

Universidade de São Paulo - USP
Universidade Federal de São Carlos - UFSCar
Universidade Estadual Paulista - UNESP

Descrição do DMSumm: um Sumarizador Automático Baseado em um Modelo Discursivo

Thiago Alexandre Salgueiro Pardo

NILC-TR-02-02

Março, 2002

Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional
NILC - ICMC-USP, Caixa Postal 668, 13560-970 São Carlos, SP, Brasil



Resumo¹

Neste relatório é apresentada a descrição de um sumário automático de textos baseado em um modelo de discurso que combina conhecimento semântico, retórico e intencional para produzir sumários coerentes. O sumário segue a arquitetura clássica da geração de textos de três passos, isto é, possui os processos de seleção de conteúdo, planejamento textual e realização lingüística, tendo como entrada uma mensagem que compreende um objetivo comunicativo, uma proposição central e um conteúdo informativo. Os dois primeiros componentes são informações pontuais; o último é uma estrutura semântica. A seleção de conteúdo focaliza essa estrutura, podendo-a e fornecendo uma mensagem reduzida para o planejador textual. Este, por sua vez, baseado no modelo de discurso, constrói as possíveis estruturas retóricas do sumário, que são, então, lingüisticamente realizadas.

¹ Este trabalho foi apoiado pela FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo.

ÍNDICE

1. INTRODUÇÃO	1
2. ARQUITETURA	1
3. O SISTEMA DMSUMM.....	3
3.1. IMPLEMENTAÇÃO	3
3.2. PERFIL DO USUÁRIO	3
3.3. PREPARAÇÃO DO TEXTO A SUMARIZAR.....	4
3.4. DADOS DE ENTRADA DO DMSUMM.....	4
3.5. DADOS DE SAÍDA DO DMSUMM	8
3.6. PROCESSAMENTO	9
4. CONCLUSÕES	15
REFERÊNCIAS BIBLIOGRÁFICAS.....	16

1. Introdução

A sumarização automática de textos tem se tornado uma área proeminente devido à crescente demanda por informação no menor tempo possível. Com o advento da internet, onde as pessoas se vêem em um mar de informação em constante expansão e atualização, um grande interesse acadêmico, comercial e governamental surgiu nessa área.

Há diversos tipos de sumários e métodos de sumarização. Previsões meteorológicas, sinopses de novelas, chamadas de notícias jornalísticas, resenhas e *abstracts* de livros e teses, por exemplo, podem ser considerados sumários. Estes podem ser classificados como *indicativos*, *informativos* ou *críticos* (Mani, 2001): sumários indicativos apenas listam ou indicam o assunto principal dos textos-fonte; os informativos são autocontidos, isto é, possuem toda a informação essencial dos textos-fonte, dispensando a leitura destes; os críticos avaliam ou apenas comentam o conteúdo de suas fontes. Os métodos de sumarização, por sua vez, podem ser divididos em duas grandes abordagens: a superficial e a profunda. A abordagem superficial utiliza dados estatísticos e empíricos para a sumarização, enquanto a profunda procura utilizar teorias formais e modelos lingüísticos.

Este relatório apresenta a implementação de uma abordagem profunda de sumarização, o sumarizador automático de textos DMSumm – *Discourse Modeling Summarizer* – que segue o modelo de discurso de Rino (1996), o qual estabelece um relacionamento entre relações profundas do discurso – as relações semânticas, retóricas e intencionais – podendo gerar sumários indicativos e informativos. Diferentemente de trabalhos baseados em métodos estatísticos (por exemplo, Larocca Neto et al., 2000; Knight and Marcu, 2000; Black and Johnson, 1988; Baxendale, 1958) ou fundamentais, baseados em algum tipo de estruturação retórica (por exemplo, Marcu, 1998a, 1998b; O'Donnel, 1997; Hovy, 1988; McKeown, 1985) ou até mesmo em cenários comunicativos limitados (por exemplo, Cawsey, 1993; Maybury, 1992; Moore and Paris, 1993), o destaque deste trabalho é a estruturação do discurso com base na interação entre os diferentes níveis de representação lingüística, consistindo em um modelo consistente e independente de língua. Para mais detalhes sobre métodos de sumarização, vide Martins et al. (2001).

A próxima seção descreve a arquitetura do DMSumm, enquanto a Seção 3 apresenta o ambiente computacional do DMSumm, juntamente com exemplos. As conclusões são mostradas na Seção 4.

