Universidade Federal de Juiz de Fora
Instituto de Ciências exatas
Bacharelado em Ciência da computação

Identificação de videoaulas utilizando técnicas de recuperação de informação com background knowledge

Jayme Siqueira Barbosa

JUIZ DE FORA
MARÇO, 2016

Identificação de videoaulas utilizando técnicas de recuperação de informação com background knowledge

Jayme Siqueira Barbosa

Universidade Federal de Juiz de Fora

Instituto de Ciências exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da computação

Orientador: Jairo Francisco de Souza

JUIZ DE FORA

MARÇO, 2016

Identificação de videoaulas utilizando técnicas de recuperação de informação com background knowledge

Jayme Siqueira Barbosa

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Jairo Francisco de Souza
Dr. em informática (PUC-RIO)

Eduardo Barrére
Dr. em Engenharia de Sistemas e Computação (COPPE/UFRJ)

Victor Ströele de Andrade Menezes
Dr.em Engenharia de Sistemas e Computação (UFRJ)

JUIZ DE FORA

10 DE MARÇO, 2016

Resumo

Recuperar informação é tema recorrente nas áreas de web e engenharia de software. O grande volume de informações publicadas na internet sem que sigam uma estrutura ou padronização vem tornando cada vez mais custoso indexar e recuperar informação na web (Brin & Page, 2012) . Assim, embora exista muita informação disponível, ela não está totalmente acessível aos usuários. Para resolver esses problemas, a comunidade de Banco de Dados tem trabalhado em soluções utilizando dados ligados para recuperar informação levando em consideração contexto e semântica associados à busca. A DBpedia oferece uma grande base de conhecimento organizada em RDF e estabelece ligações com assuntos correlatos de diversas fontes de dados externos, (Harth, 2010). Nossa abordagem é utilizar a DBPedia como fonte de conhecimento para estabelecer relações e identificar videoaulas que possuam assuntos relacionados e possibilitar a navegação pelo conteúdo através facetas.

Palavras-chave: RDF, Busca Facetada, Dados Ligados, Web Semântica, Recuperação de Informação, DBpedia.

Abstract

Information Retrieval is one of the most recurrent subjects in Web and Software Engeneering fields. The huge amount of unstructured information available on the internet have been making indexing and retrieving information on the web increasingly costly (Brin & Page, 2012). Therefore, although there is plenty of information available, it is not fully accessible to users. In an attempt to solve these problems, the database community has been working on solutions using linked data to retrieve information taking into account context and semantics associated with the search. The DBpedia offers a wide knowledge base organized in RDF and links them to related subjects from several different external data sources, (Harth, 2010). Our approach is refer to DBpedia as a source of knowledge to establish relationships to identify video lessons that have related topics and allow to browse by content through facets.

Keywords: RDF, Information Retrieval, Linked Data, Web Semantic, Faceted Search, DBPedia.

Agradecimentos

A Deus pelo dom da vida e por ter permitido que tudo isso acontecesse.

A Maria de Siqueira Barbosa*(in memoriam)*, mãe e mulher guerreira que um dia sonhou ter o filho formado e fez o impossível para que isso acontecesse. Será sempre meu exemplo a ser seguido.

À minha família por ser meu porto seguro nos momentos difíceis, especialmente a minha filha Larissa pela compreensão nos momentos em que me ausentei.

Aos amigos por toda ajuda concedida e pelas palavras de incentivos que foram decisivas para eu continuar e chegar até aqui;

Ao Sandro Coelho, que além de amigo, é responsável por idealizar e iniciar o projeto Qodra, com participação fundamental nos módulos ASR e Anotação Semântica tornando viável desenvolver este trabalho;

A todos os demais integrantes do projeto Qodra, que trabalharam e/ou ainda trabalham para a conclusão deste trabalho e dos demais módulos do projeto;

Ao professor Jairo de Souza pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria;

Aos demais professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

"Nada se sabe sobre o potencial que carrega este poderoso sistema; algum dia poderá chegar a executar música, compor sinfonias e complexos desenhos gráficos."

Ada Lovelace.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

UFJF	Universidade Federal de Juiz de Fora
DCC	Departamento de Ciência da Computação
RI	Recuperação de Informação
RDF	Resource Description Framework
URI	Uniform Resource Indentifier
SPARQL	Sparql Protocol And Rdf Query Language
HTTP	Hyper Text Transfer Protocol
REST	Representational State Transfer
QA	Question Answering
PLN	Processamento de Linguagem Natural
ASR	Automatic Speech Recognition
LIS	Logical Information System
LISQL	LIS Query Language
SQUALL	Semantic Query and Update High-Level Language
GLC	Gramática Livre de Contexto
URL	Uniform Resource Locator
HTML	HyperText Markup Language

1 Introdução

Recuperar informação é tema recorrente nas áreas de web e engenharia de software. O grande volume de dados disponibilizados diariamente, principalmente por usuários, na rede mundial de computadores, podem servir como fontes de dados para inúmeras aplicações, pesquisas e base de conhecimento para outros usuários (Marx et al, 2014).

Porém nem sempre é fácil, para um usuário comum, acessar a informação desejada (Marx et al, 2014). Hoje em dia, em muitas situações, é mais fácil publicar uma informação do que recuperar a mesma. Isso porque para se publicar um novo conteúdo, não é necessário seguir nenhuma padronização. Ou seja, a informação publicada é completamente desestruturada, o que torna cada vez mais custoso indexar e recuperar essa informação. Aliado a isso, os mecanismos de busca utilizados não são capazes de identificar o contexto em que uma informação está associada, tornando os resultados ainda piores.

Um exemplo disso são de sites provedores de conteúdo que geralmente, disponibilizam um mecanismo de busca muito pobre. Na maioria das veze esse mecanismo se resume a um casamento da *string* buscada no texto publicado. Tal experiência fatalmente fará com que um usuário desista de procurar o que deseja ou recorra a outros mecanismos de pesquisa. Nos cenários em que a página já foi indexada é mais fácil encontrar um conteúdo pesquisando em sites de busca do que no próprio site que publica o conteúdo.

Uma alternativa ao problema é estruturar a informação a ser publicada de forma que facilite a busca, para isso, o formato RDF (*Resource Description Framework*) oferece recursos para a estruturação das informações em conformidade com o princípio de dados ligados, permitindo que uma consulta possa levar em consideração o contexto em que a informação está inserida.

1.1 Apresentação do tema e contextualização do problema

Usuários de sistemas de Recuperação de Informação (RI) podem possuir necessidades de informação de complexidade variada. No caso mais simples, eles precisam apenas do endereço de um site associado à informação desejada. Em outros casos, eles necessitam de uma resposta mais precisa (Baeza-Yates & Ribeiro-Neto et al, 2010). Neste último caso, a consulta se torna bem mais complexa pois o sistema precisa retornar um documento ou conjunto de documentos que seja útil ou relevante para o usuário.

Para ser preciso, um sistema de Recuperação de Informação deve interpretar o conteúdo dos documentos nos quais serão pesquisados e ranqueá-los de acordo com a consulta realizada pelo usuário. Essa interpretação envolve a extração de características sintática e semântica do documento em questão, para então determinar sua relevância ou não para uma determinada consulta.

O principal objetivo de um sistema de RI é recuperar todos os documentos relevantes para a consulta do usuário e o mínimo possível de documentos não relevantes (Baeza-Yates & Ribeiro-Neto et al, 2010).

O maior problema aqui é descobrir como utilizar as informações extraídas do documento para determinar sua relevância para a consulta. Isso porque relevância é um conceito pessoal e as métricas de relevância geralmente dependem da formulação da consulta e de seu contexto. Relevância pode mudar dependendo do local e data em que uma consulta é formulada.

Ao utilizar um sistema de recuperação de informação, um usuário precisa traduzir a informação de interesse em uma consulta. Frequentemente, essa consulta não é clara, pois em geral, o usuário não domina completamente o assunto pelo qual está buscando.

Neste contexto, determinar um conjunto de documentos relevantes que contenham palavras-chave existentes na consulta do usuário nem sempre é suficiente para entregar a informação que o usuário precisa (Bizer & Berners-Lee, 2009). Assim, é conveniente para um sistema de RI, determinar um conjunto de palavras que possuam o mesmo sentido semântico das palavras existentes na consulta do usuário, para então identificar a

relevância ou não de um documento para aquela consulta.

Muitas das publicações atualmente são realizadas em formato RDF *(Resource Description Framework)*. Este formato permite que sejam feitas ligações entre dados semelhantes ou anotações de entidades previamente conhecidas através das URIs *(Uniform Resource Indentifier)* que identificam essas classes ou entidades. Em outras palavras, o formato RDF é um grafo que pode ser usado para ligar documentos de classes semelhantes e/ou ligar documentos a entidades das quais eles possuem alguma correlação (Klyne et al, 2006).

Ao recuperar uma informação em RDF pode-se navegar pelo seu grafo de correlações e reconhecer a qual entidade ou classe um determinado recurso pertence, tornando possível reconhecer o contexto e semântica no qual aquele recurso está inserido (Klyne et al, 2006). No entanto é preciso conhecer qual contexto é relevante para o usuário, para ser possível decidir se a informação é suficientemente relevante para ser retornada.

1.2 Justificativa

Uma forma de facilitar a busca por informação é a utilização de facetas. A medida que o usuário navega pelas facetas, pode-se construir a consulta já conhecendo a categoria ou tema que é relevante para o usuário (Guyonvarch et al, 2013). Facetas podem utilizar dados facilmente encontrados em bancos de dados. Por exemplo, facetas para consulta em uma biblioteca poderiam ser: idioma, nível de ensino, editora etc.

Se um sistema utiliza uma base de dados que é construída em RDF, através da navegação pelas facetas pode-se gerar uma consulta em linguagem SPARQL *(Sparql Protocol And Query Language)* para recuperar informação nesta base. Com RDF pode-se ainda consultar recursos externos através de dados ligados (Harth, 2010) para expandir a consulta e sugerir o resultado dessa expansão ao usuário, ou para construir novas facetas que melhor se encaixem aos dados do sistema. Assim, a medida com que os dados externos vão se modificando, o sistema vai se moldando a esses dados, estando portanto em constante aprimoramento.

1.3 Objetivos gerais e específicos

Este projeto possui como objetivo geral melhorar a busca em sistemas de recuperação de informação que utilizam bases de dados em RDF.

Como objetivo específico, o projeto pretende verificar a viabilidade de construir facetas dinamicamente e recomendar videoaulas por meio de consultas a dados externos. E ainda, mostrar como dados ligados podem contribuir para o desenvolvimento da web semântica e verificar que dados externos podem ser utilizados para melhorar os resultados de busca em sistemas de recuperação de informação.

1.4 Metodologia

Para a realização do presente trabalho será implementado o módulo de consultas do Qodra, um *framework* de anotação semântica de vídeos para indexação e busca que está sendo desenvolvido na Universidade Federal de Juiz de Fora (UFJF). Em resumo, o Qodra, que será especificado em detalhes no capitulo 3, é um repositório de videoaulas cujas informações dos vídeos são armazenadas em grafo RDF.

Em nossa abordagem será utilizada a DBpedia como base de conhecimento para ligar o vídeo a recursos ou entidades que representem seu contexto e assuntos relacionados com o tema principal. E uma vez que esses dados estiverem presentes no vídeo, procurar em dados ligados por categorias ou classes que permitam relacionar um recurso a outro e por conseguinte um vídeo a outro.

Para aprimorar a experiência do usuário, planeja-se construir facetas para facilitar a navegação pelos conteúdos do repositório e assim que o assunto de interesse do usuário for identificado, recomendar os vídeos relacionados. Por exemplo, em uma navegação cujo vídeo escolhido possui o tema bioinformática, é conveniente sugerir para o usuário uma videoaula relacionada a biologia.

Espera-se com isso que essa consulta seja mais precisa do que uma busca por palavras-chave contidas no título ou na descrição do vídeo e ainda que retorne um maior número de resultados relacionados. Para avaliar nossa abordagem iremos analisar o *recall* de nossa aplicação.

1.5 Organização do trabalho

Para melhor entendimento das tecnologias utilizadas neste projeto, no capítulo 2 serão detalhados os conceitos relacionados a dados ligados e à DBpedia.

No capítulo 3 será apresentado o *framework* que está em desenvolvimento, especificando em detalhes o módulo Busca e Recomendação e dando uma visão geral dos demais módulos do projeto, assim como a integração dos mesmos.

No capítulo 4 serão apresentados e discutidos os resultados mostrando os pontos positivos e negativos de cada abordagem que foi utilizada para construção dos mesmos e os passos percorridos ate chegar ao resultado final.

Por fim, no capítulo 5 apresenta-se as conclusões, lições aprendidas e deixa sugestões de trabalhos futuros.

2 Pressupostos e Revisão da Literatura

Neste capítulo são abordadas os conceitos básicos e necessários para o desenvolvimento do projeto. A seção 2.1 apresenta os pressupostos teóricos que embasam as tecnologias envolvidas. A seção 2.2 realiza uma revisão dos trabalhos relacionados ao tema.

