UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

DESCOBERTA AUTOMÁTICA DE TEMAS UTILIZANDO LEGENDAS

Nuno José Santos de Almeida Neves da Silva

PROJECTO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2012

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

DESCOBERTA AUTOMÁTICA DE TEMAS UTILIZANDO LEGENDAS

Nuno José Santos de Almeida Neves da Silva

PROJECTO

Trabalho orientado pelo Prof. Doutor Thibault Nicolas Langlois

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2012

Agradecimentos

Aos pais, família, amigos, professores e colegas por todo o apoio ao longo da vida académica.

Nesta fase que decorreu estes últimos meses cruzei-me com várias pessoas a quem quero agradecer:

Ao meu orientador, Professor Thibault Langlois, por todo o acompanhamento que foi sempre excelente.

Às Professoras Teresa Chambel e Paula Carvalho por terem ajudado com opiniões importantes.

Aos colegas da sala 6.3.33 pelo companheirismo que fez com que houvesse sempre um bom ambiente de trabalho ao longo do ano.

Aos colegas do projecto VIRUS, André, Eduardo, Jorge, Marta, Nuno e Pedro por directa ou indirectamente contribuírem para este trabalho.

Aos amigos que responderam à entrevista feita para avaliar o trabalho, Ana, Bruno, Cátia, Eurico, Jeferson e Vanessa. E pela mesma razão, um *thank you very much* ao *staff* do *Wall Street Institute* de Almada, em especial à Ana, Mara, Sheila e Tim.

À colega e amiga Ana Teixeira, que com toda a ajuda ao longo do ano fez com que este trabalho ficasse mais rico.

Ao LaSIGE e FCT por fornecerem os meios necessários para o desenvolvimento do projecto.

Para os amantes da 7ª arte…ou simplesmente para quem goste de apreciar uma boa série.

Resumo

Este trabalho insere-se no projecto VIRUS (*Video Information Retrieval Using Subtitles*).

O projecto VIRUS tem como objectivo o desenvolvimento de um sistema de Recuperação de Informações Vídeo que irá funcionar em bibliotecas de vídeos compostas por documentos legendados. Contrastando com projectos anteriores, limitamo-nos a processar filmes e séries de televisão para as quais as legendas estão disponíveis. Aspectos diferenciais deste projecto incluem a recuperação de informação com base na análise simultânea de três fluxos de informação: sinal de vídeo, legendas e sinal de áudio. O sistema permite visualizar vídeos de forma significativa e aceita consultas do utilizador para encontrar partes dos documentos de vídeo às quais correspondem as consultas. Os domínios de aplicação de um sistema como este são vastos. Pode ser usado por profissionais da indústria cinematográfica para aceder e visualizar cenas que partilham algumas características, ou para produzir uma descrição concisa e detalhada de um filme, o que poderia ser um valioso contributo para um sistema de recomendação. Outro domínio de aplicação deste sistema é o anúncio contextual. A análise semântica das cenas fornece uma poderosa ferramenta para colocar anúncios relacionados nos documentos de vídeo.

O trabalho "DESCOBERTA AUTOMÁTICA DE TEMAS UTILIZANDO LEGENDAS" explora um dos fluxos de informação que se pretende abordar no projecto VIRUS, as legendas. O seu objectivo é desenvolver algoritmos capazes de descobrir automaticamente o tema de uma conversa e sugerir quais os temas mais relevantes. Além desse objectivo principal, há outras particularidades das legendas que podem ser analisadas e que diferenciam as séries de TV. Os textos usados foram legendas de séries como o 24, Anatomia de Grey, Os Sopranos, e muitas outras.

O trabalho foi desenvolvido em Java e os resultados que obtemos são apresentados na interface *web* do MovieClouds, o protótipo do projecto VIRUS.

Apesar do projecto ainda não estar terminado, concluímos, através de testes com utilizadores que o processamento das legendas são um excelente contributo para identificar temas nos vídeos.

Palavras-chave: Legendas, Séries, Sistema de Recuperação de Informações Vídeo, Projecto VIRUS, Temas

Abstract

This work is inserted in the VIRUS (Video Information Retrieval Using Subtitles) project.

The VIRUS project has as objective the development of a Video Information Retrieval system that will operate on video libraries composed of subtitled documents. Contrasting with previous projects, we restrict ourselves to movies and television series for which subtitles are available. Distinguishing aspects of this project include information retrieval based on the simultaneous analysis of three information streams: video signal, subtitles and audio signal. The system allows visualizing videos in meaningful ways and accepts queries from the user to find portions of video documents that match the queries. The domains of application of such a system are vast. It can be used by movie industry professionals to access and visualize scenes that share some characteristic, or to produce a concise and detailed description of a movie that could be a valuable input for a recommendation system. Another domain of application of this system is contextual advertisement. The semantic analysis of scenes provides a powerful tool for placing related advertisements on video documents.

The work "*DESCOBERTA AUTOMÁTICA DE TEMAS UTILIZANDO LEGENDAS*" exploits one of the information fluxes that we want to approach on the VIRUS project, the subtitles. The purpose of this work is to develop algorithms that would be able to automatically identify the theme of a conversation and suggest the most relevant ones. Besides this main objective, there are other particularities of the subtitles that can be observed and that differentiate the TV series. The texts we have used are subtitles from series as 24, Grey's Anatomy, The Sopranos and many others.

The work has been developed in Java and the results we obtained are shown in the web interface of MovieClouds, the prototype of the VIRUS project.

The results are still preliminary, however we can conclude, by testing with users, that subtitles processing is an excellent contribution to identify themes in videos.

Keywords: Series, Subtitles, Theme, Video Information Retrieval system, VIRUS project

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Este capítulo introduz o projecto, inserido na cadeira de Projecto de Engenharia Informática. Para além da motivação para construir esta solução, são apresentados os objectivos do projecto, as contribuições deste, a instituição onde foi desenvolvido, e por fim, a organização do presente documento.

1.1 Motivação

Numa altura em que os consumidores de filmes e séries estão cada vez mais exigentes, começa a ser muito importante ter bons mecanismos de recomendação para ajudar o utilizador a não perder tempo a ver algo que, em princípio, não irá apreciar. E se mesmo assim o utilizador tiver dúvidas da recomendação é interessante a ideia de poder fazer consultas detalhadas sobre características do filme ou série.

Além do consumidor de informação, também os profissionais da indústria de cinema podem ser beneficiados com sistemas que lhes permitem aceder a cenas com características idênticas para efectuar alguma comparação.

1.2 Objectivos

O trabalho tem como objectivo principal, descobrir automaticamente o tema de uma conversa e sugerir quais os temas mais relevantes. Além desse objectivo principal tínhamos em mente poder explorar várias particularidades das legendas que podiam ser utilizadas também para distinguir séries.

1.3 Contribuições

Este trabalho contribui para perceber que informações podem ser retiradas das legendas e qual a melhor maneira para o fazer. Para isso foram estudadas várias técnicas que já contribuíram, em parte, para a publicação do artigo "Going Through the Clouds: Search Overviews and Browsing of Movies". Esta publicação vai ser apresentada em Outubro na conferência "Academic MindTrek 2012", em Tampere, Finlândia.

1.4 Enquadramento institucional

Este trabalho foi desenvolvido no LaSIGE (Laboratório de Sistemas Informáticos de Grande Escala), uma unidade de investigação associada ao Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa (FCUL).

A equipa do LaSIGE é composta aproximadamente por 100 pessoas, entre doutorados, alunos de doutoramento e de mestrado, entre outros investigadores.

Hoje em dia esta unidade está separada em alguns grupos e o projecto VIRUS está a ser desenvolvido pelo grupo HCIM (*Human-Computer Interaction and Multimedia*).

1.5 Estrutura do documento

Este documento está organizado da seguinte forma:

- Capítulo 2 – Neste capítulo apresentam-se em pormenor os *objectivos* do trabalho, o contexto em que ocorreu, a metodologia utilizada no seu desenvolvimento, bem como o planeamento efectuado para o concretizar e é apresentada uma confrontação com o plano de trabalho inicial analisando as razões dos desvios ocorridos;
- Capítulo 3 – Neste capítulo é apresentado o trabalho relacionado;
- Capítulo 4 – Neste capítulo é descrito detalhadamente o trabalho realizado, desde a análise do problema, passando pela implementação e terminando na avaliação do trabalho;
- Capítulo 5 – Neste capítulo são apresentadas as conclusões que retiramos sobre o projecto e também é apresentado o trabalho futuro.

Capítulo 2

Contextualização do Projecto

Neste capítulo começa-se por apresentar o contexto em que está inserido este projecto. Na segunda secção são detalhados os objectivos que guiaram o projecto. De seguida é descrita a metodologia utilizada para a realização do projecto desenvolvido e o capítulo termina com a comparação do planeamento que tínhamos com a respectiva execução.

2.1 Contexto

No âmbito do projecto VIRUS (Video Information Retrieval Using Subtitles) havia ainda áreas para serem exploradas e escolhi trabalhar com as legendas, que são uma parte integrante do projecto. Paralelamente ao meu trabalho estão a ser desenvolvidos outros que farão com que no final tenhamos um projecto como um todo, valorizado por a soma das partes.

O VIRUS começou em 2010 e tem até ao final do ano para ser realizado. Sendo o seu financiamento suportado pela Fundação para a Ciência e a Tecnologia (FCT).

O projecto VIRUS tem como objectivo o desenvolvimento de um sistema de Recuperação de Informações Vídeo que irá funcionar em bibliotecas de vídeos compostas por documentos legendados.

Fazem parte da equipa alguns Professores da FCUL e não só, tendo trabalhado em conjunto com mais proximidade com o Professor Thibault Langlois, Professora Teresa Chambel e Professora Paula Carvalho.

Como resultado deste projecto de investigação, já foram publicados mais de 10 artigos científicos ao longo destes quase 3 anos.

2.2 Objectivos específicos

Tínhamos como objectivo principal o desenvolver algoritmos capazes de descobrir automaticamente o tema de uma conversa e sugerir quais os temas mais relevantes.

A par deste objectivo, foram aparecendo outros como:

- Saber quantas palavras tem cada sequência (frases que aparecem ao mesmo tempo no ecrã) e o tempo que demoram a ser ditas;
- Saber quantas palavras são ditas, por segundo, em cada sequência;
- Saber os *stems* (radicais) de determinadas palavras existentes nas legendas;
- Permitir associar *tags* a cada sequência de legendas;
- Mostrar nuvens de palavras ordenadas, quer seja determinado pelo número de ocorrências, quer como por outros algoritmos desenvolvidos;
- Explorar exaustivamente as capacidades dos algoritmos de *stemming;*
- Estudar e utilizar a biblioteca do *WordNet,* uma base de dados lexical*:*
- Estudar e implementar o algoritmo TF*IDF (*Term Frequency – Inverse Document Frequency).*

2.3 Metodologia

Neste projecto seguimos uma metodologia ágil que foi continuamente acrescentando funções ao interpretador que desenvolvemos.

A ideia inicial foi construir um interpretador. Ao faze-lo temos um programa que lê as legendas fornecidas e com o qual podemos interagir para realizar várias operações, como pesquisar por palavras, associar *tags* às legendas, obter nuvens de palavras, etc.

Numa segunda fase, cada ideia que tínhamos era analisada, desenvolvida e testada usando o interpretador. Cada ideia desenvolvida gerou um novo comando incrementando as funcionalidades do programa.

2.4 Planeamento

De seguida são apresentadas duas tabelas, uma com o plano inicial do trabalho (Tabela 1) e outra com a calendarização (Tabela 2), bem como a explicação do que levou às diferenças entre o plano inicial e a execução das tarefas.

Tarefas	2011			2012					
	10	11	12	1	2	3	4	5	6
Pesquisa bibliográfica									
Familiarização com os algoritmos e *software* usado									
Desenvolvimento do programa utilizando as legendas									
Aplicação às legendas das séries TV									
Desenvolvimento de um protótipo com interface *web*									
Escrita do relatório									

Tabela 1: Planeamento inicial apresentado no relatório preliminar

Tarefas	2011			2012								
	10	11	12	1	2	3	4	5	6	7	8	9
Pesquisa bibliográfica												
Familiarização com os algoritmos e *software* usado												
Desenvolvimento do programa utilizando as legendas												
Aplicação às legendas das séries TV												
Desenvolvimento de um protótipo com interface *web*												
Integração com o projecto MovieClouds												
Escrita do relatório												

Tabela 2: Calendário das tarefas realizadas

Na tarefa "pesquisa bibliográfica", que estava pensada ser os primeiros 3 meses, arrastou-se ao longo de 8 meses. A justificação é fácil de entender, este projecto não foi desenvolvido em "cascata", não foram apenas os primeiros meses que serviram para pensar e desenhar o projecto, e depois implementar. Ao contrário, até Maio houve muita pesquisa bibliográfica sobre os temas em questão. Durante o desenvolvimento do projecto foram surgindo problemas que tiveram de ser ultrapassados, assim como ideias que foram amadurecendo ao longo do tempo. Um bom exemplo disso foi a leitura de vários artigos de Porter, o criador do algoritmo que utilizei e há muita literatura sobre WordNet que tinha de ser analisada para perceber o que nos podia interessar.

A tarefa "Familiarização com os algoritmos e *software* usado está muito relacionada com a tarefa da pesquisa bibliográfica, uma vez que muitos algoritmos que foram usados, eram-me desconhecidos e tiveram de ser explorados.

Para a tarefa "Desenvolvimento do programa utilizando as legendas", estava pensada ser realizada no 3º e 4º mês mas já no relatório preliminar tinha feito uma correcção ao plano, porque desde o 1º mês que estava a desenvolver o programa. E agora posso constatar que começou no primeiro e foi praticamente até ao fim, sempre intervalado com as tarefas descritas anteriormente, isto porque à medida que estudava nova bibliografia com novos algoritmos, surgiam ideias e novas dificuldades para ultrapassar, daí ter sido um desenvolvimento iterativo.

A tarefa "Aplicação às legendas das séries TV" é um pouco mais ambígua porque desde o início foram realizados testes com legendas das séries de TV, mas indico que esta tarefa começou em Maio porque foi quando começaram a ser feitos testes com outras legendas que não as que serviram de base ao desenvolvimento do programa, permitindo assim perceber a robustez do programa com legendas de séries que nunca tinham sido utilizadas.

Uma tarefa que deixou de existir e foi substituída por outra foi a "Desenvolvimento de um protótipo com interface *web*", porque o projecto VIRUS é composto por mais elementos e chegou-se à conclusão que era mais interessante integrar o meu programa já na interface utilizada no **MovieClouds** (o protótipo do projecto VIRUS) em vez de desenvolver um protótipo só para a minha tese.

Em consequência do descrito anteriormente, apareceu a tarefa "Integração com o projecto **MovieClouds**"

A "Escrita do relatório" estava pensada para ser realizada no 9º mês. Já no relatório preliminar (como este acaba por ser uma base de trabalho para a escrita do relatório final) tinha corrigido o plano em que coloquei também o 2º mês com esta tarefa. No final constato que não foi no 9º mês que escrevi o relatório, mas sim nos seguintes.

Muito importante também é deixar uma nota final sobre o atraso ocorrido. Em teoria o trabalho terminaria a 30 de Junho mas como estou inserido no projecto VIRUS e este termina apenas no final do ano, sendo que a parte que desenvolvo também tem planos até ao final do projecto, havia duas datas que tinham de ser coordenadas, a entrega da tese e a conclusão do trabalho em si.

Neste momento continuo a trabalhar no projecto e por isso tudo o que penso fazer até ao final estará descrito na secção "Trabalho Futuro". Para este documento tivemos de delinear uma fase do projecto que já tivesse resultados visíveis, embora preliminares.

Capítulo 3

Trabalho Relacionado

Ao longo dos anos, as legendas de vídeos têm sido utilizadas para alguns estudos em outras áreas, como a sua utilização para a aprendizagem de línguas estrangeiras ou a aprendizagem de línguas por crianças. Contudo, nos últimos anos, com o aumento exponencial do número de vídeos disponíveis, a importância de os conseguir catalogar automaticamente cresceu e a utilização das legendas para esse fim também já foi tratada em alguma literatura mas menos do que as outras vertentes do vídeo (áudio e visual). Uma vantagem de tratar as legendas é aproveitar todo o trabalho que já tinha sido desenvolvido para categorização de textos em geral.

No âmbito do tratamento do texto em vídeo há trabalhos que vão buscar a informação não apenas a legendas mas também à imagem do vídeo com sistemas de reconhecimento de caracteres (OCR).

Estudos publicados que utilizam apenas legendas foram sempre feitos com documentários ou noticias, com a particularidade de estes vídeos normalmente serem monólogos. Também há estudos com filmes mas que misturam as legendas com o visual ou áudio.

Na literatura presente pode-se separar em duas abordagens diferentes que são utilizadas na análise de legendas, a categorização por aplicação de técnicas de processamento de linguagem natural e a categorização por aprendizagem. Depois o que existe são pequenas variações.

No trabalho (Demirtas, Cicekli, & Cicekli, 2010) são tratadas essas duas abordagens para categorização de vídeo utilizando texto. O primeiro método é a categorização do vídeo por aplicação de técnicas de processamento de linguagem natural em legendas de vídeo e utiliza o WordNet (Miller, 1995). O método é baseado num algoritmo já existente (Katsiouli, Tsetsos, & Hadjiefthymiades, 2007) de categorização de vídeo, com algumas alterações. A segunda abordagem é a categorização por aprendizagem, tem os mesmos passos para extrair informação, mas realiza a categorização usando um módulo de aprendizagem.

De acordo com o algoritmo da primeira abordagem, o vídeo é catalogado com uma categoria utilizando a base de dados lexical WordNet e os recursos do WordNet Domains (Bentivogli, Forner, Magnini, & Pianta, 2004), bem como a aplicação de técnicas de processamento de linguagem natural nas legendas. No segundo algoritmo a categorização é feita por aprendizagem. Um módulo de aprendizagem é implementado, o qual pode ser treinado usando vídeos de categorias conhecidas. O algoritmo é iniciado com as etapas de pré-processamento do primeiro algoritmo e a categorização é executada pelo módulo de aprendizagem.

