BCLaaS: implementação de uma base de conhecimento linguístico *as-a-service*

Mário Mourão[1] and José Saias[2]

[1] Cortex Intelligence - Évora, Portugal
mario.mourao@cortex-intelligence.com
[2] Departamento de Informática - ECT
Universidade de Évora, Portugal
jsaias@uevora.pt

Resumo Na área de Processamento de Linguagem Natural existem operações muito frequentes, independentemente do maior ou menor grau de análise linguística praticada. Um caso muito comum é a consulta da lista de sinónimos de um termo. Num ambiente com várias aplicações deste género, como Sistemas de Pergunta-Resposta, Análise de Sentimentos e outros, a manutenção destes recursos de apoio linguístico junto de cada aplicação torna-se pouco eficaz. Para cada ajuste numa coleção de sinónimos, por exemplo, seria necessário gerir o processo de atualização dos recursos individuais instalados junto das aplicações.
Este trabalho descreve a conceção de uma base de conhecimento linguístico *as-a-service*, considerando aspetos de armazenamento, comunicação e gestão de conteúdo, que permitam uma solução evolutiva e eficiente.

Keywords: Rede semântica, NoSQL, Grafos, *SaaS*

1 Introdução

A cada dia, pessoas e processos geram novos dados que se avolumam e propiciam o surgimento de novos serviços de procura de informação. Estes serviços aplicam técnicas complexas e semanticamente ricas na análise e relacionamento de dados, sejam estruturados, como quantitativos ou categóricos, ou não estruturados como uma publicação textual num fórum. Para tal, é comum a existência de bases de conhecimento, para o apoio na interpretação dos dados. A título de exemplo, uma tabela de sinónimos pode ajudar a captar o significado relevante de um termo. Numa pesquisa de notícias sobre compras de veículos, saber que *comprar* é sinónimo de *adquirir*, permitirá a identificação de mais casos de negócio, o que é fundamental neste serviço.
Na área de Processamento de Linguagem Natural (PLN), o uso destes serviços é cada vez mais comum, para recuperação de documentos, tradução automática e outras aplicações. Ao invés de cada aplicação ter o seu próprio repositório de conhecimento linguístico, este pode ser fornecido como um serviço autónomo que as aplicações consultam, evitando redundância e facilitando a atualização do conteúdo, com repercussão imediata em todas as aplicações cliente.
A centralização do conteúdo poderá trazer, por outro lado, eventuais limitações de desempenho, manifestadas por exemplo no tempo de resposta quando vários clientes geram múltiplas operações de leitura e escrita. Pretende-se que a base de conhecimento assente num serviço eficiente e escalável, cujo repositório tenha capacidade de evoluir sem prejuízo da consistência. Neste artigo, descrevemos uma solução para um serviço deste género, estudada no âmbito de um projeto entre a Cortex Intelligence e o Departamento de Informática da Universidade de Évora.

2 Trabalho Relacionado

O SentiLex-PT é um léxico de sentimento [7] com lemas e formas flexionadas, em Português. Cada entrada do léxico tem indicação de polaridade do sentimento e informação sobre o alvo dessa manifestação de sentimento, para adjetivos, nomes, verbos e expressões idiomáticas. Este recurso é

disponibilizado em ficheiro de texto CSV[3], deixando a cada aplicação cliente a responsabilidade de processar esse ficheiro e representar os dados num formato funcional próprio.
George Miller liderou o projeto WordNet [1], na Universidade de Princeton. Neste recurso, para o Inglês, cada conceito tem associado um *synset*, que aglomera um conjunto de sinónimos. Existem mais de 100.000 *synsets* e entre eles existem relações, incluindo sinonímia, hiperonímia e meronímia. Esta rede de conceitos pode ser consultada via Web[4] ou obtida para fins académicos ou comerciais, em formato de *scripts* Prolog, ficheiros de texto ou XML. A WordNet.PT [2] é uma base de dados de conhecimento lexical do Português desenvolvida no Centro de Linguística da Universidade de Lisboa, e que surge na sequência da WordNet de Princeton. Um conceito corresponde a um nó da rede, que pode ser representado por várias expressões lexicais, e cujo significado poderá ser deduzido pela posição relativa na rede, de acordo com as relações existentes. Existe uma interface Web para consultas aos serviço[5].
Os três casos referidos são bases de conhecimento com reconhecido valor, que funcionam como coleção de dados, que cada aplicação usará à sua maneira.
O WorldCat[6] é uma rede internacional de bibliotecas que dispõe de uma vasta base de conhecimento sobre dados bibliográficos e institucionais, cujo conteúdo evolui diariamente. Para facilitar a partilha da informação e a implementação de diversas aplicações junto dos parceiros desta rede, o acesso à base de conhecimento é normalizado, através de um serviço REST[7]. Na area da linguística, há vários exemplos da utilização de REST em serviços de dicionários[8] e *thesaurus*[9].
O sistema Wolfram|Alpha[10] tem uma abrangente base de conhecimento formada pela aplicação de múltiplos algoritmos sobre diferentes fontes de dados. Permite encontrar resposta a questões, não pela via de pesquisa na Web, mas através de cálculos dinâmicos sobre a base de conhecimento. Para colocar este serviço ao dispor da comunidade, a base de conhecimento é consultada com uma API baseada em REST, que uniformiza e facilita a integração das funcionalidades em aplicações móveis ou Web.

