7
Conclusão

Este trabalho propõe um processo para recomendação de fontes RDF, que usa como base a tecnologia de computação na nuvem para processamento de dados intensivo. Uma ferramenta foi desenvolvida para avaliar o processo proposto, que reúne tecnologias da Web semântica, juntamente com técnicas de processamento distribuído e indexação de dados para dar suporte na busca de fontes na LOD.

7.1
Contribuições

Um dos grandes desafios quando se trata de pesquisa na LOD é o entendimento das diferentes tecnologias usadas no desenvolvimento de aplicativos sobre a Web semântica [71]. É ainda mais complicado integrar tecnologias em um processo de recomendação de fontes de dados RDF. O trabalho desenvolvido consegue coordenar em diferentes níveis as capacidades independentes de cada umas das tecnologias usadas, desde a captura das necessidades do usuário, e guiá-lo através de um processo estruturado, até o consumo e interconexão dos dados de forma escalável.

Nossa abordagem possibilita a geração automatizada de descritores de fontes de dados RDF através do vocabulário VoiD, onde informação estatística da fontes de dados é extraída tais como, *links* a outras fontes de dados, número de triplas, vocabulário usado, quantidade de indivíduos, número de propriedades, número de entidades, e algumas outras que o usuário pode considerar importantes. Os descritores em VoiD contribuem na otimização de consultas federadas [2]. A abordagem possibilita a análise de arquivos de tipo RDF *dump* e inspeção paralela da fonte de dados através do seu serviço SPARQL *endpoint*. Nesta análise são consideradas as diferentes restrição que os responsáveis da publicações aplicam sobre as fontes [2].

Nossa abordagem extrai informação de uma amostra de indivíduos de uma determinada fonte de dados, um processo de análise é aplicado nas propriedades de tipo texto para cada um dos indivíduos escolhidos. Neste processo conseguimos identificar as palavras-chave que representam um

individuo, estas palavras-chaves são ordenadas pelo grau de importância. Sobre um índice semântico, um número de buscas são executadas usando o conjunto de palavras-chave extraídas, visando identificar fontes de dados segundo o interesse do usuário. Para resolver a complexidade de tempo e espaço do processo de análise, nossa abordagem implementa uma solução baseado em computação em nuvem [31].

O desenvolvimento da uma ferramenta baseada em uma arquitetura que apoie no processo de publicação de dados, usando como base o modelo de programação de MapReduce para processamento de dados intensivos, é outra contribuição deste trabalho. O MapReduce é um modelo fortemente ligado com o paradigma de computação na nuvem, que fornece à arquitetura desenvolvida uma abstração sobre os recursos computacionais usados, sem uma dependência de um provedor de recursos em nuvem. A principal justificativa da utilização do modelo MapReduce é a natureza do processo de busca abordado neste estudo. Cada consulta enviada a um serviço SPARQL *endpoint* e as buscas de dados sobre os diferentes serviços usados na arquitetura são independentes uma de outra. O MapReduce sob o principio de "mapeamento e redução", resolve este problema paralelizando o processo de consulta e busca [34]. Além disso, o MapReduce é implementado por Hadoop que permite o processamento paralelo e distribuído de grandes volumes de informação. Assim, nossa abordagem desenvolvida consegue lidar com a complexa natureza da Web de dados.

Disponibilizar um processo de recomendação de fontes RDF, sendo este materializado numa ferramenta de software, que tem como o principal objetivo possibilitar a criação de conexões entre diferentes fontes de dados, colaborando assim com a comunidade científica. Além disso, torna possível que tanto o consumidor quanto o responsável da publicação dos dados possam interagir através de um processo recomendação de fontes RDF. Sendo assim, acreditamos que a ferramenta proposta representa uma peça importante dentro do processo de publicação de dados em RDF [8]. A utilidade de nossa abordagem foi demonstrada através de um estudo de caso.

7.2 Comparação com trabalhos relacionados

O trabalho de Böhm [37] apresenta uma abordagem para geração de descritores VoiD usando MapReduce. A abordagem trabalha sobre um grande conjunto de dados como é o BTC 2010. Uma grande diferença com nosso trabalho é o processamento *online* sobre os dados. Nossa abordagem consegue gerar estatísticas das fontes de dados usando a linguagem SPARQL, através da execução paralela das consultas apoiada no modelo MapReduce.

