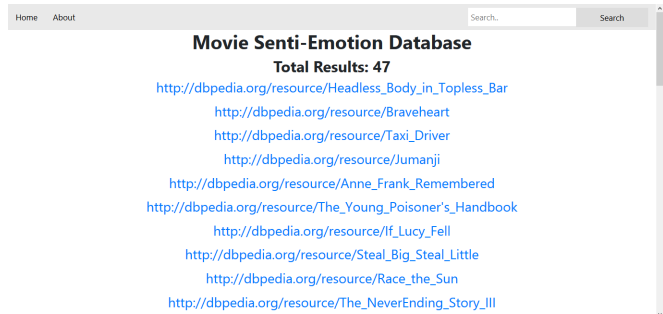


Fig. 5. Resultados de filmes negativos na interface web

A *interface web* apresenta ainda um método de pesquisa, introduzindo texto e através do módulo de pesquisa descrito na secção anterior devolve os filmes com emoções semelhantes às introduzidas pelo utilizador. Na figura 6 temos o exemplo dessa pesquisa, que através do **URL** verificamos que foi efetuada uma pesquisa por *Sorrowfull movies* que traduzindo é uma pesquisa por filmes tristes, lembrando que a emoção de tristeza associada aos filmes é de "*Sadness*" logo são termos diferentes para a mesma emoção. Para esta caso as emoções extraídas desta *query* foram as de **Sadness** e de **Anger**, bem como o sentimento de **Negativo**, sendo visível na figura 7.

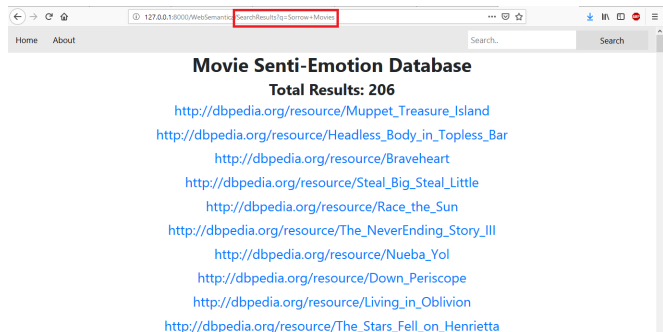


Fig. 6. Resultados da query na interface web

```

Terminal
+ movie
x 0.5
Neutral
movie
0.5333333333333333
Negative
['Anger', 'Sadness']

```

Fig. 7. Output do módulo correspondente à pesquisa

Estas *queries* também estão desenvolvidas de maneira a extrair o significado da frase introduzida como descrito

anteriormente, logo a introdução de palavras ou textos sem significado não terão resultados como demonstrado na figura 8.

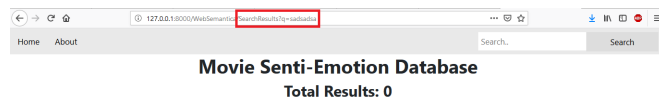


Fig. 8. Resultados de uma query incorreta na interface web

O conteúdo de cada filmes pode ser visto selecionando o *link* de o filme desejado, onde o utilizador será levado à página do recurso da **DBPedia** contendo todas as informações sobre este, como visível na figura 9.

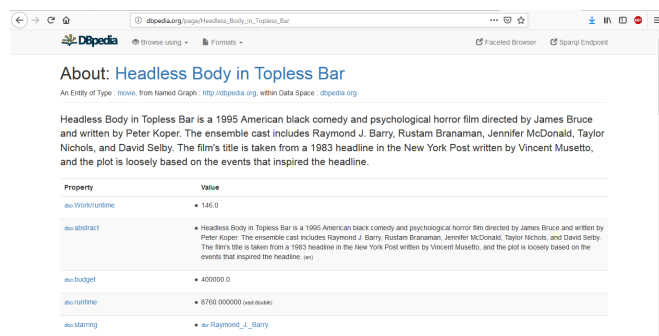


Fig. 9. Detalhe de um filme da DBPedia

Por fim é importante referir que toda a interface poderá sofrer alterações nomeadamente a construção de uma página de detalhe de um filme específico em vez de disponibilizar a ligação direta para a **DBPedia**. Por fim a ação **About** presente no canto superior direito leva o utilizador a uma página onde está descrito os autores do projeto.

VI. CONCLUSÕES

Após a realização deste trabalho, os conceitos principais da **Web Semântica** ficaram solidificados e ampliados em relação ao funcionamento da Web. O processo de extração de sentimento e de emoção permitiu a aplicação de vários métodos de **Processamento de Linguagem Natural** e verificamos que existem bastantes bibliotecas que suportam a classificação de sentimentos mas não de emoções. Também observamos que estas facilitam bastante os processos de *tokenização*, *lemmatização* e de *PoS tagging* poupando desta forma, muito trabalho.

A nível das ontologias apesar da ideia inicial ser a junção da **Onyx** e da **Marl** verificamos que a existência da entidade que faça a ligação a um recurso facilitou imenso o trabalho, bem como a pequena dimensão destas ontologias que suavizou todo o processo de população. Por fim a utilização do *Python* revelou-se um enorme

desafio visto que a utilização do **RDFLib** não é fácil e tendo em conta a documentação rudimentar da mesma, bem como a existência de alguns *bugs* dificultou o nosso trabalho, nomeadamente a nível das *queries SPARQL* à *triple store* criada.

No fundo estes problema não foram um impedimento mas a utilização de *Java* e do **Jena** possivelmente teria facilitado esta parte do trabalho. Por fim percebemos a complexidade de **Processamento de Linguagem Natural** que está por detrás dos motores de pesquisa, onde aplicamos processos complexos de maneira a tentar chegar ao que é pretendido pelo utilizador, onde os problemas de má ortografia, ironia e aplicação de calão dificultam imenso a extração de informação correta do pretendido pelo o utilizador.

Concluindo verificamos que desenvolver este projeto foi trabalhoso, mas gratificante pois conseguimos aplicar quase todos os processos aprendidos ao longo desta Unidade Curricular.

REFERENCES

- [1] Plutchik, Robert. "The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice." American scientist, volume 89, pages 344 to 350. 2001.
- [2] Zhao, Lili, and Chunping Li. "Ontology based opinion mining for movie reviews." International Conference on Knowledge Science, Engineering and Management. Springer, Berlin, Heidelberg, 2009.
- [3] Klein, Michel. "Combining and relating ontologies: an analysis of problems and solutions." IJCAI-2001 Workshop on ontologies and information sharing. 2001.
- [4] Hastings, Janna, et al. "The emotion ontology: enabling interdisciplinary research in the affective sciences." International and Interdisciplinary Conference on Modeling and Using Context. Springer, Berlin, Heidelberg, 2011.