SODQ - Um sistema de perguntas e respostas para a I OlinCom

Daniel Feitosa[1], Vinícius Rodrigues de Uzêda[1]

[1]Núcleo Interinstitucional de Lingüística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo
CP 668, 13.560-970 São Carlos-SP, Brasil

{feitosa.daniel,vruzeda}@gmail.com

Abstract. *This paper describes the SODQ, a system for Question Answering developed for I OLinCom (www.nilc.icmc.usp.br/~arianidf/olincom).*

Resumo. *Este artigo descreve o SODQ, um sistema para solucionar questões desenvolvido para I OLinCom (www.nilc.icmc.usp.br/~arianidf/olincom).*

1. Introdução

A I Olimpíada Brasileira de Linguística Computacional (I OLinCom) é uma competição científica vinculada ao Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL2009). Este evento possui duas trilhas, no qual, a primeira consiste no desenvolvimento de um sistema de perguntas e respostas de acordo com especificações presentes no site do evento.

Portanto, apesar do SODQ ser um sistema para responder questões de propósito geral, foi desenvolvido nos moldes propostos para I OLinCom, um sistema que recebe um conjunto de perguntas e deve respondê-las, baseado em um conjunto de textos também previamente fornecidos, e exibir como saída a resposta juntamente como texto onde esta foi encontrada.

2. O Sistema

O SODQ foi desenvolvido utilizando a linguagem *Perl*, pelo fato desta ser uma linguagem poderosa para processamento de textos.

Para o sistema, decidiu-se utilizar uma arquitetura simples e que seguisse os moldes da I OLinCom. Desta forma, foi criada a arquitetura exibida na Figura 1, uma adaptação aprimorada de um trabalho previamente desenvolvido para esta mesma finalidade [Balage et al., 2007]. Nesta arquitetura se destacam 3 etapas mais importantes: identificação do tipo de pergunta; extração das palavras-chave da pergunta; e busca da resposta nos textos.

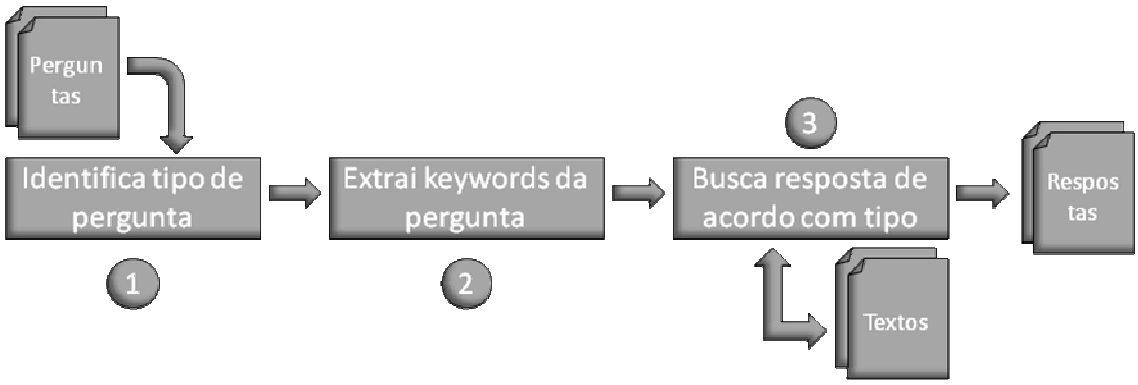

Figura 1: Arquitetura do SODQ

Com base nesta arquitetura, buscou-se tornar o sistema o mais modular possível, deixando seu crescimento mais simplificado. Assim, cada etapa compõe um módulo independente do sistema, responsável apenas por passar sua saída para o módulo seguinte.

Uma característica comum a todos os módulos é a utilização de filtros, pois as tarefas de cada módulo são realizadas a partir destes. E a fácil escalabilidade do sistema está justamente no fato da simplicidade em se alterar ou acrescentar novos filtros em cada um dos módulos.

