Procedimentos para compilação de um *corpus* composto por legendas e construção de uma ferramenta de *corpus on-line*: o *Corpus of English Language Videos*

Procedures for compiling a corpus composed of subtitles and building an on-line corpus tool: the Corpus of English Language Videos

Lucas Maciel Peixoto[*]
Luiz Fernando Afra Brito[**]

RESUMO: O Corpus of English Language Videos (CELV) é composto por legendas de vídeos em inglês do YouTube, e tem como objetivo servir como recurso didático para o ensino e aprendizagem da língua por meio de uma ferramenta disponibilizada on-line. Este texto apresenta os procedimentos linguísticos e computacionais que foram realizados para o desenvolvimento do CELV, desde a coleta de textos até a construção da ferramenta. Buscou-se embasamento teórico-metodológico na Linguística Computacional e áreas relacionadas, como a Linguística de Corpus, o Processamento de Linguagem Natural e a Recuperação de Informação. Espera-se que a metodologia descrita no texto apresente detalhes suficientes para demonstrar as etapas mais importantes na construção desse tipo de sistema, permitindo a replicação do processo por outros pesquisadores.

PALAVRAS-CHAVE: Linguística Computacional. Linguística de *Corpus*. Ensino de línguas baseado em *corpora*. Compilação de *corpus*. Ferramentas de *corpora* on-line.

ABSTRACT: The Corpus of English Language Videos (CELV) is composed of YouTube video subtitles in English, and aims to provide a resource for the teaching and learning of the language by means of an on-line tool. This text presents the linguistic and computational procedures used for the development of CELV, from collecting texts to building the tool. Theoretical and methodological basis was sought for in Computational Linguistics and related areas, such as Corpus Linguistics, Natural Language Processing and Information Retrieval. It is expected that the methodology described in the text presents enough details to demonstrate the main steps for building this type of system, enabling the replication of the process by other researchers.

KEYWORDS: Computational Linguistics. Corpus Linguistics. Language teaching based on *corpora*. Compilation of *corpora*. On-line corpus tools.

1. Introdução

Este artigo apresenta os procedimentos que foram utilizados para a compilação do *Corpus of English Language Videos* (CELV) e para a construção da ferramenta *on-line* que permite buscas nesse *corpus*. No momento da redação deste texto, o CELV está sendo

[*] Mestrando do Programa de Pós-Graduação em Estudos Linguísticos da Universidade Federal de Uberlândia. Possui Graduação em Letras – Inglês pela mesma universidade.
[**] Graduando em Sistemas de Informação pela Faculdade de Computação da Universidade Federal de Uberlândia.

desenvolvido no âmbito do Programa de Pós-Graduação em Estudos Linguísticos do Instituto de Letras e Linguística da Universidade Federal de Uberlândia. O *corpus* é composto por legendas em inglês de vídeos do site YouTube, selecionadas conforme o país de origem (EUA, Inglaterra, Canadá e Austrália), gênero (tutorial, *vlog*[1] e palestra) e tema (beleza, culinária, música, meio ambiente, política, ciência, tecnologia, viagens e tópicos gerais).

O objetivo do CELV é servir como recurso didático para o ensino e aprendizagem de língua inglesa com base em *corpus*, conforme a abordagem conhecida como Aprendizagem Direcionada por Dados, do inglês *Data-Driven Learning* (JOHNS, 1991). A ferramenta para consulta ao *corpus* foi disponibilizada *on-line*[2], na forma de um sistema de busca com funções encontradas em outros *corpora on-line*, como o *Corpus of Contemporary American English* – COCA[3] (DAVIES, 2014). Tais funções incluem: buscas simples por palavras ou frases, buscas contendo parâmetros variáveis[4], buscas contendo etiquetas morfossintáticas[5], enumeração dos resultados de busca em ordem de frequência, apresentação dos resultados de busca em forma de gráficos comparativos e exibição de linhas de concordância.

Além dessas funções, o CELV possui um novo recurso: é possível acessar os vídeos nos quais se encontram as palavras de interesse, no momento de sua elocução. Tendo em vista o objetivo de se usar essa ferramenta para o ensino de língua inglesa, esse é um recurso que trará novas possibilidades para o trabalho com *corpora* em sala de aula, uma vez que tornará possível analisar as estruturas linguísticas de interesse dentro de um *corpus*, não apenas de forma escrita, mas também ouvindo as palavras e frases sendo enunciadas e assistindo aos vídeos. Sabe-se que esse tipo de recurso audiovisual pode enriquecer o ambiente de aprendizagem de língua estrangeira: segundo Azevêdo (2008, 2009), vídeos oferecem maneiras dinâmicas de aferir conhecimentos gramaticais, ativar *schemata* e realizar atividades de *brainstorming*. De acordo com Quevedo (1994), vídeos são úteis para o ensino de línguas pelo fato de apresentarem elementos da comunicação que não estão disponíveis por meio da escrita, como gestos corporais e expressões e movimentos faciais.

[1] Neste trabalho, entende-se por *vlog* o tipo de vídeo no qual o produtor fala diretamente à câmera sobre determinado tema, que pode ser relacionado à sua vida pessoal ou a viagens. Também existem *vlogs* especializados em assuntos científicos.
[2] www.celvonline.com
[3] http://corpus.byu.edu/coca/
[4] Por exemplo: no CELV, "*open the **"* é uma busca que pode encontrar qualquer palavra no lugar do símbolo *.
[5] Por exemplo: no CELV, "*[v*] the door*" pode encontrar qualquer verbo no lugar da etiqueta [v*].

Como mencionado anteriormente, este artigo apresentará os procedimentos adotados para a construção do CELV, desde a seleção e coleta de legendas do YouTube até o desenvolvimento da ferramenta de busca, procurando incluir detalhes metodológicos tanto linguísticos quanto computacionais, de maneira que pesquisadores interessados em projetos semelhantes possam replicar os procedimentos demonstrados em suas próprias investigações.

2. Opção por arquivos de legenda para compor o *corpus*

Do ponto de vista computacional, legendas são arquivos de texto em formato SubRip (extensão *.srt)*, que seguem uma estrutura própria de organização. A concepção do CELV partiu da observação de um desses arquivos, cuja estrutura é como no seguinte exemplo:

> 1
> 00:00:11,448 --> 00:00:16,259
> Hello once again, percussionists of the internet.
> In this cajon tutorial we'll be looking at a fairly simple hip hop beat.

A primeira linha da estrutura traz uma numeração que estipula a sequência de legendas que aparecerão ao longo do vídeo (neste caso, a legenda acima será a primeira a aparecer). A segunda linha contém uma marcação de tempo que determina o intervalo durante o qual essa legenda será exibida na tela do vídeo (neste caso, a legenda aparecerá na tela no momento 00:00:11,448, e desaparecerá no momento 00:00:16,259). As linhas seguintes contêm a informação textual da legenda propriamente dita.

O motivo da escolha por arquivos de legenda foi a existência das marcações de tempo. Notou-se que essas marcações podem ser usadas para a criação de um *corpus* capaz de exibir informações linguísticas em formato audiovisual. Devidamente formatadas e implementadas em um sistema computacional capaz de recuperá-las, as marcações de tempo permitem a reprodução dos vídeos a partir dos momentos específicos em que cada elocução é feita. Tem-se, assim, um *corpus* no qual a informação textual pode ser visualizada em forma escrita, e, adicionalmente, assistida e ouvida em forma de vídeo.

A escolha pelo YouTube foi feita devido à sua grande seleção de vídeos disponíveis gratuitamente, muitos dos quais contam com legendas escritas por seus produtores. Além disso, o YouTube possui recursos que facilitaram a implementação do sistema, como incorporação de vídeos e reprodução das gravações a partir de momentos específicos.

3. Revisão da literatura

A descrição metodológica que será exposta neste texto é, em parte, relacionada à Linguística, e, em parte, relacionada à computação, o que situa o conteúdo em uma área interdisciplinar, a Linguística Computacional. Portanto, nesta seção de revisão da literatura, serão apresentadas referências relacionadas ao trabalho linguístico e computacional necessário para compilação e análise de *corpora* e para a construção de ferramentas computacionais nesta área, focando alguns conceitos da Linguística de *Corpus*, do Processamento de Linguagem Natural e da Recuperação de Informação.