O método para a extracção de domínios do WordNet começa com "pré-processamento de texto". Nesta etapa, as frases das legendas são divididas, as palavras em cada frase são marcadas com *tags* de "*part of speech*" (POS) (Toutanova & Manning, 2000) e as *stop words* são removidas. O texto processado passa a um módulo de "extracção de palavras-chave" que encontra palavras-chave no texto dado. Uma vez que as palavras-chave podem ter mais do que um significado, o módulo "desambiguação do sentido da palavra" encontra o sentido correcto. Depois o módulo "extracção de domínios do WordNet" localiza os domínios do WordNet das palavras-chave correspondentes aos seus sentidos correctos. Além dos domínios este módulo considera o título do vídeo na categorização, já que os títulos podem dar pistas importantes sobre a categoria.

Para seleccionar as palavras mais importantes das legendas para classificar o vídeo, é utilizado um algoritmo de selecção de palavras-chave, naquele caso o TextRank (Mihalcea & Tarau, 2004). Este algoritmo baseia-se num grafo que representa o texto e aplica-se um algoritmo de classificação para os vértices do grafo (as palavras). Dois vértices são conectados se tiverem uma relação de co-ocorrência. Dois vértices co-ocorrem se tiverem dentro de uma janela máxima de N palavras, onde N pode ser definido como um valor entre 2 e 10. Na implementação de (Demirtas, Cicekli, & Cicekli, 2010) N é definido como 2. Depois de construir o grafo, um algoritmo de classificação baseado no grafo, derivado do algoritmo PageRank (Page, Brin, Motwani, & Winograd, 1998), é usado para decidir a importância de um vértice. A ideia básica do algoritmo é a "votação": quando um vértice se liga a outro, dá um voto a esse vértice. Depois de calcular a pontuação de cada vértice, estes são classificados com base nos seus resultados e os vértices do Top T são seleccionados como palavras-chave. Geralmente, T é definido a um terço do número de vértices no grafo.

A desambiguação do sentido da palavra (WSD) é a tarefa de determinar o sentido correcto de uma palavra num texto e é usado para encontrar os sentidos correctos das palavras-chave. O algoritmo utilizado (Banerjee & Pedersen, 2002) é uma adaptação do algoritmo de Lesk baseado no dicionário. O algoritmo adaptado usa WordNet para incluir as definições das palavras que estão relacionadas com a palavra que está a ser desambiguada através de relações semânticas, como hiperónimo, hipónimo, holónimo, merónimo, tropónimo, e atributo de cada palavra. Isso fornece uma fonte rica em informações e aumenta a precisão de desambiguação. O algoritmo compara as definições entre cada par de palavras num certo contexto. Estas definições são associadas ao *synset*, hiperónimo, hipónimo, holónimo, merónimo, tropónimo, e atributo de uma outra palavra. Por exemplo, a definição de um *synset* de uma palavra pode ser comparada com a definição de um hiperónimo de outra.

Ao utilizar a catalogação por domínio associado ao WordNet, os *synsets* do WordNet são anotados com pelo menos uma etiqueta de domínio, usando um conjunto de cerca de 200 etiquetas hierarquicamente organizadas.

Para atribuir uma categoria a um documentário, o mapeamento é definido entre as etiquetas das categorias e os domínios do WordNet. Primeiro, os sentidos relativos a cada etiqueta da categoria foram obtidos pelo WordNet através de relações hiperónimos e hipónimos e os domínios do WordNet correspondentes aos sentidos de cada etiqueta da categoria são obtidos. Para cada categoria, a pontuação de ocorrência dos domínios derivados é calculada e são classificados por ordem decrescente de ocorrência.

A segunda abordagem para categorização automática de vídeos baseados nas legendas é por aprendizagem. O algoritmo apresentado por (Demirtas, Cicekli, & Cicekli, 2010) usa um mecanismo de aprendizagem para catalogar as categorias dos vídeos. As etapas de pré-processamento do algoritmo são as mesmas que as utilizadas na forma de catalogação que vimos anteriormente. No entanto, a fase seguinte inclui um passo de aprendizagem. Quando um vídeo está a ser catalogado, a distribuição do domínio do vídeo é comparada com a distribuição de domínios aprendidos de categorias e a categoria mais semelhante é atribuída.

Na fase de aprendizagem da distribuição de categoria do domínio, documentários com categorias conhecidas são usados como um conjunto de treino. Primeiro de tudo, as legendas dos documentários pertencentes a uma categoria específica são processadas utilizando o módulo de extracção de domínios do WordNet utilizando o WordNet Domains. Assim, os domínios e as pontuações das ocorrências do domínio de uma categoria são colectados

A fim de determinar a distribuição de domínios por categoria, utilizaram a ponderação da frequência dos termos (TF) dos domínios para determinar a distribuição dos domínios por categorias. A ponderação TF é calculada para cada par categoria-domínio. Daí uma matriz que mostra as ponderações TF dos domínios para todas as categorias é obtida.

Quando se cataloga um vídeo, podemos comparar a distribuição dos domínios dos vídeos com a distribuição de domínios por categorias e a categoria que tem a distribuição de domínios mais próxima com a vídeo é seleccionada. A legenda do vídeo é processada de modo a obter domínios do WordNet e as pontuações das ocorrências do domínio (valores TF dos domínios) do vídeo. Utilizando a matriz das categorias aprendidas, pretende-se encontrar a categoria que tem a distribuição de domínios mais semelhante ao vídeo. Para este fim, utilizaram a similaridade de co-seno que é uma medida de semelhança entre dois vectores.

Como referido as abordagens utilizadas por (Demirtas, Cicekli, & Cicekli, 2010) melhoram um pouco os resultados relativamente aos obtidos com outros algoritmos mais antigos, e espelham o que foi feito até hoje utilizando só legendas. Para se ter noção de mais algumas variações descrevo mais duas:

Zhu et al. (Zhu, Toklu, & Liou, 2001) realizaram uma categorização automática de notícias, em que aproveitavam as palavras mais utilizadas, extraiam as palavras-chave, retirando as *stop words* e cruzavam essas palavas chave com uma base dados de conhecimento, que iria sendo incrementada com a utilização por intermédio de votação, o que lhes permitiria indicar qual a categoria da notícia.

Brezeale e Cook (Brezeale & Cook, 2008) utilizaram texto e recursos visuais separadamente na classificação vídeo. Na parte do texto, são removidas as *stop words* e é encontrada a raiz da palavra através da remoção de sufixos. A classificação é realizada utilizando uma máquina de suporte vectorial (SVM).

Capítulo 4

Trabalho Realizado

Este capítulo começa com a análise do problema que tínhamos para resolver e na segunda secção temos detalhadamente os passos que foram realizados na elaboração do trabalho. Na terceira secção é avaliado o projecto com a ajuda de utilizadores.

4.1 Análise do problema

Como o programa para tratar as legendas ia ser feito de raiz, havia a possibilidade de escolher a linguagem, tendo eu preferido o Java, por ser uma linguagem que servia para o efeito e estar mais à vontade.

Como *input* tínhamos ficheiros de legendas e queríamos extrair o máximo informação possível sobre as séries. Para poder explorar a informação que as legendas contém, desenvolvemos um interpretador que permite em *back-end* fazer *queries* ao programa e extrair resultados.

Para o projecto VIRUS a ideia os utilizadores interagirem com a interface do MovieClouds e para tal o utilizador não pode fazer directamente *queries* ao interpretador (que fornece os dados ao MovieClouds, como mostra a Figura 1) mas o nosso trabalho nessa área é cingir ao utilizador um leque de *queries* possíveis para as quais via interface, o nosso programa está pronto a responder.

Na Figura 1 também está expressa a possibilidade de alguns utilizadores acederem directamente ao Interpretador apenas para *debugging* ou nesta fase do projecto para acederem a todas as funções que o Interpretador já disponibiliza.

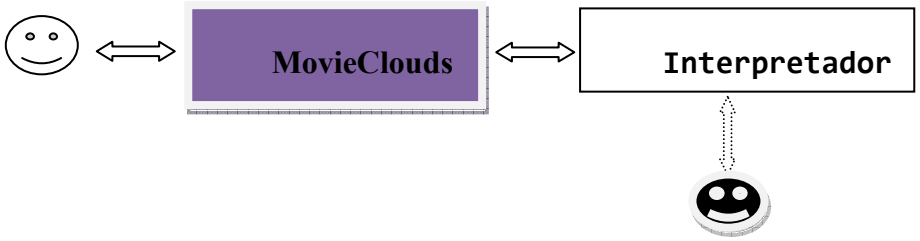

Figura 1: Esquema da interacção dos utilizadores com o MovieClouds e com o Interpretador

4.2 Cronologia do trabalho realizado

Primeiro que tudo, era preciso desenvolver um programa que lesse as legendas, normalmente no formato ".srt" (SubRip – explicado no Anexo C). O programa foi desenvolvido de maneira a ser flexível ao ponto de ler todas as legendas da base de dados ou subconjuntos escolhidos.

Na Figura 2 é apresentado um esquema que de forma resumida mostra como está estruturado o programa.

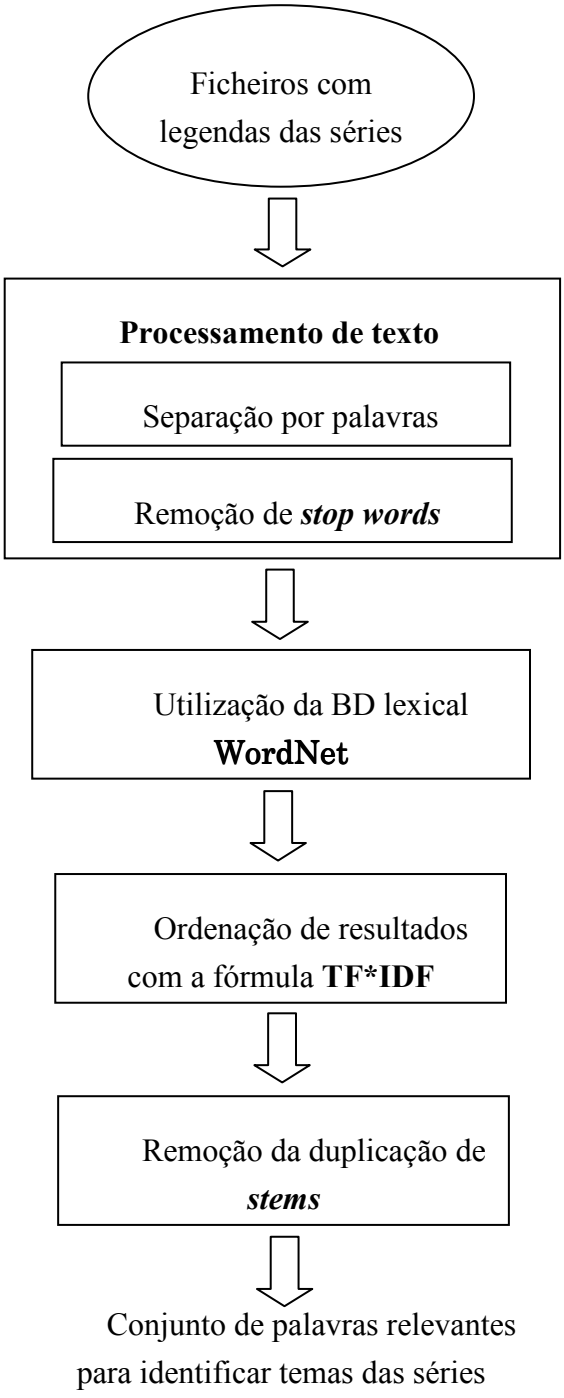

Figura 2: Esquema resumido da estrutura do funcionamento do interpretador

Com esta figura temos ideia do processamento actual mas nas próximas secções do relatório vamos apresentar os nossos desafios seguindo uma ordem cronológica, só assim se percebe melhor as escolhas que iam sendo tomadas na altura, ao acrescentar comandos ao interpretador.

4.2.1 Procura por palavras

A nossa primeira abordagem na utilização das legendas, foi dar ao utilizador a possibilidade de fazer uma *query* para encontrar as legendas que contêm ocorrências de determinada palavra.

Como já referido, como o utilizador comum não vai poder fazer *queries* directamente ao interpretador no projecto VIRUS, construímos uma nuvem de palavras mais frequentes em determinada série, depois de retiradas as *stop words* - maioritariamente artigos e preposições sem relevância semântica - como se pode ver neste exemplo:

```
> ncloud 40
{{"dr", 763}, {"good", 570}, {"yeah", 559}, {"time", 359}, {"surgery", 348}, {"gonna", 345}, {"hey", 325}, {"meredith", 279}, {"burke", 274}, {"heart", 268}, {"uh", 266}, {"fine", 252}, {"george", 236}, {"life", 212}, {"patient", 207}, {"talk", 205}, {"grey", 203}, {"people", 199}, {"bailey", 196}, {"izzie", 187}, {"shepherd", 186}, {"guy", 185}, {"day", 182}, {"baby", 174}, {"stop", 171}, {"malley", 158}, {"god", 157}, {"um", 154}, {"today", 154}, {"love", 153}, {"wait", 151}, {"home", 147}, {"chief", 147}, {"man", 145}, {"doctor", 142}, {"bad", 142}, {"work", 137}, {"mother", 136}, {"feel", 135}, {"talking", 129}}
```

Com a nuvem de palavras elaborada e apresentada no interface, o utilizador pode fazer *queries* ao interpretador mas apenas das palavras que lhe são apresentadas.

Nessa nuvem queríamos perceber se as palavras mais frequentes das séries teriam conotação com os temas das mesmas mas percebemos que havia algumas limitações que tinham de ser ultrapassadas.

Um problema é o caso das palavras na fronteira das *stop words,* palavras que não costumam estar nas listas de *stop words* mas que aparecem muitas vezes em qualquer série, como são os casos das palavras good e time e da expressão yeah. Será explicado mais à frente o que foi feito para ultrapassar o problema.

Outro problema é a não capacidade de lidar com palavras de raiz semelhante. Na nuvem que é apresentada como exemplo, temos talk e talking, o que nos retira espaço para aparecerem outras palavras relevantes na nuvem. Na secção seguinte vai ser explicado o que foi feito na busca de solução para este problema.

4.2.2 *Stemming*

Como procurar pela correspondência exacta entre a palavra introduzida na *query* e o texto contido nas legendas para procurar determinado tema era insuficiente, pensámos numa maneira de construir uma relação de "um-para-muitos" entre uma determinada palavra e outras correspondentes que estão relacionadas. Se o processo de *stemming* reduz duas palavras para o mesmo *stem*, diz-se que confluem. No caso das palavras referidas na secção anterior, talk e talking, são relacionadas pelo *stem* talk.

Abaixo mostro um exemplo de um caso em que utilizar a técnica *stemming* faz com que se encontre uma palavra relacionada (adjustment) com o que procuramos (adjust), caso contrário pensaríamos que ao longo da série nunca se refeririam a tal palavra.

```
> word adjust
NOTICE: 0 sentences found with the word "adjust"

> stem adjust

24.S01.E12 (subtitle 26 144.653 147.804 (sentence (w "I'm") (w "sure") (w
"my") (w "being" "be") (w "here") (w "has") (w "been") (w "a") (w "big") (w
"adjustment" "adjust") (w "."))
```

O comando `word` permite-nos procurar por uma palavra no conjunto de legendas seleccionado, e neste caso não existe a palavra adjust nas legendas. Quando utilizamos o comando `stem`, que nos permite obter palavras com um determinado *stem*, obtemos a resposta que existe uma frase que tem a palavra adjustment, que tem como *stem* a palavra adjust.

O *stemming* das palavras é uma característica importante suportada hoje em dia pelos sistemas de indexação e de pesquisa. Indexação e pesquisa, que por sua em vez, são parte de aplicações de *Text Mining*, sistemas de Processamento de Linguagem Natural (NLP) e sistemas de Recuperação de Informação (IR). *Text Clustering*, categorização e sumarização também exigem esta conversão como parte do pré-processamento antes de realmente aplicar qualquer outro algoritmo.

O principal objectivo do *stemming* é reduzir diferentes formas gramaticais / "formas da palavra" de uma palavra como o seu substantivo, adjectivo, verbo, advérbio, etc., para a sua forma de raiz para que se consiga uma melhoria no momento da indexação e pesquisa, uma vez que o *stem* permite-nos abranger um leque mais amplo de palavras obtidas, como já visto com as palavras talking e talk, ou adjust e adjustment. (Jivani, 2007)

Há três categorias às quais se costumam associar os algoritmos de *stemming*:

- Estatísticos;
- Mistos;
- Remoção (de afixos).

Os algoritmos baseados em técnicas e análise estatística são mais recentes do que os de remoção. Estes algoritmos normalmente removem os afixos mas depois de algum processo estatístico que conta com o conhecimento da morfologia da língua.

Os algoritmos mistos podem combinar várias abordagens. Por exemplo, um algoritmo de remoção pode ser reforçado com o uso de um dicionário de verbos irregulares. (Smirnov, 2008)

No nosso trabalho foram utilizados algoritmos de remoção, pelo qual vão ser mais detalhados. Ao falar das desvantagens voltamos a comparar com os outros tipos.

Os algoritmos de remoção aplicam um conjunto de regras de transformação para cada palavra, tentando cortar prefixos ou sufixos (afixos) conhecidos. O primeiro algoritmo deste tipo foi descrito por J. B. Lovins (Lovins, 1968). De seguida, mais alguns algoritmos de remoção foram sugeridos.

O mais utilizado é o algoritmo Porter (Porter, 1980) e, eventualmente o Porter 2 (Porter, Snowball: A language for stemming algorithms, 2001), algoritmos que vão ser explicados mais à frente.

Resumidamente, o primeiro algoritmo de *stemming* descrito, o algoritmo de Lovins, funciona da seguinte maneira:

Define 294 terminações, cada uma delas ligada a uma das 29 condições, mais 35 regras de transformação. A palavra a ser processada quando encontra uma terminação com condição satisfatória, remove-a e o resultado é o *stem* da palavra.