2.1 Fundamentação Teórica

Para melhor entendimento das tecnologias envolvidas neste projeto, na subseção 2.1.1 são apresentados os conceitos principais sobre dados ligados, a DBpedia é apresentada na subseção 2.1.2.

2.1.1 Dados Ligados

Segundo Bizer & Berners-Lee (2009), dados ligados é uma forma de estruturar a web para criar ligações entre dados de acordo com seu tipo. A proposta da web semântica é realizar ligações entre os dados de fontes diferentes de forma que pessoas e máquinas possam reutilizá-los. Como estes dados estão publicados na web, podemos ter fontes de informações em bancos de dados localizados em diferentes posições geográficas e ainda assim legíveis por máquina, uma vez que temos seu significado explícito por um esquema.

Dados Ligados conta com duas tecnologias que são fundamentais para a web: *Uniform Resource Identifier* (URI) e o protocolo de transferência de hipertexto HTTP. Enquanto (URLs) *Uniform Resource Locator* tornaram-se familiares como endereços de documentos e outras entidades que podem ser localizados na Web, URIs fornecem um meio mais genérico para identificar qualquer entidade que exista no mundo.

URIs e HTTP são complementados por outra tecnologia fundamental, o RDF. Enquanto na web de documentos temos *links* HTML para outros documentos, RDF fornece um modelo genérico baseado em grafo que permite estruturar e interligar dados de diferentes domínios. O resultado é descrito por Bizer & Berners-Lee (2009) como Web de

Dados.

O modelo RDF codifica dados na forma da tripla (sujeito, predicado, objeto), onde o sujeito é a URI que especifica um recurso, o predicado é a URI que especifica a característica do recurso que será descrito e o objeto é o valor resultante entre o relacionamento do recurso com sua propriedade. Um exemplo é mostrado na figura abaixo.

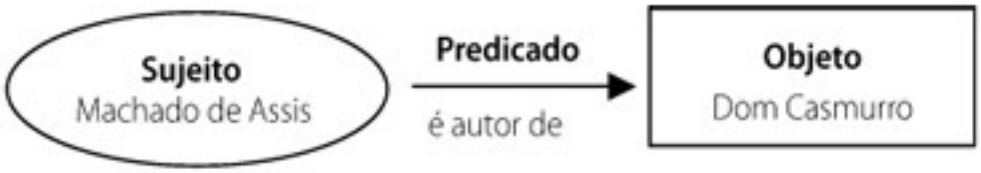

Figura 2.1: Grafo de triplas (sujeito, predicado e objeto)

Para se publicar dados na forma de RDF, os quatro princípios básicos criados por Berners-Lee (2006) devem ser seguidos, são eles:

1. Utilize URIs para identificar recursos;
2. Utilize HTTP URIs para que usuários possam encontrar estes recursos;
3. Ao procurar por URIs, deve-se fornecer informações úteis usando os padrões (RDF, SPARQL);
4. Crie ligações para outras URIs para que seja possível encontrar mais recursos.

Para realizar consultas em dados representados em RDF, utilizamos a linguagem SPARQL. Ela é capaz de realizar consultas em diferentes bases RDF e obtem como resultado um grafo RDF (Hommeaux et al, 2008).

Uma consulta utilizando SPARQL simples é composta por uma cláusula SELECT, onde se especifica quais variáveis devem ser retornadas e uma cláusula WHERE, onde o padrão desejado dos relacionamentos a serem buscados é informado. É utilizando consultas SPARQL que mecanismos de busca rastreiam recursos na Web navegando pelos *links* dos documentos RDF e retornando os recursos relacionados.

2.1.2 DBPedia

O projeto DBPedia extrai conhecimento da Wikipédia e o torna disponível livremente estruturado na forma RDF (Daiber et al, 2013). Utilizando-se dos conceitos de dados ligados e Web Semântica, o projeto extrai conhecimento das 111 diferentes edições da Wikipédia.

A mais importante base de conhecimento da DBpedia é a base extraída a partir da edição em inglês da Wikipédia. Isso é consequência direta do fato desta ser utilizada para consultas e publicações por pessoas de praticamente todas as nacionalidades. Apenas nesta edição, a DBpedia descreve mais de 3,7 milhões de recursos que estão relacionados a mais de 400 milhões de fatos (Lehmann et al, 2014).

A ontologia da DBpedia consiste de 320 classes e 1.650 propriedades. Os mapeamentos são criados através de um esforço de *crowd-sourcing* em todo o mundo e possibilita que o conhecimento das diferentes edições da Wikipédia possam ser combinados nesta única ontologia.

De tempos em tempos o projeto disponibiliza lançamentos de todas as bases de conhecimento da DBpedia e fornece recursos para consultas SPARQL em suas bases. O conteúdo é atualizado periodicamente à medida em que as páginas da Wikipédia são atualizadas ou incrementadas. A DBpedia define mais de 27 milhões de *links* RDF que apontam para mais de 30 fontes de dados externos. Dessa maneira permite que os dados a partir destas fontes possam ser utilizados em conjunto com os dados da DBpedia. Existem centenas de *datasets* que publicam links RDF apontando para DBpedia. Isso faz com que a DBpedia cada vez mais se torne uma central de dados ligados, interligando outros centros de conexões em nuvem (Lehmann et al, 2014).

Segundo Alexa Internet Inc (2015), a wikipédia é o 6º mais popular website do mundo e o 10º mais popular do Brasil. Lehmann et al (2014) descrevem a wikipédia como a enciclopédia mais amplamente utilizada e um dos melhores conteúdos colaborativos já criados. Os artigos da wikipédia consistem de texto livre, *infoboxes*, tabelas e listas de categorização. A grande vantagem da BDpedia é que esta disponibiliza estes dados de forma estruturada e oferece suporte para consultas SPARQL em seu *endpoint*.

O *endpoint* da DBpedia é servido pelo software Virtuoso. Para acessar a interface

de consulta da DBpedia, basta acessar a URL http://dbpedia.org/sparql/. Além da interface de consultas, a DBpedia também oferece suporte a requisições REST, ampliando ainda mais as possibilidades de exploração de seu conteúdo.

2.2 Trabalhos Relacionados

No contexto deste trabalho serão apresentados três trabalhos relacionados, cada um deles com sua parcela de correlação com projeto Qodra. Seguindo a linha de recuperação de informação de maneira mais genérica utilizando dados ligados, é apresentado o OpenQA. No que tange à construção de consultas SPARQL utilizando facetas, apresenta-se o Scalewiles. De outro lado em uma linha de trabalhos relacionadas à recuperação de vídeos, associa-se o YoVisto.

2.2.1 Open QA

Com o significante volume de dados ligados que vem sendo publicados na web, a utilização de dados ligados em sistemas de recuperação de informação já é uma realidade. Dados ligados são fundamentais para agregar valor conjunto de dados originais, enriquecendo-o com associação a outros conjuntos de dados conexos. Dados ligados podem ser utilizados de diferentes maneiras e para diferentes propósitos, é neste contexto que o OpenQA está associado ao Qodra.

O OpenQA (Marx et al, 2014) tem como objetivo recuperar informação para o usuário, baseado em *Question Answering* (QA). Isto é, fornecer uma resposta para uma pergunta do usuário realizada em linguagem natural.

Marx et al (2014) divide o OpenQA em quatro estágios: Interpretação, Recuperação, Sintetização e Resolução. Ainda Segundo Marx et al (2014), a fase de Interpretação é um estagio crucial, em que deve-se entender o que o usuário precisa e determinar como essa pergunta será processada pelas demais partes do sistema. De acordo com (Lopes et al, 2013), a informação que o usuário precisa deve ser traduzida de uma forma em que possa ser avaliada posteriormente. Neste sentido, Burger et al (2003) inclui a interpretação como um dos principais desafios na área de QA. No OpenQA, esse estágio

é realizado através do processamento de linguagem natural realizada sobre a consulta do usuário. No Qodra, o estágio semelhante é realizado através da navegação por facetas.

A fase de recuperação é responsável por coletar a informação na base de dados. Da mesma forma que o Qodra, o OpenQA recupera essas informações através de consultas SPARQL. Em seguida, passa à fase de Sintetização, onde propõe-se a filtrar e eliminar ambiguidades de documentos ou de partes de múltiplas sentenças nas respostas recuperadas, pois elas podem vir de diferentes fontes de dados (Marx et al, 2014).

Assim como o Qodra, o OpenQA também utiliza dados ligados para consultas a bases externas. Enquanto o OpenQA consulta dados externos no intuito de recuperar informação e formular uma resposta para o usuário, no Qodra, consulta-se dados externos para descobrir relações entre os vídeos existentes na base.

2.2.2 Scalewiles

A utilização de facetas para geração de consultas SPARQL não é trivial, porém existem trabalhos relacionados que já utilizaram dessa estratégia para gerar consultas SPARQL. Neste sentido, o Scalewiles relaciona se ao Qodra.

Scalewiles utiliza facetas associadas uma gramática livre de contexto (GLC) para gerar consultas SPARQL, que pode ser aplicada a diferentes bases de dados RDF. Ele é inspirado no mecanismo de busca Sewelis. De acordo com Guyonvarch et al (2013), o Sewelis (Ferré & Hermann, 2012) utiliza linguagens formais para gerar consultas SPARQL incrementalmente e fornece um método seguro e completo no que diz respeito à construção de consultas SPARQL consistentes. A linguagem de consulta de Sewelis é chamada de LISQL *(LIS Query Language)*. Ainda segundo Guyonvarch et al (2013) LISQL é facil de aprender por usuário casuais.

Scalewiles define LISQL2, uma nova versão de LISQL e associa inspirações da linguagem SQUALL. SQUALL que é definida em detalhes em (Ferré, 2012), é uma linguagem natural controlada que pode ser convertida em SPARQL cobrindo a maior parte desta última. Como resultado dessa associação, LISQL2 que também é uma sub linguagem de SQUALL, é na prática uma gramática livre de contexto (GLC) composta por quinze regras de produção, que levam à construção de consultas em linguagem SPARQL.

Scalewiles fornece facetas que casam com a gramática de LISQL2. Assim, a medida em que se navega por essas facetas, Scalewiles vai construindo incrementalmente uma consulta SPARQL válida. A semântica das consultas é obtida traduzindo a gramática para SPARQL. Cada regra de produção é equivalente a uma regra de tradução para SPARQL.

Scalewiles se assemelha ao Qodra no que diz respeito a navegação por facetas para identificar o interesse do usuário. No Qodra facetas serão construídas dinamicamente e a medida que a base de vídeos muda ou aumenta, as facetas vão sendo rearranjadas automaticamente, ao contrário de Scalewiles, onde as facetas são independente dos dados e dependentes da gramática. Por ser muito genérico, a consulta SPARQL gerada por Scalewiles apresenta limitações dependendo da base de dados na qual ela será aplicada.

2.2.3 YoVisto

Para o fim específico de recuperação de vídeos acadêmicos o YoVisto é o principal mecanismo de busca relacionado ao Qodra. Especializado em vídeos acadêmicos e conferências, o YoVisto é um buscador que também utiliza ligações a dados externos para enriquecer a busca e oferecer uma melhor experiência ao usuário. Ele difere da arquitetura do Qodra, na abordagem que é utilizada para anotação de entidades e assuntos que se relacionam com os vídeos.

Sua principal contribuição é utilizar a indexação do conteúdo com baixa granularidade, segmentando e definindo *tags* no vídeo por quadro ou trechos do vídeo. Para extrair a informação são utilizadas tecnologias de de processamento de imagens, *Optical Character Recognition e Folksonomia* (Waitelonis et al, 2010). Os metadados extraídos compõem o índice utilizado pela ferramenta. Finalmente, estes metadados são então relacionados com entidades da DBPedia para ajudar o motor de busca a sugerir conceitos e apoiar os usuários durante a busca. Combinando estas técnicas, o YoVisto faz uso das vantagens do conceito de buscas exploratórias apoiadas por filtros geográficos e *faceted browsing* utilizando propriedades da DBpedia.

O presente trabalho assemelha ao YoVisto no que se refere à recuperação de vídeos acadêmicos, o principal avanço da arquitetura do Qodra é poder combinar técnicas

e não somente explorar um par de estratégias de indexação. O YoVisto ao não publicar o seu índice ou não disponibilizar dados em outros formatos, além do HTML, perde no aspecto de integração e não se beneficia por completo dos recursos da Web Semântica, além de estar altamente acoplado à sua interface de consultas não abrindo chances para explorar novas abordagens (Coelho & Souza, 2015).

3 Especificação do Framework

O projeto Qodra terá como resultado um *framework* de buscas e anotação semântica de videoaulas. O desafio aqui é indexar, formatar e disponibilizar tal conteúdo de forma que possa aprimorar a experiência do usuário(Coelho & Souza, 2015).