3.1. Linguística de *Corpus*

Segundo autores brasileiros (BERBER SARDINHA, 2004; VIANA, TAGNIN, 2011) e estrangeiros (BIBER, CONRAD, REPPEN, 1998; TOGNINI-BONELLI, 2001), a Linguística de *Corpus* (LC) é a área da Linguística que se ocupa do trabalho com *corpora*, incluindo sua compilação e análise. *Corpora*, por sua vez, são grandes coleções de textos em formato eletrônico, podendo ser abrangentes ou específicos, dependendo do que se pretende estudar. Podem ser classificados conforme o número de línguas (uma única língua ou várias línguas, permitindo análises paralelas e contrastivas). Também podem ser orais e/ou escritos, sincrônicos ou diacrônicos, entre outras classificações. Um pesquisador interessado em usar *corpora* em investigações linguísticas tem a opção de usar *corpora* existentes, como o COCA, ou compilar seu próprio *corpus*, caso os existentes não atendam às suas necessidades.

A LC e a existência de *corpora* de grande tamanho seriam inconcebíveis sem que houvesse o auxílio de ferramentas computacionais, uma vez que a área trabalha com grandes quantidades de informação textual, impossíveis de se processar manualmente com rapidez, mesmo que por grandes equipes. Portanto, são usados programas de análise lexical, como o Wordsmith Tools (SCOTT, 2008) e o AntConc (ANTHONY, 2014), para efetuar operações de processamento da linguagem, tais como contagem de palavras, geração de listas de frequência, geração de listas de palavras-chave e exibição de linhas de concordância.

Linhas de concordância são extratos de texto retirados de um *corpus*. São exibidas a partir de determinada palavra de busca e listadas em uma tela de concordância. Uma linha de concordância “mostra a palavra de busca no centro da listagem ladeada pelas palavras que ocorreram no texto junto com ela” (BERBER SARDINHA, 2004, p. 272). Isso permite que a

palavra de interesse seja colocada em foco, centralizada na tela do computador, facilitando a observação dos padrões lexicogramaticais que ocorrem no cotexto ao seu redor.

De acordo com Viana (2011), a leitura de uma linha de concordância é diferente da leitura comum que se faz, por exemplo, das frases em um texto. Normalmente, lê-se uma frase a partir de seu início, que é marcado, por exemplo, por uma palavra com letra inicial maiúscula, até o seu fim, marcado por algum sinal de pontuação. Essa leitura é feita com o objetivo de entender o significado da frase. Uma linha de concordância, por outro lado, deve ser lida do centro para as extremidades esquerda e direita, observando as palavras que co-ocorrem com a palavra de interesse em busca de formas de uso dessa palavra específica e não do sentido geral da frase. Quando esse tipo de leitura é feito em várias linhas de concordância, listadas uma abaixo da outra, é fácil detectar os padrões lexicogramaticais que emergem e descrever como determinada palavra é usada. Por esse motivo, as linhas de concordância são uma das ferramentas mais importantes da LC e devem sempre figurar entre os recursos disponíveis em sistemas computacionais de análise de *corpora*.

3.2. Linguística Computacional e Processamento de Linguagem Natural

Segundo Biemann (2007), a Linguística Computacional (LCO) e o Processamento de Linguagem Natural (PLN) são áreas complementares de estudo que lidam com dados linguísticos e seu processamento por meio de ferramentas computacionais. O que distingue as duas áreas é o foco de cada uma e o tipo de especialista que a desempenha. A LCO é uma subárea da Linguística e, portanto, preocupa-se com a solução de problemas linguísticos, usando ferramentas computacionais para alcançar esse objetivo. O PLN é uma subárea da Ciência da Computação que se encarrega do desenvolvimento dessas ferramentas computacionais. Portanto, há uma relação de complementaridade entre as duas áreas: para utilizar ferramentas computacionais na análise linguística, é interessante que o linguista tenha familiaridade com conceitos computacionais; e para desenvolver sistemas de processamento da linguagem, é necessário que o cientista da computação possua conhecimento sobre o funcionamento das línguas. Nesse contexto, do ponto de vista de um especialista em Linguística, um *corpus* é um repositório de dados linguísticos que serve como objeto de estudo. Do ponto de vista de um especialista em computação, um *corpus* é um material amostral que fornece dados para a construção de sistemas computacionais, a partir do qual regras podem ser derivadas e reaplicadas a novos conjuntos de dados.

Halliday (2005) explica que uma das primeiras aplicações dessas áreas foi a construção de sistemas de tradução automática de línguas. A partir de então, foram desenvolvidos outros tipos de sistema, tais como corretores ortográficos e gramaticais, etiquetadores e ferramentas para extração de termos. A metodologia de desenvolvimento dessas aplicações é predominantemente empírica, isto é:

> (...) uma abordagem é empírica se envolve a observação de dados reais, ao invés do uso de exemplos artificialmente construídos ou intuição. O empirismo também é conhecido como um método para levantar ou refutar hipóteses usando observações e experimentos, ou como o raciocínio indutivo (e não dedutivo) baseado nessas observações. (BORDAG, 2007, p. 14)[6]

Uma das atividades de processamento de linguagem focadas neste estudo é a etiquetagem, ou seja, a "atribuição de categorias a porções de texto" (OLIVEIRA, FREITAS, 2006, p. 179). A etiquetagem é importante na análise de *corpora* porque permite que informações linguísticas específicas sejam recuperadas por computador, possibilitando a observação de dados textuais de maneira mais detalhada do que se fosse analisado o texto sem nenhum tipo de categorização.

Um dos tipos mais comuns de etiquetagem automática é a morfossintática (em inglês, *part-of-speech tagging*), que atribui categorias gramaticais a cada palavra de um texto. Uma busca em um *corpus* assim etiquetado pode responder a perguntas gramaticais como, por exemplo, "quais são os substantivos mais comumente usados após o verbo *fazer*?", ou "qual preposição e artigo devem ser usados na frase *ir _ cinema*?". Esses exemplos demonstram a utilidade de um *corpus* com etiquetagem morfossintática para o ensino e aprendizagem de línguas, uma vez que pode ser usado como um recurso didático disponibilizado a professores e alunos para consultas em busca de exemplos de uso de diversas estruturas linguísticas.

3.2.1. O Etiquetador CLAWS

Um dos etiquetadores morfossintáticos existentes para a língua inglesa é o CLAWS[7] (*Constituent Likelihood Automatic Word-tagging System*), desenvolvido na Universidade de

[6]Todas as traduções contidas neste texto são de nossa autoria. No original: Simply put, an approach is empirical if it involves observing real-world data, as opposed to using artificially constructed examples or intuition. It is also known as a method to construct hypotheses or disprove them using observations and experiments, or as the inductive (contrary to deductive) reasoning or formulation of hypotheses based on such observations.
[7]http://ucrel.lancs.ac.uk/claws/

Lancaster pelo centro de pesquisa UCREL[8] (*University Centre for Computer Corpus Researchon Language*) e usado para etiquetar o COCA.

Garside (1996) explica que o CLAWS aplica a etiquetagem morfossintática seguindo seis etapas, que são:

1. o sistema faz a leitura do texto inserido pelo usuário, com reconhecimento de palavras (*tokens*)*;*
2. é atribuída uma lista de possíveis etiquetas a cada palavra do texto inserido, a partir de um *lexicon* que contém um grande número de palavras do inglês, associadas às classificações gramaticais que podem assumir na língua;
3. é possível que algumas palavras do texto inserido não sejam encontradas no *lexicon*. Portanto, na etapa 3, o sistema segue um conjunto de regras pré-estabelecidas para determinar etiquetas aplicáveis a essas palavras;
4. o sistema analisa o cotexto imediato ao redor de cada palavra do texto e o compara com uma biblioteca de padrões lexicogramaticais previamente construída, ajustando as listas de etiquetas possíveis atribuídas nas etapas 2 e 3 para acomodar melhor os padrões conhecidos;
5. o sistema realiza um cálculo probabilístico de cada combinação possível de etiquetas a serem atribuídas a uma dada sequência de palavras e seleciona a combinação mais provável, com base em dados estatísticos;
6. o texto inserido inicialmente é retornado ao usuário, com cada palavra acompanhada de uma etiqueta gramatical escolhida pelo sistema.