O *stemmer* de Lovins remove o maior sufixo da palavra. Uma vez que o final é removido, a palavra é recodificada com uma tabela diferente que faz vários ajustes para converter estes *stems* em palavras válidas. O algoritmo remove sempre o máximo de um sufixo de uma palavra, devido à sua natureza como algoritmo de passagem única.

As vantagens deste algoritmo são, ser muito rápido e poder lidar com a remoção de consoantes duplas e também com muitos plurais irregulares.

Como desvantagem principal desta abordagem, tinha o facto de muitos sufixos não estarem disponíveis na tabela de terminações. Vários artigos, entre eles, (Jivani, 2007) e (Smirnov, 2008) referem que a razão para muitas falhas era o vocabulário utilizado pelo autor ser muito técnico.

Algoritmo Porter

Para explorar os benefícios do *stemming*, numa primeira abordagem utilizámos o algoritmo descrito em (Porter, An algorithm for suffix stripping, 1980). Este algoritmo foi escrito em BCPL, uma linguagem já não muito utilizada. Martin Porter escreveu-o em 1979 e o trabalho foi publicado em 1980.

Para o nosso trabalho utilizámos a implementação em Java, desenvolvida por o próprio Martin Porter uns anos mais tarde e que está disponível em (http://tartarus.org/~martin/PorterStemmer/).

No anexo A é explicado em detalhe os vários passos do algoritmo.

Para consolidar a explicação do algoritmo, apresento alguns exemplos (Tabela 3) de palavras que aparecem muito em duas séries testadas (24 e Anatomia de Grey), palavras essas que seriam alteradas da seguinte forma:

Palavra	Stem	Passo(s)	Condições
president	presid	4	(m>1) ENT ->
people	peopl	5a	(m=1 and not *o) E ->
hey	hei	1c	(*v*) Y -> I
minutes	minut	1a; 5a	S -> ; (m>1) E ->
today	todai	1c	(*v*) Y -> I
stay	stai	1c	(*v*) Y -> I
family	famili	1c	(*v*) Y -> I
senator	senat	2; 4	(m>0) ATOR -> ATE ; (m>1) ATE ->
house	hous	5a	(m=1 and not *o) E ->
day	dai	1c	(*v*) Y -> I
happened	happen	1b	(*v*) ED ->
wanted	want	1b	(*v*) ED ->
surgery	surgeri	1c	(*v*) Y -> I
talking	talk	1b	(*v*) ING ->
hospital	hospit	4	(m>1) AL ->

Tabela 3: Remoção de sufixos com o algoritmo Porter

Para perceber melhor os resultados, é preciso entender os supostos erros do *stemming*:

Apesar de que muitas vezes os *stemmers* se basearem no conhecimento da morfologia da linguagem, o seu objectivo não é encontrar uma raiz que seja significativa de uma palavra. Em vez disso, uma palavra pode ser truncada numa posição "incorrecta" do ponto de vista da linguagem natural.

Os resultados não são formas morfologicamente correctas de palavras. No entanto, uma vez que o índice de documentos e consultas são feitos sem o utilizador saber qual é o stem, esta particularidade não deve ser considerada como uma falha, mas sim como uma característica distinguindo *stemming* (radicalização) de lematização (que é a tarefa de encontrar uma forma canónica de um lexema). (Smirnov, 2008)

Na secção sobre o trabalho futuro é explicado o que ainda pode ser feito com uma abordagem de lematização.

O facto de os *stems* produzidos por algoritmos de remoção de sufixos muitas vezes não serem palavras, faz com que seja difícil usá-los para outros fins que não a recuperação da informação. Técnicas interactivas que exigem um *input* do utilizador, como a selecção de termos para a expansão duma consulta, vai sofrer muito se o utilizador tiver de trabalhar com *stems* em vez de palavras reais. Também se torna difícil pesquisar num dicionário sem palavras reais. (Hull & Gregory, 1996)

Alem deste aspecto da morfologia da palavra, existem erros bem conhecidos - reportados por exemplo por (Jivani, 2007) e (Willett, 2006) - associados ao *stemming*, os quais vão ser detalhados de seguida. Os erros que mais acontecem são de dois tipos: "de mais" (*over-stemming*) e "de menos" (*under-stemmaing*). Também podemos chamar por "*over-truncation*" e "*under-truncation*"

Over-stemming ocorre quando permanece um *stem* muito curto após o truncamento e pode resultar em palavras nada relacionadas a ser confundidas com a mesma raiz, como medical e media que são recuperados pela raiz **med***. Isto também é conhecido como um falso positivo.

Under-stemming, por outro lado, surge se a *string* removida é muito curta e pode resultar em palavras relacionadas a serem descritos por diferentes *strings*, como bibliographically a ser truncada como bibliographic, em vez da raiz mais curta **bibliograph*** que também englobaria a raiz bibliography. Também é conhecido como um falso negativo.

Além dos erros que estão em quase toda a literatura sobre *stemming*, Porter mais tarde (Porter, Snowball: A language for stemming algorithms, 2001) adicionou outro tipo de erro: *mis-stemming*. ***Mis-stemming*** é tirar o que parece ser um fim, mas é realmente parte do *stem* enquanto *Over-stemming* é tirar um verdadeiro final que resulta na confluição de palavras de significados diferentes.

Para o exemplo **ly** pode ser retirado de cheaply, mas não de reply, porque em reply, **ly** não é um sufixo. Se for removido, reply confluiria para **rep** (que geralmente é a forma curta de representative).

Com um algoritmo misto, utilizando um dicionário, podemos tentar evitar *mis-stemming* e *over-stemming*. O dicionário pode nos dizer que reply não deriva de **rep**. É importante perceber, porém, que um dicionário não dá uma solução completa, mas pode ser uma ferramenta para melhorar o processo de confluência dependendo também da qualidade do mesmo. Um dicionário terá de ser muito abrangente, totalmente actualizado, e com boas definições de palavras para alcançar os melhores resultados.

Para as palavras referidas anteriormente, podemos fazer uma análise critica sobre o resultado (apresentado na Tabela 4) com a justificação para os resultados menos bons (depois da tabela).

Palavra	Stem	Avaliação
president	presid	Bom
people	peopl	Bom
hey	hei	Mau
minutes	minut	Bom
today	todai	Mau
stay	stai	Mau
family	famili	Bom
senator	senat	Bom
house	hous	Bom
day	dai	Discutível
happened	happen	Bom
wanted	want	Bom
surgery	surgeri	Discutível
talking	talk	Bom
Hospital	hospit	Discutível

Tabela 4: Análise crítica dos *stems*

Na palavra hey, a transformação de **y** em **i** faz com que se conseguisse retornar muitas palavras, mas nada relacionado com a expressão hey.

Na palavra today, a mesma transformação, faz com que seja impossível encontrar mais alguma palavra.

Em stay a transformação em **stai** até permitiria encontrar a palavra staid – uma palavra que praticamente já não se usa - mas faz com que não encontre várias palavras que estão presentes nas legendas, como: stays, staying e stayed.

A palavra day transformada em **dai** permitiria encontrar várias palavras relacionadas com day, por isso podemos pensar que não é um *stem* mau. Como exemplos de palavras que este *stem* encontraria temos: daily, dailiness, dailies. Como aspecto negativo, temos que algumas palavras que começam por **dai** mas não têm nada relacionado com a palavra day iriam ser retornadas, além de que nas legendas este *stem* faz com que não encontremos a palavra days.

A transformação de surgery em **surgeri** também é discutível. É um *stem* óptimo para encontrar a palavra surgeries porque de certeza que não vai encontrar mais palavras. Mas consideramos que pode haver *under-stemming* porque não são encontradas palavras como surgeon ou surgeons. Claro que para isso o *stem* tinha de ser **surg**, com as implicações negativas de encontrar mais palavras que não são relacionadas com surgery.

Em hospital, embora o *stem* **hospit** permita encontrar palavras como hospitia, podemos pensar que peca por *over-stemming*. Isto porque com o *stem* **hospit**, podemos obter palavras como hospitals ou hospitably, esta ultima nada relacionada.

Como se pode ver por estes exemplos, as palavras para as quais se considerou que o *stem* resultante era "mau", eram as palavras terminadas em **y**.

De seguida surgiram dois pontos que queríamos trabalhar. Um era procurar alternativas ao *stemmer* utilizado e outro era começar a pensar num *stemmer* para outras línguas, começando pelo português.

Algoritmo Porter 2

O segundo algoritmo que foi utilizado acabou por ser também desenvolvido por Porter, um algoritmo conhecido como Porter 2, no qual tinham sido corrigidos alguns dos erros apontados para o seu antigo algoritmo. Martin Porter nos mais de 20 anos que intervalaram os 2 algoritmos, desenvolveu uma *framework* para ser mais fácil criar *stemmers* de várias línguas utilizando a base construída por ele. A *framework* de nome Snowball (http://snowball.tartarus.org/index.php), além de contar com a actualização do algoritmo que falei anteriormente (para a língua inglesa), hoje em dia tem representadas dezenas de línguas.

As mudanças não são muito extensas:

- A terminação **y** é alterada para **i** com menos frequência;
- O sufixo us não perde o **s**;
- Alguns sufixos foram adicionados nas regras para poderem ser removidos (p. ex. o sufixo ly);
- Além disso, uma pequena lista de formas excepcionais foi incluída.
- Os passos 5a e 5b do antigo *stemmer* - que está no Anexo A - foram combinados num único passo. Isto significa que a transformação de ll em l não é feito com a remoção do último **e**;
- No passo 3, ative é removido apenas quando está depois da primeira consoante;
- Inclui um novo passo 0 para lidar com apóstrofo.

Nos testes feitos com este algoritmo, podemos confirmar o primeiro ponto, em que a nova regra é:

Step 1c: *

substituir o sufixo **y** ou **Y** por **i** se precedido por uma consoante, que não seja a primeira letra da palavra, por exemplo: cry -> cri, by -> by, say -> say

Realmente as palavras que terminam com **y** passaram menos vezes a ser transformadas para acabar em **i**, como é mostrado na Tabela 5:

Palavra	Stem	Passo(s)	Condição
hey	hei	1c	(*v*c) Y -> I
today	today	1c	(*v*c) Y -> I
stay	stay	1c	(*v*c) Y -> I
family	famili	1c	(*v*c) Y -> I
day	day	1c	(*v*c) Y -> I
surgery	surgeri	1c	(*v*c) Y -> I

Tabela 5: Remoção de sufixos com o algoritmo Porter 2

Os *stemmers* disponíveis no Snowball são puramente algorítmicos. Podem ser alargados de modo a incluir listas de excepções, que poderiam ser usadas em combinação com um dicionário completo. Ao serem puramente algorítmicos deveriam ter, um desempenho inferior ao desempenho dos *stemmers* bem construídos baseados no dicionário mas as vantagens fazem com que continue a ter muita adesão. Os *stemmers* algorítmicos em geral são muito rápidos, conseguem ter bons resultados apesar de terem alguns erros e exigem muito menos actualizações do que um *stemmer* baseado no dicionário, pelo problema da língua estar em constante mudança.

Porter em (Porter, Snowball: A language for stemming algorithms, 2001) esclareceu algumas das questões levantadas ao longo dos anos e a razão para haver determinados "erros":

- O *stemmer* não processa prefixos. Normalmente, os prefixos alteram radicalmente o significado, de modo que é melhor deixar como está. Exemplo: unhappy e happy;

- O *stemmer* não processa *stop words*. O *stemming* dessas palavras não é útil. Há uma ligação gramatical entre be e being e também têm uma conexão morfológica, mas que não é verdade para am e was, apesar de terem uma ligação gramatical;

- A morfologia de uma língua altera-se menos rapidamente do que os significados da palavra. Transpire passou a significar "acontecer", e o seu antigo significado de "exalar" ou "expirar" agora praticamente não é usado. Mas transpiration ainda carrega o significado anterior. Então, o que era anteriormente um *stemming* aceitável, pode ser julgado agora como um *over-stemmer*, não, porque a palavra depois do processo tenha alterado o seu significado, mas porque a palavra mudou o seu significado.

- Todos os idiomas contêm irregularidades, mas até que ponto eles devem ser acomodados num algoritmo de *stemming*? Um *stemmer* em inglês, por exemplo, pode converter plurais regulares em forma singular sem dificuldade em palavras como: boys, girls, hands, etc. Deveria fazer o mesmo com plurais irregulares (men, children, feet, ...)? Aqui temos casos irregulares, com sufixos flexionais, mas há irregularidades com sufixos derivacionais, a que Lovins chama "excepções de soletração". absorb/absorption e conceive/conception são exemplos disso. Etimologicamente, a explicação do primeiro é a raiz latina, sorbere, é um verbo irregular, e do segundo é que a palavra conceive vem do francês, em vez de directamente do latim. O *stemmer* de Porter não lida com irregularidades, mas da própria experiência do autor, esta nunca foi uma área de reclamação. Reclamações de facto são sempre sobre confluições falsas, por exemplo new e news.

- As irregularidades dos sufixos derivacionais em inglês são com palavras curtas e antigas, que são muito comuns (man/men, woman/women, see/saw ...) ou são utilizadas apenas raramente (ox/oxen, louse/lice, forsake/forsook ...). A última classe pode ser ignorada, e a primeira tem os seus próprios problemas que nem sempre são resolvidos por *stemming*. Por exemplo, man pode ser um verbo, e saw pode significar um instrumento de corte, ou, como um verbo, pode significar usar tal instrumento. A confluência dessas formas leva a erros frequentemente como o *mis-stemming*.

- Outro ponto que Porter refere é sobre as formas linguísticas raras. Os *stemmers* não precisam de lidar com formas linguísticas que aparecem muito raramente. Por este motivo não se deve preocupar muito com a sua presença ocasional. Aparecem em todos os livros de gramática, e em qualquer caso, podem ser encontradas em textos mais antigos. O hábito de colocar as formas raras "completar o quadro" é normal, e geralmente passa despercebido. Por exemplo yourselves, por analogia com himself, herself, etc., embora yourselves seja realmente muito raro em Inglês.

Vários artigos, entre eles, (Jivani, 2007) e (Hull & Gregory, 1996) referem que dos vários algoritmos de *stemming* a diferença de resultados é pouca. Até comparando os 2 algoritmos de Porter, é referido em (The English (Porter2) stemming algorithm) que a diferença de resultados é inferior a 5%. Uns são melhores nos erros de *over-stemming*, outros são melhores em *under-stemming*.

Sendo a diferença tão pequena, achámos melhor tentar outras abordagens, em vez de tentarmos ter resultados melhores com outros algoritmos de *stemming*.

4.2.3 WordNet

A fim de melhorar a qualidade das respostas às consultas que fazíamos nas legendas, experimentámos o WordNet (Miller, 1995) para enriquecer o relacionamento entre palavras, utilizando sinónimos ou outras relações possíveis.

A origem do WordNet foi a construção de um dicionário on-line que começou a ser desenvolvido em 1985 por psicolinguistas e linguistas seguindo as ideias que tinham sido já estudadas por Miller. Depois começou a evoluir e tornou-se uma ferramenta muito mais completa do que um simples dicionário. A ideia era ter uma combinação mais eficaz da informação lexicográfica tradicional que está nos dicionários com a computação de alta velocidade.

A diferença mais óbvia entre WordNet e um dicionário padrão é que o WordNet divide o léxico em quatro categorias: substantivos, verbos, adjectivos e advérbios. Sendo que as palavas podem aparecer em mais do que uma categoria, dependendo do contexto em que apareçam, uma característica conhecida como "polissemia". Ou seja, uma única palavra pode ter significados muito diferentes em contextos diferentes e é mesmo possível que uma única palavra possa ser usada em diferentes partes do discurso (verbo, substantivo, etc.). Por exemplo, fly pode-se referir a um insecto (substantivo) ou ao acto de mover-se através do ar (um verbo).

Cada palavra é associada a vários *synonyms sets* (synsets), abrangendo assim sinónimos em vários contextos possíveis para a utilização da palavra.

Tipicamente um synset além de incluir word forms – que vão ser explicados mais à frente - também inclui uma colecção de relações com base em características tais como antonímia (opostos, tais como "para cima" e "para baixo"), hipónimos / hiperónimos (subtipo / supertipo - "computador", por exemplo é um subtipo de "máquina"), etc.. As relações específicas aplicáveis a um synset dependem da parte do discurso associado ao conceito que o synset representa. Além disso, dois tipos diferentes de relacionamentos são definidos, especificamente semântico e lexical. A relação semântica é aquela que existe entre dois synsets e que se presume que se aplica a todas as formas de palavras dentro dos synsets. Em contraste, a relação lexical existe entre duas formas de palavras específicas dentro de dois synsets separados.

O synset também inclui uma definição curta e, geralmente, fornece um ou mais exemplos de como as formas da palavra no synset são usados.

Para melhor percepção deste conceito, apresentamos de seguida o resultado da utilização do WordNet 2.1 Browser ao pesquisarmos pela palavra people:

```
The noun people has 4 senses (first 4 from tagged texts)

1. (559) people -- ((plural) any group of human beings (men or women or
children) collectively; "old people"; "there were at least 200 people in the
audience")

2. (94) citizenry, people -- (the body of citizens of a state or country;
"the Spanish people")

3. (40) multitude, masses, mass, hoi polloi, people, the great unwashed --
(the common people generally; "separate the warriors from the mass"; "power
to the people")

4. (5) people -- (members of a family line; "his people have been farmers for
generations"; "are your people still alive?")

The verb people has 2 senses (first 1 from tagged texts)

1. (1) people, populate -- (fill with people or supply with inhabitants;
"people a room"; "The government wanted to populate the remote area of the
country")

2. people, populate -- (furnish with people; "The plains are sparsely
populated")
```

Cada um destes conjuntos representa um synset, como se pode observar, tem os sinónimos em cada sentido, definição e exemplo de utilização.

O WordNet é uma plataforma grande que foi explorada aos poucos e de seguida vão ser descritos os passos que tomámos.