Considerando que a maior parte dos buscadores ainda faz uso de uma base textual para recuperar informação (Fensel, 2005), há uma necessidade de se indexar de forma eficaz arquivos de áudio e vídeo (Croft et al, 2010). Para muitos buscadores a recuperação é realizada através de metadados que dizem respeito ao contexto do vídeo. Esses metadados geralmente trazem informações relativas ao título, autor, data, localidade, palavras-chave referentes ao tema central, e em alguns casos a categorização dos mesmos (Tumbull et al, 2015).

Ocorre que esses metadados não estão presentes nativamente no vídeo. Logo, para que esta técnica funcione um processo manual de seleção e marcação palavras que identificam e descrevem o conteúdo do vídeo precisa ser realizado. Essa prática facilita o processo de busca por deixar o contexto explícito para o vídeo. No entanto ela não resolve completamente o problema pois além do tempo empregado para realizar este trabalho, os vídeos são catalogados e classificados de forma subjetiva, reduzindo a eficácia dos métodos de busca atuais (Sack, H. & Waitelonis). Dessa forma há uma necessidade de se indexar arquivos de áudio e vídeo de forma eficaz (Stamou et al, 2005).

De forma menos precisa, sistemas web podem fazer uso da indexação de *tags* encontradas em *wikis* e blogs para tratar esse problema sem muito esforço manual (Specia & Motta, 2007).

Outra técnica usada para melhorar a busca em vídeos é a utilização dos textos transcritos a partir do vídeo (ou áudio) permitindo que o mesmo possa ser recuperado através de palavras faladas ao longo da sua reprodução. Segundo Raimond el al (2012) essa técnica é muito custosa pois gera demanda de trabalho humano ainda maior que criação de metadados e raramente está disponível para sistemas de busca.

Considerando a problemática acima, o Qodra propõe uma abordagem de busca

semântica em arquivos de áudio e vídeo. Essa abordagem emprega técnicas de extração de informação visando maximizar a precisão de sistemas de busca em vídeos sem demanda de esforço manual para a indexação dos mesmos.

Para eliminar a demanda de esforço manual nossa abordagem propõe: transcrever a fala do vídeo para texto de maneira automática usando ASR (*Automatic Speech Recognition*); processar este texto visando anotar entidades de dados ligados cujo vídeo as referencia e através dessas referências, encontrar relacionamentos em dados ligados de maneira que seja possível classificar/relacionar os vídeos entre si. As relações encontradas serão persistidas em Banco de Dados RDF.

Na figura 3.1 estão representados os módulos que compõem o *framework*, nas seções seguintes são apresentados o funcionamento de cada um deles.

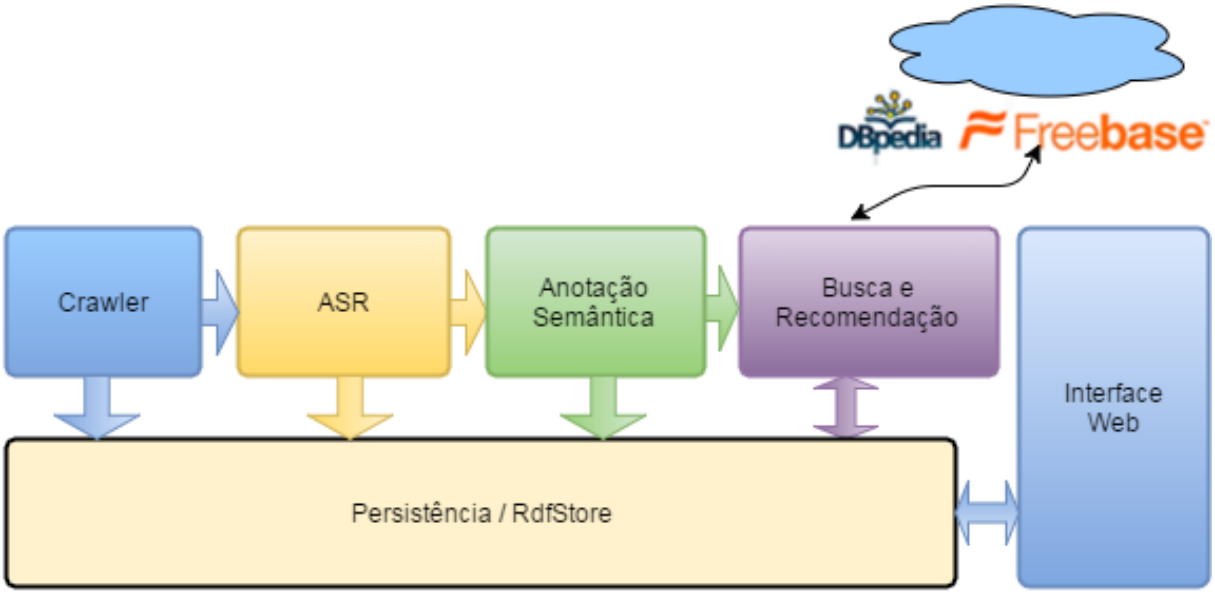

Figura 3.1: Esquema de módulos do *Framework*

O funcionamento do *framework* se dá em dois cenários: um deles é iniciado pela execução do Módulo *Crawler* seguido pelos Módulos ASR e Anotação Semântica respectivamente; o outro, através da interação do usuário com o Módulo Interface Web, que fará acesso à base de dados para recuperar a informação conveniente e apresentar ao usuário.

3.1 Módulo Crawler

O módulo *Crawler* é responsável por recuperar as videoaulas de determinadas fontes. Para o presente trabalho ele foi configurado para recuperar arquivos do projeto VídeoAula@RNP da Rede Nacional de Pesquisas, onde estão disponibilizados videoaulas do Instituto de Ciências Exatas da UFJF e de outras instituições (Coelho & Souza, 2015). O objetivo é verificar a existência de uma videoaula que não que não se encontra na base local, fazer o download da mesma e disponibilizá-la para que seja processada pelos demais módulos do sistema.

3.2 ASR (Módulo de Transcrição)

Para indexar os arquivos de vídeo nossa abordagem propõe utilizar o texto transcrito de forma automática. Para isso, o módulo ASR possui papel fundamental pois é capaz de reconhecer palavras faladas durante a execução do vídeo e transcrevê-las para texto. Dessa maneira, elimina o alto custo gerado pela transcrição manual.

O *Automatic Speech Recognition (ASR)* abrange um conjunto de técnicas de processamento de sinais que juntamente com modelos estatísticos realizam o reconhecimento do fala humana resultando no texto transcrito. Essa transcrição não será integralmente fiel à fala uma vez que o resultado do ASR são transcrições que mais se aproximam do som analisado (Coelho & Souza, 2015). A melhoria desse processo não será abordada neste trabalho e poderá ser tratada em trabalhos futuros.

3.3 Módulo Anotação Semântica

Após transcrito, o texto é analisado que pelo módulo de Anotação Semântica, cuja finalidade é anotar automaticamente *tags* que representem entidades da DBPedia de acordo com as palavras contidas na transcrição. O módulo permite que o desenvolvedor utilize diferentes técnicas que possam fornecer *tags, scores* e tipos da ontologia da DBPedia (Coelho & Souza, 2015).

3.4 Módulo Persistência

Para armazenamento dos metadados o módulo Persistência permite o uso de qualquer gerenciador de Banco de Dados que permita o armazenamento em formato RDF e possuam uma interface de consultas SPARQL e de inserção SPARQL Update. Esse módulo foi configurado para uso do Allegrograph como Gerenciador de Banco de Dados.

3.5 Módulo Busca e Recomendação

O objetivo central deste trabalho é realizar buscas em vídeos dadas as condições anteriores. Uma vez indexados, deseja-se apresentar o melhor resultado para o usuário e sugerir videoaulas que complementem o seu estudo, este módulo foi configurado para consultar a DBpedia para expandir os recursos relacionados aos vídeos, encontrar relacionamentos entre estes recursos e em função disso inferir quais vídeos possuem assuntos semelhantes.

Como visto na seção 2.1.2, a edição em inglês da DBpedia (http://dbpedia.org) é sua versão mais completa e por este motivo foi escolhida para ser referenciada por este módulo.

A figura a seguir esquematiza as entidades e repositórios de dados ligados visitados pelo algoritmo.

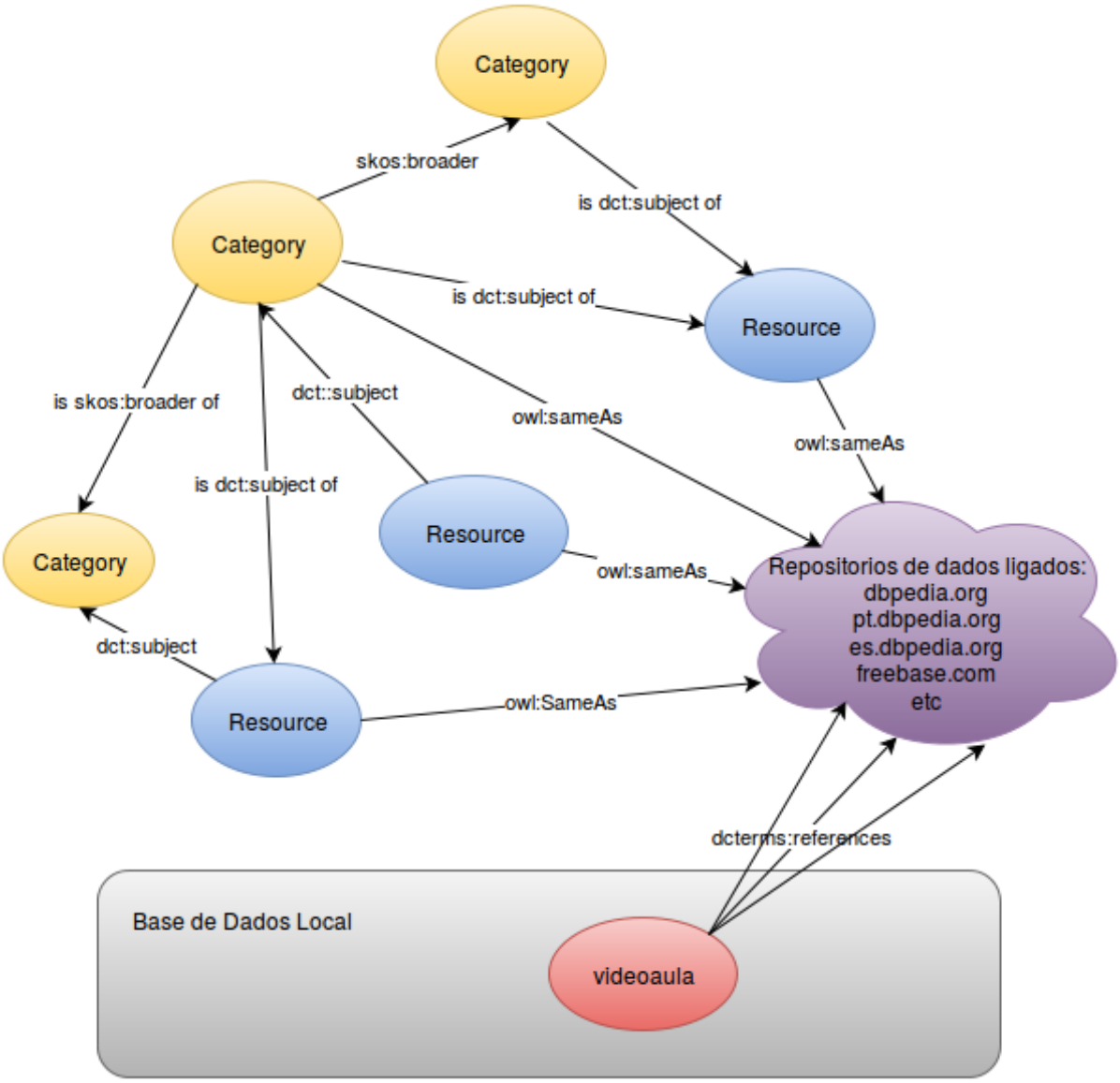

Figura 3.2: Esquema de navegação por entidades de dados ligados

Em nossa base de dados as *tags* presentes na propriedade $< dcterms : references >$, são referências que apontam para recursos de diferentes edições da DBpedia ou outros repositórios de dados ligados.

O primeiro passo será identificar os recursos que apontam para outros repositórios e encontrar a sua relação com a edição está sendo utilizada, para isso será consultada a propriedade $< owl : sameAs >$ do esquema da DBpedia. Essa propriedade mantêm ligações para outras fontes de dados de todos os recursos que são descritos nesta edição. Assim, para cada recurso presente em um vídeo que aponte para qualquer repositório que não seja a DBpedia em inglês, será realizada a seguinte consulta SPARQL:

$select\ distinct\ ?x\ where\ \{?x\ owl:sameAs\ <uri>\}\ limit\ 100$

Onde $<uri>$ pode ser uma referência que aponta para um recurso da edição em português (por exemplo: http://pt.dbpedia.org/resource/Algoritmo).