Em suma, o CLAWS considera, inicialmente, todas as palavras do texto de forma isolada e lista todas as etiquetas possíveis para cada palavra (etapas 2 e 3). Nesse ponto, de acordo com Garside (1996), considera-se que a etiquetagem está ambígua, pois há mais de uma etiqueta associada a cada palavra. Para alcançar o objetivo final do processo, que é escolher uma única etiqueta para cada palavra, é feito um procedimento de desambiguação (etapas 4 e 5), no qual o sistema considera o cotexto ao redor de cada palavra para decidir quais combinações de etiquetas são impossíveis, eliminando-as, e quais são possíveis, selecionando a mais provável. As etiquetas mais prováveis são aceitas como corretas e retornadas ao usuário

[8] http://ucrel.lancs.ac.uk/

juntamente com o texto inserido. Segundo o autor, esse processo tem uma precisão de, aproximadamente, 95%, dependendo do tipo de texto inserido.

3.3. Recuperação de Informação

A Recuperação de Informação (RI) é uma subárea da Ciência da Computação que se dedica a "encontrar materiais (...) de natureza não estruturada (geralmente textos) que satisfaçam uma demanda por informação, a partir de grandes coleções (geralmente armazenadas em computadores)" (MANNING, RAGHAVAN, SCHÜTZE, 2009, p. 1)[9]. Os autores explicam que materiais de natureza não estruturada são dados que não possuem estrutura clara o suficiente para serem interpretados por um computador. Portanto, cabe ao especialista em RI analisar e organizar esses materiais de maneira a torná-los processáveis por um computador, possibilitando o uso das informações de interesse.

Uma das técnicas envolvidas nesse processo é a indexação, que é a transformação de estruturas textuais em estruturas computacionais, a fim de organizar as informações de maneira otimizada para a implementação de um sistema de busca. Para exemplificar um processo de indexação, considere-se uma coleção de documentos de texto, cada um com um número variável de palavras, armazenada em um computador. A indexação consiste no isolamento de todas as palavras contidas nos documentos, reorganizando-as em uma lista enumerada (índice), na qual cada palavra aparece associada aos documentos em que ocorre. O resultado é uma listagem semelhante a um glossário do tipo que se encontra nas páginas finais de livros técnicos e científicos, no qual os termos específicos que foram usados no livro aparecem listados em ordem alfabética, seguidos das páginas em que aparecem na obra. Essa estrutura de dados permite o acesso rápido a informações específicas, possibilitando a consulta.

Esta seção do texto buscou apresentar as principais referências teóricas e metodológicas relacionadas ao desenvolvimento do CELV, atribuindo especial relevância aos conceitos de linhas de concordância, etiquetagem morfossintática e indexação. Esses conceitos serão necessários para a compreensão dos procedimentos que serão detalhados na seção seguinte.

[9]No original: Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

4. Metodologia

Conforme previamente mencionado, o trabalho com *corpora* comumente requer o uso de recursos computacionais. Sobre esse assunto, Danielsson (2004) comenta que:

> Você não precisa ser um faz-tudo para se tornar um linguista de *corpus*, mas pode ser que você se encontre, depois de anos na área, precisando aprender um pouco de tudo. Além dos desafios teóricos, linguistas de *corpus* enfrentam problemas de ordem prática com editores de texto, ferramentas de concordância, formato de marcação e anotação linguística. (...) Se o trabalho manual não é uma opção, especialmente quando se lida com *corpora* de tamanho muito grande, então outra opção é aguardar até que os funcionários técnicos da sua instituição estejam disponíveis para investigar o problema. Isso geralmente envolve momentos um pouco embaraçosos de dificuldade de comunicação, onde o poder está com aquele que sabe quais tarefas são impossíveis de serem implementadas, enquanto outras tarefas são realizadas quase instantaneamente. (DANIELSSON, 2004, p. 225)[10]

Desse modo, embora o trabalho com *corpora* seja essencialmente linguístico, pode ser útil, para o linguista de *corpus,* familiarizar-se com ferramentas computacionais, de forma a facilitar a comunicação com os profissionais da computação que desenvolverão os sistemas de processamento da linguagem natural. Assim, serão apresentados os procedimentos para o desenvolvimento do CELV, buscando explicitar os aspectos linguísticos e computacionais do processo e demonstrar a parceria interdisciplinar formada para a construção do sistema.

A Figura 1 apresenta as etapas iniciais do trabalho com as legendas, que são arquivos em formato SubRip, posteriormente convertidos para texto não estruturado (em formato *.txt*) e, por fim, transformados em documentos computacionais por meio da indexação. Os subitens seguintes detalharão as etapas do processo: seleção e coleta das legendas; categorização, formatação e etiquetagem; indexação; e o funcionamento do sistema de pesquisa.

[10]No original: You do not need to be a Jack-of-all-trades to become a corpus linguist but you may well find yourself, after years in the field, having had to learn a bit of everything. Alongside the theoretical challenges, corpus linguists fight many practical battles with text editors, concordance tools, mark-up format and linguistic annotation. (…) If manual labour is not an option, particularly when working with very large corpora, then another option is often to wait for your organisation's over-worked technical staff to allocate time for investigating the problem. This usually involves some more or less embarrassing moments of mis-communication, where the power is with the one who knows which tasks are impossible to implement, while other tasks are achievable almost instantaneously.

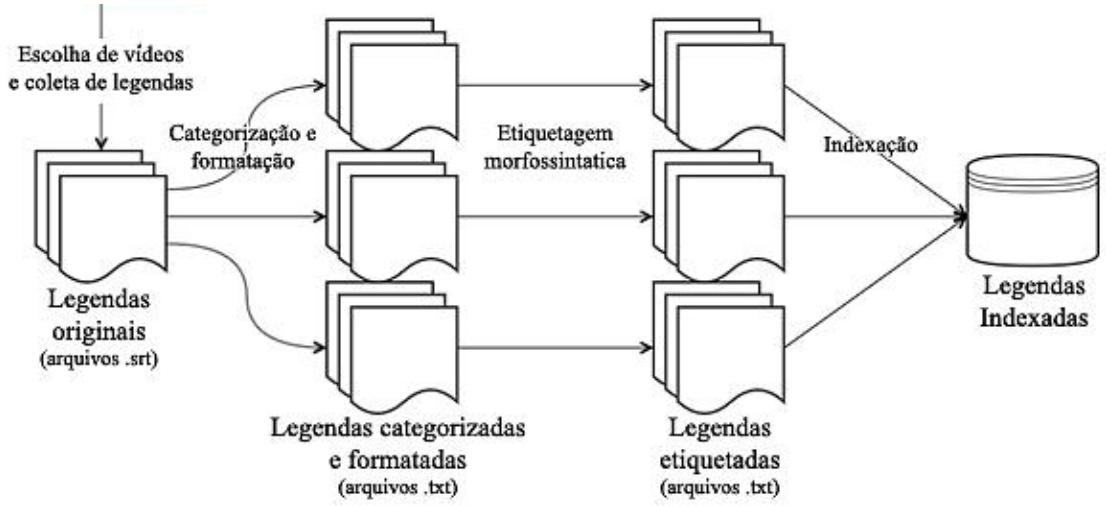

Figura 1. Etapas para compilação do CELV, da coleta à indexação dos textos.

