PyGER: Uma Ferramenta Geradora de Expressões Regulares a partir de um Conjunto de Expressões em Linguagem Natural

Valéria Delisandra Feltrim[1], Vinícius Mourão Alves de Souza[1]

[1]Departamento de Informática – Universidade Estadual de Maringá (UEM)
Av. Colombo, 5.790 – 87020-900 – Maringá – PR – Brasil

{valeria.feltrim, vsouza}@din.uem.br

Abstract. *A very common procedure used in the natural language processing is the analysis of natural language expressions as well as their transformation into regular expressions. The transformation mentioned before can help finding other important expressions into the body of a text article and also contributes as input data to lexical analyzer generators (scanners). Unfortunately, the mentioned tasks are still made manually and for that reason the development of new strategies are needed, in a way of overcoming the mentioned problems, the present work aims at presenting a computational tool with this goal In a more specific context, the proposed software will reduce the cost of adaptation of the SciPo texts to other domains.*

Resumo. *Um procedimento recorrente no processamento de linguagem natural, consiste na análise de expressões em linguagem natural e transformação das mesmas em expressões regulares para que se possa identificar outras expressões de importância no texto, e também servindo de entrada para geradores de analisadores léxicos (scanners). Essa tarefa é até o momento realizada de forma manual e neste trabalho foram investigadas formas de automatizar tal processo, resultando na implementação de uma ferramenta com esse objetivo. Em um contexto mais específico, este trabalho reduzirá o custo de adaptação do ambiente SciPo a textos de outros domínios.*

1. Introdução

Diferentes ferramentas têm sido desenvolvidas com o objetivo de auxiliar o processo de escrita de textos do gênero acadêmico. Um exemplo é o ambiente SciPo – *Scientific Portuguese* [Feltrim 2004], que consiste em um conjunto de ferramentas integradas com o propósito de auxiliar o processo de escrita de resumos e introduções de textos acadêmicos na área de Ciência da Computação em português.

O ambiente SciPo foi inspirado no projeto AMADEUS – *Amiable Article Development for User Support* [Aluísio *et al*. 2001], um ambiente de auxílio à escrita acadêmica voltado para escritores não-nativos da língua inglesa. Diferentemente do AMADEUS, o SciPo enfoca a escrita em português e tem por objetivo apoiar a estruturação de textos científicos e a sua realização linguística de forma flexível, deixando o usuário livre para escolher entre dois modos de trabalho, a saber: (i) um processo *top-down*, que parte do planejamento estrutural para a escrita propriamente dita, herdado do projeto AMADEUS; ou (ii) um processo *bottom-up*, em que se submete um texto já escrito à análise automática da estrutura.

Para permitir a análise estrutural automática de textos prontos, o SciPo utiliza um classificador estatístico de estrutura esquemática chamado AZPort [Feltrim 2004], [Feltrim *et al.* 2006]. Tal classificador atribui uma categoria retórica para cada sentença do resumo apresentado, com base em um conjunto de oito atributos que são extraídos automaticamente de cada sentença do texto, apresentados na Tabela 1.

Tabela 1. Conjunto de atributos utilizados pelo AZPort

Atributo	Descrição
1. Tamanho	Qual é o tamanho da sentença em comparação aos dois limiares (20 e 40 palavras)?
2. Localização	Qual a posição da sentença no resumo?
3. Citação	A sentença contém citações?
4. Expressão	Que tipo de expressão padrão a sentença contém?
5. Tempo	Qual o tempo do primeiro verbo finito da sentença
6. Voz	Qual a voz do primeiro verbo finito da sentença?
7. Modal	O primeiro verbo finito da sentença é modal?
8. Histórico	Qual a categoria da sentença anterior?

O atributo com maior poder de distinção, denominado “Expressão”, detecta a presença de expressões indicativas do papel retórico da sentença com base em uma lista de expressões regulares gerada manualmente, por meio da observação e anotação de córpus. Uma expressão pode ser considerada indicativa do papel retórico da sentença quando ela (ou suas variações) ocorre com certa frequência em sentenças da mesma categoria retórica.