3 Solução Proposta

Esta secção enumera alguns aspetos e opções tomadas para os três pontos fulcrais do serviço: armazenamento, comunicação e gestão de conteúdo.

3.1 Armazenamento dos dados

No trabalho de Saias e Quaresma [5], a escolha automática do resultado para perguntas em língua natural é baseada numa rede semântica. As respostas candidatas, previamente extraídas pelo sistema, são validadas e ordenadas em função da afinidade semântica, entre o conjunto de hipóteses, e entre cada uma e elementos da pergunta. As técnicas empregues no sistema baseiam-se na ativação semântica ao longo das relações da base de conhecimento. Este repositório inclui conceitos associados a diversos domínios e relações como *hiperónimo*, *merónimo*, *sinónimo*, *antónimo*, *instânciaDe* [3]. Em trabalho posterior, sobre interfaces em língua natural [6,4] e análise de sentimentos, outros aspetos sobre o léxico (flexão de vocábulos) ou a semântica (novas relações) foram gradualmente acrescentados a esta base de conhecimento.
Uma vez que o serviço pode ser integrado em várias aplicações, com natureza distinta, e que poderão fazer acessos simultâneos ao sistema, a solução de armazenamento emerge como fator crítico

[3] CSV: *comma-separated values*, é um formato onde há um conjunto de valores em cada linha, separados por vírgula ou outro separador textual como ";" ou ":".
[4] http://wordnetweb.princeton.edu/perl/webwn
[5] http://www.clul.ul.pt/wn/
[6] http://www.oclc.org/uk/en/worldcat/
[7] REST significa *Representational State Transfer*, um protocolo de comunicação cliente-servidor sobre HTTP, alternativo a *Web Services*
[8] Merriam-Webster's Medical dic: http://www.dictionaryapi.com/products/api-medical-dictionary.htm
[9] Big Huge Thesaurus: http://words.bighugelabs.com/api.php
[10] http://products.wolframalpha.com/docs/WolframAlpha-API-Reference.pdf

para o serviço. Uma aplicação pode estar interessada apenas em sinónimos, outra apenas em conjugações de verbos, e outra poderá consultar polaridades de sentimento para alguns vocábulos. Assim, um dos aspetos a considerar é a ocorrência de acessos simultâneos a zonas diferentes do repositório. Depois, pela intenção de tornar o recurso multilingue, espera-se um crescimento substancial no repositório. E este crescimento não é necessariamente uniforme e organizado. Pode haver oportunidade de aumentar informação sobre variações lexicais, definir relações para traduções entre idiomas, adicionar uma entrada num catálogo de entidades mencionadas, ou outras. Assim, foi escolhida uma base de dados (BD) NoSQL[11] baseada em grafos, o Neo4J[12] . O modelo conceptual desta base de conhecimento, que é constituída de conceitos e relações, mapeia na própria estrutura de grafo da BD [8], com nós, propriedades e relações. Desta afinidade, espera-se uma vantagem na gestão dos conteúdos, e também no desempendo. A Figura 1 mostra a interface nativa do Neo4J para visualização da BD. Em Neo4J, as relações entre os nós podem ter um determinado tipo, tal como interessa neste serviço. Na figura, observamos relações do tipo *hiperónimo* e *sinónimo*. Tanto os nós como as relações da BD podem ter propriedades, com um nome e um valor. É através das propriedades Neo4J que se representa a maioria dos dados, como os vocábulos de um certo domínio do conhecimento, ou metainformação para identificar o idioma ou o contexto.
Para encontrar rapidamente o nó de base para um qualquer processo de análise da base de conhecimento, o Neo4J dispõe de um sistema de indexação flexível, adaptável a cada caso e baseado na tecnologia *Apache Lucene*[13]. A título de exemplo, há vantagem em indexar as palavras em Português num índice, usando outro para a indexação desses nós noutro idioma.