A partir das uri's relacionadas, deseja-se consultar relações existentes no esquema da DBpedia em vários níveis, de tal maneira que uma vez encontrado um recurso relacionado ao recurso inicial, pode-se procurar por novos recursos.

Através de inspeção observou-se que as propriedades presentes na ontologia da DBpedia $<dcterms:subject>$, $<skos:broader>$ e $<is\ skos:broader\ of>$ seriam as mais indicadas para procurarmos as correlações existentes. A primeira delas, $<dct:subject>$, nos fornece a(s) categoria(s) de um recurso a partir da qual podemos utilizar as demais propriedades $<skos:broader>$ e/ou $<is\ skos:broader\ of>$ para relacionar esta com outras, mais ou menos específicas, obtendo assim um conjunto de categorias que se relacionam ao vídeo. As consultas SPARQL realizadas nesta etapa são as seguintes:

(1)$select\ distinct\ ?categoria\ where\ \{<referencia><dct:subject>?categoria\}$

(2)$select\ distinct\ ?categoriaMaisAmpla\ where\ \{<categoria><skos:broader>$ $?categoriaMaisAmpla\}$

(3)$select\ distinct\ ?categoriaMaisEspecifica\ where\ \{?categoriaMaisEspecifica$ $<skos:broader><categoria>\}$

Será formado um conjunto de categorias por vídeo e aqueles que apresentarem interseção de categorias serão recomendados na forma de assuntos relacionados.

A utilização de outras propriedades da ontologia da DBpedia, como por exemplo $<rdf:type>$, poderia levar a um resultado demasiadamente genérico e portanto não interessante, como no caso da uri (http://dbpedia.org/resource/Algorithm), que está mapeada na propriedade $<rdf:type>$ como uma entidade de *thing* (coisa).

Para melhor ilustrar o funcionamento desse algoritmo, a figura 3.3 apresenta o fluxograma de execução do mesmo.

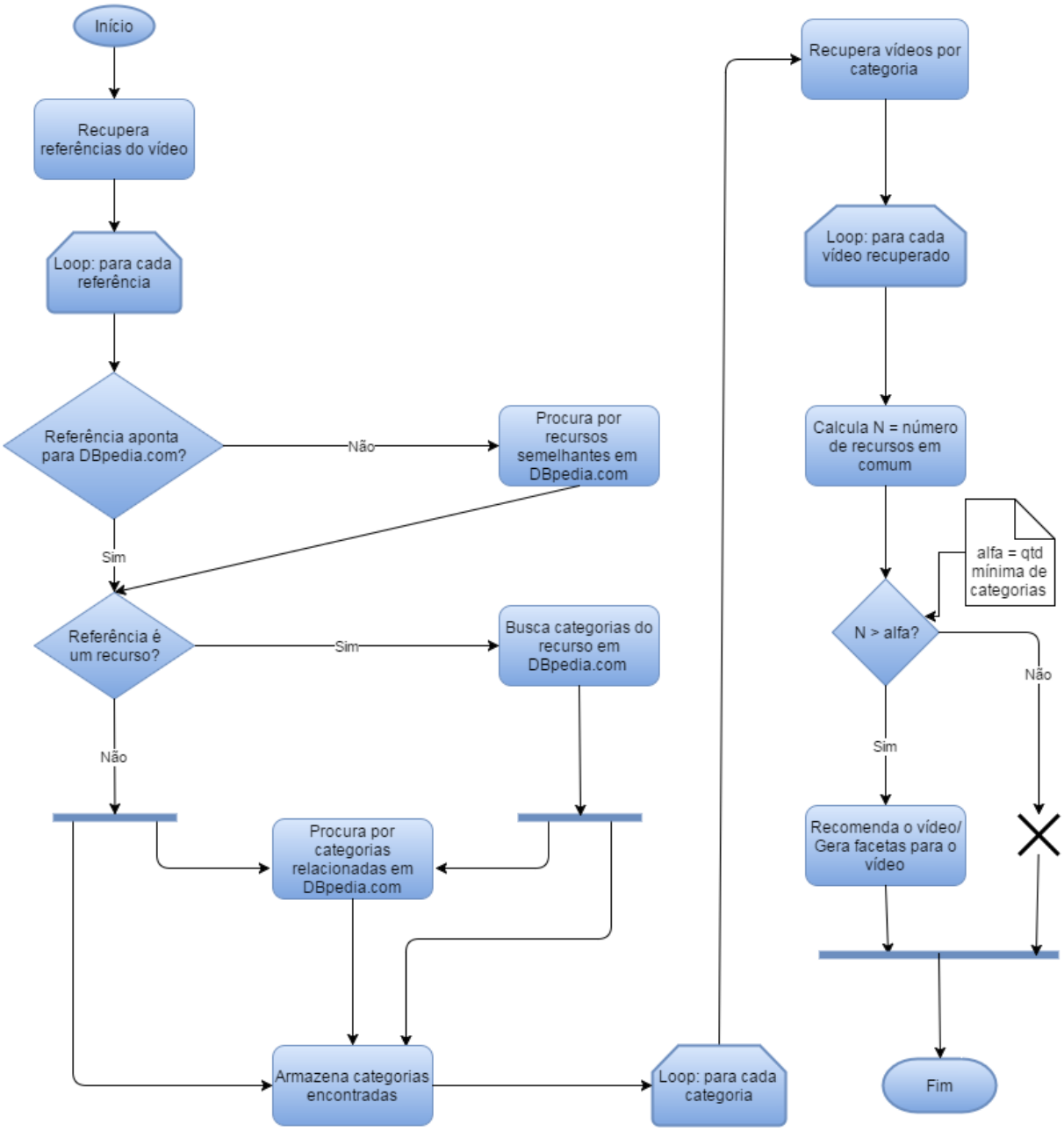

Figura 3.3: Fluxograma do algoritmo de busca e recomendação

3.5.1 Construindo Facetas

As facetas serão o meio de navegação pelo conteúdo dos vídeos existentes no repositório, para construir as facetas foi usada a seguinte estrategia: dado um vídeo qualquer, suas facetas serão obtidas consultando propriedade $< rdfs : label >$ na DBpedia para cada uma das referências do vídeo através da consulta SPARQL a seguir.

$$Select\ ?label\ where\ \{< resource >\ rdfs : label\ ?label$$
$$filter((LANG(?label)\ =\ "\ ")\ ||\ LANGMATCHES(LANG(?label),\ "pt"))\}$$

Onde $< resource >$ é o recurso obtido pela consulta a propriedade $< owl : sameAs >$ para uma referência ($< dcterms : references >$) do vídeo ou a própria referência, caso ela já esteja apontando para a edição em inglês da DBPedia. O resultado é a descrição de um assunto do vídeo no idioma português, se houver.

De posse das facetas realiza-se a busca por vídeos relacionados e aos vídeos que foram identificados como relacionados, são associados as mesmas facetas do vídeo inicial, dessa maneira possibilita apresentar aos usuários os vídeos relacionados ao se escolher uma faceta no módulo Interface Web.

3.6 Módulo Interface web

O módulo Interface Web será responsável por permitir a interação do usuário com o sistema, nele o usuário poderá navegar pelas facetas até encontrar o tema de seu interesse. Assim que o usuário iniciar a execução de uma videoaula, as videoaulas já identificadas como relacionadas serão sugeridas. O usuário poderá ainda se cadastrar e deixar *likes* para o vídeo e sua opinião sobre o sistema.

Nas figura 3.4 abaixo é mostrada a interface principal do sistema apresentando os principais vídeos e na figura 3.5 é ilustrado a execução de uma videoaula com as sugestões de vídeos relacionados ao lado.

Para utilizar/testar o sistema basta acessar o seguinte endereço:
(http://200.131.219.35).

Figura 3.4: Tela inicial do sistema

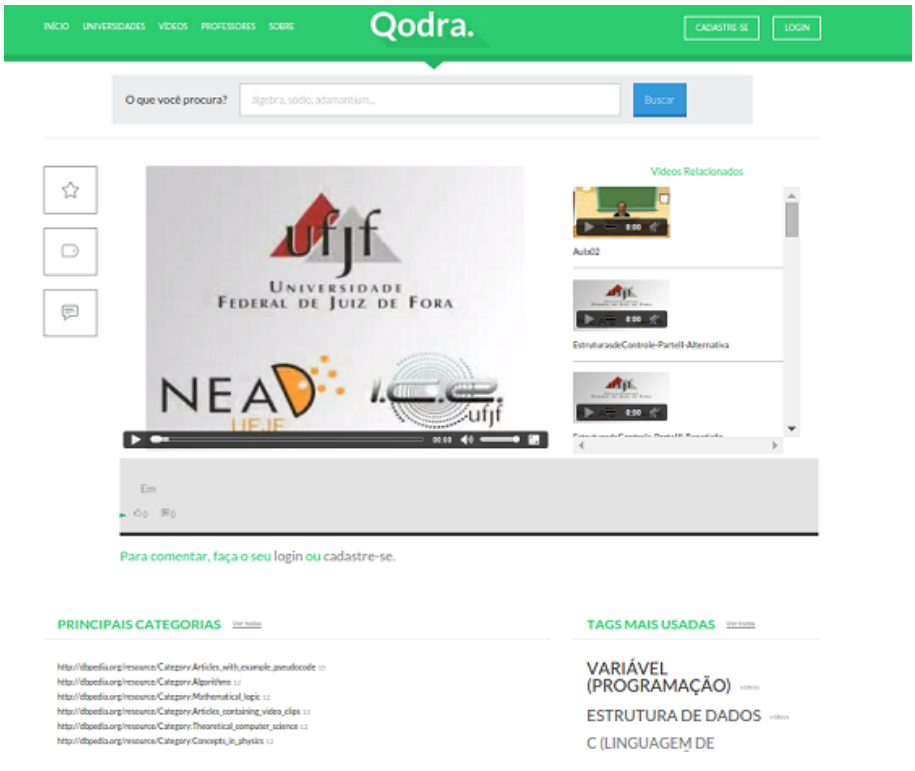

Figura 3.5: Tela de vídeos relacionados

3.7 Integração dos módulos

A arquitetura do Qodra é projetada para que cada módulo possa realizar suas funções de maneira independente, permitindo que seja aplicada diferentes soluções tecnológicas, sendo possível experimentar a aplicabilidade de novas abordagens e ainda favorecer a combinação das melhores técnicas de recuperação de informação (Coelho & Souza, 2015).

Dessa forma, se por alguma questão tecnológica for preciso alterar o gerenciador de banco de dados por exemplo, apenas o módulo Persistência precisará ser reconfigurado. Para a mediação e integração dos módulos foi adotada como solução de *middleware*, o Apache ActiveMQ (http://activemq.apache.org).

3.7.1 Apache ActiveMQ

O Apache ActiveMQ é um software que gerencia mensagens através de filas em que a cada fila deve ser associado um consumidor. No caso do Qodra, este consumidor será um módulo do projeto que ficara como *listenner* (ouvinte) em cada fila. Quando o ActiveMQ receber uma mensagem ele irá acordar este módulo e entregar-lhe a mensagem para que este realize a funcionalidade associada.

O ActiveMQ fornece Apis para diferentes plataformas de desenvolvimento através das quais os módulos do Qodra poderão enviar e/ou consumir mensagens, tornando a implementação mais simples e objetiva.

3.7.2 Formatos das mensagens trocadas entre os módulos

Para a definição das filas do Apache ActiveMQ a nomenclatura segue o padrão: [Projeto].[Módulo de Origem].[Módulo de Destino].[Idioma*]. Os três primeiros itens são obrigatórios, o idioma é opcional e aplicável somente ao módulos de ASR e Anotação Semântica.

Assim, uma mensagem enviada para a fila Qodra.Crawler.RdfStore, o módulo de origem é o Crawler e o módulo Persistência é quem receberá essa mensagem. As mensagens devem possuir o formato N-Triple, com a seguinte estrutura:

$< URL\_Video\_Aula >< Propriedade >< Valor > .$

Na figura 3.6 abaixo é mostrado o ActiveMQ em funcionamento com o módulo Persistência configurado como ouvinte em todas as filas que ele deve consumir as mensagens. O Status da fila Qodra.Busca.RdfStore indica que o módulo Busca e Recomendação enviou 2469 mensagens que foram tratadas pelo ActiveMQ e entregues ao módulo Persistência.

Queues

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
qodra.AnnotationTool.RdfStore	0	1	0	0	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
qodra.ASR.RdfStore	0	1	0	0	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
qodra.Busca.RdfStore	0	1	2469	2469	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
qodra.Crawler.RdfStore	0	1	0	0	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
qodra.Feedback.RdfStore	0	1	0	0	Browse Active Consumers Active Producers atom rss	Send To Purge Delete

Figura 3.6: Apache ActiveMQ enfileirando e tratando mensagens recebidas

4 Resultados

Conforme foi apresentado na seção 3.5, O módulo de Busca e Recomendação irá procurar por categorias de recursos na DBPedia e formar um conjunto de categorias por vídeo para poder recomendar aqueles que se relacionam. Nas seções seguintes são apresentados e discutidos os resultados obtidos a partir do emprego das técnicas propostas para a resolução do problema.