As expressões analisadas podem ser classificadas entre seis categorias retóricas possíveis, utilizadas pelo AZPort e descritas na Tabela 2.

Tabela 2. Categorias retóricas, descrição e exemplos de expressões indicativas

Categoria	Descrição	Exemplo de expressão
1. Contexto	Conhecimento aceito pela comunidade científica	A partir do ano...
2. Lacuna	Problema de pesquisa, necessidade, ...	Contudo, é necessário...
3. Propósito	Propósito da pesquisa	Esta tese apresenta...
4. Metodologia	Metodologia utilizada	Foi utilizado o modelo...
5. Resultado	Resultados obtidos	Os resultados mostram...
6. Conclusão	Conclusão, recomendação, contribuição, ...	Concluímos que...

Visto que o classificador foi treinado para textos de Ciência da Computação, nota-se um gargalo para a adaptação do AZPort a textos de outros domínios, devido ao

alto custo para coleta e análise manual do córpus, a fim de se gerar a lista de expressões que será usada para a extração do atributo “Expressão”. Além disso, tais expressões devem ser reescritas na forma de expressões regulares, para que reconhecedores (*scanners* ou analisadores léxicos) possam ser implementados computacionalmente.

Dessa forma, este trabalho dedica-se a apresentar uma parte da automatização deste processo, realizado pela ferramenta intitulada PyGER – *Python Gerador de Expressões Regulares*, tendo como objetivo final minimizar o custo de adaptação do classificador AZPort a outros domínios.

Na Seção 2 é descrita a metodologia empregada para o desenvolvimento do trabalho. Os resultados obtidos são relatados na Seção 3, e, por fim, as conclusões são apresentadas na Seção 4.

2. Metodologia

A primeira tarefa a ser realizada para a automatização do processo de criação do recurso necessário ao cálculo do atributo “Expressão” é a extração automática das expressões indicativas a partir de um córpus retoricamente anotado. Tal tarefa vem sendo realizada pela ferramenta GERBBoW, ainda em fase de desenvolvimento e que faz uso do modelo *bag of clusters* proposto por Anthony & Lashkia (2003).

Uma vez extraídas as expressões indicativas, uma segunda tarefa a ser realizada é a transformação automática do conjunto de expressões em linguagem natural para um conjunto de expressões regulares (ER’s). Tal tarefa é realizada pela ferramenta PyGER. Esse conjunto de ER’s servirá como entrada para um gerador automático de analisadores léxicos (*scanners*), que serão utilizados no cálculo automático dos atributos de classificação utilizados pelo AZPort. Vale ressaltar que tais *scanners* podem ser utilizados em qualquer contexto em que se faça necessário o reconhecimento de expressões indicativas.

Atualmente, o PyGER realiza a leitura de um arquivo com um conjunto de *clusters* gerado pelo GERBBoW, onde cada *cluster* possui as seguintes informações: i) sentença extraída; ii) quantidade de vezes que tal sentença ocorre no córpus; e iii) em quais categorias retóricas. Por exemplo: “do critério análise de mutantes 5 PPPBP”, onde P representa a categoria *Purpose* (Propósito) e B representa a categoria *Background* (Contexto).

São selecionadas e agrupadas apenas as sentenças dos *clusters* de acordo com a categoria retórica desejada. As sentenças são novamente agrupadas de acordo com sua semelhança. Para isso, foi criado um atributo chamado GS – Grau de Semelhança. O GS representa a quantidade de palavras iguais levando em conta a sua posição, que também deve ser a mesma, entre diferentes sentenças.