4.1. Procedimentos para seleção, coleta e categorização das legendas

O primeiro passo para iniciar a compilação do CELV foi estabelecer critérios para decidir de quais tipos de vídeo seriam extraídas legendas para compor a amostra. Isso foi feito com base nas recomendações da literatura a respeito do assunto:

> (...) um *corpus* que objetiva representar a totalidade de uma língua precisa abarcar uma ampla gama de gêneros discursivos, contextos de produção, participantes (de diversas faixas etárias, origens geográficas, sexos, classes sociais etc.), entre outros. Ao mesmo tempo, a diversidade deve ser temperada com a concepção de equilíbrio. (VIANA, 2011, p. 28)

Considerando-se que a única fonte de textos do CELV é o YouTube, é evidente que não se pretende com esse *corpus* representar a totalidade de uma língua. No entanto, o objetivo traçado foi elaborar uma amostra a mais representativa e diversa possível, dentro dessa limitação. Tendo isso em vista, estudou-se o que o YouTube tem a oferecer para compor uma boa amostra, por meio de pesquisas preliminares por canais que contivessem um número considerável de vídeos com legendas[11], e selecionaram-se três critérios principais que condizem com as recomendações de Viana (2011):

[11] No YouTube, os produtores de vídeos publicam seu conteúdo em canais, que são associados às suas contas de usuário. Qualquer pessoa navegando pelo site pode acessar o canal de determinado usuário e visualizar uma lista contendo todos os seus vídeos, e os que possuem legendas são identificados por um ícone CC (*Closed Captions*). Essas legendas são, geralmente, escritas pelos próprios produtores dos vídeos, com o objetivo de alcançar um número maior de espectadores, como, por exemplo, pessoas surdas.

- origem geográfica (país): constatou-se que o YouTube possui uma grande quantidade de usuários que produzem vídeos com legendas em (pelo menos) quatro países falantes de língua inglesa: Austrália, Canadá, Estados Unidos, Reino Unido. Os quatro foram incluídos na amostra final;
- gênero discursivo: observou-se que diferentes produtores de conteúdo do YouTube produzem vídeos de diferentes tipos, como tutoriais, *vlogs* e palestras. Cada um desses tipos de vídeo tem sua própria estrutura e formato de apresentação, e os três gêneros foram incluídos na amostra final;
- tema dos vídeos: na amostra selecionada, os temas estão associados aos gêneros, de forma que há: tutoriais sobre culinária, beleza e técnicas musicais; *vlogs* sobre tópicos gerais, viagens e ciência; e palestras sobre meio ambiente, política e tecnologia.

Estabelecidos os critérios, a próxima etapa foi procurar mais canais provenientes de cada um dos países selecionados e especializados nos diversos gêneros e temas escolhidos, incluindo aqueles encontrados nas pesquisas preliminares. Uma vez encontrada uma quantidade de canais que fosse, ao mesmo tempo, de tamanho razoável e relativamente equilibrada entre cada critério, iniciou-se a extração das legendas dos vídeos de cada canal.

A coleta das legendas foi feita com uso do *software* gratuito Google2SRT[12], que extrai arquivos em formato *.srt* a partir do endereço eletrônico dos vídeos no YouTube, bastando copiar o endereço de cada vídeo e inseri-lo na interface do programa usando os atalhos *Ctrl + C* e *Ctrl + V* do teclado. No entanto, o programa não possui recursos para extrair as legendas de vários vídeos com um único clique, tornando necessária a repetição do procedimento para todos os vídeos que compuseram o CELV.

À medida que as legendas eram coletadas, foram adotados procedimentos para organizar os arquivos *.srt* resultantes do processo. Primeiramente, os arquivos foram renomeados seguindo um padrão de nomenclatura que inclui todas as informações importantes sobre cada texto: país de origem, gênero, tema, canal de origem e endereço eletrônico. Posteriormente, os arquivos foram armazenados em diretórios (pastas no computador) correspondentes aos três critérios de organização escolhidos (país, gênero e tema), de maneira a facilitar o acesso a um

[12] http://google2srt.sourceforge.net/pt-br/

texto específico, caso necessário. A Figura 2 demonstra um exemplo de diretório contendo arquivos de legenda nomeados e organizados.

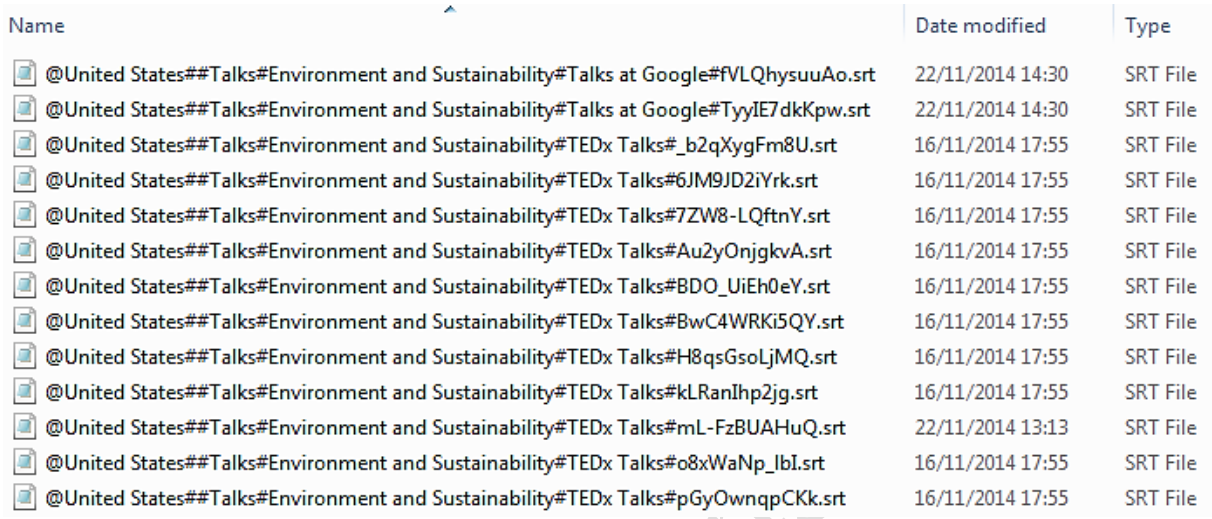

Figura 2. Demonstração da nomenclatura e organização dos arquivos SubRip em um diretório.

O resultado dessa etapa foi um *corpus* composto por cerca de 4.100.000 palavras, provenientes de cerca de 5.300 vídeos do YouTube, organizado em diretórios contendo os arquivos originais de legenda em formato SubRip.

4.2. Procedimentos para formatação e etiquetagem das legendas

Os arquivos originais foram formatados com o objetivo de: (i) simplificar e padronizar sua estrutura para que pudessem ser recuperados mais facilmente durante a indexação e (ii) preparar os arquivos para a etiquetagem do CLAWS. Para isso, foram realizados os seguintes procedimentos:

1. conversão do formato *.srt* para *.txt* com padrão de codificação UTF-8[13];
2. combinação das legendas, formando arquivos maiores e em número menor;
3. inserção de cabeçalhos no início de cada legenda;
4. eliminação da numeração existente nos arquivos de legenda;
5. formatação das marcações de tempo.

[13] Codificação de caracteres é um padrão que permite que caracteres do alfabeto humano sejam armazenados e interpretados por um computador. Um outro exemplo de codificação é o Unicode. A codificação escolhida para determinado arquivo de texto pode interferir na sua compatibilidade com editores de texto e outras ferramentas.

O primeiro programa usado nessa etapa foi o Notepad++[14], um editor de texto gratuito semelhante ao Bloco de Notas, porém com um número maior de funções.

A conversão de todos os arquivos *.srt* foi feita a partir de um *script* escrito em linguagem de programação Python e executado a partir da interface do Notepad++. O *script* encontra todos os arquivos de texto armazenados em um determinado diretório, e os converte para *.txt* em codificação UTF-8.

Após a conversão, foi usado o programa gratuito TXTcollector[15], cuja função é combinar vários arquivos em formato *.txt* em um único arquivo de tamanho maior. Essa combinação foi feita com o objetivo de facilitar as etapas seguintes do processo. O TXTcollector possui o recurso de acrescentar, no arquivo final, um separador entre cada um dos arquivos de texto originais. Esse separador pode ser configurado para ser igual ao nome dos arquivos originais. Como o *corpus* original foi previamente organizado seguindo a nomenclatura de arquivos explicada na seção anterior, o resultado dessa etapa é um número menor de arquivos *.txt* de grande tamanho, contendo várias legendas separadas uma da outra por um cabeçalho gerado automaticamente a partir do nome de cada arquivo.