2. Arquitetura

No modelo de Rino, procura-se lidar com restrições típicas não só da sumarização (Sparck Jones, 1993), mas da geração de textos também, a saber: a) preservação da proposição central (PC) e b) satisfação do objetivo comunicativo (OC). A PC se refere ao que se quer comunicar com o discurso, enquanto o OC representa a própria motivação para a existência de qualquer discurso. Juntamente com uma base de conhecimento (BC), referente ao conteúdo informativo do texto-fonte que se quer sumarizar, o OC e a PC formam a entrada do sumarizador, a mensagem-fonte (MF). Dessa forma, a MF do DMSumm fornece os principais componentes para a produção do discurso: o OC é responsável por selecionar os componentes textuais que se relacionarão à PC nos sumários e a BC fornece informação, isto é, conhecimento para ser manipulado durante a sumarização. No DMSumm, tanto o OC quanto a PC são informações pontuais; a BC é um componente hierárquico semanticamente estruturado

que também contem a PC. Como podem existir diferentes interpretações para um texto, a MF pode variar, dando origem a diferentes sumários (conforme Mushakoji, 1993).

É importante notar que o DMSumm, caracterizado como um gerador textual, não possui o processo de interpretação automática, sendo a MF produzida a partir da interpretação manual (subjetiva) do texto que se quer sumarizar. O DMSumm sumariza, portanto, a MF correspondente a um texto-fonte, e não o texto-fonte propriamente dito.

O DMSumm abrange os processos clássicos da geração, ou seja, os processos de seleção de conteúdo, planejamento textual e realização lingüística, executados em *pipeline*, sendo o planejamento textual o processo principal que, de fato, é a implementação do modelo de discurso. A Figura 1 apresenta a arquitetura do DMSumm.