No caso de Flores [18] é usada uma busca por palavras-chave para encontrar entidades de interesse em fontes de dados governamentais. O processo de busca é complementado com um conjunto de consultas SPARQL que são executadas sequencialmente sobre um conjunto de dados. Este último conjunto de dados é um oraculo criado manualmente e usado para ligar as entidades encontradas. Nosso processo supera as limitações de um processo sequencial, e o processamento de grandes volumes de dados, mas não conta com uma abordagem para a ligação automatizada dos dados.

No trabalho de Maali [16] é apresentada uma comparação entre diferentes abordagens de busca de dados RDF, mas esta comparação é limitada à ligação com DBPedia. O trabalho não consegue aproveitar as capacidades de índice semântico Sindice. A principal diferença com o trabalho de Maali, é que nosso trabalho aproveita os dados de forma *online* da fonte em questão.

O trabalho de Nikolov [15] não materializa a abordagem apresentada numa ferramenta. No trabalho de Nikolov não consegue-se sustentar muitas das afirmações feitas no seu estudo, e que nosso trabalho consegue avaliar. A abordagem de Nikolov propõe o uso de *ontology matching* para o refinamento dos rankings gerados, o que não é abordado pelo nosso estudo..

Uma das características do estudo feito por Ding [7] é a participação do consumidor e do responsável da publicação de dados no mesmo processo de interconexão. Neste aspecto, nosso trabalho tem a vantagem de interagir diretamente estes agentes.

7.3 Limitações

No desenvolvimento deste trabalho foi necessário criar uma série de artefatos para amenizar as limitações das tecnologias usadas. Por exemplo, nossa arquitetura habilita o uso ilimitado de recurso na nuvem, mas a capacidade de resposta dos serviços consultados pode ser restringida, ou simplesmente ficar indisponíveis, como foi o caso do Sindice.

O processamento de grandes volumes de informação precisa de um tempo considerável, o que pode ser crucial para um usuário leigo em computação na nuvem.

O coordenação de diferentes ferramentas trouxe conflitos integração e versionamento pelo uso de bibliotecas em comum. Por exemplo a API de Jena contém alguns artefatos em comum com o Hadoop *framework*, mas em diferentes versões, o que gera alguns conflitos no momento de integrar estes componentes. Recomenda-se aproveitar as funcionalidades de cada uma das tenologias usadas de forma independente.

7.4 Trabalhos futuros

Durante o desenvolvimento do projeto desta dissertação, identificamos as seguintes oportunidades para trabalhos futuros:

- Acrescentar o conjunto de palavras-chave usadas no processo de recomendação de fontes RDF. Em outras palavras, possibilitar ao usuário inserir um conjunto de palavras-chave relacionadas com o tópico de interesse para gerar buscas além das oferecidas pelo conjunto de indivíduos escolhidos no processo de recomendações. E também, evoluir a gestão de recursos em nuvens, conseguindo abranger mais de um provedor em nuvem.
- Possibilidade de criar um conjunto de dados como foi feito em Flores [18], para conseguir conectar os dados de forma confiável, com entidades de fontes conhecidas. Por exemplo, usar um corpus oferecido como o BTC 2010, para obter informações úteis sobre diferentes fontes da LOD.
- Seguindo o trabalho de Nikolov [15], confirma-se a necessidade de aplicar técnicas de *ontology matching* para melhorar os ranking de classes e de fontes RDF. Para aplicar este tipo de técnicas pode-se considerar o uso de *frameworks* como Silk [13] ou Limes [14], especializados na identificação de *links* entre duas fontes.
- Conseguir abranger a busca em diferentes serviços de busca na Web. Como por exemplo, o motor de busca proposto por Hogan [11, 53] *Semantic Web Search Engine* (SWSE), que disponibiliza os dados através de uma API de tipo SPARQL REST [53, 42], ou serviços de SPARQL *endpoint*.
- Evoluir nossa abordagem proposta para um motor de busca, aproveitando que os modelos MapReduce podem coletar grandes quantidades de dados e estando o serviço de indexação e busca de nossa arquitetura apoiado pelas abordagens de SIREn e Solr [43, 47] que possibilitam recuperação de documentos em formato RDF em grande escala. Para isto é necessário considerar o desenvolvimento de componentes que este tipo de tecnologia precisa [11, 75].