Nesse momento, os arquivos de texto não estruturado de tamanho maior foram acessados, usando novamente o Notepad++, para formatar as marcações de tempo e os cabeçalhos. Essa etapa é mais facilmente visualizada por meio de um exemplo. Considere-se a seguinte legenda:

> 1
> 00:00:00,190 --> 00:00:05,920
> Hey guys it's Joanne from FifteenSpatulas.com,
> today we are going to make some cheesecake

Após a formatação da marcação de tempo com o uso do Notepad++, essa legenda adquire o seguinte formato:

> <00:00:00>
> Hey guys it's Joanne from FifteenSpatulas.com,
> today we are going to make some cheesecake

[14] http://notepad-plus-plus.org/
[15] http://bluefive.pair.com/txtcollector.htm

O número 1, na primeira linha, foi eliminado, pois a informação sobre a sequência de legendas não é importante para o CELV. A marcação de tempo final foi eliminada, porque apenas a marcação inicial é necessária. Por fim, a marcação inicial foi simplificada, eliminando a informação correspondente aos milissegundos (,190), e colocada entre chaves anguladas (< e >). As chaves anguladas são necessárias porque o etiquetador CLAWS é configurado para ignorar informações que estejam entre essas chaves. A informação que se deseja etiquetar é o conteúdo linguístico das legendas, e não as marcações de tempo.

Abaixo, o resultado final da formatação é exemplificado com a legenda demonstrada acima, antecedida por seu cabeçalho:

> <‡United States†HowTo†Cooking†Fifteen Spatulas†C6VuNiqVr6w>
> <00:00:00>
> Hey guys it's Joanne from FifteenSpatulas.com,
> today we are going to make some cheesecake

Como explicado anteriormente, o cabeçalho foi inserido pelo programa TXTcollector a partir do nome do arquivo. No entanto, também foi necessário inseri-lo entre chaves anguladas, pois essa informação também deve ser ignorada pelo etiquetador. Além disso, foi usado o caractere ‡ para simbolizar o início de cada cabeçalho, e o caractere † para separar as informações de categorização, canal e endereço da legenda. Esses caracteres foram escolhidos por serem incomuns (não aparecem em nenhum outro momento no conteúdo textual dos documentos). Isso possibilita que, durante a indexação, os cabeçalhos sejam facilmente identificáveis e distinguíveis do restante do texto. Qualquer outro par de caracteres esdrúxulos poderia ter sido escolhido para esse fim. Todas essas formatações foram feitas usando expressões regulares[16] no Notepad++.

Por fim, o conteúdo dos arquivos de texto, devidamente formatado, foi inserido na interface *on-line*[17] do CLAWS, que é uma página simples da *internet* contendo algumas opções de etiquetagem e uma caixa de texto para inserção das informações. Essa versão *on-line* do etiquetador etiqueta um número máximo de 100 mil palavras por uso. Considerando-se que o

[16] Expressão regular é um recurso computacional que permite a identificação de padrões textuais ou sequências de caracteres. Por exemplo: as marcações de tempo nos arquivos de legenda seguem o padrão xx:xx:xx,xxx --> xx:xx:xx,xxx, onde x representa um algarismo qualquer que faz parte da informação de tempo. Esse padrão é recuperável pelo processador de expressões regulares do Notepad++, permitindo a formatação de todas as marcações de tempo em um dado documento com um único clique.
[17] http://ucrel.lancs.ac.uk/claws/trial.html

corpus do CELV possui mais de 4 milhões de palavras, foi necessário inserir pequenas porções de texto múltiplas vezes até que toda a amostra estivesse etiquetada. A legenda demonstrada nos exemplos anteriores, quando etiquetada, tem o seguinte formato:

<‡United States†HowTo†Cooking†Fifteen Spatulas†C6VuNiqVr6w>
<00:00:00>
Hey_UH guys_NN2 it_PPH1 's_VBZ Joanne_NP1 from_IIFifteenSpatulas.com_NP1 ,_, today_RT we_PPIS2 are_VBRgoing_VVGKto_TOmake_VVIsome_DD cheesecake_NN1

Observa-se que as informações contidas entre chaves anguladas foram mantidas intactas, e foi inserida uma etiqueta gramatical após cada palavra do texto. A lista de todas as etiquetas usadas (e seu significado) é chamada de *tagset*, e está disponível *on-line*[18].

4.3. Procedimentos para indexação das legendas

Conforme explicitado na seção de revisão da literatura deste texto, a indexação produz uma lista, chamada de índice, que contém as informações de interesse e onde encontrá-las em determinada amostra. Nessa etapa, cada legenda do CELV é chamada de documento, e não está mais associada a um tipo de arquivo como *.srt* ou *.txt*. O índice produzido lista todas as palavras contidas nesses documentos e em quais deles cada palavra pode ser encontrada[19]. Todas as informações que devem ser recuperadas pelo sistema foram indexadas e associadas a cada documento. Há seis tipos principais de informação: país, gênero, tema, canal, endereço na *internet* e conteúdo. A Figura 3 exemplifica um documento contendo essas informações:

[18] http://ucrel.lancs.ac.uk/claws7tags.html
[19] O processo de indexação do CELV, e também os processos de pesquisa, foram implementados utilizando a biblioteca para consulta textual Lucene (http://lucene.apache.org/core/), escrita em linguagem de programação Java e sob licença Apache (http://www.apache.org/licenses/LICENSE-2.0).

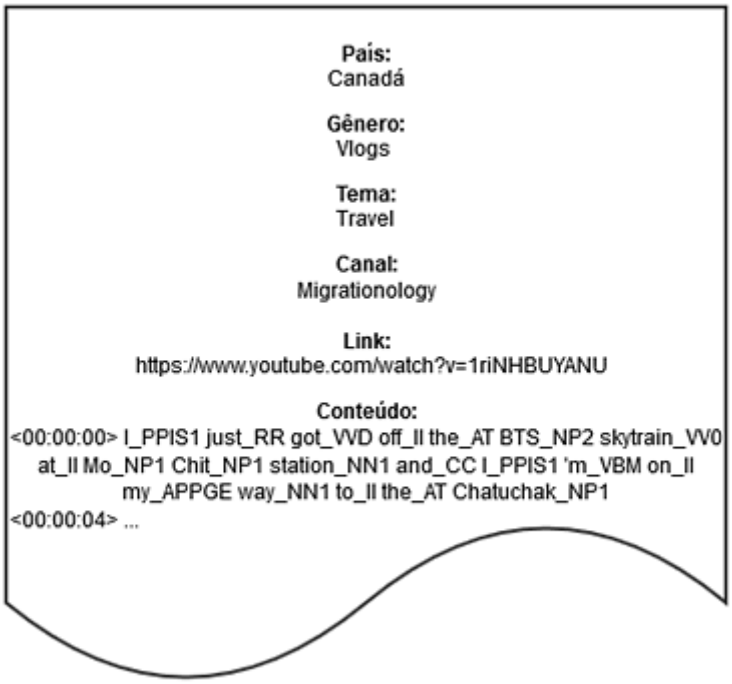

Figura 3. Informações associadas a cada documento após a indexação

Outras informações armazenadas no sistema com a finalidade de otimizar o processo de busca foram: frequência de cada palavra em cada legenda, total de palavras no escopo "país", total de palavras no escopo "gênero", total de palavras no escopo "tema" e uma lista de lemas e seus lexemas correspondentes[20], em língua inglesa.

4.4.Funcionamento do sistema de pesquisa

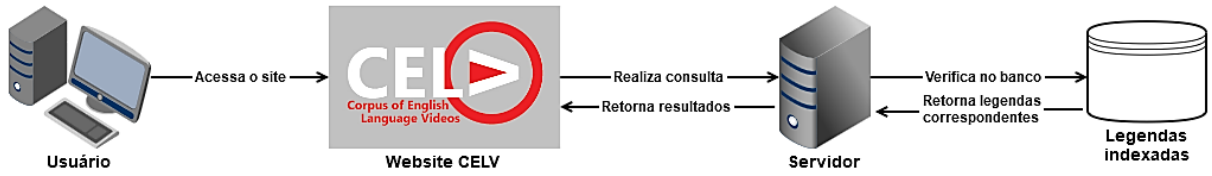

Figura 4. Interação dos elementos do sistema durante o processo de busca.