Sinonímia

No primeiro teste explorámos as word forms, sendo na prática os sinónimos, que nos parecia ser a relação óbvia que precisávamos. Cada conjunto de word forms é associado a um sentido (*senses*), mas nesta fase interessava-nos saber todos os sinónimos possíveis.

Nesta tabela mostro algumas das palavras já referidas como exemplo anteriormente e os sinónimos correspondentes, a tabela completa está na tabela 1 do Anexo B.

Palavra	Sentido	Word Forms
president	1	president
	2	President of the United States, United States President, President, Chief Executive
	3	president
	4	president, chairman, chairwoman, chair, chairperson
	5	president, prexy
	6	President of the United States, President, Chief Executive
house	1	house
	2	house
	3	house
	4	family, household, house, home, menage
	5	theater, theatre, house
	6	firm, house, business firm
	7	house
	8	house
	9	house
	10	house
	11	sign of the zodiac, star sign, sign, mansion, house, planetary house
	12	house
	13	house
	14	house, put up, domiciliate
surgery	1	surgery
	2	surgery
	3	operating room, OR, operating theater, operating theatre, surgery
	4	operation, surgery, surgical operation, surgical procedure, surgical process

Tabela 6: Parte da tabela com Word Forms

De referir que pode parecer haver várias word forms iguais, mas cada linha desta tabela está associada a um sentido de utilização ou contexto. Para cada um destes contextos temos uma definição, o que vai ser apresentado mais à frente.

A tabela anterior mostra um aspecto positivo e outro negativo para o que pretendíamos. O aspecto positivo é que realmente o WordNet tem uma grande potencialidade a encontrar sinónimos. Como aspecto negativo, temos o facto de serem sinónimos a mais fora do possível contexto de utilização, por exemplo, como sinónimo de house, temos a expressão "*sign of the zodiac*".

Esse aspecto negativo é transversal a todos os métodos e por isso uma decisão que tomámos foi trabalhar apenas com palavras presentes nas legendas, com isso os resultados centram-se nos temas da série.

A estrutura do WordNet está dividida em synsets de vários tipos:

- Genéricos (Synset);
- Substantivos (NounSynset);
- Verbos (VerbSynset);
- Adjectivos (AdjectiveSynset e AdjectiveSatelliteSynset);
- Advérbios (AdverbSynset).

Os advérbios vão ter menos atenção porque, por norma são palavras com pouco conteúdo informativo. Miller – o principal criador do WordNet – em (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990) referiu que não ia abordar os advérbios, dando-lhes pouca importância

Cada tipo de synset contem vários métodos associados. Esses métodos vão ser explicados mais à frente.

Como já referido anteriormente, os primeiros testes foram com as word forms. Estes incluem-se no tipo genérico, o que faz com que o WordNet retorne sinónimos de todas as palavras que tem na base de dados, independentemente se são substantivos, verbos, adjectivos ou advérbios.

Outro método que pertence à categoria genérica é o:

getDerivationallyRelatedForms - Retorna formas de palavras que derivam de outras e estão relacionadas com a palavra da legenda.

Exemplo: uma forma derivadamente relacionada de "*meter*" é "*metrical*".

Exemplo com palavras das legendas na tabela 2 do Anexo B.

De seguida são detalhados alguns testes feitos para explorar as potencialidades do WordNet.

Definições das palavras

Cada synset tem uma definição da palavra em determinado contexto, e uma abordagem que pensámos para extrair mais informação foi retirar as palavras da definição, exceptuando as *stop words*. Nesta tabela mostro o resultado para duas palavras e o restante está na tabela 3 do Anexo B.

Palavra	Definição da palavra num certo contexto	Palavras existentes na definição
president	an executive officer of a firm or corporation	executive, officer, firm, corporation
	the person who holds the office of head of state of the United States government	person, holds, office, head, state, United, States, government
	the chief executive of a republic	chief, republic
	the officer who presides at the meetings of an organization	presides, meetings, organization
	the head administrative officer of a college or university	administrative, college
	the office of the United States head of state	
surgery	the branch of medical science that treats disease or injury by operative procedures	branch, medical, science, treats, disease, injury, operative, procedures
	a room where a doctor or dentist can be consulted	room, doctor, dentist, consulted
	a room in a hospital equipped for the performance of surgical operations	hospital, equipped, performance, surgical, operations
	a medical procedure involving an incision with instruments; performed to repair damage or arrest disease in a living body	procedure, involving, incision, instruments, performed, repair, damage, arrest, living, body

Tabela 7: Parte da tabela com definições

Na coluna das palavras existentes na definição, retirei as repetições apenas por uma questão do tamanho da tabela. No algoritmo, da frase "*the office of the United States head of state*" são extraidas as palavras office, United, States, head e state.

Para confirmar que podíamos utilizar as palavras da definição quisemos perceber se havia bijecção na definição das palavras existentes na definição da palavra da legenda. Apresento um pouco do que se obtém a fazer este teste para a palavra president. Para ver a tabela completa, ver tabela 4 do Anexo B.

Palavra	**Palavra na definição**	**Palavras existentes na definição** (excepto Stop Words)
president	executive	person, responsible, administration, business, persons, administer, law, manages, government, agency, department, function, carrying, plans, orders
	person	human, body, including, clothing, grammatical, category, pronouns, verb, forms
	college	body, faculty, students, establishment, seat, higher, learning, housed, including, administrative, living, quarters, facilities, research, teaching, large, diverse, institution, created, educate, life, profession, grant, degrees

Tabela 8: Parte da tabela com palavras existentes nas definições das definições

Com a palavra president percebemos que não é muito intuitivo utilizar as palavras existentes na definição, porque são frases que mesmo retirando as *stop words*, tem muitas palavras que não são relacionadas com a palavra, por exemplo, president tem na definição person mas por sua vez esta tem grammatical, o que não é relacionado com president.

Hiperonímia/hiponímia

Outra propriedade que quisemos explorar foi a "hiponímia". Para isso utilizámos três métodos, um que retorna hiperónimos para os substantivos e verbos, outro que retorna hipónimos para os substantivos e outro equivalente para os verbos.

A hiponímia é a relação entre um termo mais específico e um termo genérico, expresso por "*is-a*". Hiperónimos são generalizações e hipónimos são especializações.

Métodos utilizados:

getHypernyms – São devolvidos os hiperónimos directos (hierarquia superior) da palavra da legenda.

Exemplo: (substantivo) o hiperónimo de "*tent*" é "*shelter*".

(verbo) "*verbalize*" é um hiperónimo de "*shout*".

getHyponyms – São devolvidos os hipónimos directos (hierarquia inferior) da palavra da legenda.

Exemplo: um hipónimo de "*shelter*" é "*tent*".

getTroponyms – São devolvidos os tropónimos directos (hipónimos/tipos subordinados) da palavra da legenda.

Exemplo: "*shout*" é um tropónimo/hipónimo/tipo subordinado de "*verbalize*".

Exemplos com palavras das legendas nas tabelas 5 a 7 do Anexo B.

Como se pode ver pelas tabelas em anexo, os resultados dos hipónimos, *troponyms* e hiperónimos têm muitas palavras, que podem ser uteis para o nosso trabalho.

Para tentar utilizar o que já sabíamos do WordNet para catalogar legenda a legenda, fizemos um cruzamento de hiperónimos relacionados com palavras da legenda e sempre que aparecessem mais do que uma vez esses hiperónimos, ficariam a catalogar a legenda. Por exemplo na seguinte frase, a palavra know e a palavra honor partilham os hiperónimos recognize, recognise e accept. Se numa frase as palavras partilhassem hiperónimos parecia-nos poder ser uma indicação de um tema falado naquela determinada frase.

```
As you KNOW, the HONOR of performing the first surgery

recognize: [know, honor] recognise: [know, honor] accept: [know, honor]
```

Nesta frase o resultado é aceitável, cruzando know com honor retorna aqueles hiperónimos. Mas este algoritmo nem sempre dá resultados tão bons para servir de catalogação para uma frase, como por exemplo:

```
It HAS nothing to DO with me being black.

make: [has, do]
```

E muitas das legendas não chegavam a ser catalogadas, por não haver cruzamento de hiperónimos, como nestas frases:

```
for God's sakes, David, running for president.

what were you thinking?!

So, bypass surgery tomorrow with Dr. Burke - I hear he's good.
```

Não sendo os resultados propriamente bons, não avançámos, por agora, em catalogar frases mas há margem nesta área para algum trabalho futuro.

Meronímia/holonímia

Os merónimos representam a "parte" e os holónimos representam o "todo".

Métodos utilizados:

getMemberHolonyms – São devolvidos termos que expressam o "todo" do qual a palavra da legenda é uma parte.

Exemplo: o termo holónimo de "*Saturn*" é "*solar system*".

getMemberMeronyms – São devolvidos termos que expressam partes das quais a palavra da legenda é o todo.

Exemplo: termos merónimos de "*Roman Alphabet*" são "A", "B", "C", etc.

getSubstanceHolonyms – São devolvidos termos que foram feitos com a substancia que a palavra da legenda representa.

Exemplo: uma substancia holónoma de "*paper*" é "*page*".

getSubstanceMeronyms – São devolvidos termos que são substâncias que podem fazer o representado na palavra da legenda.

Exemplo: uma substancia merónimo de "*chocolate*" é " *cocoa*".

getPartHolonyms – São devolvidos termos que são o todo que incluem a palavra da legenda.

Exemplo: uma parte holónoma de "*fuselage*" é "*airplane*".

getPartMeronyms – São devolvidos termos que são partes da palavra da legenda.

Exemplo: parte merónimos para "*airplane*" incluem "*wing*" e "*fuselage*".

Exemplos com palavras das legendas nas tabelas 8 a 13 do Anexo B.

Os merónimos e holónimos são importantes na procura de palavras relacionadas porque nos descrevem o todo e as partes que compõem esse todo mas, nos nossos testes percebemos que estes métodos não devolvem muitas respostas, assunto, que vai ser explicado mais à frente.

Adjectivos

Outro tipo de palavras que queríamos explorar melhor eram os adjectivos. O WordNet permite-nos obter algumas relações entre os adjectivos contidos no texto.

Os métodos que são descritos a seguir apenas estão disponíveis para os adjectivos. Mais à frente são apresentados mais alguns métodos que são transversais a vários tipos de palavras.

getPertainyms – Retorna os *pertainyms* (palavras de onde a palavra da legenda derivou).

Exemplo: um *pertainym* de "*academic*" é "a*cademia*".

getRelated – Retorna palavras relacionadas ("*see also*") com a palavra da legenda.

Exemplo: "*aggressive*" está relacionado com "*hostile*".

getSimilar – Retorna palavras com o significado semelhante à palavra da legenda.

Exemplo: "*abridged*" é semelhante a "*shortened*".

getParticiple – Devolve o verbo do qual a palavra da legenda derivou.

Exemplo: "*breaking*" é o particípio presente de "*break*".

Exemplos com palavras das legendas nas tabelas 14 a 17 do Anexo B.

O problema destes métodos é o mesmo do que acontece com os merónimos, e holónimos. Isto é, os adjectivos são palavras que regra geral não têm realce nas nuvens elaboradas com os algoritmos que utilizamos.

Atributos

Também quisemos explorar um método que está disponível para substantivos e adjectivos:

getAttributes –

- Nos substantivos: são devolvidos adjectivos que descrevem estados associados à palavra da legenda;
- Nos adjectivos: são devolvidos termos, cuja palavra da legenda é um atributo.

Exemplo: (adjectivo) "accurate" é um atributo de "truth".

(substantivo) atributos de "seriousness" incluem "serious" e "frivolous".

Exemplo com palavras das legendas na tabela 18 no Anexo B.

Os atributos retornados por o WordNet eram também palavras com pouca relevância.

Tópicos / Termos dos Tópicos

O método que nos permite obter tópicos associados à palavra está presente nos três tipos de palavras que temos acompanhado, verbos, substantivos e adjectivos. Também é analisado um método que apenas está disponível para os substantivos que retorna termos associados a determinado tema.

getTopics - Identifica tópicos com os quais a palavra da legenda está associada.

Exemplo: (verbo) "*bandage*" está no domínio da "*medicine*".

(substantivo) o tópico associado a "*periodic table*" é "*chemistry*".

(adjectivo) "*acidic*" está no domínio da "*chemistry*".

getTopicMembers – São devolvidos termos associados ao tópico.

Exemplo: termos do domínio da "*medicine*" incluem "*acute*" e "*chronic*".

Exemplos com palavras das legendas nas tabelas 19 e 20 do Anexo B.

Os tópicos são palavras que vamos utilizar mais à frente porque como o nome e os exemplos dados mostram, são uteis para categorizar uma determinada palavra. Os membros dos tópicos dão resultados menos bons.

Verbos

Os verbos têm alguns métodos que apenas são usados com este tipo de palavras e também quisemos perceber a sua importância.

getEntailments - São devolvidos termos que a palavra da legenda implica.

Exemplo: "*snore*" implica "*sleep*".

getOutcomes – São devolvidos termos que expressam o que a palavra da legenda pode causar.

Exemplo: "*remind*" é uma causa de "*remember*".

getVerbGroup – São devolvidos verbos que fazem parte de um grupo de verbos com significado semelhante à palavra da legenda.

Exemplo: "*talk*" e "*write*" pertencem ao mesmo grupo de verbos.

Exemplos com palavras das legendas nas tabelas 21 a 23 do Anexo B.

Destes métodos, só o último nos deu respostas melhores, os outros retornavam poucas palavras relevantes.

Usos

Este método permite-nos saber que utilização damos a determinada palavra.

getUsages - Identifica os tipos de uso associados à palavra da legenda.

Exemplo: (substantivo) O uso associado a "*stuff*" é "*slang*".
(verbo) "*play hooky*" é um termo "*slang*".
(adjectivo) "*hot under the collar*" é um "*colloquialism*".

Exemplo com palavras das legendas na tabela 24 do Anexo B.

Esta tabela tem uma particularidade em relação às outras todas em que utilizei o WordNet. Os termos que estão relacionados com a palavra da legenda não são apenas os que aparecem na legenda mas todos o que o WordNet retorna.

A utilização deste método pode ter mais usos num futuro porque conseguimos separar as palavras por algumas categorias mas neste momento a utilização é para apenas poder retirar os palavrões da nuvem de palavras a apresentar. Também tínhamos a categoria "*slang, cant, jargon, lingo, argot, patois, vernacular*" que podíamos filtrar mas por agora fazemos apenas uma lista com as palavras da legenda que pertencem à categoria: "*obscenity, smut, vulgarism, filth, dirty word*".

A lista de palavras obscenas serve para que no interface o utilizador possa seleccionar se quer ver determinada lista com um filtro "parental".

Exemplo da utilização do WordNet no interpretador

Das várias nuvens à nossa disposição, vamos apresentar um exemplo com a utilização duma delas – **Related** - o WordNet neste método encontra apenas adjectivos e retorna as palavras relacionadas com esse adjectivo. Para tal, um primeiro comando que introduzimos no interpretador retorna os adjectivos encontrados por o WordNet em determinado texto, tendo nós limitado a escolha aos 40 que apareciam mais.

```
> selectCloud 5

Related (Adjectives) - TOP 40

152 : good
55 : fine
34 : kind
33 : bad
32 : wrong
22 : wanted
22 : hard
19 : open
19 : clear
17 : happy
16 : shut
16 : important
15 : ready
15 : left
14 : lost
13 : hot
13 : clean
11 : stupid
9 : worse
9 : stable
9 : real
9 : normal
9 : light
9 : easy
8 : true
8 : short
8 : fat
8 : easier
8 : difficult
7 : white
7 : moving
6 : worst
6 : significant
6 : healthy
5 : sound
5 : simple
5 : saved
5 : personal
5 : free
5 : early
```

Com um segundo comando podemos obter os momentos em que aparecem palavras relacionadas com uma das palavras da lista anterior. Se escolhermos a palavra white, temos:

```
> getCloud white
Greys.Anatomy.S01.E02 (subtitle 525 1854.069 1857.07 (sentence (w "I") (w "could") (w
"ve") (w "done") (w "a") (w "better") (w "job") (w "if") (w "I'd") (w "had") (w
"more") (w "light"))
Greys.Anatomy.S01.E03 (subtitle 163 547.567 550.995 (sentence (w "I") (w "wish") (w
"he'd") (w "just") (w "go") (w "into") (w "the") (w "light") (w "so") (w "I") (w
"can") (w "get") (w "on") (w "another") (w "case"))
Greys.Anatomy.S01.E03 (subtitle 278 903.316 906.598 (sentence (w "and") (w "you're")
(w "feeling") (w "this") (w "big") (w "push") (w "to") (w "go") (w "towards") (w
"the") (w "light"))
Greys.Anatomy.S01.E04 (subtitle 82 213.747 216.752 (sentence (w "You") (w "always")
(w "come") (w "in") (w "like") (w "that") (w "bang") (w "the") (w "light") (w "on"))
Greys.Anatomy.S01.E07 (subtitle 712 2012.884 2015.04 (sentence (w "You") (w "know")
(w "Taylor") (w "her") (w "anesthesia") (w "s") (w "awful") (w "light"))
Greys.Anatomy.S01.E08 (subtitle 53 148.457 150.554 (sentence (w "It's") (w "a") (w
"light") (w "rotation") (w "Can") (w "you") (w "get") (w "me") (w "in") (w "then"))
Greys.Anatomy.S01.E08 (subtitle 258 783.482 787.083 (sentence (w "I'll") (w
"lighten") (w "up") (w "when") (w "I") (w "feel") (w "light"))
Greys.Anatomy.S01.E08 (subtitle 308 927.894 930.208 (sentence (w "You") (w "guys") (w
"don't") (w "even") (w "light") (w "candles") (w "friday") (w "nights"))
Greys.Anatomy.S01.E08 (subtitle 710 2453.413 2455.835 (sentence (w "My") (w
"favorite") (w "color") (w "is") (w "blue") (w "I") (w "don't") (w "like") (w
"light") (w "blue") (w "indigo"))
Greys.Anatomy.S01.E01 (subtitle 97 343.969 346.356 (sentence (w "No") (w "no") (w
"The") (w "white") (w "lead") (w "is") (w "on") (w "the") (w "right"))
Greys.Anatomy.S01.E01 (subtitle 346 1148.397 1149.922 (sentence (w "Well") (w "if")
(w "the") (w "white") (w "cap") (w "fits"))
Greys.Anatomy.S01.E01 (subtitle 489 1562.743 1566.291 (sentence (w "No") (w
"there's") (w "no") (w "white") (w "count") (w "and") (w "she") (w "has") (w "no") (w
"C.T.") (w "lesions") (w "no") (w "fevers"))
Greys.Anatomy.S01.E07 (subtitle 665 1843.832 1847.694 (sentence (w "Digby") (w "s")
(w "post-op") (w "CBC") (w "shows") (w "a") (w "severe") (w "spike") (w "in") (w
"the") (w "white") (w "blood") (w "cell") (w "count"))
Greys.Anatomy.S01.E08 (subtitle 3 14.844 19.825 (sentence (w "white") (w "dress") (w
"prince") (w "charming") (w "who'd") (w "carry") (w "you") (w "away") (w "to") (w
"a") (w "castle") (w "on") (w "a") (w "hill"))
Greys.Anatomy.S01.E08 (subtitle 157 467.417 468.774 (sentence (w "It's") (w "the") (w
"other") (w "white") (w "meat"))
Greys.Anatomy.S01.E08 (subtitle 328 994.336 997.889 (sentence (w "Maybe") (w "one")
(w "of") (w "those") (w "fudgey") (w "things") (w "with") (w "the") (w "white") (w
"squiggle") (w "on") (w "the") (w "frosting"))
```

Como se pode observar o programa encontrou frases com a palavra white mas também com light, uma palavra relacionada.