4.1 Base de dados para análise dos resultados

Para que fosse possível verificar a eficiência das abordagens foi construída uma base para testes. Esta base possui vídeos das áreas de Ciência da Computação, Estatística, Química e Física e a relação que foi definida entre eles. Para criar os relacionamentos foi usada a seguinte estratégia: os vídeos foram assistidos na integra e para cada assunto explicitamente falado, procurou-se por recursos que os identificassem na DBPedia; os recursos encontrados foram persistidos na base de dados como triplas RDF na propriedade $< dcterms : references >$; através da análise desses recursos foram definidos quais vídeos deveriam ser recomendados ao usuário como vídeos que contêm assuntos relacionados. Estes, são os vídeos que serão considerados como itens relevantes de serem retornados pelo algoritmo e serão a base para a aplicação das métricas que serão usadas medir a eficiência do mesmo.

Admitiu-se que um recurso só estaria relacionado se ele é literalmente expressado no vídeo pois essa premissa é valida para os módulos de ASR e Anotação Semântica. Se um assunto está implícito em algum vídeo mas ele não é verbalizado, o módulo ASR não irá decodificar nenhuma palavra referente ao assunto. Logo, o módulo de Anotação Semântica não irá anotar nenhum recurso referente ao conceito implícito.

Uma pré-condição para a execução do algoritmo de Busca e Recomendação é que exista ao menos uma *tag* de referência. os recursos persistidos durante a construção da base serão as referências que o módulo de Buscas utilizará como entrada para navegar por

dados ligados e relacionar os vídeos.

Como já foi dito, a classificação manual de vídeos não é o ideal para se trabalhar. Contudo essa base teve que ser construída manualmente pois os módulos ASR e de Anotação Semântica ainda estão em fase de desenvolvimento. A utilização destes módulos neste momento poderia comprometer os resultados ao invés de melhorá-los.

Cabe ressaltar ainda que embora essa classificação tenha sido feito de forma cuidadosa, ainda assim ela é subjetiva. Este fato deverá ser considerado na análise dos dados.

4.2 Métricas utilizadas

Foram propostas duas métricas para avaliação dos resultados: a primeira delas é o *recall* ou cobertura. Esta métrica é calculada conforme a fórmula a seguir:

$$recall = \frac{\sum(itens\ relevantes \bigcap itens\ recuperados)}{\sum(itens\ relevantes)}$$

O *recall* será máximo (igual a 1) quando todos os itens relevantes forem retornados, cobrindo todo o espaço dos resultados esperados, e mínimo (igual a 0) quando nenhum item relevante for retornado (sem cobertura). Como o objetivo é recomendar o maior número de vídeos com assuntos relacionados possíveis, espera-se maximizar o *recall*.

Exemplo:

Vídeos relevantes: $\{a,\ b,\ c,\ d\}$

Vídeos retornados: $\{e,\ c,\ b,\ a\}$

$$recall = \frac{\sum(\{a,\ b,\ c,\ d\} \bigcap \{e,\ c,\ b,\ a\})}{\sum(\{a,\ b,\ c,\ d\}} = \frac{3}{4} = 0.75$$

A segunda métrica proposta é o TopN. Ela será importante principalmente quando o algoritmo retornar um número maior de vídeos do que os definidos manualmente. O TopN fornece o quão próximo das primeiras posições estão os itens relevantes, ele será máximo (igual a 1) se todos os itens relevantes forem retornados e ocuparem exatamente as posições iniciais. Admitindo-se que a base manual está correta, uma situação ótima

será aquela em que a busca obtém *recall* e TopN máximos.

Para calcular o TopN é preciso que os itens retornados estejam ranqueados para determinar a posição em que devem aparecer. Os vídeos retornados foram ranqueados considerando o número de categorias em comum que possuem com o vídeo inicial,(aquele que está procurando relacionar com outros). Espera-se que quanto maior for esse número, mais correlatos serão os os assuntos tratados nos dois vídeos, o que justifica uma melhor colocação no *ranking*.

O TopN é calculado de acordo com a seguinte fórmula:

$$TopN = \frac{\alpha^{i_1} + \alpha^{i_2} + \alpha^{i_3} + ... + \alpha^{i_k}}{\alpha^1 + \alpha^2 + \alpha^3 + ... + \alpha^n}$$

Onde: $0 < \alpha < 1$, para o estudo foi escolhido $\alpha = 0.8$; n é o total de itens relevantes; k é o k-ésimo vídeo relevante que foi retornado e i é a posição em que um vídeo relevante foi ranqueado.

Exemplo:

Vídeos relevantes: $\{a,\ b,\ c\}$

Vídeos retornados: $\{e,\ c,\ b,\ d,\ a\}$, nesta ordem.

$$TopN = \frac{0.8^2 + 0.8^3 + 0.8^5}{0.8^1 + 0.8^2 + 0.8^3} = 0,758$$

4.3 Algoritmo de Busca sem expansão de categorias

Nesta etapa o algoritmo foi configurado para navegar pelos recursos ligados e recuperar as categorias na DBPedia mas apenas na propriedade $< dct : subject >$. Inicialmente os vídeos foram relacionados pela comparação de quais possuem pelo menos uma categoria em comum.

Embora a estratégia usada para realizar os relacionamentos seja relativamente simples, o valor médio de *Recall* foi igual a 0.589 e TopN igual a 0.551. Outro dado importante que pôde ser observado é que em 38 das 42 consultas realizadas o total relaci-

onamentos feitos pelo algoritmo (coluna total retornados) foi maior que o total de vídeos relacionados de forma manual (coluna classif. manual), o que de certa forma já era esperado. No entanto a maioria predominante de consultas retornando muitos relacionamentos fez surgir a necessidade de analisar o valor de TopN das consultas.

Verificou se um valor médio de TopN igual a 0.538 contudo 25 das 42 consultas tiveram TopN com valores muito satisfatórios, entre 0.5 e 1.0, cuja média foi de 0.842 e cobertura média de 0.793. Essas consultas foram consideradas favoráveis a execução do algoritmo e a aplicação das métricas por possuírem mais relacionamentos que as demais. Outra situação que pode ter contribuído é a qualidade das referencias anotadas, mesmo que não sejam muitas, se na DBpedia o recurso possuir mais ligações facilitará a identificação de assuntos correlatos.

Entre as consultas cujo TopN ficou abaixo da média pôde ser notada uma característica comum, o número muito baixo de vídeos que foram relacionados manualmente contrapondo um número alto de vídeos relacionados pelo algoritmo. Enquanto a média de vídeos relacionados manualmente é aproximadamente 3 sendo o valor máximo igual a 5, a média de vídeos retornados pelo algoritmo é maior que 10 com um valor máximo de 21. Isso pesou muito negativamente no valores médios, principalmente no TopN que considerando apenas as 25 primeiras consultas possui média de 0.842. Os dados supracitados são dispostos na tabela 4.1 abaixo e nos gráficos apresentados nas figuras 4.1 e 4.2, onde pode-se perceber a dispersão da quantidade de consultas em relação às faixas de valores de *recall* e TopN.

Tabela 4.1: resultados do algoritmo de busca sem expansão de categorias

Vídeo	Ref. anotadas	Classif. Manual	Retornados Certos	Total Retornados	Recall	TopN
Dcc119 Aula2	13	9	9	17	1.000	1.000
Dcc119 Aula3	6	9	9	16	1.000	1.000
Dcc119 Aula4	7	9	9	16	1.000	1.000
Dcc119 Aula-Teste	6	9	9	16	1.000	1.000
Fis2TempCalor	15	1	1	15	1.000	1.000
Fis2cap18parte2	7	1	1	17	1.000	1.000
Qui125 Aula10	4	1	1	8	1.000	1.000
Qui125 Aula7	7	1	1	11	1.000	1.000
Dcc119 Aula1	11	11	10	17	0.909	0.966
Dcc119 Aula5	10	14	10	16	0.714	0.913
Dcc119 Aula6	10	14	10	16	0.714	0.913
Dcc119 Aula7	9	14	10	16	0.714	0.913
Dcc119 Aula8	9	14	10	17	0.714	0.911
Dcc008 Aula01	5	4	3	8	0.750	0.827
Qui125 Aula5	8	4	3	13	0.750	0.827
Quim131 Aula3	7	5	4	10	0.800	0.804
Dcc008 Aula02	8	4	3	19	0.750	0.783
Dcc008 Aula03	7	4	3	19	0.750	0.783
Dcc008 Aula04	7	4	3	19	0.750	0.783
Dcc119 Aula9	8	14	9	18	0.643	0.701
Dcc116 Aula04	11	5	4	8	0.800	0.695
Qui125 Aula4	4	3	2	14	0.667	0.578
DCC116 Aula09	13	2	1	9	0.500	0.556
Qui125 Aula8	3	2	1	10	0.500	0.556
DCC116 Aula01	18	5	2	17	0.400	0.535
Quim131 Aula2	5	2	2	10	1.000	0.472
Qui125 Aula9	6	7	3	8	0.429	0.395
Dcc119 Ex02	4	1	1	15	1.000	0.328
Qui125 Aula3	5	2	1	17	0.500	0.284
DCC116 Aula06	14	4	1	10	0.250	0.271
DCC116 Aula02	26	4	1	11	0.250	0.173
Dcc119 Ex01	3	1	1	15	1.000	0.134
Quim131 Aula1	4	2	1	12	0.500	0.048
Est029 Aula2	6	5	0	0	0.000	0.000
Qui125 Aula6	2	5	0	3	0.000	0.000
DCC116 Aula05	7	5	0	1	0.000	0.000
DCC116 Aula10	31	4	0	14	0.000	0.000
Aula08 Aula08	21	2	0	21	0.000	0.000
Est029 Aula1	4	1	0	3	0.000	0.000
Fis2 Grav	22	1	0	12	0.000	0.000
Qui125 Aula1	6	1	0	12	0.000	0.000
Qui125 aula16	5	1	0	0	0.000	0.000
Média					0.589	0.551
Média 25 primeiras consultas					0.793	0.842

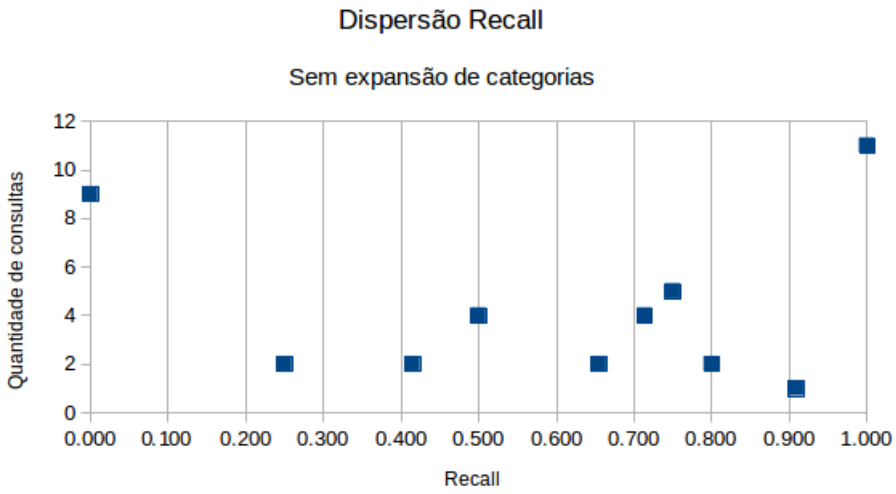

Figura 4.1: Gráfico de dispersão dos valores de *recall* para consultas sem expansão de categorias

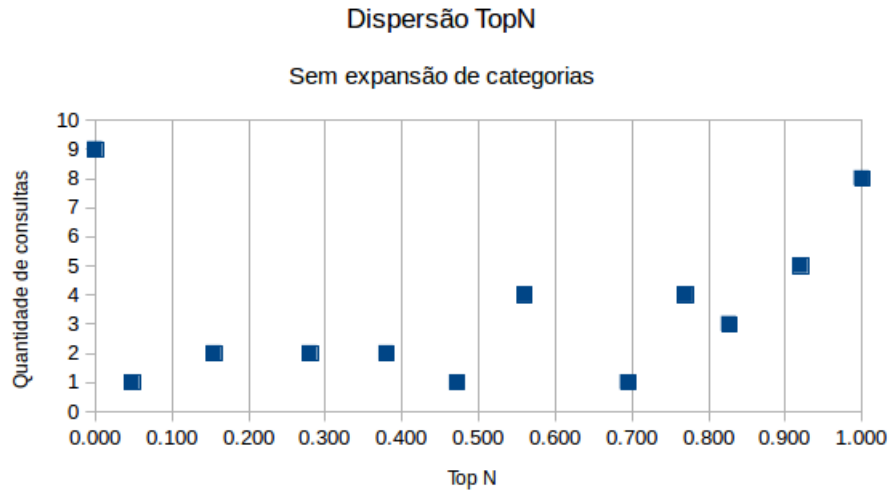

Figura 4.2: Gráfico de dispersão dos valores de TopN para consultas sem expansão de categorias

4.4 Algoritmo de Busca com expansão de categorias

Embora os dados discutidos na seção anterior ponderam positivamente no valor médio do TopN, foi realizado um ajuste no algoritmo na tentativa aumentar o *recall* médio obtido na abordagem anterior e observar os efeitos que ocorreriam nos valores de TopN.