A Figura 4 demonstra o funcionamento básico do sistema de busca do CELV. Primeiramente, o usuário realiza a sua consulta na página do sistema e envia uma requisição ao servidor. O servidor processa a requisição recebida e busca por legendas compatíveis à consulta

[20] Disponível em http://www.lexically.net/downloads/BNC_wordlists/e_lemma.txt.

no banco de legendas indexadas. A seguir, serão brevemente descritos os tipos de consulta possíveis e as opções de pesquisa disponíveis.

Consultas no CELV podem ser simples ou complexas. Esses dois tipos de consulta possuem limite de até cinco palavras. Essa restrição é necessária para que não haja sobrecarga do servidor, uma vez que o tempo de processamento aumenta conforme a quantidade de palavras na consulta. A consulta simples é a forma mais básica de consulta, usando somente palavras ou sintagmas. Consultas complexas permitem a utilização de parâmetros avançados de pesquisa, que podem ser inseridos no campo de pesquisa por meio de quatro símbolos ou operadores:

- * - qualquer palavra (exemplo: *do* * faz uma busca pelo verbo *do* seguido de qualquer palavra);
- | - uma ou outra palavra (exemplo: *a*|*an* faz uma busca por *a* ou *an*);
- {} – permite buscas lematizadas (exemplo: *{break}* encontra *break*, *breaks*, *breaking*, *broke*, e/ou *broken*);
- [] – insere uma etiqueta gramatical do *tagset* do CLAWS. (exemplo: *work [v*]* faz uma busca pelo verbo *work* seguido de qualquer verbo);

O CELV possui dois modos de pesquisa: lista e gráfico (Figuras 5 e 6). No modo lista, os resultados da pesquisa são listados em ordem decrescente de frequência. Escolher um dos resultados gera uma tabela contendo linhas de concordância. No modo gráfico, os resultados são exibidos em forma de gráficos de barras, que informam a distribuição de ocorrências da palavra consultada entre os países, gêneros e temas da amostra. Assim como no modo lista, selecionar uma das barras do gráfico gera uma tabela contendo linhas de concordância. Nas tabelas de linhas de concordância geradas por qualquer um dos modos de pesquisa, é possível escolher uma das linhas para reproduzi-la em vídeo.

Outras funcionalidades do CELV incluem: filtros de busca, possibilitando limitar os resultados da consulta por gênero, país, tema e canal do YouTube (Figura 7); ordenação alfabética de linhas de concordância a partir de até três palavras à esquerda ou à direita da palavra de busca (*sort* L1, L2, L3, R1, R2, R3); especificação de um número máximo de vídeos para visualização de legendas; especificação de um número máximo de exemplos a serem retirados de cada vídeo; e especificação do momento de início da reprodução dos vídeos. A

próxima seção descreverá, de forma simplificada, os algoritmos usados para a implementação das principais funcionalidades do sistema.

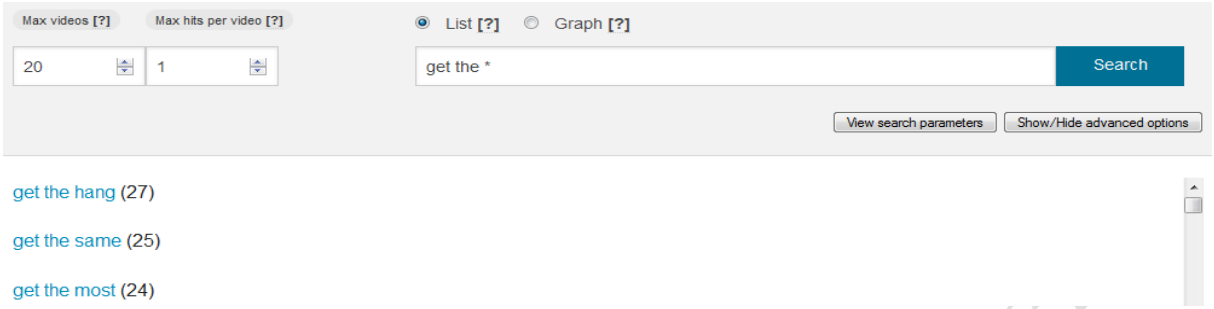

Figura 5. Exemplo de busca com o modo Lista do CELV: os três primeiros resultados da busca por "*get the* *". O número de ocorrências de cada resultado aparece entre parênteses.

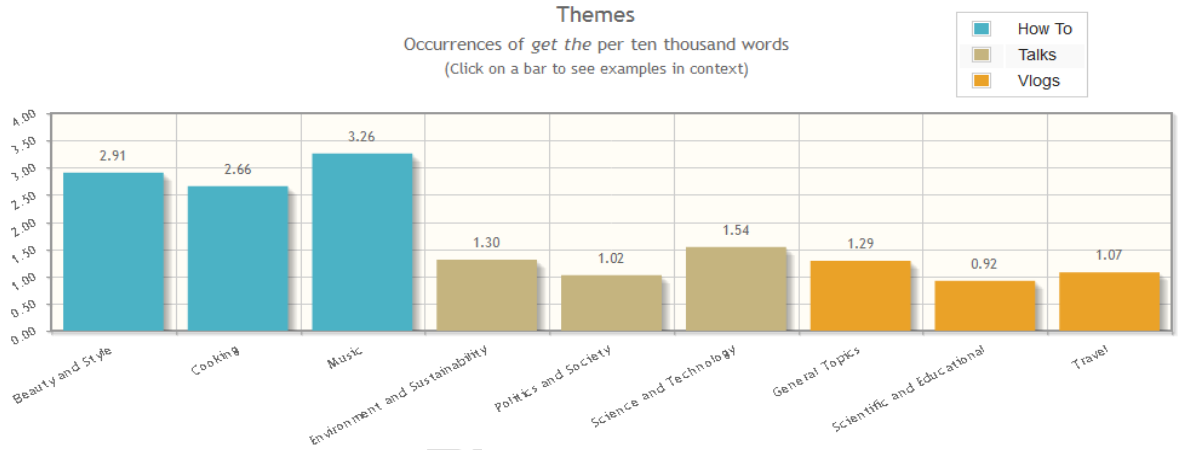

Figura 6. Exemplo de busca com o modo Gráfico do CELV: distribuição de ocorrências de "*get the*" entre os temas que compõem a amostra do *corpus* (a cada dez mil palavras).

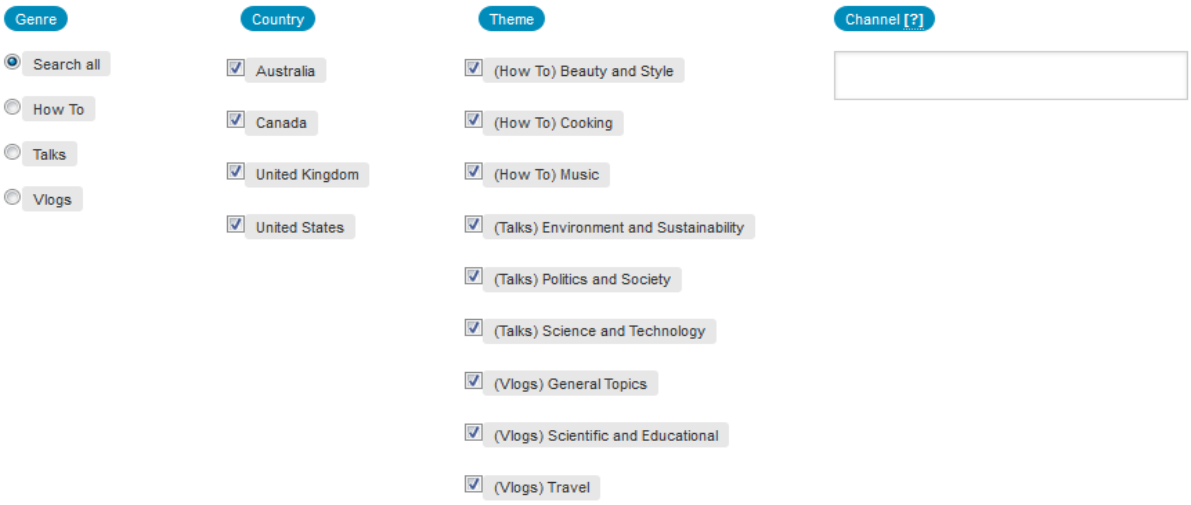

Figura 7. Filtros de busca: gênero, país, tema e canal.