Dessa forma, ao processar *clusters* com sentenças possuindo N palavras, realizamos os seguintes cálculos: GS = (N – 1), GS = (N – 2), ..., GS = (N - N), sendo que, as sentenças que se enquadram em um determinado GS, não são verificadas novamente em um GS menor. Assim, o GS = 0 considera as sentenças restantes que não possuem nenhum grau de semelhança entre si. Não se verifica o GS = N, pois espera-se que não exista nenhuma sentença totalmente semelhante nesse arquivo, ou seja, a mesma sentença não ocorre mais de uma vez no arquivo analisado.

Após o cálculo e manuseio dos *clusters*, a ferramenta gera como saída um arquivo de descrição para a ferramenta Flex, que se trata de uma ferramenta para geração automática de analisadores léxicos.

A Flex basicamente transforma um arquivo contendo expressões regulares em um programa em linguagem C que reconhece os padrões descritos no arquivo ou pode ser modificado para ser usado em conjunto com outros programas.

3. Resultados e Discussão

Como principal resultado, temos a conclusão do desenvolvimento da ferramenta multiplataforma PyGER, utilizando a linguagem de programação Python em conjunto com o módulo PyGTK e construtor de interfaces Glade

A PyGER possui duas versões, sendo a primeira com interface gráfica e a segunda em modo texto, com o objetivo de facilitar a sua interação com diferentes ferramentas além do classificador AZPort. Além do arquivo de saída gerado pela ferramenta, é retornado um conjunto de informações sobre o processamento realizado, como o número de sentenças avaliadas e a quantidade de expressões regulares geradas em cada grau de semelhança.