Na DBpedia a categoria de um recurso pode estar ligada a outras categorias que sejam mais amplas (propriedade $< skos : broader >$) ou mais específicas (propriedade $< is\ skos : broader\ of >$). O algoritmo foi ajustado para que além de recuperar as categorias da propriedade $< dct : subject >$, partisse deste ponto em busca de outras categorias navegando pelas propriedades citadas acima. Os resultados dessa abordagem são apresentados na tabela 4.2.

De modo geral os resultados melhoraram substancialmente, a média geral do *recall* passou de 0.576 na abordagem anterior para 0.898 superando inclusive a cobertura média das consultas consideradas favoráveis e o TopN médio passou de 0.538 para 0.633. Outro ponto positivo foi o número de consultas com TopN abaixo de 0.5 que caiu de 18 para 16 e das 11 consultas que tinham apresentado *recall* e TopN iguais a 0, apenas uma persiste nessa condição. Ou seja, mesmo para consultas com baixo número de vídeos relacionados manualmente o algoritmo obteve boas medidas de cobertura e TopN.

Porém essa abordagem trouxe um efeito indesejado pois algumas consultas retornaram um conjunto muito grande de resultados. Isso não quer dizer que o algoritmo errou ao relacionar muitos os vídeos, ele apenas generalizou muito ao expandir pelas categorias da propriedade $< skos : broader >$. Obviamente, ao considerar recursos mais abrangentes mais vídeos terão assuntos relacionados, ainda que distantes.

Para minimizar esse efeito um novo ajuste foi realizado no algoritmo, dessa vez foi retirada a expansão para categorias mais amplas ($< skos : broader >$) e mantidas as buscas por categorias específicas ($< is\ skos : broader\ of >$).

Nos resultados apresentados na tabela 4.3 percebe-se que conforme foi previsto houve uma diminuição no total de vídeos retornados, 5 em média. Os valores médios de TopN uma pequena variação de apenas 2 pontos percentuais para menos. Outro dado positivo é que o número de consultas com TopN Maior que 0.5 passou de 27 para 29, conforme pode ser observado nos gráficos de dispersão (figura 4.3 e 4.4). Somente no

valor médio do *recall* percebe-se uma pequena queda que passou de 0.896 para 0.847, muito em função das duas últimas consultas.

Tabela 4.2: resultados do algoritmo de busca expandindo por categorias mais amplas e mais específicas na DBpedia

Vídeo	Ref. anotadas	Classif. Manual	Retornados Certos	Total Retornados	Recall	TopN
Dcc119 Aula3	6	9	9	34	1.000	1.000
Dcc119 Aula4	7	9	9	34	1.000	1.000
Dcc119 Aula teste	6	9	9	34	1.000	1.000
Dcc119 Aula9	13	2	2	30	1.000	1.000
Qui125 Aula10	4	1	1	20	1.000	1.000
Fis2Cap18parte2	7	1	1	35	1.000	1.000
Fis2tempcalor	15	1	1	42	1.000	1.000
Qui125 Aula7	7	1	1	43	1.000	1.000
Dcc119 Aula2	13	9	9	36	1.000	0.992
Qui125 Aula5	8	4	4	35	1.000	0.965
Dcc119 Aula1	11	11	11	36	1.000	0.964
Dcc119 Aula6	10	14	13	34	0.929	0.949
Dcc119 Aula5	10	14	13	34	0.929	0.947
Dcc119 Aula7	9	14	13	35	0.929	0.945
Dcc119 Aula8	9	14	13	35	0.929	0.939
Qui125 Aula4	4	3	3	24	1.000	0.840
Dcc008 Aula1	5	4	3	17	0.750	0.827
Est029 Aula2	6	5	5	31	1.000	0.808
Qui131 Aula3	7	5	5	41	1.000	0.797
Qui125 Aula3	5	2	2	40	1.000	0.783
Dcc119 Aula9	8	14	12	34	0.857	0.726
Dcc008 Aula2	8	4	3	32	0.750	0.721
Dcc008 Aula3	7	4	3	32	0.750	0.721
Dcc008 Aula4	7	4	3	32	0.750	0.721
Qui125 Aula9	6	7	7	20	1.000	0.622
Qui125 Aula8	3	2	2	41	1.000	0.603
Dcc116 Aula4	11	5	4	40	0.800	0.560
Dcc008 Aula6	14	4	3	41	0.750	0.497
Dcc116 Aula1	18	5	5	44	1.000	0.442
Est029 Aula1	4	1	1	33	1.000	0.410
Qui125 Aula16	5	1	1	27	1.000	0.410
Qui125 Aula1	6	1	1	24	1.000	0.328
Dcc119 Ex02	4	1	1	28	1.000	0.328
Qui125 Aula6	2	5	5	40	1.000	0.312
Dcc116 Aula2	26	4	3	35	0.750	0.187
Fis2 Grav	22	1	1	24	1.000	0.168
Dcc119 Ex01	3	1	1	32	1.000	0.168
Qui131 Aula2	5	2	2	25	1.000	0.165
Quim 131 Aula1	4	2	2	24	1.000	0.132
Dcc116 Aula10	31	4	3	47	0.750	0.044
Dcc116 Aula8	21	2	1	43	0.500	0.020
Dcc116 Aula5	7	5	0	2	0.000	0.000
Média					0.898	0.644
Média 27 primeiras consultas					0.940	0.868

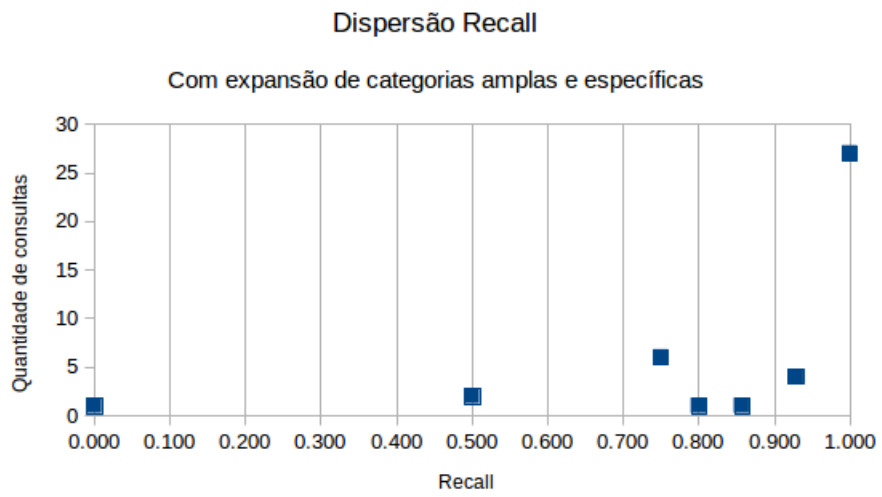

Figura 4.3: Gráfico de dispersão dos valores de *recall* para consultas com expansão por categorias amplas e específicas

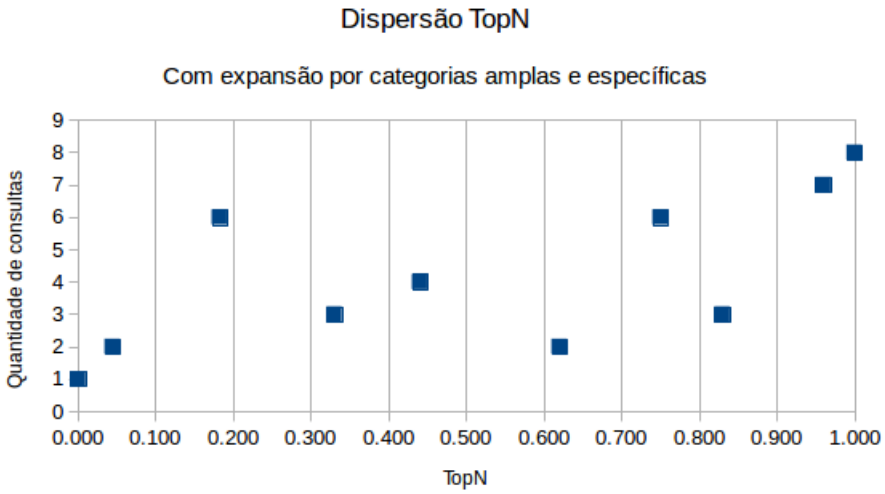

Figura 4.4: Gráfico de dispersão dos valores de TopN para consultas com expansão por categorias amplas e específicas

Tabela 4.3: resultados do algoritmo expandindo apenas por categorias mais específicas

Vídeo	Ref. anotadas	Classif. Manual	Retornados Certos	Total Retornados	Recall	TopN
Dcc119 Aula3	6	9	9	24	1.000	1.000
Dcc119 Aula4	7	9	9	24	1.000	1.000
Dcc119 Aula-teste	6	9	9	24	1.000	1.000
DCC116 Aula09	13	2	2	16	1.000	1.000
Fis2cap18parte2	7	1	1	27	1.000	1.000
Fis2 Tempecalor	15	1	1	25	1.000	1.000
Qui125 Aula7	7	1	1	24	1.000	1.000
Qui125 Aula10	4	1	1	15	1.000	1.000
Dcc119 Aula2	13	9	9	25	1.000	0.955
Dcc119 Aula6	10	14	13	24	0.929	0.942
Dcc119 Aula7	9	14	13	24	0.929	0.942
Dcc119 Aula5	10	14	13	24	0.929	0.940
Dcc119 Aula8	9	14	13	25	0.929	0.935
Dcc119 Aula1	11	11	10	25	0.909	0.928
Dcc008 aula01	5	4	3	15	0.750	0.827
Quim131 Aula3	7	5	5	18	1.000	0.820
Qui125 Aula5	8	4	3	32	0.750	0.783
Est029 Aula2	6	5	4	24	0.800	0.731
Dcc008 Aula02	8	4	3	26	0.750	0.721
Dcc008 Aula03	7	4	3	26	0.750	0.721
Dcc008 Aula04	7	4	3	26	0.750	0.721
Dcc119 Aula9	8	14	12	25	0.857	0.721
Qui125 Aula1	6	1	1	24	1.000	0.640
Qui125 Aula4	4	3	2	20	0.667	0.620
DCC116 Aula04	11	5	4	33	0.800	0.616
Qui125 Aula8	3	2	2	20	1.000	0.615
Qui125 Aula9	6	7	6	15	0.857	0.533
Est029 Aula1	4	1	1	29	1.000	0.512
Fis2 Grav	22	1	1	16	1.000	0.512
DCC116 Aula06	14	4	3	31	0.750	0.494
DCC116 Aula01	18	5	5	36	1.000	0.440
Dcc119 Ex02	4	1	1	24	1.000	0.328
Qui125 Aula3	5	2	1	36	0.500	0.284
Quim131 Aula2	5	2	2	18	1.000	0.230
Qui125 Aula6	2	5	4	15	0.800	0.200
DCC116 Aula02	26	4	2	24	0.500	0.192
Quim131 Aula1	4	2	2	20	1.000	0.170
Dcc119 Ex01	3	1	1	24	1.000	0.134
DCC116 Aula10	31	4	3	39	0.750	0.035
DCC116 Aula08	21	2	1	34	0.500	0.008
DCC116 Aula05	7	5	0	2	0.000	0.000
Qui125 aula16	5	1	0	0	0.000	0.000
Média					0.837	0.625
Média 29 primeiras consultas					0.909	0.818

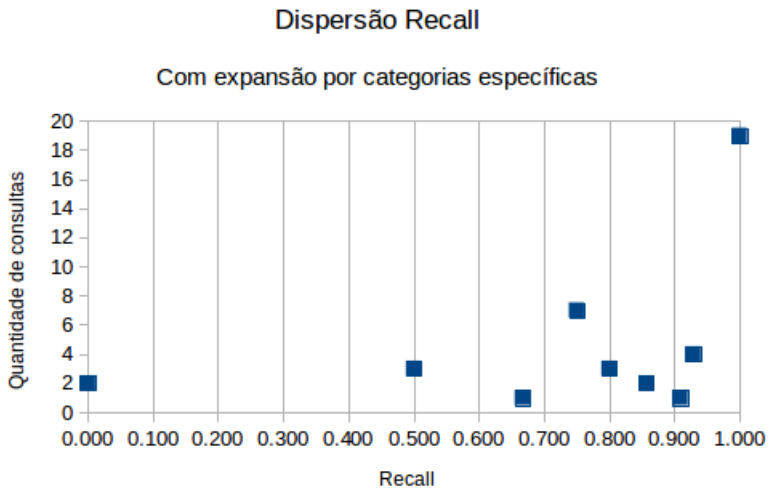

Figura 4.5: Gráfico de dispersão dos valores de *recall* para consultas com expansão por categorias específicas