O WordNet tem dois dicionários disponíveis actualmente, um que está na versão actual – 3.0 – e outro que está na versão 2.1. O mais recente apenas está disponível em Unix. Como a diferença nos testes era pequena, o trabalho foi continuado em Windows por isso todos os testes apresentados ao longo deste relatório são da versão 2.1 mas na tabela seguinte são apresentadas algumas diferenças encontradas:

Métodos	2.1	3.0
getHyponyms		die
getPertainyms	operative, monthly	
getRelated	clear	
getParticiple	living, breaking, pulled, exploded, drawn	
getTopics	head, point	
getTopicMembers		offense, science
getAttributes	sign	
getMemberHolonyms	parents (em duplicado)	parents
getPartMeronyms		couple

Tabela 9: Comparação de resultados entre a versão 2.1 e 3.0 do WordNet

Isto é só um exemplo mas em 800 palavras que foram comparadas nesta pesquisa por os vários tipos de relações que o WordNet retorna, encontrámos menos de 20 diferenças entre a versão 2.1 e 3.0, ou seja, a diferença entre os resultados que o nosso programa retorna se tiver a correr em Windows ou em Unix/Linux não é significativa.

Com o WordNet conseguimos enriquecer as respostas às *queries* dos utilizadores, relacionando as palavras de várias maneiras possíveis mas nesta fase deparámo-nos com um problema, os resultados davam muitas palavras iguais em qualquer série que pesquisávamos. Por exemplo a palavra good ou a expressão yeah não ajudam o utilizador a identificar temas e aparecem em muitas séries.

Esse problema foi resolvido com a introdução do algoritmo que apresento na secção seguinte.

4.2.4 TF*IDF

No nosso programa utilizámos a função de ponderação TF*IDF (*term frequency - inverse document frequency*) para resolver um problema que tínhamos notado ao longo do trabalho, mesmo retirando as stop words, nem todas as palavras que aparecem mais são relevantes. Das muitas listas de stop words que existem, talvez uma ou outra até tenha algumas destas palavras, mas para esse processo não ser manual, o uso do TF*IDF é muito eficaz.

A função TF*IDF teve origem na função de ponderação IDF (*Inverse Document Frequency*), proposta por Sparck Jones (Jones, 1972)

O IDF baseia-se na contagem do número de documentos, numa colecção onde estamos a pesquisar, que contêm o termo em questão. A intuição era de que um termo numa *query* que ocorre em muitos documentos não é um bom discriminador, e deve ser dado menos peso do que um que ocorre em alguns documentos, e a medida foi uma implementação heurística desta intuição.

Essa medida juntamente com o TF (frequência do termo no próprio documento) construíram uma função que está presente em muitos dos esquemas de ponderação de termos.

A fórmula da função é:

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

Legenda:

- f(t, d) – Nº de vezes que o termo **t** aparece no documento **d**;
- max{f(w,d) : w ϵ d} – Nº de vezes que aparece a palavra que tem maior frequência no documento **d**;
- |D| - Nº de documentos que a colecção contem;
- |{d ϵ D : t ϵ d}| – Nº de documentos onde o termo **t** aparece.

A nossa primeira dúvida na utilização deste algoritmo foi associar à fórmula teórica a nossa utilização prática. Fizemos testes em que considerámos que um documento era um episódio mas decidimos considerar cada série para que ficasse mais coerente quando testássemos o programa com várias séries.

Por exemplo ao testar uma colecção de documentos de duas séries com 10 episódios cada, no primeiro teste tínhamos 20 documentos, e na segunda abordagem teríamos 2.

A diferença de resultados ao escolher uma ou outra abordagem é muito grande, por isso é importante saber o que pretendemos ao utilizar a fórmula TF*IDF. Por exemplo a palavra juju, como num dos episódios da Anatomia de Grey aparece muito, se fosse utilizado o d = episódio, esta palavra teria um lugar de realce numa comparação entre séries, num teste feito aparece em terceiro lugar, ao invés, utilizando d = série, já passa para o lugar 129.

As duas abordagens podem ser importantes, a primeira pode servir para eliminar as palavras na fronteira das *stop words* quando fazemos queries apenas a uma série, enquanto a segunda é a que começámos a utilizar quando queremos comparar séries.

Depois da escolha de como iriamos utilizar a fórmula, surgiram os primeiros resultados e tivemos duas constatações imediatas.

Como esta fórmula realça as palavras que só aparecem numa das séries comparadas, em primeiro lugar em quase todas as comparações apareciam nomes de personagens. Este foi um problema que teve de ser resolvido acrescentando os nomes das personagens principais à lista de *stop words*. Esta lista no futuro pode ser utilizada como opcional por exemplo dando ao utilizador a hipótese de querer ver ou não nomes de personagens.

Para se ter a noção, comparando a Anatomia de Grey e o 24, as 10 primeiras palavras que aparecem como resultado do TF*IDF, são todas personagens, o que ia tirar protagonismo a palavras mais interessantes para os temas das séries.

Outra constatação pela mesma razão de serem realçadas as palavras que aparecem mais só numa das séries, foi ao compararmos uma série com muitos palavrões, e outra em que eles não apareciam, essas palavras iam ter muito realce. Por exemplo ao comparar Os Sopranos com a Anatomia de Grey, temos nas primeiras 10 posições, quatro palavrões, e logo nos primeiros 4 lugares.

Embora tivéssemos estes problemas, nas posições seguintes às personagens - e aos palavrões se fosse caso disso – a fórmula TF*IDF relevou-se muito importante para realçar palavras do tema, eliminando as palavras que apenas com o WordNet estavam em destaque, as palavras na fronteira das *stop words*.

4.2.5 Integração com o MovieClouds

A integração com o interface do **MovieClouds** numa fase do trabalho era feita através de ficheiros no formato ".json", neste momento é um ficheiro de texto normal mas que tem algumas regras acordadas. O nosso programa responde dinamicamente às *queries* mas falta da parte do **MovieClouds** alguns detalhes para se poder interligar com o *back-end*. Os ficheiros de texto são a solução no momento.

Há dois tipos de ficheiros acordados, embora só um neste momento esteja a funcionar no **MovieClouds**.

O ficheiro que está a ser utilizado contém até 80 palavras que são as 80 primeiras depois da ordenação por resultado do TF*IDF e tem o seguinte formato:

```
Número : Palavra – Série

[…]

Número : Palavra – Série
```

O número é o resultado do TF*IDF multiplicado por um factor apenas para serem números mais fáceis de trabalhar sem expoente.

A palavra é a que aparece na série e é obtida através do getWordForms, que é o método que mais palavras retorna, isto porque cada método tem a capacidade de retornar determinadas palavras. Os métodos genéricos como é o caso do getWordForms e também do getDerivationallyRelatedForms são os que mais retornam porque devolvem substantivos, verbos, adjectivos e advérbios.

Estudámos a hipótese de utilizar outros tipos de palavras misturando a utilização de mais métodos mas os resultados não eram melhores, por isso manteve-se a utilização apenas do getWordForms para este fim.

Quanto à serie, como a fórmula TF*IDF realça as palavras de cada série, estas palavras só aparecem numa das séries. Apenas muito mais abaixo na ordenação do TF*IDF teremos palavras que aparecem em mais do que uma série.

Se as palavras aparecerem em todas as séries, o resultado é igual a zero, e se tivermos a comparar mais do que duas séries, se a palavra estiver em todas as séries, menos numa, o resultado vai ser perto de zero.

Exemplo do que é enviado para ser processado no **MovieClouds** para comparar as séries, neste caso, Lie to Me, Game of Thrones e Sherlock, e ser apresentado como uma nuvem de palavras aproveitando o número correspondente às palavras como medida:

```
469.2734058474058 : gods - Game.of.Thrones
354.45118952304057 : landing - Game.of.Thrones
349.4589192480682 : sword - Game.of.Thrones
331.5809166579932 : fbi - Lie.to.Me
294.54394622337173 : throne - Game.of.Thrones
284.5594056734269 : iron - Game.of.Thrones
259.59805429856493 : kingdoms - Game.of.Thrones
239.62897319867528 : gates - Game.of.Thrones
230.19088671873791 : mobile – Sherlock
(…)
```

O outro ficheiro que já foi trabalhado mas ainda não é utilizado no interface, contém os momentos da série em que aparecem as palavras do top e as palavras relacionadas para que no interface o utilizador possa escolher uma palavra e ter como resultado todos os momentos do vídeo assinalados em que aparece a palavra e as relacionadas desta.

Como já referido as palavras que aparecem neste top só pertencem a uma série mas as palavras relacionadas já podem pertencer a várias e por isso o utilizador pode no interface ver esses momentos de qualquer série.

O formato enviado é o seguinte:

```
“Número de vezes que aparece”: “Palavra da query”
Série.Temporada.Episódio “Momento inicial” “Momento final” – Legenda
“Palavra relacionada” “Número de vezes que aparece”
Série.Temporada.Episódio “Momento inicial” “Momento final” – Legenda
(…)
“Palavra relacionada” “Número de vezes que aparece”
Série.Temporada.Episódio “Momento inicial” “Momento final” – Legenda
```

Por exemplo:

```
    127 : alright

    TheSopranos.S02.E01 00:19:30,102 00:19:32,604 Alright, tell her she can
stay with us if she wants.

    TheSopranos.S02.E01 00:21:17,709 00:21:21,713 "Alright, just this one
time.

    TheSopranos.S02.E01 00:22:43,795 00:22:45,797 How was the guest bed, did
you sleep alright?

    (...)

    TheSopranos.S02.E13 00:19:30,102 00:19:31,103 It's alright.

    TheSopranos.S02.E13 00:45:17,147 00:45:18,148 Med, it's alright,

    TheSopranos.S02.E13 00:54:32,702 00:54:34,704 Alright, David, take care
of yourself.

    alright 127

    TheSopranos.S02.E01 00:19:30,102 00:19:32,604 Alright, tell her she can
stay with us if she wants.
```

TheSopranos.S02.E01 00:21:17,709 00:21:21,713 "Alright, just this one time.

TheSopranos.S02.E01 00:22:43,795 00:22:45,797 How was the guest bed, did you sleep alright?

(...)

TheSopranos.S02.E13 00:19:30,102 00:19:31,103 It's alright.

TheSopranos.S02.E13 00:45:17,147 00:45:18,148 Med, it's alright,

TheSopranos.S02.E13 00:54:32,702 00:54:34,704 Alright, David, take care of yourself.

fine 508

Greys.Anatomy.S01.E01 00:07:01,579 00:07:02,414 He'll be fine?

Greys.Anatomy.S01.E01 00:07:02,516 00:07:03,329 You'll be fine.

Greys.Anatomy.S01.E01 00:23:34,363 00:23:35,945 He'll be fine, right?

(...)

Greys.Anatomy.S02.E27 00:22:32,300 00:22:32,800 it's fine.

Greys.Anatomy.S02.E27 00:22:54,800 00:22:55,600 That's fine. Go ahead.

Greys.Anatomy.S02.E27 00:29:09,900 00:29:11,000 And I've been fine with that.

24.S01.E01 00:06:24,173 00:06:26,971 OK, fine. Get a hold of the others and bring 'em in.

24.S01.E01 00:13:29,053 00:13:31,044 Fine.

24.S01.E01 00:28:00,053 00:28:04,046 - How you doing? - Fine. Just a few dozen more to go.

(...)

24.S02.E23 00:37:28,733 00:37:30,644 Fine.

24.S02.E23 00:39:39,293 00:39:41,488 - You OK? - I'm fine.

24.S02.E24 00:07:33,893 00:07:36,327 - Fine. - In writing.

TheSopranos.S01.E01 00:03:14,043 00:03:17,547 Fine. Back at work.

TheSopranos.S01.E01 00:12:57,543 00:12:59,545 That's fine. But i will say this:

TheSopranos.S01.E02 00:18:53,599 00:18:56,101 How you doin'? I'm fine, thank you.

(...)

TheSopranos.S02.E12 00:23:11,356 00:23:12,856 I'm good, I'm fine.

TheSopranos.S02.E12 00:32:01,384 00:32:03,887 No it's not, you're gonna be fine.

TheSopranos.S02.E13 00:26:33,525 00:26:35,527 but you're gonna be fine.

okay 492

Greys.Anatomy.S01.E01 00:01:25,135 00:01:27,393 I'm gonna go upstairs and take a shower, okay?

Greys.Anatomy.S01.E01 00:03:02,303 00:03:05,937 Okay, Martin, Robinson, Bond, Hawkins.

```
    Greys.Anatomy.S01.E01 00:06:43,040 00:06:44,761 - I just thought you
wanna know. - Okay.

    (...)

    Greys.Anatomy.S02.E27 00:30:27,900 00:30:30,200 No! I'm not all right.
Okay?

    Greys.Anatomy.S02.E27 00:31:55,300 00:31:56,500 Oh, okay.

    Greys.Anatomy.S02.E27 00:34:15,900 00:34:16,600 Okay, go.

    TheSopranos.S01.E01 00:13:47,459 00:13:50,963 We overindulge him, okay?

    TheSopranos.S01.E01 00:17:40,359 00:17:42,861 You just speak to uncle
junior about Artie, okay?

    TheSopranos.S01.E01 00:20:20,919 00:20:22,905 Okay.

    (...)

    TheSopranos.S02.E13 00:42:48,999 00:42:51,001 Don't snap at me, okay?

    TheSopranos.S02.E13 00:44:00,571 00:44:02,573 Okay?

    TheSopranos.S02.E13 00:47:56,306 00:47:59,309 saying,"yes, ma, okay, ma,
i hear you, ma",
```

4.3 Avaliação

Para avaliar algumas áreas deste trabalho foi pedido a um grupo de 10 pessoas – com média de idades de 34 anos, 4 do sexo masculino e 6 do sexo feminino - que realiza-se algumas tarefas.

Em média os entrevistados tinham um nível de inglês "muito bom", pelo menos referente a vocabulário, ponto muito importante para conseguir avaliar o trabalho.

Outro ponto que queríamos ter em conta era entrevistar pessoas com algum gosto por séries, tendo sido entrevistadas pessoas que apenas viram algumas e outras que são fans que não perdem um episódio de dezenas de séries.

Os mais fanáticos por séries são pessoas que vão a *websites* como o IMDB (www.imdb.com) e que têm alguma curiosidade sobre as séries e por isso lêem secções como "*Storyline*", "*Plot Summary*", "*Synopsis*" e que observam as "*Plot Keywords*" das séries.

De seguida vou seguir o guião para explicar os resultados obtidos.

Pergunta 1, 2 e 3: Qual é a série que pensa que está representada? Com que grau de confiança entre 1 e 5?

No primeiro teste feito, foram apresentadas nuvens com palavras apenas de uma série. Com esse teste queríamos perceber se com a ordenação por frequência das palavras - retirando apenas *stop words* e nomes de personagens principais - era intuitivo identificar a série.

As nuvens eram deste tipo:

Figura 3: Nuvem de palavras da série Donas de Casa Desesperadas

Com 3 nuvens para cada pessoa associar a uma série entre a lista das que já viu, tivemos um resultado de vinte e duas (22) certas e 8 erradas.

As respostas erradas foram maioritariamente entre séries da mesma área, como se pode constatar pela Tabela 10 que tem apenas as respostas erradas.

Nuvem apresentada	Resposta do entrevistado
24	Os Sopranos
Donas de Casa Desesperadas	Family Guy
Lie To Me	CSI
Anatomia de Grey	Dr. House
Lie To Me	Walking Dead
Dr. House	Serviço de Urgência (ER)
Anatomia de Grey	Dr. House
Nip/Tuck	Anatomia de Grey

Tabela 10: Comparação entre os resultados da nuvem apresentada e a resposta do entrevistado

Quatro das oito respostas erradas são entre séries médicas (Anatomia de Grey, Dr. House, Serviço de Urgência e Nip/Tuck).

As palavras relacionadas com família também tiveram na origem dos dois primeiros enganos apresentados na tabela (24 – Os Sopranos e Donas de Casa Desesperadas – Family Guy).

As outras duas respostas erradas (Lie to Me – CSI e Lie to Me – Walking Dead) tiveram como origem palavras da área policial.

O grau de confiança expresso por os entrevistados nas respostas certas foi sempre na casa dos 4-5 (numa escala de 1 a 5), isto porque os entrevistados conseguiam associar várias palavras à série.