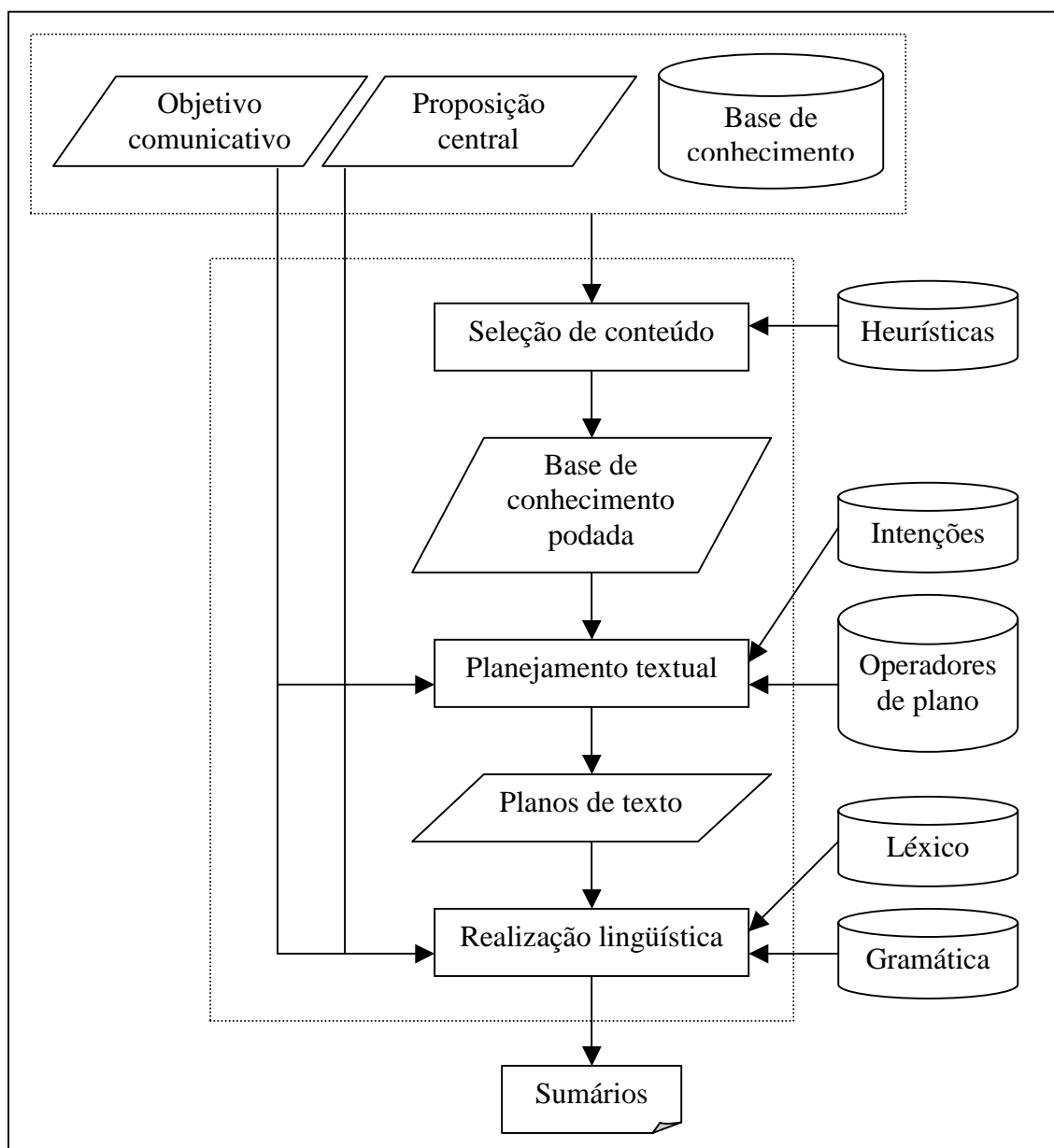


Figura 1 – Arquitetura do DMSumm

A seleção de conteúdo recebe como entrada a MF, tendo a função de reduzir o conteúdo informativo disponível para a produção dos sumários. Esse processo é

composto de duas tarefas: a) podar a BC por meio de heurísticas e b) reproduzir tanto o OC quanto a PC na BC podada. A saída deste processo, a BC podada, e o OC e a PC originais constituem os dados de entrada para o planejamento textual, denominados mensagem-fonte do sumário (MFS).

O planejamento textual, cuja função é estruturar o discurso, recebe como entrada a MFS. A partir desta, são montados os planos de texto (estruturas retóricas) de possíveis sumários com base no modelo de discurso de Rino (1996), mapeando relações semânticas (da BC) e intencionais em relações retóricas. Os modelos semânticos utilizados são o Modelo Problema-Solução (Winter, 1976; Jordan, 1980, 1984) e as relações clausais de Jordan (1992); as intenções são as da GSDT (*Grosz and Sidner Discourse Theory* – Grosz and Sidner, 1986); as relações retóricas são basicamente as da RST (*Rhetorical Structure Theory* – Mann and Thompson, 1987). O mapeamento entre as relações é feito por operadores de plano (OPs), que são um artifício computacional para se montar a estrutura e o conteúdo de textos, conforme o trabalho de Moore e Paris (1993).

O processo de realização lingüística possui a função de produzir os sumários propriamente ditos a partir dos planos de texto, expressando estes últimos em língua natural pela aplicação de *templates*.

Este relatório descreve somente a funcionalidade do DMSumm a sua forma de interação com o usuário, não abordando os conceitos fundamentais e os modelos utilizados em cada processo da sumarização. Detalhes dessa natureza podem ser encontrados em Pardo e Rino (2001a, 2001b) e Pardo (2002a).

A próxima seção descreve o ambiente computacional do DMSumm e a forma como este foi projetado.

3. O Sistema DMSumm

Esta seção explora o ambiente computacional do DMSumm. São discutidos a implementação, o perfil do usuário, a preparação do texto a sumarizar, a forma dos dados de entrada e de saída do sistema e o processamento computacional em si.

3.1. Implementação

A implementação do DMSumm pode ser dividida em duas partes: o engenho de inferência e a interface com o usuário. O engenho de inferência foi desenvolvido em PROLOG; a interface com o usuário em DELPHI.

O engenho de inferência é responsável por todo o processamento do DMSumm. A interface com o usuário é responsável por passar os componentes da MF e outros parâmetros para o engenho de inferência e por apresentar, de forma amigável, os resultados automáticos ao usuário.

3.2. Perfil do Usuário

O DMSumm é destinado a usuários especialistas, visto que é um gerador textual e, portanto, não tem o processo de interpretação automática. O usuário deve conhecer os modelos semânticos, para que possa estruturar a BC a partir do texto-fonte, e as relações intencionais da GSDT.

3.3. Preparação do Texto a Sumarizar

Como o processo de interpretação não é automático, o texto-fonte precisa ser interpretado manualmente para produzir a MF do DMSumm.

A primeira etapa para a interpretação é a segmentação do texto a sumarizar. O texto pode ser segmentado em parágrafos, em sentenças ou, conforme Mann e Thompson (1987), em unidades mínimas de significado. É importante notar que as proposições elementares (folhas) da BC são os segmentos delineados durante essa fase.

Durante a segmentação, deve-se identificar os segmentos com números seqüências. Nos casos em que uma mesma sentença é partida em mais de um segmento, esses segmentos devem possuir os mesmos números identificadores acrescidos de letras distintas, sendo estes colocados entre apóstrofos.