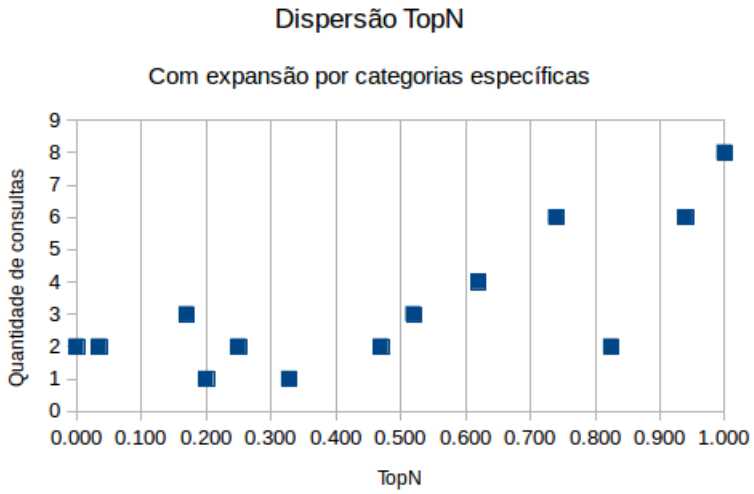

Figura 4.6: Gráfico de dispersão dos valores de TopN para consultas com expansão por categorias específicas

4.5 Análise de consultas específicas

Nesta seção será analisado em detalhes os motivos que levaram algumas consultas obterem resultados muitos satisfatórios enquanto outras os resultados obtidos ficaram muito abaixo da média principalmente no caso da métrica TopN. Para isso dividiu-se as consultas em três casos facilmente identificados nas tabelas 4.1, 4.2 e 4.3. Em cada um dos casos, pelo menos um exemplo será mostrado.

4.5.1 Caso 1: consultas com *recall* alto e TopN alto

Neste caso destacam-se o grupo de consultas realizadas para os vídeos da disciplina de algoritmos (DCC119). Um fator que certamente contribuiu para essa boa colocação, são as características dessas videoaulas que estão relacionadas a quase tudo em Ciência da Computação e com outras áreas. O que facilitou a classificação manual possibilitando realizar um número alto de relacionamentos se comparado com as demais consultas.

Como a DBpedia explicita esses relacionamentos, já era esperado que a Busca encontrasse essas relações, casando com as relações feitas de forma manual, ou seja, para essas consultas a classificação manual foi mais precisa do que para as demais.

O fato dessas videoaulas terem muitos relacionamentos explicitados na DBPedia também justifica o alto número de vídeos relacionados (coluna total retornados) que foi realizado pela Busca, nas abordagens em que houve expansão de categorias (resultados expostos nas tabelas 4.2 e 4.3).

Outras consultas que se destacaram neste caso são as videoaulas de química, possivelmente por terem relações entre si muito fortes e estatística que por ser uma matéria interdisciplinar também possui relação com várias das videoaulas de computação, o que possibilitou as videoaulas tidas como relevantes ficarem bem colocadas no *ranking* e em consequência o valor de TopN também alto.

4.5.2 Caso 2: consultas com *recall* alto e TopN baixo

Neste caso destacam se as consultas cuja a classificação manual considerou um número muito baixo de vídeos relacionados. Além disso, os vídeos referentes a essas consultas geralmente apresentam conteúdos muitos abrangentes como no caso da disciplina de Introdução a Ciência da Computação (DCC116) que por se tratar de uma disciplina introdutória, aborda muitos assuntos de forma superficial como Computação Gráfica e Processamento de Imagens (aula 01), Otimização Combinatória(aula 08) e Bioinformática(aula 10).

Como o conteúdo dessas aulas é bastante genérico, em uma visão subjetiva o classificador manual escolheu relacionar apenas os vídeos mais fortemente relacionados como por exemplo no caso da videoaula de Computação Gráfica (DCC116 Aula01) que possui 5 vídeos que foram relacionados manualmente. Contudo essas videoaulas possuem sim relações com muitas outras coisas e isso fica evidenciado pelos relacionamentos da DBpedia. Tanto que na abordagem em que se expandiu categorias, esta mesma videoaula chegou a ter 44 relacionamentos na abordagem em que se expande por categorias amplas e específicas (tabela 4.2) e 40 relacionamentos na abordagem em que se expande categorias apenas categorias mais específicas (tabela 4.3).

Assim, embora os itens relevantes tenham sido todos retornados (tabelas 4.2 e 4.3) obtendo *recall* máximo, os itens relevantes ficaram espalhados entre pelo resultado resultando numa medida de TopN baixo.

4.5.3 Caso 3: consultas com *recall* baixo e TopN baixo

Apenas uma consulta obteve TopN e *recall* iguais a zero nas três abordagens, a consulta para o vídeo DCC116 Aula05, como pode ser visto nas tabelas 4.1, 4.2 e 4.3.

O vídeo em questão é da disciplina de Introdução a Ciência da Computação e a aula é sobre Métodos Numéricos e os vídeos que foram relacionados a ele na classificação manual são das quatro aulas de Cálculo Numérico(DCC008) e da videoaula de Inteligência Artificial(DCC116 Aula06).

Ao procurar pelos motivos do mal desempenho da busca para este vídeo descobriu-se que, por um erro durante a construção da base, as referências anotadas manualmente

para esse vídeo diziam respeito à videoaula de Redes de Computadores ao invés de Métodos Numéricos. Por isso, mesmo que encontrasse relações com vídeos relacionados, eles seriam relacionados a Redes, diferente dos vídeos que são relevantes para a computação das métricas.

Além de não encontrar relacionamentos para essa videoaula este problema comprometeu a a consulta para a videoaula

e também acarretou uma queda no valor do *recall* em todas as consultas que tinham como videoaula relacionada, a aula sobre Métodos Numéricos, são elas: DCC008 aula 1 a aula 4 e DCC116 aula 6. Em todas elas o número de relacionamentos encontrados pelo algoritmo(coluna retornados certos) foi exatamente 1 a menos que o número de vídeos relacionados manualmente (coluna classif. manual). Nestes casos a videoaula que o faltou ser relacionada pelo algoritmo foi exatamente a problemática aula sobre Métodos Numéricos conforme pode ser observado nas tabelas 4.2 e 4.3.

Embora essa situação, do ponto de vista da avaliação por meio das métricas seja ruim e não tenha sido planejada, do ponto de vista do sistema como um todo ela é muito boa pois mostra que o sistema não irá relacionar vídeos desconexos desde que as referências estejam corretas.

Outra consulta que apareceu nas últimas colocações nas tabelas 4.1 e 4.3 foi uma videoaula de Química (Qui125 aula 16). Ela possui as seguintes recursos anotados como referências.

```
(1) http://pt.dbpedia.org/resource/Reagente
(2) http://es.dbpedia.org/resource/Ácidos
(3) http://es.dbpedia.org/resource/Óxidos_básicos
(4) http://pt.dbpedia.org/resource/Balanceamento_de_equações_químicas
(5) http://pt.dbpedia.org/resource/Reação_ácido-base
```

Como pode ser visto os recursos são da edição em português e espanhol. Analisando esses recursos na DBPedia edição em inglês (http://dbpedia.org), percebe-se que somente o item 5 possui ligação com na propriedade $< owl : sameAs >$. Assim, somente para este recurso foi possível procurar relações.

Um ponto positivo a ser observado é que na abordagem em que permitiu-se a

expansão por categorias mais abrangentes($< skos : broader >$), esta videoaula encontrou relacionamentos, como pode ser visto na tabela 4.2. Diante dessa situação conclui-se que a abordagem que em se permite a expansão por categorias mais abrangentes é interessante para os casos há poucas referências para se iniciar o processo de busca e realizar os relacionamentos entre vídeos.

4.6 Algoritmo de busca com expansão de categorias e poda dos resultados

Após a análise dos resultados das primeiras abordagens decidiu-se que a expansão por categorias mais abrangentes deveriam ser mantidas e para evitar retornar muitos vídeos com assunto pouco relacionados foi usada a estratégia de poda pelo número de categorias em comum.

Para o estudo de caso os resultados que tinham menos de 10 categorias em comum foram descartados mas esse número pode ser alterado dependendo da necessidade do sistema.

Os resultados finais são dispostos na tabela 4.4 abaixo, para essa execução, foi retirado da base o vídeo que continha erro nas referências, mantida a expansão por categorias amplas e específicas e definido um número mínimo de categorias que um vídeo deveria apresentar para ser considerado relacionado a outro.

Pode-se considerar que este resultado é o melhor dentre as quatro abordagens pois apresenta o maior TopN médio igual a 0.644 e *recall* médio de 0.865 e pelos gráficos de dispersão (figura 4.7 e 4.8) percebe-se que a maior parte das consultas se concentram próximas dos resultados ótimos, tanto no caso do *recall* quanto no caso do TopN.

Tabela 4.4: resultados do algoritmo expandindo por categorias amplas e específicas com poda dos vídeos que possuem menos de 10 categorias em comum

Vídeo	Ref. anotadas	Classif. Manual	Retornados Certos	Total Retornados	Recall	TopN
Dcc119 Aula3	6	9	9	21	1.000	1.000
Dcc119 Aula4	7	9	9	21	1.000	1.000
Dcc119 Aula teste	6	9	9	21	1.000	1.000
Dcc119 Aula9	13	2	2	10	1.000	1.000
Qui125 Aula7	7	1	1	19	1.000	1.000
Fis2 Cap18 parte2	7	1	1	17	1.000	1.000
Qui125 Aula10	4	1	1	17	1.000	1.000
Fis2 Tempcalor	15	1	1	16	1.000	1.000
Dcc119 Aula1	13	9	9	21	1.000	0.992
Qui125 Aula5	8	4	4	17	1.000	0.965
Dcc119 Aula1	11	11	10	22	0.909	0.963
Dcc119 Aula6	10	14	13	28	0.929	0.949
Dcc119 Aula5	10	14	13	28	0.929	0.947
Dcc119 Aula7	9	14	13	28	0.929	0.945
Dcc119 Aula8	9	14	13	28	0.929	0.939
Qui125 Aula4	4	3	3	17	1.000	0.840
Dcc008 Aula1	5	4	3	10	0.750	0.827
Quim131 Aula3	7	5	4	12	0.800	0.788
Qui125 Aula3	5	2	2	18	1.000	0.783
Est029 Aula2	6	5	3	10	0.600	0.726
Dcc008 Aula2	8	4	3	17	0.750	0.721
Dcc008 Aula3	7	4	3	17	0.750	0.721
Dcc008 Aula4	7	4	3	17	0.750	0.721
Qui125 Aula9	6	7	7	17	1.000	0.622
Qui125 Aula8	3	2	2	14	1.000	0.603
Dcc116 Aula4	11	5	4	21	0.800	0.560
Dcc116 Aula6	14	4	3	18	0.750	0.497
Dcc116 Aula1	18	5	5	21	1.000	0.442
Est029 Aula1	4	1	1	17	1.000	0.410
Qui125 aula16	5	1	1	12	1.000	0.410
Qui125 Aula1	6	1	1	16	1.000	0.328
Dcc119 Ex2	4	1	1	12	1.000	0.328
Dcc116 Aula3	7	2	1	10	0.500	0.284
Qui125 Aula6	2	5	2	10	0.400	0.219
Dcc116 Aula2	26	4	2	20	0.500	0.185
Dcc119 Ex1	3	1	1	17	1.000	0.168
Fis2 Grav	22	1	1	17	1.000	0.168
Quim131 Aula2	5	2	2	19	1.000	0.165
Quim131 Aula1	4	2	1	16	0.500	0.117
Dcc116 Aula10	31	4	2	36	0.500	0.044
Dcc116 Aula8	21	2	1	24	0.500	0.020
Média					0.865	0.644

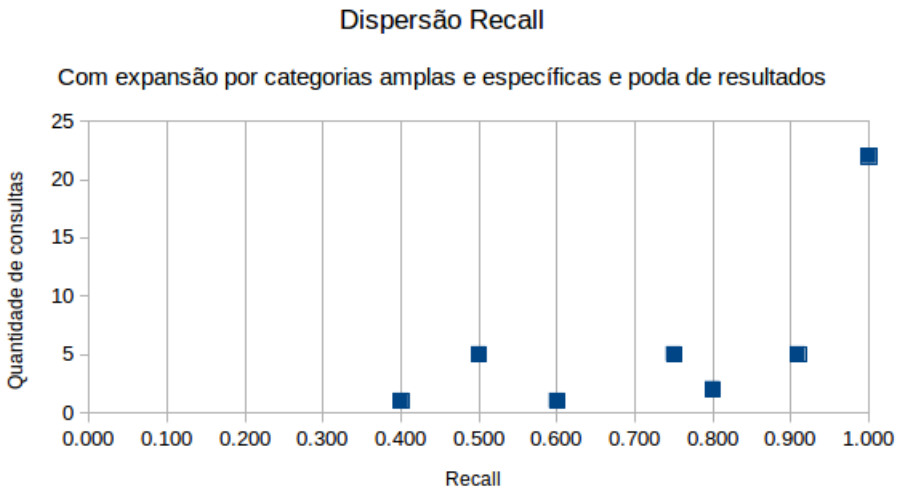

Figura 4.7: Gráfico de dispersão dos valores de *recall* para consultas com expansão por categorias amplas e específicas e poda dos resultados

