Objectivo:

Desenvolver a capacidade de representar e processar conhecimento recorrendo aos modelos e técnicas que estão atualmente ligadas ao conceito de Web Semântica. Construir modelos de representação de conhecimento, com ênfase para: a) descrição de recursos (via RDF[S]), e b) especificação de ontologias (via OWL). Explorar o princípio da "Linked Data" integrando diversas perspectivas locais, disponíveis via "SPARQL-endpoint", para compor uma perspectiva global (que unifique as diversas locais) gerida por repositório (RDF4J; *ex Sesame*) também ele disponível como "SPARQL-endpoint". Usar interrogação (via SPARQL) e realizar inferência (e.g., via Fact++, HermiT, Pellet).

Problema:

O principal objectivo de uma publicação científica (ou técnica) é o de documentar e comunicar novas abordagens e novos resultados. O conhecimento cresce sustentado pelas publicações mais válidas e qualquer novo trabalho de investigação (ou de desenvolvimento) irá necessariamente suportar-se nessas publicações; e eventualmente contribuirá também para o seu crescimento através de uma nova publicação (alicerçada nas já existentes).

Na área das Ciências da Computação e da Informática o DBLP (Digital Bibliography & Library Project) [Ley, 2002] é um repositório de referência onde apenas são registadas as publicações de maior qualidade. O repositório DBLP está disponível (às pessoas) via página Web [dblp-webpage] e via SPARQL interativo [dblp-rkbexplorer]. Este último segue a perspectiva "Linked Data" dos repositórios de tripletos RDF(S) para acesso aos dados (por humanos e por máquinas) O repositório da DBLP é um grafo RDF(S) que utiliza conceitos e relações definidos em diversas "namespaces" (taxonomias), tais como [foaf], [swrc], [d2r], [dc], [dcterms] e [akt].

Outro repositório de referência é o CiteSeerX [citeseer-webpage] com SPARQL interativo em [citeseer-rkbexplorer]. Notar que os acessos [citeseer-rkbexplorer] e [dblp-rkbexplorer] já adoptam alguma uniformidade nas taxonomias que usam. Por exemplo, em ambos os acessos um autor é um indivíduo em akt:Person.

Notar que por vezes existe diferença no tempo de resposta (e mesmo na disponibilidade) dos diferentes "SPARQL-endpoint". O estado de disponibilidade de vários "SPARQL-endpoints" é registado e monitorado em [sparqles].

Para introduzir rigor formal e unificar os "namespaces" (taxonomias) o laboratório SWAT (Semantic Web and Agent Technologies Lab) [SWAT] propõe ontologias para diversos contextos incluindo proposta para DBLP [SWAT-owl]. No entanto, esta ontologia [SWAT-owl] apenas consiste numa TBox e atualmente não está a ser usada pelo DBLP.

Admitindo a utilidade e importância de se construir um repositório de referências bibliográficas (em especial durante a realização de uma dissertação ou projeto) vamos combinar a ontologia (ou melhor, a TBox) proposta pelo SWAT e os dados (ABox) fornecidos pelo DBLP e CiteSeerX e construir um repositório de referências bibliográficas pessoais que estará também acessível como um "SPARQL-endpoint". Para isso devem ser implementadas as seguintes 3 fases:

1. Construir TBox adaptando proposta "SWAT Lab" para DBLP (em "`dblp_owl.xml`", e original [SWAT-owl]).
2. Construir ABox usando os dados disponíveis nos "endpoint" da DBLP e CiteSeerX (RDF "Linked Data")
3. Realizar interrogações e inferências e tornar a ontologia disponível como um "SPARQL-endpoint".

Fase 1 (construir uma TBox adaptando a proposta do "SWAT Lab"). Simplificando alguns aspectos e estendendo outros de acordo com as afirmações a seguir enunciadas (alguns termos em Inglês como no original). A ontologia resultante pode ser editada a partir de **`dblp.owl.xml`** mas deve ser guardada no ficheiro "**`rpc_owl.xml`**".

a) Os conceitos primitivos são apenas: `Document`, `Organization` e `Person`. Existem 2 tipos de `Document`: `Publication` e `Unpublished`.

b) Existem 3 tipos de `Publication`: `Article-in-Journal`, `Article-in-Proceedings` e `Book`. Cada documento (i.e., indivíduo em `Document`) pertence sempre a 1 e 1 único dos possíveis tipos (i.e., todos os tipos são disjuntos entre si).