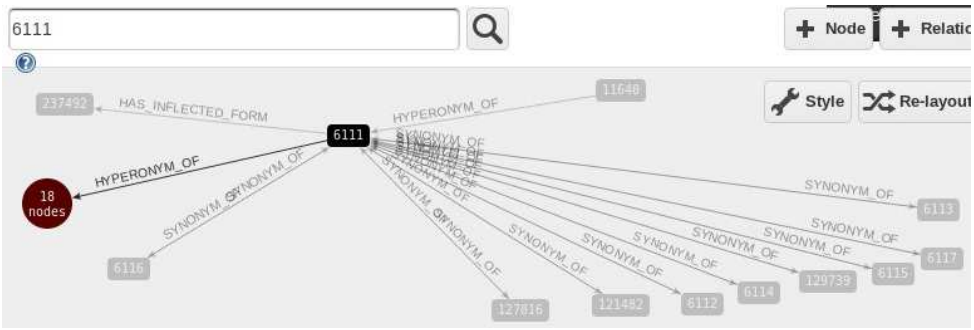

Figura 1. Interface Neo4j para consulta do repositório

3.2 API do serviço

Tendo em mente o futuro uso do sistema em diversas aplicações, possivelmente em ambiente *cloud*, era imperativa a adoção de um protocolo de comunicação universalmente aceite, que não dificultasse o processo de integração, independente da tecnologia da aplicação cliente. Assim, foi estabelecido que o protocolo do serviço seria REST, com formato de dados à escolha do cliente, entre JSON[14] e XML. Por omissão, o formato de envio dos dados é JSON, por ser mais compacto, como ilustrado na Figura 2 para um teste de equivalência semântica entre *adquirir* e *comprar*.
A maioria dos acessos ao serviço são consultas. Outros acessos destinam-se à gestão do conteúdo, com operações para editar os conceitos, relações, ou metainformação. A interface do serviço prevê uma separação destes perfis, um de consulta e outro de gestão de conteúdo.

[11] NoSQL: *not only SQL*. É uma classe de sistemas de gestão de bases de dados que não se baseiam no modelo relacional, nem usam SQL.
[12] Neo4j é uma tecnologia BD em grafos, *open-source* e de alto desempenho. http://www.neo4j.org/
[13] http://lucene.apache.org/
[14] *JavaScript Object Notation*: formato compacto para representação de dados. http://www.json.org/

Pela observação dos pedidos ao serviço, cedo se notou que as aplicações que requerem alguma análise linguística repetem operações. Por exemplo, em Sistemas de Pergunta-Resposta, a verificação da relação de sinónimo entre *t1* e *t2* pode ser necessária N vezes, apenas para o tratamento de uma pergunta. Como a comunicação até ao serviço atravessa a rede, o uso de cache tornou-se necessário. Isto poderia ser gerido pela aplicação cliente, decidindo quando realizar uma consulta ao serviço e quando usar alguma indicação prévia, que teria de manter localmente. A multiplicação desta necessidade, nos ambientes heterogéneos das diferentes aplicações cliente, levou à implementação de um sistema de cache no próprio serviço. Assim, a gestão da cache é transparente para a aplicação cliente, ficando disponível um conjunto de operações: ativar, desativar, ajustar tamanho máximo e o tempo de validade (*lease time*) das entradas.
As entradas em cache ficam em memória, do lado da aplicação cliente, representadas em objetos integrantes da própria API cliente do serviço, e ajudam a minimizar a comunicação com o repositório.

```
$ curl -HAccept:application/xml "http://localhost:8080/synonyms?t1=adquirir&t2=comprar"
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<BooleanValue>true</BooleanValue>

$ curl "http://localhost:8080/synonyms?t1=adquirir&t2=comprar"
{"value":true}
```

Figura 2. Resposta do serviço a um pedido de verificação de sinónimos: formatos XML e JSON