De referir que estas nuvens foram escolhidas entre um leque que os entrevistados tinham avisado anteriormente que viam frequentemente ou que tinham visto vários episódios. Assim sendo, em média cada pessoa tinha um leque de 15 séries, das quais eram escolhidas as 3.

Destas nuvens de palavras representativas da série podemos constatar que em 73,3% das respostas foram acertadas.

Um problema que detectámos a fazer estes testes é que fosse que série fosse, acabávamos por ter várias palavras na fronteira das stop words - good, time, people, yeah, etc. - em destaque, tirando brilho a palavras relevantes para a série. É um assunto que nos fez utilizar o TF*IDF ao comparar séries. Da maneira que estamos a utilizar essa função não funciona para uma série mas se adaptarmos como foi testado no início, podemos utilizar essa função para apenas uma série. Outra abordagem seria acrescentar palavras que aparecem muito em quase todas as séries à lista de stop words.

Pergunta 4: Identifica 5 palavras de cada série? E 10? Quais?

A pergunta 4 era baseada numa nuvem em que estavam representadas 3 séries. Estas séries eram escolhidas do leque de cerca de 15 que o entrevistado tinha escolhido, mas desta vez sabia quais eram as 3 em causa.

As nuvens eram deste estilo:

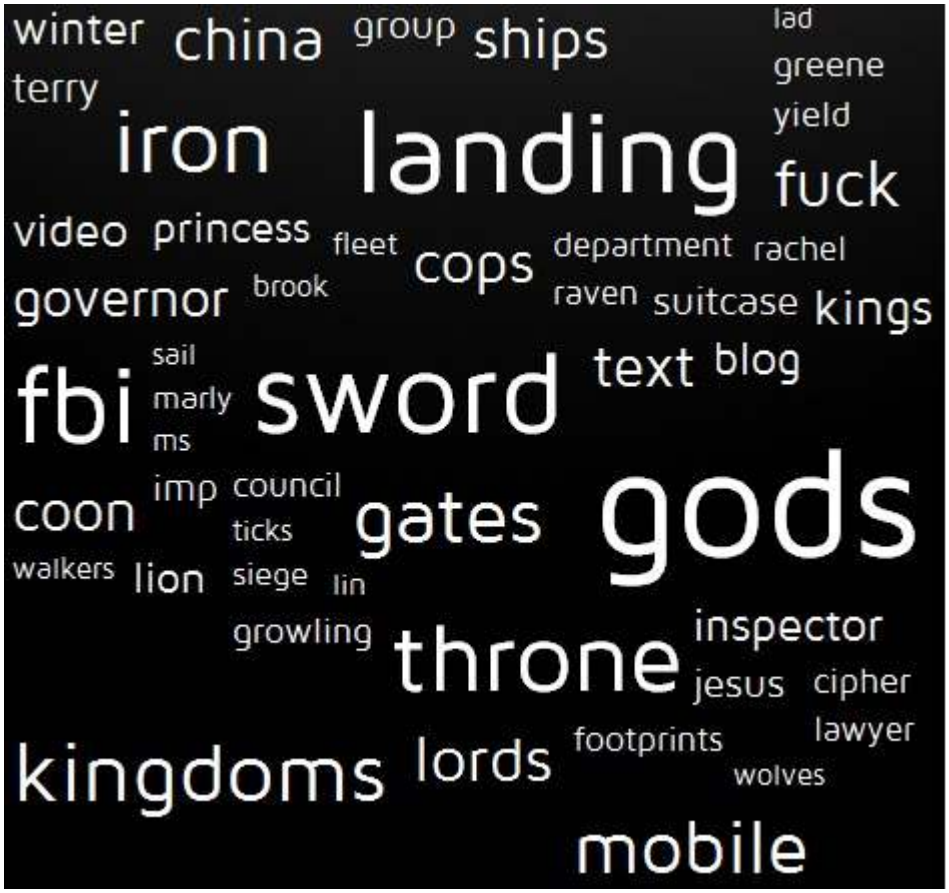

Figura 4: Nuvem de palavras das séries Game of Thrones, Sherlock e Lie to Me

Resumindo os resultados por ocorrências, foram os seguintes:

Nº de palavras certas	Nº de vezes que ocorreu
< 5	11
5	10
5-10	7
10	1
> 10	1

Tabela 11: Resultados com o número palavras ditas associadas a cada série

Os entrevistados disseram sempre pelo menos 5 palavras, o que acontece é que por vezes as palavras eram associadas à série errada.

O que podemos constatar deste teste é que em 19 vezes os entrevistados conseguiram identificar pelo menos 5 palavras duma determinada série. Também não o conseguiram por 11 vezes, e há duas justificações.

A primeira foi os erros entre séries da mesma área, como aconteceu no primeiro conjunto de perguntas com nuvens de uma só série. Confundir palavras entre a Anatomia de Grey e o Dr. House foi o mais vulgar. A segunda justificação foi os entrevistados tentarem acertar nos poucos nomes que aparecem em cada nuvem e muitas vezes erraram.

Palavra	Palavra pertencente à série	Resposta do entrevistado
inflammation	Dr. House	Anatomia de Grey
lymphoma	Dr. House	Anatomia de Grey
lumbar	Dr. House	Anatomia de Grey
mitchell	Lie To Me	Dr. House
finn	Anatomia de Grey	Dr. House
terry	Lie To Me	Donas de Casa Desesperadas
knitting	Anatomia de Grey	Donas de Casa Desesperadas
prom	Anatomia de Grey	Donas de Casa Desesperadas
bongo	Donas de Casa Desesperadas	Lie To Me
immune	Dr. House	Anatomia de Grey
puncture	Dr. House	Anatomia de Grey
andrews	Lie To Me	Anatomia de Grey
interferon	Dr. House	Anatomia de Grey
hahn	Anatomia de Grey	Donas de Casa Desesperadas
santiago	Donas de Casa Desesperadas	Nip/Tuck

Tabela 12: Resultados com as palavras que foram associadas à série errada

O algoritmo TF*IDF permite-nos fazer uma distinção das palavras que aparecem em cada série. Numa nuvem como esta em que só aparecem as primeiras 50 posições do resultado do TF*IDF, essas palavras só aparecem numa série, e por isso, mesmo entre duas séries médicas, as palavras que são apresentadas só existem numa delas, mas é normal que mesmo assim não saibamos com muita certeza de qual é.

De seguida foi utilizada sempre a mesma nuvem para todos os entrevistados.

Pergunta 5: As palavras podem ser agrupadas por temas? Quais?

A nuvem era a seguinte:

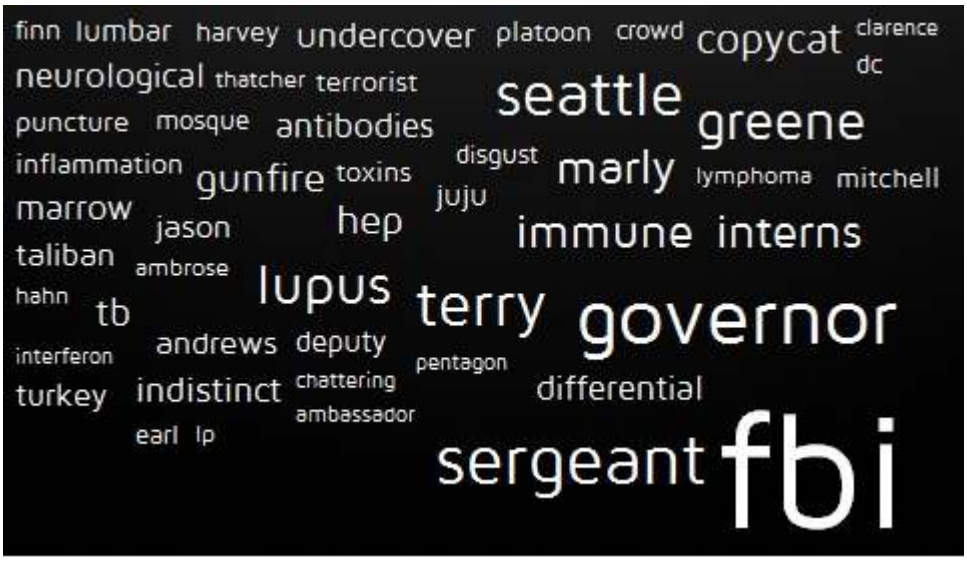

Figura 5: Nuvem de palavras das séries Lie to Me, Dr. House e Anatomia de Grey

Como o exercício de escolha de temas pode ser subjectivo, para avaliar utilizámos as *plot keywords* do IMDB para cada série.

Nas tabelas 13 e 14 apresento as respostas dos entrevistados e das *plot keywords* que são identificadas por os utilizadores ao responderem livremente a que temas associam as palavras. As tabelas estão separadas apenas por ser mais conveniente a formatação desta maneira mas podiam ser paralelas.

	Temas
1	Medicina
2	Medicina
3	Doenças/Sintomas
4	Medicina
5	Hospitais
6	Medicina
7	Medicina
8	Medicina
9	Tratamentos/Doenças
10	Medicina/Saúde
IMDB	**Medicine/Hospital/Medical Treatment/Illness/Medical**

Tabela 13: Resultados com as palavras que correspondem às *plot keywords* do IMDB da Anatomia de Grey e Dr. House

	Temas		
1	Religião	Politica	Polícia/Guerra
2		Lei	
3			
4			Segurança
5			Forças armadas /Guerra/Estado-Maior
6		Politica	Criminal
7	Religião	Politica	
8			Policial
9		Governo/Politica	Investigação/Crime/Terrorismo
10		Politica	Investigação Criminal
IMDB			**Investigation/Police Investigation/Crime Investigation**

Tabela 14: Resultados com as palavras que correspondem às *plot keywords* da Lie to Me e outras que não tiveram correspondência com nenhuma série

Como se pode observar pelas duas tabelas anteriores, há 2 temas que os entrevistados regra geral identificaram e que se cruzam com as *plot keywords* do IMDB que estão também referidas na tabela.

Todos os entrevistados identificaram com muita clareza o que podemos resumir como "Medicina". As palavras às quais os entrevistados associaram este tema, têm uma dupla importância, porque ao cruzarmos duas séries médicas podíamos recear que muitos termos desaparecessem por efeito do TF*IDF e ficarmos com uma nuvem maioritariamente com termos do Lie to Me. Mas o que constatamos, é que mesmo dentro do mesmo tema, cada série acaba por realçar determinadas palavras que não aparecem na outra.

Há uma parte do trabalho que ainda não funciona no MovieClouds mas que quisemos testar com os entrevistados. Como já referido na secção de Integração com o MovieClouds, queremos ter uma nuvem de palavras em que o utilizador ao escolher uma delas, pode aceder a todos os momentos em que aparecem as palavras relacionadas.

O conceito de palavras relacionadas é o que queremos avaliar com este teste. No trabalho realizado avaliei muitas listas de palavras que o WordNet retorna e poderíamos pensar que misturar as todas as listas era o ideal mas há alguns pormenores que ainda têm de ser pensados e os entrevistados deram-nos algum feedback.

Um ponto importante é a capacidade que cada método tem em retornar palavras. Como já foi referido cada método retorna determinadas palavras porque alguns recebem qualquer tipo de palavra como *input* enquanto outros só recebem adjectivos por exemplo. Na tabela 15 é mostrado um comparativo de muitos dos métodos disponíveis no WordNet e a sua capacidade de retorno de palavras. As colunas representam intervalos de realce nos resultados TF*IDF, sendo cada coluna um intervalo dos resultados multiplicado por um factor. Ao analisar esta tabela em que foram testados os vários métodos com o mesmo texto, fazendo um top 60, vemos que por exemplo o método getWordForms retorna palavras sempre relevantes, ao contrário, o método getParticiple só retorna palavras com um resultado TF*IDF muito baixo.

Por esta razão, quando cruzamos os vários métodos, alguns têm pouca probabilidade de contribuir com resultados e por isso decidimos não acrescentar à lista.

Métodos	>100	50-99	30-49	20-29	<20
getWordForms	25	35			
getDerivationallyRelatedForms	9	15	36		
getHypernyms	18	37	5		
getHyponyms	2	18	27	13	
getTroponyms	5	5	22	24	4
getVerbGroup	5	5	6	3	41
getTopics	1	12	15	17	15
getPartHolonyms	2	7	14	13	24
getSimilar			3	2	55
getPartMeronyms		3	9	14	34
getEntailments	2	1	8	5	44
getOutcomes			1	2	57
getRelated			2	1	57
getTopicMembers		1	2	6	52
getAttributes			1		59
getMemberHolonyms			2	3	55
getSubstanceHolonyms				1	59
getPertainyms		2		6	52
getParticiple					60

Tabela 15: Número de palavras associadas a cada tipo de lista

Para o teste cruzamos o resultado de 7 métodos (getWordForms, getDerivationallyRelatedForms, getHypernyms, getHyponym, getTroponyms, getVerbGroup e getTopics), que têm muita probabilidade de contribuir para o resultado final. Depois desse processamento temos como *output* uma lista de palavras que estão ordenadas por a fórmula TF*IDF e para cada uma dessas palavras, é associada uma lista de palavras que são o resultado dos 7 métodos se aplicável para a palavra.

Apresentámos aos entrevistados um conjunto de 50 palavras escolhidas através das palavras com mais realce da nuvem conjunta das séries Lie to Me, Anatomia de Grey e Dr. House.

Pergunta 6: Para cada palavra p está associado um conjunto de palavras. Desse conjunto pode indicar as que pensa serem menos relacionadas com p?

Como para os entrevistados não é transparente o que cada um dos métodos retorna, foram misturados todos os conceitos e foi pedido para assinalarem palavras que considerassem menos relacionadas com a palavra correspondente. Em teoria são todas relacionadas porque estão directamente ligadas à palavra por um dos tipos de listas obtidas por o WordNet, mas o que queríamos perceber era se algum desses 7 métodos retorna palavras que as pessoas identificassem menos com a palavra.

Os resultados foram os seguintes:

Word Forms	20,8%
Derivationally Related Forms	15,3%
Hypernyms	32,5%
Hyponyms	31,2%
Troponyms	39,1%
Verb Group	39,4%
Topics	21,4%

Tabela 16: Resultados com a percentagem de palavras menos relacionadas com determinada palavra

Temos resultados entre 15,3% e 39,4% de palavras que os entrevistados consideraram não relacionar com a palavra. São resultados que não são claros para tirar conclusões finais mas podemos observar que há dois métodos (getTroponyms e getVerbGroup) com resultados muito perto dos 40%.

O que tentámos estudar com este teste, como já referido, ainda não está em funcionamento no interface do **MovieClouds,** quando estiver esperamos que com mais testes feitos se possa confirmar a qualidade de cada lista de palavras.

A ideia, e já está feito do nosso lado, é permitir ao utilizador fazer uma *query* por gunfire e a resposta conterá todos os momentos em que as palavras gunfire, gunshot, shooting e shot aparecem, e no interface esses momentos ficam assinalados no *timeframe* do vídeo, podendo o utilizador ver esses momentos.

Podíamos misturar as listas todas (estas 7 e muitas outras), mas com isso íamos ter muitas palavras nas legendas com a indicação de estarem relacionadas com outra e muitas vezes os utilizadores não iam entender o porquê, por isso queremos tentar encontrar o compromisso óptimo entre encontrar palavas relacionadas mas não todas as que o WordNet consegue obter. Por exemplo, se o utilizador fizer uma *query* por puncture, não é muito benéfico aparecerem todos os momentos em que aparece a palavra make no vídeo.

Para se ter uma noção da divisão por tipos de listas que estavam presentes na lista apresentada aos entrevistados, de seguida é apresentado com cada tipo assinalado por uma cor, e apenas com uma diferença do que foi apresentado aos entrevistados, porque algumas palavras pertencem a duas ou três listas e no teste apareciam só uma vez.

Legenda:

- Word Forms
- Derivational Related Forms
- Hypernyms
- Hyponyms
- Troponyms
- Verb Group
- Topics

fbi - law

governor - governor, rule, regulate, order, politician, control, timer

terry - terry, fabric, cloth, material

immune - immune, resistant, immunity, resistance, person, individual, someone, somebody, mortal, soul, carrier

interns - intern, work, doctor, doc, physician

copycat - copycat, person, individual, someone, somebody, mortal, soul, parrot

hep - hep, hip

gunfire - gunfire, gunshot, shooting, shot

tb - tuberculosis, metal

undercover - secret, undercover, underground

neurological - neurological, neurology

marrow - marrow, substance, core, center, essence, heart, meat, sum, goody, treat, content, stuff

differential - differential, difference, difference, figuring, partial, math

antibodies - antibody, protein, immunoglobulin

turkey - turkey, bomb, dud, flop, bust, tom

deputy - deputy, lieutenant, surrogate, substitute, deputize, assistant, helper, help, agent

juju - juju, voodoo, fetish, magic, charm

terrorist - terrorist, terror, threat, panic, terrorism, radical, sleeper

toxins - toxin, poison

platoon - platoon, military

puncture - puncture, puncture, deflate, pierce, make, create, decompress, break, separate, hole, activity, perforation, prick

inflammation - inflammation, redness, excitement, ignition, firing, lighting, inflame, wake, heat, symptom, arousal, burning, appendicitis, cellulitis, cholecystitis, conjunctivitis, diverticulitis, encephalitis, gastritis, myelitis, pancreatitis, pneumonitis, prostatitis, tendinitis, uveitis, vasculitis, sensation

disgust - disgust, disgust, stimulate, stir, dislike, repulsion, revulsion, horror, nausea, shock, offend, outrage

dc - electricity

earl - earl, peer

crowd - crowd, crowd, crew, gang, bunch, herd, herd, herd, push, move, fill, meet, gather, approach, near, gathering, army, crush, jam, press, drove, swarm, swarm, mob, mob, phalanx, mass, pour, pack, pile

lp - record, disk, disc

lymphoma - lymphoma, cancer

thatcher - thatcher, thatch

interferon - interferon, antiviral

clarence - clarence, rig

pentagon - pentagon, military

chattering - chatter, chatter, chattering, click, chat, chitchat, gossip, jaw, visit, tattle, sound, go, cut, talk, speak, utter, mouth, noise, jawbone

ambassador - ambassador, representative, voice

neurologist - neurologist, neurology, specialist

anthrax - anthrax, disease

brooks - creek, digest, stomach, bear, stand, tolerate, support, suffer, suffer allow, let, accept, swallow, take, undergo, submit, pay

dings - ding, gouge, nick, ring, sound, defect, dig

cane - cane, flog, beat, stalk, stem, switch

bacterial - bacterial, bacteria, bacterium

sec - second, sec, s

projects - project, task, labor, plan, plan, visualize, visualize, fancy, see, see, figure, picture, image, cast, throw, communicate, transmit, transfer, transport, channel, show, draw, imagine, send, direct, assign, work, program, breeze, picnic, snap, pushover, adventure, assignment, baby, enterprise, marathon, no-brainer, thrust, bag, concert, map, offer, introduce, shoot, understand, realize, realise, psychology

hardy - hardy, daring

groom - groom, groom, prepare, prepare, train, train, dress, dress, curry, curry, training, dressing, hand, newlywed, qualify, dispose, clean, shave, comb, arrange,arrange, set, set, do, do, gel, manicure, pedicure, spruce, perfume, scent, develop, discipline, check, condition

groaning - moan, groaning, moaning, utter

bomber - bomber, hero, sub, bomb, airplane, plane, person, individual, someone, somebody, mortal, soul, sandwich, military

psychosis - psychosis, psychotic, paranoia, schizophrenia

hallucinations - hallucination, delusion, delusion, object, trip, disorientation

thatch - thatch, thatch, thatcher, roof

sew - sew, stitch, sewing, fix, secure, fashion, forge, overcast, gather, pucker, tuck, fell, tick

Capítulo 5

Conclusões e Trabalho Futuro

Neste capítulo são apresentadas algumas conclusões sobre o trabalho realizado e são apresentadas algumas das possibilidades que temos como trabalho futuro.