c) Os restantes conceitos (em "`dblp_owl.xml`") podem-se ignorar (e.g., eliminar do `xml` usado o Protégé). O(s) conceito(s) acima definidos e que não estejam em "`dblp_owl.xml`" devem ser aí adicionados.

d) As propriedades a considerar são, no mínimo, aquelas que se obtêm por execução das directivas SPARQL disponíveis em "`z_exemplosSPARQL.txt`". Note que `journal_name`, `title` e `book_title` (em "`dblp_owl.xml`") se referem aos títulos, respectivamente, de `Article-in-Journal`, `Article-in-Proceedings` e `Book`. É também importante notar que a propriedade `has-affiliation` está definida em "`dblp_owl.xml`" como `affiliation` e tem como domínio `Person` e como contradomínio `Organization`.

e) Sugere-se que adicione outras propriedades que considerar relevantes. Deve eliminar, de "`dblp_owl.xml`", todas as propriedades que não considerar relevantes (mantendo naturalmente as impostas pela alínea (d)).

f) É necessário acrescentar (em "`dblp_owl.xml`") o conceito de `Autor` (note que está em Português) que se define como qualquer `Person` que seja autor de algum (pelo menos um) `Document`. Note o domínio e contradomínio de `author`; *sugestão* – quando precisar definir a propriedade inversa de `p` chame-lhe `p_inv`.

g) É necessários acrescentar (em "`dblp_owl.xml`") o conceito de `Autor_Org` que se define como qualquer `Autor` que tenha registo da sua afiliação (i.e., que tenha pelo menos uma relação `affiliation`).

h) O conceito de `Publication` representa qualquer `Document` publicado (`publisher`) por `Organization`.

i) O conceito de `Unpublished` representa qualquer `Document` publicado por `none` (restrição definida no Protégé como: `publisher value none`); i.e., indivíduo `none` representa uma organização nula. Assim definimos a propriedade `publisher` com domínio `Document` e contradomínio `Organization or {none}`.

Ao construir a TBox deve ir testando adicionando indivíduos um-a-um e analisando as inferências obtidas.

Deve testar inserindo os indivíduos: `doc01`, `doc02`, `doc03` em `Document`; `pub01` em `Publication`; `org01`, `org02`, `none` em `Thing`; `person01`, `person02`, `person03` em `Person`.

Deve testar inserindo as relações: (`doc01 publisher org01`), (`doc02 publisher none`), (`doc02 author person02`), (`pub01 author person01`), (`person01 affiliation org01`).

Deve guardar estas testes no ficheiro "**`rpc_owl_TESTES.xml`**". Ficheiro para apresentar na discussão do trabalho.

Sugestão: para testar o resultado das inferências é bastante útil o "**DL Query**" ("plug-in" já instalado) que permite usar expressões "DL" (mesma sintaxe do "*expression editor*") e aí ative "*Instances*" para obter os indivíduos (instâncias da ABox); para além disso pode analisar os indivíduos usando o separador "*Individuals by class*".

Fase 2 (construir uma ABox usando os dados disponíveis nos "endpoint" da DBLP e CiteSeerX).

Vamos tara tirar partido da "RDF Linked Data" construindo uma ABox (da ontologia definida na Fase 1) obtendo informação disponível nos "SPARQL-endpoint" que oferecem dados sobre publicações (científicas ou outras que deseje incorporar) disponível em [dblp-rkbexplorer] e [citeseer-rkbexplorer].

O processo para obter esta informação é o seguinte:

a) Construir um grafo local com os dados a usar nas interrogações aos "SPARQL-endpoint".

b) Interrogar os "SPARQL-endpoint" e estender o grafo local com o resultado dessas interrogações.

Para apoiar o desenvolvimento está disponível um protótipo (**`x01_prototipo_expandir_grafoLocal.py`**) que exemplifica o processo acima descrito. Notar que o protótipo segue no essencial o trabalho da "aula prática 6 (e 5)".

Para obter dados dos "SPARQL-endpoint" é necessário explorar o essencial das taxonomias usadas. Tem especial importância identificar os conceitos e propriedades mais relevantes para o nosso objectivo. A tabela seguinte lista algumas dessas ontologias (ou apenas taxonomias) e o prefixo normalmente usado para a referir.