3.3 Aplicação Web para análise e edição do conteúdo

À medida que as aplicações cliente usavam o serviço, foram detetados alguns erros (devido à construção automática de parte do repositório) ou elementos em falta, como o simples estabelecimento de uma relação de sinonímia entre dois termos. Estas situações justificavam a validação, e eventual atualização, de alguns segmentos do repositório, por um humano, possivelmente um linguista. A interface da Figura 1 é demasiado técnica para o nível de abstração necessário a esta análise. Como tal, foi implementada uma aplicação Web para consultas e atualizações à base de conhecimento. Na Figura 3 podemos ver informação sobre um conceito da rede semântica, que, em Português, é *aluno*. Na zona central encontramos um círculo com vários segmentos de cor diferente. Cada segmento representa um tipo de relação. Em cada segmento há um conjunto de fatias, que representam as ligações individuais entre conceitos. No caso, o segmento de sinónimos está ativo. Esta forma amigável de visualização de informações em grafo foi implementada com uma versão modificada da biblioteca neovigator[15]. No canto superior direito podemos ver as principais propriedades do nó: um identificador numérico e a designação em Português. Depois surgem os tipos de relação, as direcionadas para o nó e as que partem daquele nó para outros. Quando é adicionada uma tradução para Inglês, o nó recebe uma nova propriedade com o nome CONCEPT_NAME_EN. Na interface adiciona-se uma tradução para determinado idioma. O modo como essa tradução é representada é escondido pela camada de armazenamento do sistema. A aplicação Web lida diretamente com os conceitos e relações, que aqui são a "lógica de negócio" da base de conhecimento, procurando ser independente da solução particular de armazenamento. Desta forma, o acesso aos dados realiza-se também através da interface REST. A Figura 4 ilustra uma operação em que se define que *aluno* é sinónimo de *discente* (em algum contexto e para a língua base).
Complementando a inserção manual de conteúdo no repositório, existe a possibilidade de importar dados de uma coleção. Uma das aplicações cliente, sobre análise de sentimentos, dispõe de um mecanismo de anotação de frases, para criação de regras para a extração de opinião. A Figura 5 tem um destes casos, onde é marcada a entidade *Universidade de Évora*, um verbo e o adjetivo *lucrativo*.

[15] https://github.com/maxdemarzi/neovigator

Daqui resultam dois elementos passíveis de importação para a base de conhecimento: a entidade e a polaridade positiva associada ao adjetivo. Este adjetivo não existe no recurso SentiLex-PT, pelo que ter esta informação na base de conhecimento será uma importante mais-valia para aquela aplicação cliente.