A próxima etapa consiste em identificar no texto o que se quer comunicar, isto é, determinar a PC, que, no modelo de Rino, sempre será um dos segmentos da BC. Deve-se também tentar reconhecer o OC do texto, o que pode se dar em função da PC. Por exemplo, caso o texto tente convencer o leitor de algo, pode-se associar o OC *discutir*. Se o texto explica um procedimento a fim de que o leitor consiga reproduzi-lo, o OC pode ser *relatar*. Se o texto detalha as características de um objeto ou evento para que o leitor seja capaz de identifica-los, o OC pode ser *descrever*. A determinação do OC é de caráter puramente subjetivo, ainda mais em textos em que os OCs estão mesclados. É importante notar que as estruturas retóricas produzidas durante o planejamento textual e, portanto, os próprios conteúdos dos sumários, são diretamente dependentes do OC escolhido, já que cada OC está associado a um grupo distinto de estratégias retóricas.

3.4. Dados de Entrada do DMSumm

Para o funcionamento completo do DMSumm, isto é, a aplicação de todos os seus processos em *pipeline* para a geração dos sumários, os dados de entrada que devem estar disponíveis são os componentes da MF, a base intencional e o texto-fonte segmentado, sendo que estes dois últimos elementos e a BC (da MF) devem estar em um arquivo do tipo texto. A BC será utilizada nos processos de seleção de conteúdo e planejamento textual; a base intencional durante o planejamento textual; o texto-fonte segmentado durante a realização lingüística. O OC e a PC não necessitam estar em arquivos, pois são selecionados diretamente na interface do DMSumm.

A BC deve ser estruturada em um arquivo seguindo uma das regras abaixo, no formato PROLOG, para cada nó da árvore semântica:

1. REL (NÓ PAI , [NÓ FILHO]).
2. REL (NÓ PAI , [NÓ FILHO ESQUERDO , NÓ FILHO DIREITO]).

A primeira regra é aplicada em casos em que um nó tem apenas um filho; a segunda em casos em que um nó tem dois filhos. Como exemplo, considere o texto segmentado da Figura 2 (extraído do *Theses Corpus* – Feltrim et al., 2001). A Figura 3 mostra uma possível BC para esse texto, enquanto a Figura 4 mostra a codificação dessa BC para o arquivo que servirá de entrada para o DMSumm segundo as regras acima. Em casos onde a relação semântica se repete na codificação, deve-se acrescentar um índice a ela. O padrão de índice é o símbolo de *underline* seguido de um número identificador da relação. Como exemplo, vide as relações *simsem* da Figura 4.

[1] A representação de grandes dicionários de língua natural, principalmente nos casos em que se trabalha com vários milhões (ou dezenas de milhões) de palavras, é um interessante problema computacional a ser tratado dentro da área de Processamento de Língua Natural (PLN). **[2a]** Autômatos finitos (AFs), largamente usados na construção de compiladores, são excelentes estruturas para representação desses dicionários, **[2b]** permitindo acesso direto às palavras e seus possíveis atributos (gênero, número, grau, etc).

[3] Um dicionário contendo mais de 430.000 palavras da língua portuguesa sem atributos, cuja representação em formato texto ocupa mais de 4.5Mb, pode ser convertido em um AF compactado de apenas 218Kb, conforme será apresentado (o processo de conversão para esse caso levou 4:17Seg, rodando em um Pentium166/Linux).

[4a] Esse trabalho tem por objetivo investigar métodos eficientes de construção, representação e minimização de AFs de grande porte (dezenas de milhões de estados), **[4b]** a fim de propor um conjunto de estruturas e algoritmos para criação de um sistema eficiente e versátil de representação de grandes léxicos de língua natural.

[5a] Outro objetivo é o de investigar métodos de representação para os atributos das entradas léxicas, **[5b]** de forma a permitir a criação de um formato otimizado capaz de ser inserido diretamente no AF, com o menor impacto possível no tamanho final desse último.

[6] Será também desenvolvido um protótipo do sistema, para efeito de testes e confirmação de eficiência do sistema.