Prefixo (usual)	Espaço de Nomes ("namespace")
`rdf`	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`
`rdfs`	`http://www.w3.org/2000/01/rdf-schema#`
`owl`	`http://www.w3.org/2002/07/owl#`
`foaf`	`http://xmlns.com/foaf/0.1/`
`akt`	`http://www.aktors.org/ontology/portal#`
`akts`	`http://www.aktors.org/ontology/support#`
`swrc`	`http://swrc.ontoware.org/ontology#`
`dcterms`	`http://purl.org/dc/terms/`
`dc`	`http://purl.org/dc/elements/1.1/`

Alguns dos conceitos e propriedades mais relevantes, em [dblp-rkbexplorer], são: `akt:Person`, `akt:full-name`, `akt:has-author`, `akt:has-title`, `akt:has-date`, `akt:article-of-journal`, entre outros. É bastante simples explorar esta ontologia usando o SPARQL interativo disponível em [dblp-rkbexplorer] (notar uniformidade na utilização da ontologia `akt`).

Para apoiar o desenvolvimento estão disponíveis alguns exemplos (em ***`z_exemplosSPARQL.txt`***) de interrogações SPARQL que exploram determinadas áreas destas ontologias.

A próxima tabela apresenta uma síntese dos elementos de apoio ao desenvolvimento deste projeto, Para além destes elementos deve considerar as aulas práticas da disciplina (especial relevo para as aulas práticas 4, 5, 6, 7).

Protótipo de Apoio	***Descrição (resumida)***
`x01_prototipo_expandir_grafoLocal.py`	Constrói grafo-RDF-local, gL, e usa recursos desse gL como "input" de interrogação ao "DBLB"; os resultados da interrogação são adicionados ao gL (grafo-local) inicial.
`x_util_JSONwithMD.py` `x_util_MINIDOM.py`	Auxiliar para conversão das estruturas devolvidas em interrogações SPARQL; alguns acessos devolvem os dados em formato JSON outros devolvem-nos em XML.
`z_exemplosSPARQL.txt`	***Exemplos de interrogações*** ao DBLP e citeseer.

Fase 3 (realizar interrogações e inferências e tornar a ontologia disponível como um "SPARQL-endpoint").

A TBox (construída na fase 1) e a ABox (construída na fase 2) devem ser analisadas usando o Protégé para inferir, por exemplo, documentos `Publication`, `Unpublished`, `Autor` e `Autor_Org`.

De modo a disponibilizar a informação obtida fazer o seguinte:

a) adicionar num repositório RDF4J a ontologia construída (TBox e ABox), e

b) disponibilizar a ontologia (e.g., usando o "cherryPy") como página *html* (ou acesso Web mais sofisticado).

Atenção: estes último aspecto (alínea b) é um desafio para trabalho mais ambicioso. No entanto, este requisito não deverá nunca prejudicar o desenvolvimento das fases anteriores (fase 1 e fase 2). Note que este aspecto segue o trabalho desenvolvido na aula prática 6.

Referências.

[akt] `http://lov.okfn.org/dataset/lov/agents/AKT%20Project`

[citeseer-rkbexplorer] `http://citeseer.rkbexplorer.com/sparql/`

[citeseer-webpage] `http://citeseerx.ist.psu.edu/`

[d2r] `http://dblp.l3s.de/d2r/snorql/`

[dblp-rkbexplorer] `http://dblp.rkbexplorer.com/sparql/`

[dblp-webpage] `http://www.informatik.uni-trier.de/~ley/db/`

[dc] `http://purl.org/dc/elements/1.1/`

[dcterms] `http://purl.org/dc/terms/`

[foaf] `http://xmlns.com/foaf/spec/`

[Ley, 2002] Ley, M. (2002). The DBLP computer science bibliography: Evolution, research issues, perspectives. In SPIRE 2002, Lisbon, Portugal, September 11-13, 2002, pp. 1–10. Springer.

[sparqles] `http://sparqles.okfn.org/`

[sparql-wrapper] `http://sparql-wrapper.sourceforge.net/`

[SWAT] `http://swat.cse.lehigh.edu/resources/onto/index.html`

[SWAT-owl] `http://swat.cse.lehigh.edu/resources/onto/dblp.owl`

[swrc] `http://swrc.ontoware.org/ontology#`

Regras e Datas:

a. Entregar um **relatório em versão .pdf** com a descrição da abordagem para cada uma das fases do trabalho (fase 1, fase 2 e fase 3).

b. Entregar uma versão electrónica compactada em ficheiro de nome RPC_XX.zip (XX é número do grupo) com todo o sistema desenvolvido (i.e., relatório, modelos, repositórios de informação e código concretizando os diversos aspectos); enviar para **rpc.isel@gmail.com**.

c. Data limite para entre: **até 27.junho.2021 (inclusive)**.

Todos os trabalhos devem ser identificados com o número de grupo.