4.4.1. Implementação das funcionalidades de pesquisa

Para que o usuário possa utilizar as funcionalidades mencionadas, foram implementados algoritmos[21] que fazem uso da base de legendas indexadas. Alguns dos principais algoritmos para o funcionamento do sistema são: contagem de frequência, geração de linhas de concordância, ordenação de linhas de concordância e criação de expressões *booleanas*. Esta seção do artigo descreverá esses algoritmos.

O algoritmo de contagem de frequência é utilizado quando uma busca é requisitada pelo usuário e possui quatro etapas de funcionamento:

1. **Pesquisa por termos -** Essa etapa é necessária para consultas complexas, pois esse tipo de consulta possui mais de um resultado a ser retornado. O texto inserido na consulta é decomposto e transformado em uma expressão regular. Posteriormente, são coletadas as ocorrências que correspondem à expressão regular formada. Caso a consulta feita pelo usuário seja simples, essa etapa não é necessária, pois há um único resultado fixo.
2. **Criação de expressão booleana[22] para filtros -** Os filtros selecionados pelo usuário são transformados em uma expressão booleana para que, posteriormente, seja utilizada a estrutura de documento construída na indexação. Desse modo, para uma pesquisa restrita ao país *Canadá*, por exemplo, verifica-se se a informação *país* contém esse valor (*Canadá*). O algoritmo de criação de expressões booleanas para filtros é explicado posteriormente.
3. **Contagem e coleta de frequências -** Nesta etapa é feita a contagem de frequência dos termos encontrados na etapa 1 (ou o termo simples inserido pelo usuário, caso a consulta seja simples), utilizando a expressão booleana criada na etapa 2. O cálculo feito nesta etapa depende do modo de pesquisa selecionado pelo usuário:
    a. **Modo lista –** Para cada termo encontrado, é feita uma consulta na base de legendas indexadas, somando-se a quantidade de vezes em que o termo aparece em cada legenda encontrada. Após essa etapa, obtém-se uma lista com os termos

[21] Na computação, um algoritmo é uma sequência de etapas especificadas por um programador para se realizar uma tarefa. (SOUZA, 2008)
[22] Expressões booleanas utilizam operadores como ∨ (ou), ∧ (e) e ¬ (negação) para expressar determinada operação lógica a ser desempenhada por um computador. (SOUZA, 2008)

de busca, acompanhados por suas respectivas frequências encontradas na base de dados e exibida em ordem decrescente.

b. **Modo gráfico -** Assim como no modo lista, a base de legendas é consultada. Entretanto, neste modo, as frequências dos termos são agrupadas com base nos filtros escolhidos pelo usuário. Para cada tipo de filtro (gênero, país e tema) será feita a soma das frequências dos resultados encontrados na etapa 1 para consultas complexas, ou a frequência do resultado único para consultas simples. Posteriormente, cada valor encontrado é normalizado utilizando a fórmula $\frac{f}{ft} x\ 10000$, onde f representa a frequência dos resultados que satisfaçam uma opção de filtro e ft o número total de palavras presentes nas legendas correspondentes. As quantias obtidas pela fórmula representam a quantidade de vezes em que os resultados aparecem em legendas que satisfaçam determinado filtro a cada dez mil palavras. Após essa etapa, são gerados gráficos de barra que correspondem aos tipos de filtro e às respectivas frequências normalizadas obtidas pela fórmula.

4. **Retorno ao usuário -** A informação obtida na etapa 3 é retornada ao usuário. Para o modo gráfico, além do gráfico de barras, também serão retornados os resultados obtidos na etapa 1 em consultas complexas ou, para consultas simples, o resultado único da busca feita pelo usuário. Esses resultados são armazenados, pois, posteriormente, serão utilizados na geração de linhas de concordância.

O algoritmo de geração de linhas de concordância é utilizado após o usuário selecionar um dos resultados obtidos no modo lista, ou após o usuário selecionar uma das barras apresentadas geradas no modo gráfico. Esse algoritmo possui seis etapas:

1. **Coleta de lista de termos -** Esta etapa somente é necessária em buscas do modo gráfico. Os termos encontrados no algoritmo de contagem de frequência, apresentado anteriormente, são armazenados para a execução deste algoritmo.
2. **Criação de expressão booleana para filtros -** Assim como no algoritmo de contagem de frequência, os filtros selecionados pelo usuário são transformados em uma expressão booleana para que, posteriormente, seja utilizada a estrutura de documento construída na indexação.

3. **Busca de legendas -** Nesta etapa, a base de dados é consultada em busca de legendas que satisfaçam a expressão booleana de filtros obtida na etapa 2. A forma como essa consulta é realizada depende do modo de pesquisa selecionado pelo usuário:
    a. **Modo lista -** No modo lista, somente uma consulta é realizada, correspondendo ao resultado selecionado pelo usuário na listagem de frequência. Após esta etapa, será obtida uma lista de legendas recuperadas.
    b. **Modo gráfico -** No modo gráfico é feita uma consulta para cada termo presente na lista de termos obtida na etapa 1. As legendas obtidas em cada consulta são agrupadas em uma única lista. Após esta etapa, será obtida uma lista de legendas recuperadas por todas as consultas feitas.
4. **Extração de linhas de concordância -** Para cada legenda encontrada na etapa 3, procura-se pelo termo consultado e armazena-se a posição relativa para que futuramente o termo de pesquisa seja destacado. Posteriormente, o cotexto ao redor do termo de pesquisa é reduzido para que caiba em uma linha.
5. **Ordenação de linhas de concordância -** Esta etapa somente é necessária caso o usuário escolha parâmetros de ordenação. Após esta etapa, é obtida uma lista de trechos ordenada alfabeticamente de acordo com os parâmetros escolhidos. Essa ordenação possui seu próprio algoritmo, explicado abaixo.
6. **Retorno ao usuário -** A informação obtida na etapa 4 (ou, caso o usuário escolha parâmetros de ordenação na busca, na etapa 5) é retornada ao usuário e apresentada em forma de tabela.

O algoritmo de ordenação de linhas de concordância é utilizado juntamente ao algoritmo de geração de linhas de concordância quando necessário. Esse algoritmo possui quatro etapas:

1. **Recebimento de argumentos -** É recebida uma lista de linhas de concordância contendo a posição do termo consultado.
2. **Atribuição de parâmetros -** Em cada linha de concordância, encontram-se as três palavras mais próximas do termo consultado à esquerda e à direita. Atribui-se o parâmetro L1 à primeira palavra à esquerda da palavra central, e o parâmetro R1 à primeira palavra à direita da palavra central. Seguindo a mesma lógica, atribuem-se os valores L2, L3, R2 e R3 às demais palavras ao redor da palavra central. Caso não

existam palavras para atribuição de algum parâmetro, é atribuído um valor para representar a falta de palavra.

3. **Ordenação** - Nesta etapa, verifica-se os parâmetros de ordenação selecionados pelo usuário. As linhas de concordância são ordenadas alfabeticamente, de acordo com esses parâmetros.
4. **Retorno** - Por fim, a informação obtida na etapa 3 é retornada ao algoritmo de geração de linhas de concordância.