O primeiro módulo é responsável por identificar o tipo de pergunta, e é fundamental para as próximas etapas, que realizam suas tarefas baseadas no tipo de pergunta recebida. Os filtros utilizados para identificação são criados com base não somente em pronomes relativos, mas também em palavras-chave como idade e pontuação, por exemplo.

O segundo módulo é responsável por extrair as palavras-chave (*keywords*) presentes no texto da pergunta. Os filtros deste módulo utilizam em comum uma *stoplist* com as palavras mais freqüentes do português do Brasil. Os demais critérios dependem do tipo de pergunta que está sendo feita.

O terceiro módulo é o mais complexo, pois utiliza informações dos dois módulos anteriores e possui os filtros mais complicados de serem criados. A busca pelas respostas corretas é feita com base no tipo de pergunta que está sendo feita e utiliza as palavras-chave da forma mais apropriada nesta busca. Esta etapa consiste de duas fases: a busca pelo local onde está a resposta e a extração da resposta deste local. Cada fase possui filtros independentes, muitas vezes parecidos, mas não necessariamente isto é o que ocorre.

Na primeira fase utilizam-se as palavras-chave para buscar o local que supostamente possui a resposta. Esta utilização constitui-se genericamente de um cálculo de concentração de palavras-chave, onde para cada tipo de pergunta é possível haver novos critérios de decisão, como determinadas palavras-chaves terem maior peso ou cálculos feitos de forma diferente. O local de maior concentração é o local candidato. Para esta fase, pode-se também utilizar ferramentas auxiliares como um sentenciador textual, lematizador ou *steamer*. No caso do SODQ foi testado um sentenciador, porém este foi retirado já que prejudicava vários dos resultados.

Na segunda fase é feita a extração da resposta no local indicado. Essa extração é feita por meio da utilização de uma ou mais expressões regulares. Geralmente busca-se por uma resposta mais completa, se esta não existe no local então se utiliza algo cada vez mais superficial, até encontrar uma resposta.

Com a resposta determinada, basta agora criar a saída no formato determinado para a trilha 1 da I OLinCom.

3. Resultados

Para teste e submissão do sistema foi utilizado um corpus constituído de 30 perguntas e 20 textos para buscar as respostas. Com base neste corpus, a avaliação foi realizada da seguinte forma: a resposta para cada pergunta foi buscada manualmente, construindo-se um corpus com as respostas para as perguntas; em seguida, o SODQ foi utilizado para gerar as saídas do sistema; na última etapa as duas respostas foram comparadas, gerando os resultados.

Dos resultados foram avaliados três pontos: o número de textos identificados corretamente, o número de perguntas respondidas corretamente e, subjetivamente, o motivo e localização das respostas incorretas. Desta forma gerou-se a tabela logo abaixo, com as porcentagens dos pontos objetivos avaliados.

	Respostas	Textos
Correto	53,33%	90,00%

Tabela 1: Avaliação do SODQ

Quanto à avaliação das respostas incorretas nota-se que em 78,6% delas as buscas ocorriam nos locais corretos, porém retornavam respostas incorretas.

4. Conclusão

Da avaliação dos resultados é possível perceber que alguns filtros para tipos de perguntas ainda precisam ser criados, porém, a necessidade maior é a de que os filtros para busca das respostas sejam aprimorados.

Apesar do resultado, o sistema possui uma arquitetura modular e robusta, o que permite sua evolução de forma simplificada. Isto torna o SODQ um sistema em potencial, tornando fácil a incorporação de vários métodos durante a criação dos filtros em cada etapa, independentemente.

Referências

Balage Filho, P.P.; Uzêda, V.R.; Pardo, T.A.S.; Nunes, M.G.V. (2007). Experiments on Applying a Text Summarization System for Question Answering. In the *Proceedings of the Cross Language Evaluation Forum 2006 Workshop* – CLEF (Lecture Notes in Computer Science 4730), pp. 372-376. Springer-Verlag Berlin Heidelberg.