5.1 Conclusões

Este projecto foi muito importante para explorar uma das áreas em que o projecto VIRUS quer marcar diferença. As legendas contidas nas séries, como vimos ao longo do trabalho, têm informação importante para o utilizador.

Neste momento o utilizador já pode ver nuvens de palavras que são retornadas depois de processadas no programa que foi concebido neste trabalho. Também já pronto para ser utilizado pelo interface temos a capacidade de indicar ao utilizador palavras relacionadas com determinada palavra à sua escolha dentro de um leque de palavras apresentadas na nuvem. As relações entre palavras que estamos a utilizar no nosso programa são relações do WordNet.

O *stemming* entre palavras que a dada altura foi colocado de lado por pensarmos não ser o ideal para relacionar palavras, acabou por ser utilizado apenas como uma etapa para não deixar duas palavras com o mesmo *stem* estarem numa nuvem de palavras.

Outro algoritmo que se revelou muito importante foi o TF*IDF, o qual nos permite comparar séries com nuvens de palavras que realça as palavras importantes de cada série e que como constatámos nos testes elaborados com utilizadores, estes conseguiram identificar os temas das séries ao observarem esse conjunto de palavras, objectivo principal deste trabalho.

5.2 Trabalho Futuro

Como trabalho futuro temos alguns pontos a destacar.

Na área do *stemming*, se em vez de utilizarmos algoritmos de *stemming* utilizássemos algoritmos de lematização, teríamos raízes das palavras lexicalmente completas, o que pode ser interessante. É uma área que tem de ser estudada para saber se vale a pena porque algumas das vantagens da lematização já são obtidas pela utilização do WordNet, como é o caso da percepção do sentido da palavra.

No âmbito do WordNet o trabalho ainda a fazer é confirmar se a lista cruzando 7 métodos disponíveis, é o mais eficaz a relacionar palavras ou se é melhor tirar uma ou duas ou até acrescentar mais alguma.

O algoritmo TF*IDF como vimos foi bastante útil para comparar séries e um dos trabalhos que este algoritmo faz, é retirar de posições de destaque, as palavras que aparecem em muitas séries, e não são consideradas *stop words* normalmente. Esse ponto é um dos que tem de ser melhorado na pesquisa por nuvens individuais e por isso um dos testes a fazer em breve pode passar por utilizar o TF*IDF associando o **d** a cada episódio e assim permitir que o algoritmo funcione para séries em separado.

Foi interessante ver que os utilizadores conseguiram identificar os temas das séries em causa, mas como trabalho futuro também será muito importante ser o nosso programa a fazer essa tarefa, como vimos no trabalho relacionado com algoritmos de aprendizagem ou utilizando o WordNet Domains e TextRank.

As legendas também permitem observar sentimentos ao longo das séries, esses sentimentos vão ser processados brevemente no decorrer do projecto VIRUS.

Capítulo 6

Bibliografia

Banerjee, S., & Pedersen, T. (2002). An adapted Lesk algorithm for word sense disambiguation using WordNet. *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-02)*. Mexico City.

Bentivogli, L., Forner, P., Magnini, B., & Pianta, E. (2004). Revising the wordnet domains hierarchy: semantics, coverage and balancing. *Proceedings of COLING Workshop on Multilingual Linguistic Resources*, (pp. 101–108). Geneva.

Brezeale, D., & Cook, D. J. (2008). Automatic Video Classification: A Survey of the Literature. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, pp. 38(3) (2008) 416–430.

Demirtas, K., Cicekli, N. K., & Cicekli, I. (2010). Automatic categorization and summarization of documentaries. *Journal of Information Science*, Volume: 36, Issue: 6, Pages: 671-689.

Gil, N., Silva, N., Dias, E., Martins, P., Langlois, T., & Chambel, T. (2012). Going Through the Clouds: Search Overviews and Browsing of Movies. *MindTrek'12*. Tampere, Finland.

Hull, D. A., & Gregory, G. (1996). *A Detailed Analysis of English Stemming Algorithms*. Meylan, France: Rank Xerox Research Centre .

Jivani, A. G. (2007). A Comparative Study of Stemming Algorithms. *International Journal*, 2, 2004, 1930-1938.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28, 11-21.

Katsiouli, P., Tsetsos, V., & Hadjiefthymiades, S. (2007). Semantic Video Classification Based on Subtitles and Domain Terminologies. *Proceedings of*

SAMT Workshop on Knowledge Acquisition from Multimedia Content (KAMC). Genoa, Italy.

Lovins, J. B. (1968). Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics*, pp. 11, 22-31.

Mihalcea, R., & Tarau, P. (2004). TextRank – bringing order into texts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Barcelona, Spain.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM* , pp. Vol. 38, No. 11: 39-41.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 235-312.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The pagerank citation ranking: bringing order to the web.* Stanford Digital Library Technologies Project.

Porter, M. (1980). An algorithm for suffix stripping. *Program*, pp. 14 (3): 130-137.

Porter, M. (2001). *Snowball: A language for stemming algorithms.* Obtido em 26 de Setembro de 2012, de Snowball: http://snowball.tartarus.org/texts/introduction.html

Smirnov, I. (03 de 12 de 2008). Overview of Stemming Algorithms. *Mechanical Translation*.

SubRip – Wikipédia, a enciclopédia livre. (s.d.). Obtido em 26 de Setembro de 2012, de http://pt.wikipedia.org/wiki/SubRip

The English (Porter2) stemming algorithm. (s.d.). Obtido em 28 de Setembro de 2012, de Snowball: http://snowball.tartarus.org/algorithms/english/stemmer.html

Toutanova, K., & Manning, C. D. (2000). Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. *EMNLP/VLC*, (pp. 63-70).

Video Information Retrieval Using Subtitles. (s.d.). Obtido em 26 de Setembro de 2012, de http://tl.di.fc.ul.pt/hp/?s=1-10

Willett, P. (2006). The Porter stemming algorithm: then and now. *Program: electronic library and information systems*, pp. 40 (3), 219-223.

Zhu, W., Toklu, C., & Liou, S.-P. (2001). *Automatic News Video Segmentation and Categorization Based on Closed-Captioned Text.* ISIS Technical Report Series 20.

Capítulo 7

Anexos

A Algoritmo Porter

Além das vogais **a**, **e**, **i, o, u**, temos o **y** mas só em algumas condições.

O **y** é considerado uma vogal se for precedido duma consoante. Na palavra toy o **y** é considerado consoante, mas se tivermos a palavra syzygy, já é considerado uma vogal.

Uma consoante será denotada por **c**, uma vogal por **v** Uma lista **ccc** ... de comprimento maior do que 0 será denotado por **C**, e uma lista **vvv** ... de comprimento maior que 0 será indicado por **V**. As palavras podem ser representadas desta forma:

[C]VCVC ... [V]

onde os parenteses rectos denotam presença arbitrária dos seus conteúdos.

Usando (VC){m} para denotar VC repetido m vezes, este pode ser novamente escrita como

[C](VC){m}[V]

m será chamado de "medida" de qualquer palavra ou parte da palavra.

O caso **m = 0** cobre a palavra nula. Aqui estão alguns exemplos:

m=0 tr, ee, tree, y, by.

m=1 trouble, oats, trees, ivy.

m=2 troubles, private, oaten, orrery.

As regras para a remoção de um sufixo será dada sob a forma

(condição) S1 -> S2

Isto significa que, se uma palavra terminar com o sufixo **S1**, e o *stem* que aparece antes de **S1** satisfaz a condição dada, **S1** é substituído por **S2**. A condição é geralmente dada em termos de m, por exemplo,

(m > 1) EMENT ->

Aqui **S1** é EMENT e **S2** é nulo. Isso iria mapear replacement para replac, desde que replac seja uma palavra para a qual **m = 2**.

A parte da condição também pode conter o seguinte:

***S** – o *stem* termina com **S** (e de modo semelhante para as outras letras).

v – o *stem* contem uma vogal.

*d – o *stem* termina com uma consoante dupla (ex: **-tt, -ss**).

*o – o *stem* termina em **cvc**, em que o segundo **c** não é **w, x** ou **y** (ex: **-wil, -hop**).

Além disso pode conter expressões como: **and, or** e **not** (ou seja, operadores de conjunção, disjunção e negação)

De seguida apresento o algoritmo original de Porter, sempre com exemplos à direita.

Passo 1a

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
	SSES	SS	caresses	caress
	IES	I	ponies	poni
			ties	ti
	SS	SS	caress	caress
	S		cats	cat

Passo 1b

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
(m>0)	EED	EE	feed	feed
			agreed	agree
(*v*)	ED		plastered	plaster
			bled	bled
(*v*)	ING		motoring	motor
			sing	sing

Se a segunda ou terceira das regras do Passo 1b for bem-sucedida, é feito o seguinte:

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
	AT	ATE	conflat(ed)	conflate
	BL	BLE	troubl(ed)	trouble
	IZ	IZE	siz(ed)	size

e

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
(*d and not (*L or *S or *Z))		(única letra)	hopp(ing)	hop
			tann(ed)	tan
			fall(ing)	fall
			hiss(ing)	hiss
			fizz(ed)	fizz
(m=1 and *o)		E	fail(ing)	fail
			fil(ing)	file

A regra para mapear para uma única letra provoca a remoção de uma segunda letra

. O **-e** é colocado de volta no **-at**, **-bl** e **-iz**, de modo a que os sufixos -ate, -ble e -ize possam ser reconhecidos depois. Este **e** pode ser removido no passo 4.

Passo 1c

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
(*v*)	Y	I	happy	happi
			sky	sky

Passo 2

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
(m>0)	ATIONAL	ATE	relational	relate
(m>0)	TIONAL	TION	conditional	condition
			retional	retional
(m>0)	ENCI	ENCE	valenci	valence
(m>0)	ANCI	ANCE	hesitanci	hesitance
(m>0)	IZER	IZE	digitizer	digitize
(m>0)	ABLI	ABLE	conformabli	conformable
(m>0)	ALLI	AL	radicalli	radical
(m>0)	ENTLI	ENT	differentli	different
(m>0)	ELI	E	vileli	vile
(m>0)	OUSLI	OUS	analogousli	analogous
(m>0)	IZATION	IZE	vietnamization	vietnamize
(m>0)	ATION	ATE	predication	predicate
(m>0)	ATOR	ATE	operator	operate
(m>0)	ALISM	AL	feudalism	feudal
(m>0)	IVENESS	IVE	decisiveness	decisive
(m>0)	FULNESS	FUL	hopefulness	hopeful
(m>0)	OUSNESS	OUS	callousness	callous
(m>0)	ALITI	AL	formaliti	formal
(m>0)	IVITI	IVE	sensitiviti	sensative
(m>0)	BILITI	BLE	sensibiliti	sensible

Passo 3

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
(m>0)	ICATE	IC	triplicate	triplic
(m>0)	ATIVE		formative	form
(m>0)	ALIZE	AL	formalize	formal
(m>0)	ICITI	IC	electriciti	electric
(m>0)	ICAL	IC	electrical	electric
(m>0)	FUL		hopeful	hope
(m>0)	NESS		goodness	good

Passo 4

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
(m>1)	AL		revival	reviv
(m>1)	ANCE		allowance	allow
(m>1)	ENCE		inference	infer
(m>1)	ER		airliner	airlin
(m>1)	IC		gyroscopic	gyroscop
(m>1)	ABLE		adjustable	adjust
(m>1)	IBLE		defensible	defens
(m>1)	ANT		irritant	irrit
(m>1)	EMENT		replacement	replac
(m>1)	MENT		adjustment	adjust
(m>1)	ENT		dependent	depend
(m>1 and (*S or *T))	ION		adoption	adopt
(m>1)	OU		homologou	homolog
(m>1)	ISM		communism	commun
(m>1)	ATE		activate	activ
(m>1)	ITI		angulariti	angular
(m>1)	OUS		homologous	homolog
(m>1)	IVE		effective	effect
(m>1)	IZE		bowdlerize	bowdler

Depois do passo 4 os sufixos já foram removidos. O passo 5 já é para uns pequenos ajustes.

Passo 5a

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
(m>1)	E		probate	probat
			rate	rate
(m=1 and not *o)	E		cease	ceas

Passo 5b

Condição	Sufixo original	Sufixo alterado	Exemplo original	Exemplo modificado
(m > 1 and *d and *L)		(única letra)	controll	control
			roll	rol

B Resultados detalhados de testes

Tabela 1) Word Forms associados a determinada palavra e contexto.

Palavra	Contexto	Word Forms
president	1	president
	2	President of the United States, United States President, President, Chief Executive
	3	president
	4	president, chairman, chairwoman, chair, chairperson
	5	president, prexy
	6	President of the United States, President, Chief Executive
people	1	people
	2	citizenry, people
	3	multitude, masses, mass, hoi polloi, people, the great unwashed
	4	people
	5	people, populate
	6	people, populate
hey		
minutes	1	minutes, proceedings, transactions
	2	minute, min
	3	moment, minute, second, bit
	4	moment, minute, second, instant
	5	minute, arcminute, minute of arc
	6	minute
	7	hour, minute
today	1	today
	2	today
	3	nowadays, now, today
	4	today
family	1	family, household, house, home, menage
	2	family, family unit
	3	family, family line, folk, kinfolk, kinsfolk, sept, phratry
	4	class, category, family
	5	family, fellowship
	6	family
	7	kin, kinsperson, family
	8	syndicate, crime syndicate, mob, family
senator	1	senator
house	1	house
	2	house

	3	house
	4	family, household, house, home, menage
	5	theater, theatre, house
	6	firm, house, business firm
	7	house
	8	house
	9	house
	10	house
	11	sign of the zodiac, star sign, sign, mansion, house, planetary house
	12	house
	13	house
	14	house, put up, domiciliate
happened	1	happen, hap, go on, pass off, occur, pass, fall out, come about, take place
	2	happen, befall, bechance
	3	happen
	4	happen, materialize, materialise
	5	find, happen, chance, bump, encounter
wanted	1	desire, want
	2	want, need, require
	3	want
	4	want
	5	want
	6	wanted
	7	cherished, precious, treasured, wanted
surgery	1	surgery
	2	surgery
	3	operating room, OR, operating theater, operating theatre, surgery
	4	operation, surgery, surgical operation, surgical procedure, surgical process
talking	1	talk, talking
	2	talk, speak
	3	talk, speak, utter, mouth, verbalize, verbalise
	4	speak, talk
	5	spill, talk
	6	spill the beans, let the cat out of the bag, talk, tattle, blab, peach, babble, sing, babble out, blab out
	7	lecture, talk
	8	talking
hospital	1	hospital, infirmary
	2	hospital

Tabela 2) getDerivationallyRelatedForms

Palavra	Derivationally Related Forms
jack	jack
time	time, clock, well-timed, timer, timing
good	good
president	presidential, presidency, administration
bomb	bomb, fail, bomber, bombing, turkey
people	people, mass
surgery	surgical, operative
guy	guy
fine	fine
grey	grey
man	man, gentleman
doctor	doctor, doc, physician

Tabela 3) Definição das palavras e aproveitamento das suas palavras

Palavra	Definição da palavra num certo contexto	Palavras existentes na definição
president	an executive officer of a firm or corporation	executive, officer, firm, corporation
	the person who holds the office of head of state of the United States government	person, holds, office, head, state, United, States, government
	the chief executive of a republic	chief, republic
	the officer who presides at the meetings of an organization	presides, meetings, organization
	the head administrative officer of a college or university	administrative, college
	the office of the United States head of state	
people	(plural) any group of human beings (men or women or children) collectively	plural, group, human, beings, men, women, children, collectively
	the body of citizens of a state or country	body, citizens, state, country
	the common people generally	common, people
	members of a family line	members, family, line
	fill with people or supply with inhabitants	supply, inhabitants
	furnish with people	furnish
today	the present time or age	time, age
	the day that includes the present moment (as opposed to yesterday or tomorrow)	day, includes, moment, opposed, yesterday, tomorrow
	in these times	times
	on this day as distinct from yesterday or tomorrow	distinct
senator	a member of a senate	member, senate
house	a dwelling that serves as living quarters for one or more families	dwelling, serves, living, quarters, families
	an official assembly having legislative powers	official, assembly, legislative, powers
	a building in which something is sheltered or located	building, sheltered, located
	a social unit living together	social, unit
	a building where theatrical performances or motion-picture shows can be presented	theatrical, performances, motion-picture, shows, presented
	the members of a business organization that owns or operates one or more establishments	members, business, organization, owns, operates, establishments
	aristocratic family line	aristocratic, family, line
	the members of a religious community living together	religious, community
	the audience gathered together in a theatre or cinema	audience, gathered, theatre, cinema
	play in which children take the roles of father or mother or children and pretend to interact like adults	play, children, roles, father, mother, pretend, interact, adults

	(astrology) one of 12 equal areas into which the zodiac is divided	astrology, 12, equal, areas, zodiac, divided
	the management of a gambling house or casino	management, gambling, house, casino
	contain or cover	cover
	provide housing for	housing
happened	come to pass	pass
	happen, occur, or be the case in the course of events or by chance	happen, occur, case, events, chance
	chance to be or do something, without intention or causation	intention, causation
	come into being; become reality	reality
	come upon, as if by accident; meet with	accident, meet
wanted	feel or have a desire for; want strongly	feel, desire, strongly
	have need of	
	wish or demand the presence of	demand, presence
	hunt or look for; want for a particular reason	hunt, reason
	be without, lack; be deficient in	lack, deficient
	desired or wished for or sought	wished, sought
	characterized by feeling or showing fond affection for	characterized, feeling, showing, fond, affection
surgery	the branch of medical science that treats disease or injury by operative procedures	branch, medical, science, treats, disease, injury, operative, procedures
	a room where a doctor or dentist can be consulted	room, doctor, dentist, consulted
	a room in a hospital equipped for the performance of surgical operations	hospital, equipped, performance, surgical, operations
	a medical procedure involving an incision with instruments; performed to repair damage or arrest disease in a living body	procedure, involving, incision, instruments, performed, repair, damage, arrest, living, body
talking	an exchange of ideas via conversation	exchange, ideas, conversation
	exchange thoughts; talk with	thoughts, talk
	express in speech	express, speech
	use language	language
	reveal information	reveal, information
	divulge confidential information or secrets	divulge, confidential, secrets
	deliver a lecture or talk	deliver, lecture
	uttering speech	uttering
hospital	a health facility where patients receive treatment	health, facility, patients, receive, treatment
	a medical institution where sick or injured people are given medical or surgical care	medical, institution, sick, injured, people, surgical, care

Tabela 4) Palavras retiradas da definição de president e definição destas.