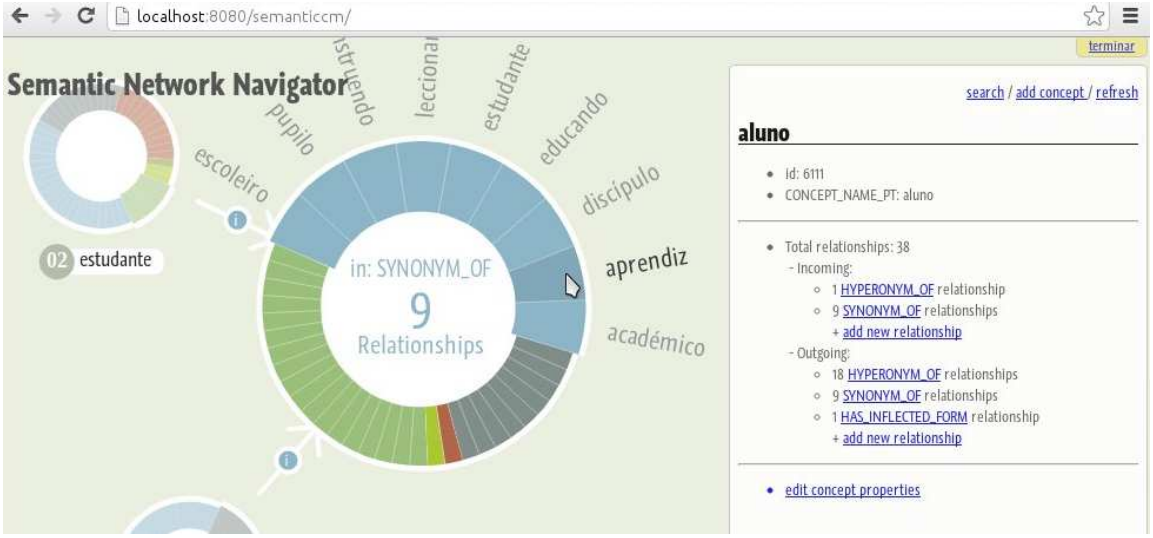

Figura 3. Visualização: resumo e relações de sinonímia para *aluno*

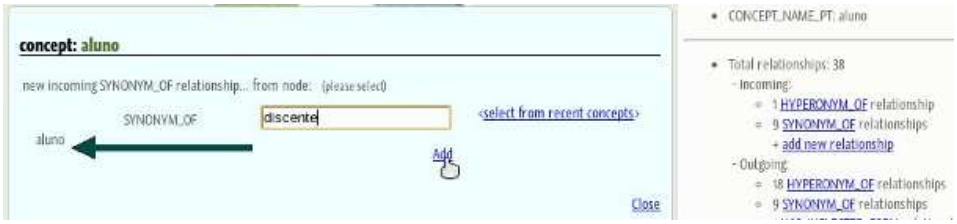

Figura 4. Edição do conteúdo: novo sinónimo

4 Conclusões

Este artigo relata um trabalho de transformação de um recurso estático convencional numa base de conhecimento *as-a-service*. Foram realçados aspetos cruciais para a implementação do serviço, envolvendo o armazenamento, a comunicação e a gestão de conteúdo. Ao evitar a distribuição de cópias do recurso estático convencional (como sucede com alguns dos exemplos referidos na secção 2), simplificamos a manutenção e coerência da base de conhecimento, que se pretende evolutiva e capaz de oferecer o mesmo grau de atualização a todas as aplicações cliente.
Em termos de trabalho futuro, existe o aspeto técnico, de monitorização da adequação desta arquitetura ao aumento do repositório e do número de pedidos a tratar, e uma vertente semântica, que incide sobre o conteúdo. A estrutura do serviço poderá evoluir, se o desempenho o determinar,

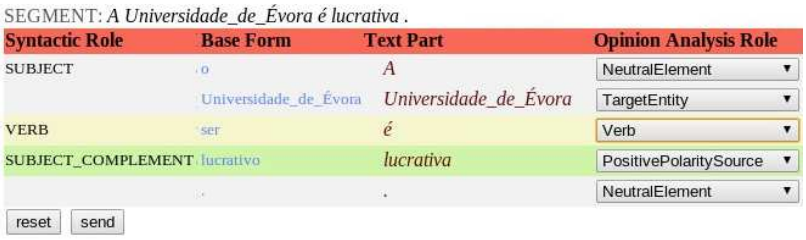

Figura 5. Evolução: importar dados da anotação em aplicações terceiras

designadamente pela introdução de replicação no repositório, o que permitiria algum balanceamento dos pedidos. O modelo de cache implementado no serviço reduz os acessos à BD, permitindo baixos tempos de resposta. Com exceção de cada primeiro pedido, o tempo de acesso à resposta local, em cache, é igual ou inferior ao verificado quando a base de conhecimento era mantida, local e integralmente, na aplicação cliente. Enquanto nesta abordagem o espaço de pesquisa era relativo a toda a base de conhecimento, a pesquisa na cache do serviço incide apenas sobre o subconjunto dos dados relevante para a aplicação cliente.
Paralelamente com a arquitetura do serviço, o conteúdo da base de conhecimento pode também evoluir. É aí que reside o valor deste serviço. A introdução de novas relações, novos conceitos ou traduções, são exemplos do trabalho contínuo de acompanhamento necessário a este recurso.

Agradecimento

Este trabalho enquadra-se numa investigação parcialmente financiada pelo programa QREN/PO Alentejo, no âmbito do projeto ALENT-07-0202-FEDER-018599.

Referências

1. Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.
2. Palmira Marrafa, Raquel Amaro, Rui Pedro Chaves, Susana Lourosa, Catarina Martins, and Sara Mendes. Wordnet.pt – uma rede léxico-conceptual do português on-line. In *XXI Encontro da Associação Portuguesa de Linguística*, Porto, Portugal, Setembro 2005.
3. José Saias. *Contextualização e Activação Semântica na Selecção de Resultados em Sistemas de Pergunta-Resposta*. PhD thesis, Universidade de Évora, 2010.
4. José Saias, P. Quaresma, P. Salgueiro, and T. Santos. Binli: An ontology-based natural language interface for multidimensional data analysis. *Intelligent Information Management*, 4(5):225–230, September 2012.
5. José Saias and Paulo Quaresma. Semantic networks and spreading activation process for qa improvement on text answers. In *Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology - STIL2011*, Cuiabá, Mato Grosso, Brasil, 2011. ISSN: 2175-6201.
6. José Saias and Paulo Quaresma. Di@ue in clef2012: question answering approach to the multiple choice qa4mre challenge. In *Proceedings of CLEF 2012 Evaluation Labs and Workshop - Working Notes Papers*, Rome, Italy, September 2012. ISBN 978-88-904810-3-1.
7. Mário J. Silva, Paula Carvalho, and Luís Sarmento. Building a sentiment lexicon for social judgement mining. In *International Conference on Computational Processing of the Portuguese Language (PROPOR)*, Lecture Notes in Computer Science (LNCS), pages 218–228. Springer-Verlag, 2012.
8. Jim Webber. A programmatic introduction to neo4j. In *Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity*, SPLASH '12, pages 217–218, New York, NY, USA, 2012. ACM.