O algoritmo de criação de expressões booleanas para filtros é utilizado nos algoritmos de contagem de frequência e de geração de linhas de concordância. Esse algoritmo possui quatro etapas:

1. **Recebimento de argumentos –** É recebida uma lista de opções referentes aos três filtros disponíveis no CELV (gênero, país e tema).
2. **Formação de disjunções –** Formam-se disjunções (que são operações lógicas com uso do operador ∨, que significa "ou") para cada lista de opções de filtros. Exemplos de disjunções obtidas nesta etapa são: Talks∨Vlogs, Australia∨Canada, Cooking∨Travel, etc. O operador lógico ∨ tem o significado de união, ou seja, para Talks∨Vlogs serão buscadas tanto legendas do gênero Talks quanto do gênero Vlogs.
3. **Formação de conjunções de disjunções –** Forma-se uma expressão *booleana* conjuntiva para cada disjunção obtida na etapa 2 utilizando o operador lógico ∧ (que significa "e"). O operador lógico ∧ tem o significado de interseção, ou seja, para Vlogs∧Australia, por exemplo, serão buscadas somente legendas que sejam do gênero Vlogs e também do país Australia. Um exemplo de expressão obtida nesta etapa é: Talks∨Vlogs∧Australia∨Canada∧Cooking∨Travel. Nesse exemplo o sistema buscaria as legendas que: sejam dos gêneros Talks ou Vlogs, dos países Astrália ou Canadá e nos temas Cooking ou Travel.
4. **Retorno** - A informação obtida na etapa 3 é retornada e usada nos outros algoritmos anteriormente descritos.

5. Considerações finais e perspectivas futuras

Este texto buscou demonstrar os procedimentos adotados durante a etapa de desenvolvimento do CELV, visando esclarecer pesquisadores interessados no funcionamento

de sistemas de Linguística Computacional. Espera-se, com isso, colaborar com a propagação da Linguística de *Corpus* como metodologia para investigações linguísticas de base empírica, e incentivar a familiarização com recursos computacionais por parte dos linguistas, uma vez que essa relação interdisciplinar entre a Linguística e a Computação tem se demonstrado cada vez mais útil para a análise, descrição, ensino e aprendizagem de línguas.

O desenvolvimento do CELV foi feito a partir da colaboração entre especialistas em linguística e especialistas em computação. Coube aos linguistas a concepção da ferramenta e o levantamento do *corpus*, incluindo o estabelecimento de critérios para seleção de legendas e a sua categorização, formatação e etiquetagem. Os especialistas em computação se encarregaram da indexação da informação textual do *corpus* e do desenvolvimento do sistema de busca, incluindo sua transferência para a *internet* e a elaboração da interface da página do CELV.

A principal dificuldade encontrada no processo de desenvolvimento do CELV foi relacionada à disponibilidade de legendas. O material necessário para a composição do *corpus* tem características bastante específicas: foi necessário buscar vídeos com fala em língua inglesa e legendas também em língua inglesa. A prática de disponibilizar legendas para os vídeos no YouTube é relativamente recente, uma vez que esse recurso passou a existir no *site* a partir de 2007, e, portanto, ainda há um número relativamente pequeno de canais cujos donos se dedicam à produção dessas legendas.

O propósito linguístico da ferramenta, embora não tenha sido o foco deste texto, é o uso do CELV no contexto de ensino-aprendizagem de língua inglesa, servindo como recurso de consulta para que professores e alunos possam obter exemplos de uso de estruturas linguísticas, tanto na forma escrita quanto na forma audiovisual. Propostas para a implementação do CELV nesses contextos serão expostas em trabalhos futuros.

Pretende-se realizar melhorias tanto para o *corpus* quanto para a ferramenta, com a inserção de novos recursos. Serão coletados novos arquivos de legenda para integrar o *corpus*, aumentando seu número total de palavras, e também será feita uma revisão qualitativa das legendas em busca da correção de possíveis erros de digitação, ortográficos ou de formatação, que podem prejudicar o funcionamento do CELV ou limitar seu uso. Também serão acrescentados recursos que possam tornar a ferramenta mais útil para professores e alunos, como um gerador automático de exercícios a partir das linhas de concordância.

Referências Bibliográficas

ANTHONY, L. **AntConc 3.4.3**. Tokyo. Waseda University, 2014. Disponível em: <http://www.laurenceanthony.net/software/antconc/>. Acesso em: 21 Fev. 2015.

AZEVÊDO, C. **Movie Segments to Assess Grammar Goals** [Internet]. Brasília: Claudio Azevêdo. 2008. Disponível em <http://moviesegmentstoassessgrammargoals.blogspot.com.br>. Acesso em: 14 Jul. 2013.

_____. **Movie Segments For Warm-Ups and Follow-Ups** [Internet]. Brasília: Claudio Azevêdo. 2009. Disponível em <http://warmupsfollowups.blogspot.com.br>. Acesso em: 14 Jul. 2013.

BERBER SARDINHA, T. **Linguística de *Corpus***. Barueri, SP: Manole, 2004.

BIBER, D.; CONRAD, S.; REPPEN, R. **Corpus linguistics**: investigating language structure and use. Cambridge: Cambridge University Press, 1998. crossref http://dx.doi.org/10.1017/CBO9780511804489

BIEMANN, C. **Unsupervised and Knowledge-free Natural Language Processing in the Structure Discovery Paradigm**. 199f. Tese (Doutorado em Ciência da Computação). Faculdade de Matemática e Ciência da Computação, Universidade de Leipzig, Leipzig, 2007.

BORDAG, S. **Elements of Knowledge-free and Unsupervised Lexical Acquisition**. 263f. Tese (Doutorado em Ciência da Computação). Faculdade de Matemática e Ciência da Computação, Universidade de Leipzig, Leipzig, 2007.

DANIELSSON, P. Simple Perl programming for corpus work. In: SINCLAIR, J. (Org.) **How to Use Corpora in Language Teaching.** Amsterdam/Philadelphia: John Benjamins B. V., v. 12, 2004, p. 225 – 246.

DAVIES,M. **The Corpus of Contemporary American English**: 450 million words, 1990-present. 2008. Disponível em:<http://corpus.byu.edu/coca>. Acesso em 25 de ago. 2014.

GARSIDE, R. The robust tagging of unrestricted text: the BNC experience. In: THOMAS, J., SHORT, M. (Orgs.) **Using corpora for language research**: Studies in the Honour of Geoffrey Leech. Londres, Inglaterra: Longman, 1996, p. 167 – 180.

HALLIDAY, M. A. K. **Computational and Quantitative Studies**. v. 6. Londres, Inglaterra: Continuum, 2005.

JOHNS, T. F. Should you be persuaded: Two samples of data-driven learning materials. In: JOHNS, T.F.; KING, P. (Orgs.) **Classroom Concordancing**. Birmingham: ELR Journal, v.4, 1991, p. 1 – 16.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. Cambridge, Inglaterra: Cambridge University Press, 2009. Disponível em: <http://nlp.stanford.edu/IR-book/>. Acesso em 22 de fev. 2015.

OLIVEIRA, S., FREITAS, M. C. Classes de palavras e etiquetagem na Linguística Computacional. In: **Calidoscópio**. São Leopoldo, RS. Unisinos. v. 4, n. 3, 2006, p. 179 – 188.

QUEVEDO, A. G. Video use possibilities in autonomous learning. In: LEFFA, V. J. (Org.) **Autonomy in Language Learning**. Porto Alegre: Ed. Universidade UFRGS, 1994, p. 89 – 94.

SCOTT, M. **WordSmith Tools**. Versão 5. Oxford: Oxford University Press, 2008.

SOUZA, J. N. **Lógica para Ciência da Computação**: Uma introdução concisa. Campinas, SP: Campus, 2008.

TOGNINI-BONELLI, E. **Corpus linguistics at work**. Amsterdam: John Benjamins, 2001. crossref http://dx.doi.org/10.1075/scl.6

VIANA, V.; TAGNIN, S. E. O. (Orgs.) **Corpora no ensino de línguas estrangeiras**. São Paulo, SP: HUB Editorial, 2011.

VIANA, V. Linguística de Corpus: Conceitos, Técnicas & Análises. In: VIANA, V.; TAGNIN, S. E. O. (Orgs.) **Corpora no ensino de línguas estrangeiras**. São Paulo, SP: HUB Editorial, 2011.

Artigo recebido em: 28.02.2015
Artigo aprovado em: 03.06.2015