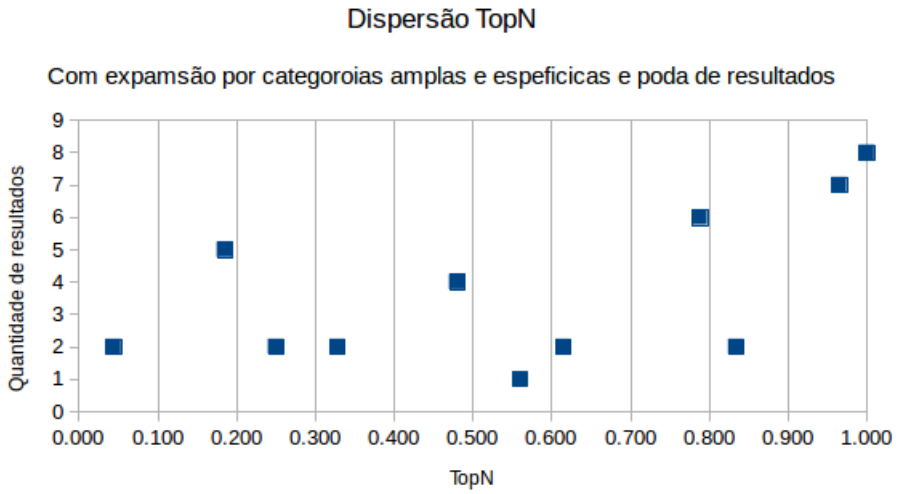

Figura 4.8: Gráfico de dispersão dos valores de TopN para consultas com expansão por categorias amplas e específicas e poda de resultados

4.6.1 Análise de falsos positivos

Conforme apresentado nas seções anteriores, em todas as versões do algoritmo foi retornado um número maior do que a base de classificação manual. Já foi dito também que grande parte dessa diferença deve-se à subjetividade da criação da base. Outro fator é devido ao algoritmo ser capaz de identificar relacionamentos que muitas vezes não são percebidas por uma pessoa.

Para exemplificar essa situação, a tabela 4.5 abaixo apresenta todos vídeos relacionados pelo algoritmo na consulta para a videoaula de física 2 cujo assunto abordado é temperatura e calor (Fis2tempcalor).

Para essa consulta a classificação manual relacionou apenas um vídeo como relevante, que é a videoaula cujo conteúdo abordado é calor específico (Fis2cap18 parte2). Além dessa videoaula o algoritmo relacionou outras 15, que se considerarmos apenas a classificação manual, seriam resultados falso positivos.

Na tabela 4.5 abaixo foram analisados os vídeos retornados como relacionados pelo algoritmo e o assunto abordado por cada um deles foi exibido na coluna conteúdo.

Analisando o conteúdo de cada videoaula é possível perceber as relações que as mesmas possuem com a videoaula de temperatura e calor(Fis2tempcalor).

Por exemplo: uma das aulas relacionadas pelo algoritmo foi a videoaula também de física cujo assunto é gravitação universal (Fis2 Grav Universal). Sabemos da Física que o Sol é fonte de calor e que a Terra é aquecida pelas radiações emitidas pelo Sol. Não por acaso esse assunto é levantado na videoaula de temperatura e calor. Também da Física sabemos que a Terra e o Sol exercem entre-se uma forca de atração que é regida pelas leis de gravitação, logo a videoaula de gravitação está sim relacionada à vídeo aula de temperatura e calor.

Um outro resultado que merece destaque é a videoaula também de física cujo conteúdo são ondas (Fis2 Ondas): uma vez que a luz solar é uma onda eletromagnética e que a radiação emitida pelo Sol chega à Terra através dessa onda, esta videoaula também está muito relacionada à videoaula temperatura e calor. Por motivos semelhantes também se relaciona as videoaulas de Química 131 Aula 2, Química 125 aula 4 e Química 125 aula 5.

Esta seção não pretende encontrar uma justificativa para cada uma das videoaulas relacionadas pelo algoritmo (mesmo que seja possível justificá-las). Pretende-se apenas mostrar com os exemplos citados acima que fica claro que estes falso positivos só estão classificados dessa forma devido à subjetividade da construção manual da base de testes. Talvez se esta base tivesse sido construída por uma pessoa externa ao projeto os resultados colhidos por meio das métricas teriam sido melhores. Não implicando porém, em melhora ou piora da qualidade do trabalho desenvolvido.

Tabela 4.5: Vídeos relacionados pelo algoritmo para a aula de Física 2 - temperatura e calor

Vídeo Relacionado	Área	Conteúdo	Presente na Classif. Manual
Fis2cap18 parte2	Física	Calor específico	Sim
Fis2 Grav	Física	Gravitação universal	Não
Fis2 Ondas	Física	Ondas Eletromagnéticas	Não
Qui125 Aula 3	Química	Modelos atômicos	Não
Qui125 Aula 7	Química	Propriedades periódicas de elementos	Não
Qui125 Aula 2	Química	Modelos atômicos	Não
Qui131 Aula 2	Química	Estrutura atômica e molecular; Luz; Espectroscopia; Radiação eletromagnética	Não
Qui125 Aula 5	Química	Modelos atômicos; Funções de onda	Não
Qui125 Aula 4	Química	Modelos atômicos; Mecânica quântica; Mecânica ondulatória	Não
Qui125 Aula 1	Química	Modelos atômicos	Não
Quim131 Aula 1	Química	Estrutura atômica e molecular	Não
Qui125 Aula 9	Química	Ligações Químicas	Não
Qui125 Aula 10	Química	Ligações iônicas	Não
Dcc116 Aula 10	Ciência da Computação	Bioinformática	Não
Qui131 Aula 3	Química	Estrutura atômica e molecular; Partículas subatômicas	Não
Qui125 Aula 8	Química	Propriedades periódicas dos elementos; Afinidade eletrônica	Não

5 Conclusões

Sistemas clássicos de busca utilizam de informação textual (ou metadados tais como: título, autor, tema central etc.) para processar consultas em vídeo (Croft et al, 2010) (Fensel, 2005). Contudo essa dependência gera demanda de trabalho humano, uma vez que estas informações não estão nativamente no vídeo. Além de a classificação manual não resolver completamente o problema por ser subjetiva e altamente dependente do indivíduo que a realiza, para uma grande base de dados torna-se inviável, devido ao enorme esforço que seria necessário para realizá-la.

Neste trabalho tratamos do problema de buscas em vídeos, utilizando dados ligados para estabelecer relações que não estão explicitas em nossa base de dados. Nossa principal fonte informações foi a DBPedia, utilizamos os relacionamentos descritos em sua ontologia e sua interface de consultas externas para expandirmos os recursos anotados no vídeo e assim expandir a consulta trazendo resultados mais abrangentes.

Conforme os dados apresentados e discutidos no capítulo anterior conclui-se que nossa abordagem é vantajosa sobre as utilizadas em sistemas tradicionais pois: os resultados apresentam aproximadamente 90% de cobertura dos itens relevantes; os vídeos retornados estão entre os primeiros colocados no *ranking* conforme indica os valores de TopN; mesmo para os vídeos com pouca informação de entrada mostrou-se que foi possível estabelecer relações e encontrar relacionamentos; mostrou-se que vídeos totalmente desconexos dificilmente serão retornados como relacionados; e por fim, não se faz necessário o uso de esforço manual para a classificação dos vídeos, já que os mesmos serão indexados pelos módulos ASR e Anotação Semântica.

A principal contribuição deste trabalho foi que teve como resultado o módulo de Buscas e Recomendação para o sistema Qodra, que está sendo desenvolvido na UFJF, o qual fará buscas em vídeo-aulas nos arquivos da RNP. Ademais, mostra-se que a utilização de dados ligados possui grande potencial para ser explorada em motores de busca ou outros sistemas computacionais.

Como trabalhos futuros planeja-se incluir funcionalidades que permitam a ex-

ploração de outras fontes de dados externos como o *Freebase*, o tratamento de consultas em linguagem natural através de Question Answering como o OpemQA, um estudo sobre quais métodos de transcrição obtêm melhores resultados no módulo ASR e quais técnicas de anotação de *tags* são mais eficientes para o módulo Anotação Semântica.

Referências Bibliográficas

Alexa Internet Inc. **The top 500 sites on the web**, Disponível em <http://www.alexa.com/topsites/>, acessado em 11/03/2015, 2015.

Baeza-Yates, R.; Ribeiro-Neto, B. **Modern information retrieval: The concepts and technology behind search (2nd edition)**. ACM Press Books, 2010.

Berners-Lee, T. Linked data-design issues. 2006.

Bizer, C.; Heath, T. ; Berners-Lee, T. Linked data - the story so far. **Int. J. Semantic Web Inf. Syst.**, v.5, 2009.

Brin, S.; Page, L. Reprint of: The anatomy of a large-scale hypertextual web search engine. **Computer networks**, v.56, n.18, p. 3825–3833, 2012.

Burger, J.; Cardie, C.; Chaudhri, V.; Gaizauskas, R.; Harabagiu, S.; Israel, D.; D.; Jacquemin, C.; Lin, C.; Maiorano, S.; Miller, G.; Moldovan, D.; Ogden, B.; Prager, J.; Riloff, E.; Singhal, A.; Shrihari, R.; Strzalkowski, T.; Voorhees, E. ; Weishedel, R. Issues, tasks and program structures to roadmap research in question answering (qa). **QA Roadmap**, october 2003.

Coelho, S. A.; de Souza, J. F. Anotação semântica de transcritos para indexação e busca de vídeos. 2015.

Croft, W. B.; Metzler, D. ; Strohman, T. **Search engines: Information retrieval in practice**. Addison-Wesley Reading, 2010.

Daiber, J.; Jakob, M.; Hokamp, C. ; Mendes, P. N. Improving efficiency and accuracy in multilingual entity extraction. p. 121–124, 2013.

Fensel, D. **Spinning the Semantic Web: bringing the World Wide Web to its full potential**. Mit Press, 2005.

Ferré, S.; Hermann, A. Reconciling faceted search and query languages for the semantic web. **International Journal of Metadata, Semantics and Ontologies**, v.7, n.1, p. 37–54, 2012.

Ferré, S. **Squall: A controlled natural language for querying and updating rdf graphs**. In: Controlled Natural Language, p. 11–25. Springer, 2012.

Guyonvarch, J.; Ferre, S. ; Ducassé, M. Scalable query-based faceted search on top of sparql endpoints for guided and expressive semantic search. **Int. J. Semantic Web Inf. Syst.**, 2009.

Harth, A. Visinav: A system for visual search and navigation on web data. **Web Semantics: Science, Services and Agents on the World Wide Web**, v.8, n.4, p. 348–354, 2010.

PrudâHommeaux, E.; Seaborne, A. ; others. Sparql query language for rdf. **W3C recommendation**, v.15, 2008.

Klyne, G.; Carroll, J. J. Resource description framework (rdf): Concepts and abstract syntax. 2006.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S. ; others. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. **Semantic Web**, 2014.

Lopes, V.; Unger, C.; Cimiano, P. ; , Motta, E. Evaluation question answering over linked data. **Web Semantics: Science, Services and Agents on the World Wide Web**, may 2013.

Marx, E.; Usbeck, R.; Ngonga, A.-C.; Höffner, K.; Lehmann, J. ; Auer, S. Towards an open question answering architecture. p. 57–60, 2014.

Raimond, Y.; Lowis, C. Automated interlinking of speech radio archives. **LDOW**, v.937, 2012.

Sack, H.; Waitelonis, J. Exploratory semantic video search with yovisto. p. 446–447, 2010.

Specia, L.; Motta, E. **Integrating folksonomies with the semantic web**. In: The semantic web: research and applications, p. 624–639. Springer, 2007.

Stamou, G.; Kollias, S. Multimedia content and the semantic web. **Joh n Wiley & Sons**, 2005.

Tumbull, D.; Barrington, L.; Torres, D. ; Lanckriet, G. Semantic annotation and retrieval of music and sound effects. **IEEE transactions on audio, speech, and language processing**, v.16, n.2, p. 467–476, 2008.

J. Waitelonis, H. Sack, J. H.; Kramer, Z. Semantically enabled exploratory video search. **Proceedings of the 3rd International Semantic Search, New York, NY EUA**, 2010.