Palavra	Palavra na definição	Palavras existentes na definição
president	executive	person, responsible, administration, business, persons, administer, law, manages, government, agency, department, function, carrying, plans, orders
	officer	person, armed, services, holds, position, authority, command, appointed, elected, office, trust, member, police, force, authorized, serve, vessel, direct
	firm	members, business, organization, owns, operates, establishments, taut, tauter, resolute, determination, marked, resolution, shakable, soft, yielding, pressure, strong, subject, revision, change, person's, physical, features, shaking, trembling, liable, fluctuate, fall, securely, established, possessing, tone, resiliency, healthy, tissue, pleasingly, fresh, making, crunching, noise, chewed, fixed, place, unwavering, devotion, friend, vow
	corporation	business, firm, articles, incorporation, approved, state, slang, paunch
	person	human, body, including, clothing, grammatical, category, pronouns, verb, forms
	holds	act, grasping, understanding, nature, meaning, quality, magnitude, power, affected, dominated, time, action, awaited, state, confined, short, stronghold, cell, jail, prison, appendage, object, designed, held, order, space, ship, aircraft, storing, cargo, organize, responsible, position, activity, hold, hands, grip, close, bounds, limit, movement, rightfully, rights, titles, offices, possess, concrete, abstract, sense, mind, convey, conviction, view, lessen, intensity, temper, restraint, limits, remain, condition, maintain, theory, thoughts, feelings, assert, affirm, committed, secure, future, application, physical, support, carry, weight, attention, exhaling, expelling, manner, room, crowding, capable, holding, valid, applicable, true, control, violent, protect, challenge, attack, declare, major, characteristic, stop, bind, obligation, indebted, cover, protection, noise, smell, drink, alcohol, showing, ill, effects, pertinent, relevant, arrange, reserve, advance, resist, confront, resistance, departing, dealing, aim, point, direct, accord, agreement
	office	place, business, professional, clerical, duties, performed, administrative, unit, government, actions, activities, assigned, required, expected, person, group, official, holding, power, workers, religious, rite, service, prescribed, ecclesiastical, authorities, job, organization
	head	upper, human, body, animals, face, brains, single, domestic, animal, responsible, thoughts, feelings, seat, faculty, reason, person, charge, military, formation, procession, pressure, exerted, fluid, source, water, stream, arises, grammar, word, grammatical, constituent, plays, role, tip, abscess, pus, accumulates, length, height, based, size, dense, clusters, flowers, foliage, educator, executive, authority, school, individual, user, soft, drugs, natural, elevation, rocky, juts, sea, rounded, compact, mass, foam, froth, pour, effervescent, liquid, container, nearest, viewer, difficult, juncture, movement, V-shaped, mark, arrow, pointer, subject, matter, issue, line, text, serving, passage, bone, bits, cavity, form, joint, skeletal, muscle, moves, science, tiny, electromagnetic, coil, metal, pole, write, read, magnetic, patterns, disk, plural, obverse, coin, bears, representation, person's,

		striking, tool, nautical, toilet, board, boat, ship, projection, membrane, stretched, taut, drum, oral, stimulation, genitals, travel, advance, leading, member, group, excel, direct, determine, direction, travelling, rise, grow, remove
	state	group, people, comprising, government, sovereign, territory, occupied, constituent, administrative, districts, nation, politically, organized, body, single, respect, main, attributes, federal, department, United, States, sets, maintains, foreign, policies, depression, agitation, chemistry, traditional, states, matter, solids, fixed, shape, volume, liquids, shaped, container, gases, filling, express, words, symbol, formula
	United	act, concert, unite, common, purpose, belief, possess, combination, joined, united, linked, bring, action, ideology, shared, situation, join, combine, characterized, unity, single, entity, involving, joint, activity, relating, people, married
	States	group, people, comprising, government, sovereign, state, territory, occupied, constituent, administrative, districts, nation, politically, organized, body, single, respect, main, attributes, federal, department, United, sets, maintains, foreign, policies, depression, agitation, chemistry, traditional, states, matter, solids, fixed, shape, volume, liquids, shaped, container, gases, filling, express, words, symbol, formula
	government	organization, governing, authority, political, unit, form, community, governed, act, exercising, study, states, units
	chief	person, charge, exercises, control, workers, important, element
	republic	political, supreme, power, lies, body, citizens, elect, people, represent, form, government, head, state, monarch
	presides	act, president
	meetings	formally, arranged, gathering, social, act, assembling, common, purpose, small, informal, casual, unexpected, convergence, joining, place, merge, flow, rivers
	organization	group, people, work, organized, structure, arranging, classifying, persons, committees, departments, body, purpose, administering, act, forming, organizing, business, activity, related, result, distributing, disposing, properly, methodically, ordered, manner, orderliness, virtue, methodical
	administrative	body, faculty, students, institution, higher, education, created, educate, grant, degrees, university, British, slang, prison, complex, buildings, housed
	college	body, faculty, students, establishment, seat, higher, learning, housed, including, administrative, living, quarters, facilities, research, teaching, large, diverse, institution, created, educate, life, profession, grant, degrees

Tabela 5) getHypernym

Palavra	Hypernym
jack	raise, lift, hunt, run, ball, flag, tool, ass
time	measure, schedule, determine, shape, influence, adjust, set, correct, case, example, period, moment, minute, second, instant, reading, indication, experience, term
good	advantage, vantage, quality
president	presidency
sir	man
bomb	attack, arms, bust
surgery	room
people	fill, live, group, family
life	being, existence, experience, history, story, person, someone, somebody, soul, need, time
guy	brace, steady, stabilize, man, image
fine	book, penalty
grey	color, wear

Tabela 6) getHyponym

Palavra	Hyponym
jack	raise, lift, hunt, run, ball, flag, tool, ass
time	measure, schedule, determine, shape, influence, adjust, set, correct, case, example, period, moment, minute, second, instant, reading, indication, experience, term
good	advantage, vantage, quality
president	presidency
sir	man
bomb	attack, arms, bust
surgery	room
people	fill, live, group, family
life	being, existence, experience, history, story, person, someone, somebody, soul, need, time
guy	brace, steady, stabilize, man, image
fine	book, penalty
grey	color, wear

Tabela 7) getTroponyms

Palavra	Troponyms
bomb	nuke, blast, shell
talk	continue, proceed, level, gossip, rap, read, begin, shout, peep, swallow, tone, deliver, present, chatter, bay, whine, sing, preach
told	present, represent, answer, respond, introduce, announce, declare, articulate, note, mention, remark, add, supply, explain, give, sum, indicate, point, signal, disclose, reveal, discover, expose, break, talk, air, repeat, leave, relate, crack, direct, command, require, call, warn, know, separate
man	crew
work	page, wait, assist, busy, cooperate, minister, serve, work, bank, fill, take, drive, labor, labour, dig, job, man, slave, freelance, double, roll, run, cut, service, tool, till, exercise, exploit, form, manipulate, colour, swing, blackmail, pressure, carry, persuade, upset, chip, layer, machine, stamp, beat, hill, throw, model, stir, proof, play, help, feed, use, answer, resolve, strike, guess, break
wait	delay, expect
house	home, accommodate, chamber
understand	comprehend, dig, sense, follow, catch, get, touch, understand, read, see, solve, work, bottom, appreciate
stay	stand, keep, be, visit, stay, stick
thought	hold, think, feel, see, consider, view, suspect, contemplate, muse, speculate, reason, conclude, judge, plan, relate, link, connect, focus, centre, give, pay, devote, know, recognize, recognise, review, refresh, design, propose
happened	break, intervene, give, operate, proceed, go, come, fall, repeat, happen, chance, shine, strike, appear
clear	declare, approve, licence, stump, clear, hop, profit, gross, pay, bear

Tabela 8) getMemberHolonyms

Palavra	Member Holonyms
people	world, humanity, man
man	force
family	order
car	train
division	corps, government, authorities, business, concern
number	series
parents	family
person	people
kid	family
order	class
child	family
beat	beats

Tabela 9) getMemberMeronyms

Palavra	Member Meronyms
people	person, individual, someone, somebody, citizen, soul
man	people
family	child, kid, parente
dead	deceased
men	gang, crew, people
set	volume
boss	bovine, ox
school	staff, fish
police	officer
wold	people
table	row
order	family

Tabela 10) getSubstanceHolonyms

Palavra	Substance Holonyms
pick	fabric, material
air	wind
water	tear, sweat, ice, water
paper	page
coffee	coffee, java
sake	rice
tissue	Being
tea	tea
soy	soy
sand	concrete, beach, spit
oxygen	water
gut	Suture

Tabela 11) getSubstanceMeronyms

Palavra	Substance Meronyms
air	neon
water	water, oxygen
road	pavement
street	pavement
coffee	coffee, coffeine
material	pick, filling
ice	water
tear	water
sutures	gut
rice	sake
sweat	water
concrete	sand

Tabela 12) getPartHolonyms

Palavra	Part Holonyms
minutes	hour, degree
car	elevator, lift
stop	camera, organ
day	day
hold	umbrella, brush, briefcase, cart, lumber, ship
room	building
morning	day
night	day
line	cable, skin, factory, mill
brain	head
hours	day
cut	booty, cards

Tabela 13) getPartMeronyms

Palavra	Part Meronyms
man	build, figure, anatomy, shape, frame, form, flesh, foot, arm, hand, face
minutes	second, sec, s
phone	receiver
car	gas, gun, wing, first, low, high, trunk, reverse, roof, third, window
stop	explosion
house	library, loft, circle, stage
surgery	suturing
life	birth, death, dying, age, past
day	day, noon, night, dark, hour, morning
heart	valve
hospital	clinic
hours	half-hour, minute

Tabela 14) getPertainyms

Palavra	Pertainyms
nuclear	nucleus
home	home
personal	personality, person
presidential	president
national	state, nation, country, land
local	neighbourhood
surgical	operation, surgery
medical	medicine
spinal	spine, back
optic	eye, optic, sight, vision
living	living
abdominal	abdomen, stomach

Tabela 15) getRelated

Palavra	Related
good	best, better, good, respectable, moral, right
dead	extinct
wanted	loved, welcome
ready	prepared, willing
safe	harmless, secure
clear	definite, clear
fine	smooth
kind	considerate, benign, merciful, soft
bad	evil, worse, worst
wrong	evil, wicked, false, inaccurate
hard	demanding, hard, difficult, merciless, tough
open	open, explicit, expressed, public

Tabela 16) getSimilar

Palavra	Similar
good	cracking, great, hot, acceptable, solid, saving, white
fine	close, tight, small
dead	asleep, deceased, gone, assassinated, cold, d.o.a., executed, fallen, late, murdered, extinct, out
wanted	desired, hot
ready	fit, set, waiting
safe	fail-safe, harmless, innocuous
kind	benign, sympathetic, gentle
bad	awful, painful, terrible, sad, sorry, hard, tough, rotten, ill, incompetent, crappy, negative, poor, pretty, rubber, severe, unsuitable
wrong	erroneous, inaccurate, false, mistaken, criminal, base, misguided
live	living, viable, vital
nice	good, pleasant
cut	shredded, punctured, severed, split

Tabela 17) getParticiple

Palavra	Participle
based	establish, base, ground, found
kidnapped	kidnap
held	hold
pulled	pull
posted	post
living	live
breaking	break
operating	function, work, operate, go, run
exploded	burst
collected	gather
filled	meet, satisfy, fill, fulfil
drawn	pull, draw, force

Tabela 18) getAttributes

Palavra	Attributes
good	quality, good
life	alive, live, dead
bad	quality
live	life, living
sex	female
dead	life, living
big	size
lives	alive, live, dead
left	place
close	distance
fun	serious
body	thick, thin

Tabela 19) getTopics

Palavra	Topics
man	military
work	agriculture, physics
stay	law
phone	telephone
stop	music
house	theater, theatre
surgery	surgery
doctor	medicine
hell	faith
feel	medicine
home	baseball
party	law

Tabela 20) getTopicMembers

Palavra	Topic Members
man	body, side, nutrition
car	tunnel, passenger, traction
recording	erase, record, enter
military	active, activated, inactive, armed, unarmed, armoured, awol, operational, fighting, effective, retreat, gun, flyover, demonstration, umbrella, drill, action, battle, conflict, fight, engagement, defense, defence, operation, resistance, maneuver, combat, campaign, mission, sally, support, reinforcement, attack, war, assault, siege, draft, base, billet, bomber, camp, fighter, fort, headquarters, mess, post, platform, attention, strength, posture, strategy, pass, taps, tattoo, order, briefing, damage, casualty, wound, injury, loss, command, army, navy, coastguard, service, force, reserve, company, wing, battery, cavalry, foot, troops, friendly, hostile, horse, rank, commando, detail, column, aviation, head, field, theater, theatre, line, position, sector, aide, captain, colonel, commander, commandant, ranger, enemy, general, lieutenant, major, marshall, officer, man, spy, issue, preparation, break, quarter
plane	log, seat, place, passenger, drift, hunt
war	side, war
surgery	operation, surgery, drain, graft, transplant, ligate, suction
history	history, story
business	dull, slow, hostile, operation, business, film, concern, player, roll
game	move, game, turn, play, side, course
films	film, shoot, take, insert
medicine	chronic, specific, invasive, local, general, vicarious, positive, negative, medicine, therapy, radiation, medication, donor, rejection, festering, infection, symptom, sign, autopsy, prescription, doctor, nurse, transfuse, cure, relieve, dress, operate, drug, dose, bleed, shoot, diagnose, feel, amputate, resect

Tabela 21) getEntailments

Palavra	Entailments
told	compare
work	reason
phone	dial
listen	hear
hold	secure
working	reason
leave	die, go, exit, pass
clear	judge
speak	speak, talk
called	dial
bring	come
left	die, go, exit, pass

Tabela 22) getOutcomes

Palavra	Outcomes
work	exercise, work, turn
stop	stop, halt
kill	die, go, exit, pass
called	meet, gather, assemble
start	begin, start, go
open	open
pressure	act, move
shut	close, shut
worry	worry
close	close, shut
remove	move
light	burn

Tabela 23) getVerbGroup

Palavra	Verb Group
bomb	fail
talk	talk, sing
told	say
man	man
work	work, bring, play, run, shape, form, forge, exercise, make, act, function, operate, go, exploit, process, turn
wait	expect
understand	project, see, figure, picture, image
phone	call
thought	think
happened	happen
clear	clear, net, gain, make, earn, realize, realise
hear	witness, find, see

Tabela 24) getUsages

Palavra	Usages
people	plural, plural form
man	colloquialism
hold	archaism, archaicism
baby	slang, cant, jargon, lingo, argot, patois, vernacular
hell	colloquialism
feel	slang, cant, jargon, lingo, argot, patois, vernacular
number	colloquialism
gave	slang, cant, jargon, lingo, argot, patois, vernacular
shot	colloquialism
fun	colloquialism
ass	obscenity, smut, vulgarism, filth, dirty word
crap	obscenity, smut, vulgarism, filth, dirty word

C SubRip

Subrip é o nome do tipo de arquivo criado pelo programa que recebe a extensão "**.srt**". Este tipo de arquivo pode ser lido pela maioria dos programas de média e de edição de legendas.

Subrip também é o nome do programa que cria os ficheiros "**.srt**".

Formato de arquivo SubRip: (SubRip – Wikipédia, a enciclopédia livre)

Número da Legenda

Momento inicial --> Momento Final

Texto da legenda (uma ou mais linhas)

Linha em branco

Exemplo:

1

00:00:04,377 --> 00:00:06,242

[Woman] The game:

2

00:00:06,312 --> 00:00:09,042

They say a person either

has what it takes to play