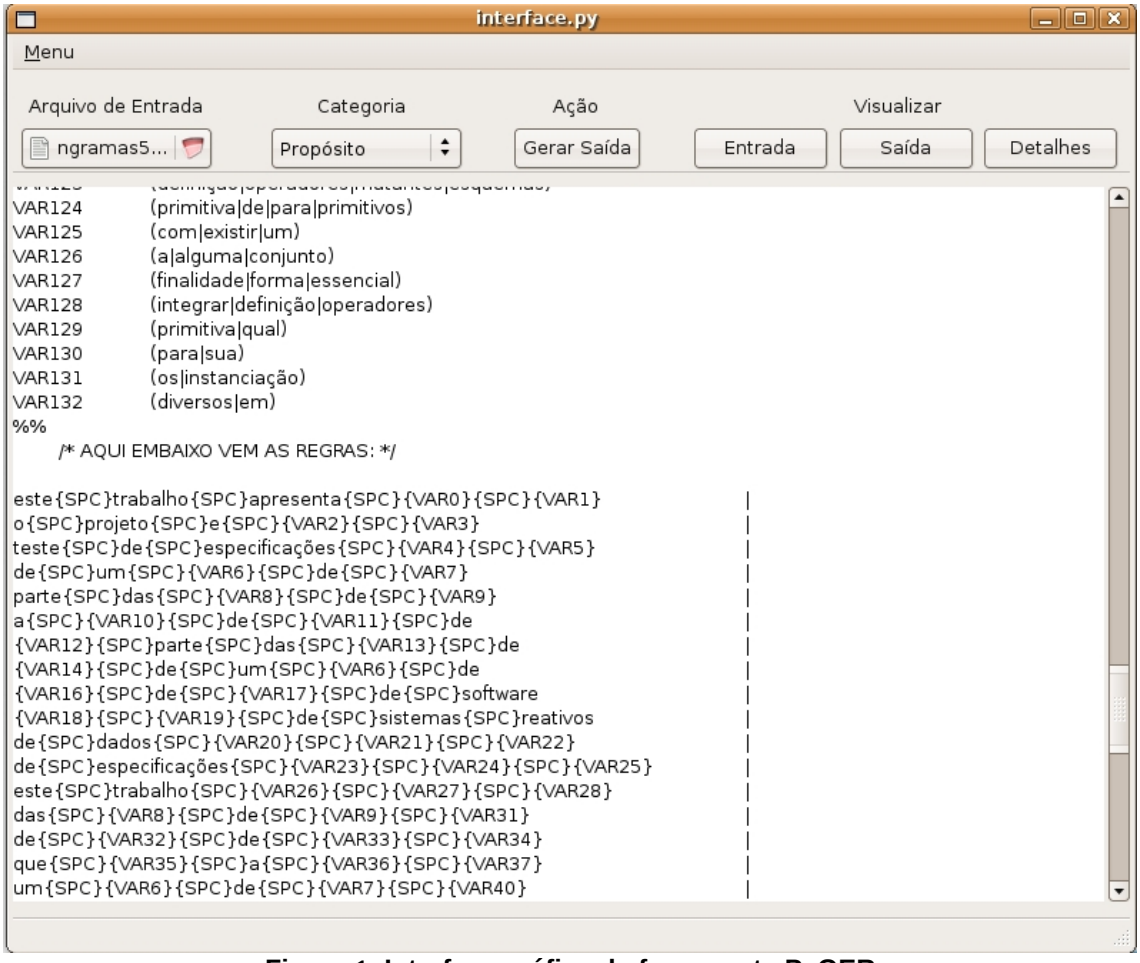

Figura 1: Interface gráfica da ferramenta PyGER

A interface gráfica da PyGER é apresentada na Figura 1, onde também é possível observar parte da lista de expressões regulares geradas como saída de um texto exemplo.

Em testes realizados, a lista de ER's gerada sempre foi menor que a lista de entrada, ou seja, as ER's geradas poderão reconhecer mais de uma expressão em linguagem natural ou, no pior caso, a mesma quantidade de expressões. Obteve-se em média, uma expressão regular para cada três sentenças em linguagem natural, independente da categoria retórica escolhida. Tal redução se faz necessária para melhorar a eficiência do *scanner* gerado. No contexto do AZPort, a eficiência do *scanner* é muito importante, uma vez que sempre que uma nova sentença é classificada, os *scanners* são invocados para o cálculo do atributo "Expressão".

4. Conclusões

Neste trabalho foi apresentada a PyGER, uma ferramenta geradora de expressões regulares a partir de um conjunto de expressões em linguagem natural. O fato de a PyGER ser de código-livre facilita a sua adequação para diferentes ferramentas, embora se pretenda em um primeiro momento beneficiar o ambiente SciPo, amenizando assim o gargalo existente para a adaptação do classificador AZPort a textos de outros domínios.

A média de redução das expressões em linguagem natural para expressões regulares pode ser melhorada por meio da verificação e mudança na ordem das diferentes possibilidades de posições ao calcular o atributo Grau de Semelhança, porém, acredita-se que haverá uma perda de desempenho na execução do aplicativo e que os ganhos ao gerar expressões não sejam significativos.

Além do ambiente SciPo, diferentes ferramentas de processamento de linguagem natural que utilizam expressões indicativas como fonte de conhecimento poderão se beneficiar com os resultados obtidos neste trabalho.

Referências

Aluísio, S.M., Barcelos, I., Sampaio, J., Oliveira Jr., O.N. (2001). How to learn the many unwritten - Rules of the Game of the Academic Discourse: A Hybrid Approach Based on Critiques and Cases. In Proceedings of the IEEE International Conference on Advanced Learning Technologies, p.257–260.

Anthony, L., Lashkia, G.V. (2003). Mover: A machine learning tool to assist in the reading and writing of technical papers. IEEE Transactions on Professional Communication, 46(3), p.185–193.

Feltrim, V.D. (2004). Uma Abordagem baseada em Corpus e em Sistemas de Crítica para a Construção de Ambientes Web de Auxílio à Escrita Acadêmica em Português. *Tese de Doutorado*. ICMC – USP, São Carlos, 181p.

Feltrim, V.D., Teufel, S., Nunes, M.G.V., Aluísio, S.M. (2006). Argumentative Zoning applied to critiquing novices' scientific abstracts. In James G. Shanahan, Yan Qu and Janyce Wiebe (Eds.) Computing Attitude and Affect in Text. Dordrecht, The Netherlands: Springer, p.233-246.