Figura 2 – Texto

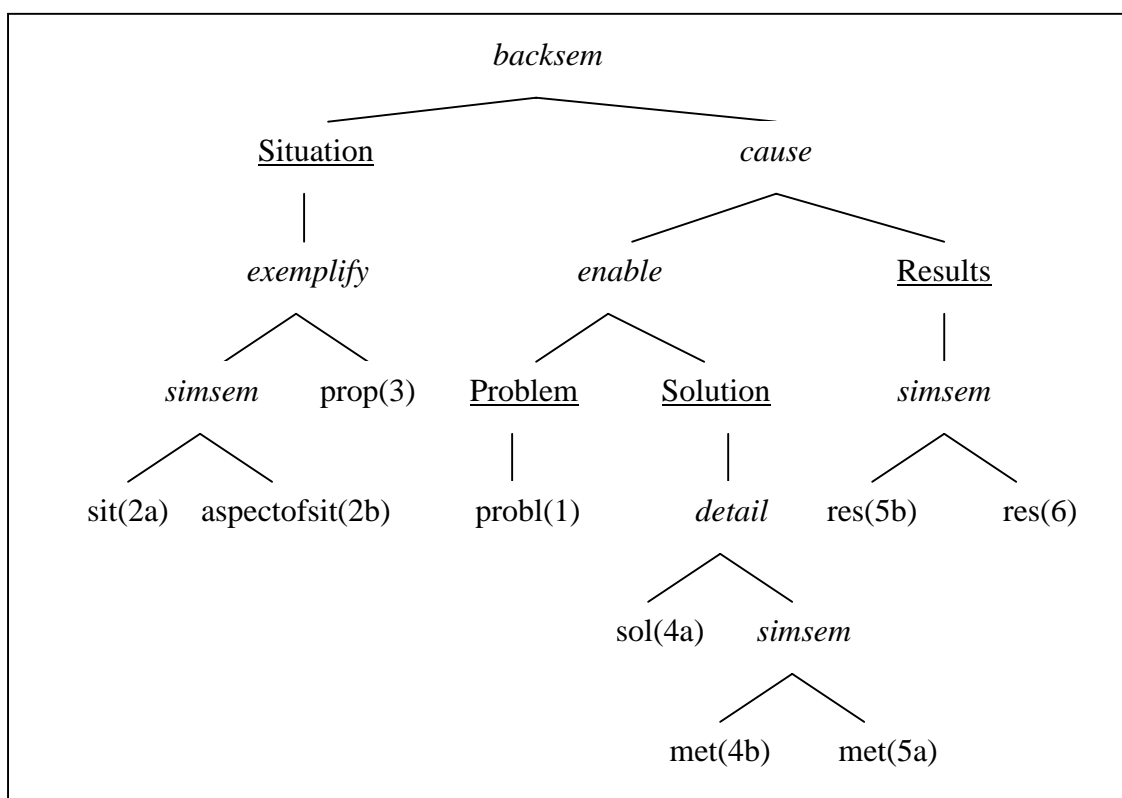


Figura 3 - BC

```

rel(backsem,[situation,cause]).
rel(situation,[exemplify]).
rel(exemplify,[simsem,prop(3)]).
rel(simsem,[sit('2a'),aspectofsit('2b')]).
rel(cause,[enable,results]).
rel(enable,[problem,solution]).
rel(problem,[probl(1)]).
rel(solution,[detail]).
rel(detail,[sol('4a'),simsem_1]).
rel(simsem_1,[met('4b'),met('5a')]).
rel(results,[simsem_2]).
rel(simsem_2,[res('5b'),res(6)]).

```

Figura 4 – BC codificada

O texto-fonte segmentado deve seguir a seguinte regra no formato PROLOG, onde a sigla *PAR* indica *PARÁGRAFO* e o *IDENTIFICADOR DO SEGMENTO* corresponde aos números identificadores dos segmentos determinados durante a preparação do texto a sumarizar:

PAR (IDENTIFICADOR DO SEGMENTO , 'SEGMENTO TEXTUAL').

A Figura 5 exibe a codificação do texto-fonte segmentado da Figura 2.

A base intencional, por sua vez, segue uma das regras no formato PROLOG abaixo, onde TAG1 e TAG2 correspondem às etiquetas do modelo Problema-Solução, conforme especificado em (Rino, 1996). A Figura 6 mostra a codificação das intenções para o texto da Figura 2. O reconhecimento das relações intencionais em um texto pode ser feito conforme sugerido por Grosz e Sidner (1986). Vale notar que, durante o desenvolvimento do DMSumm, uma base intencional genérica foi especificada, podendo ser utilizada, a princípio, para quaisquer textos do gênero científico que seguem o modelo Problema-Solução.

1. DOMINATES (TAG1 , TAG2).
2. SATISFACTION-PRECEDES (TAG1 , TAG2).
3. SUPPORTS (TAG1 , TAG2).
4. GENERATES (TAG1 , TAG2).

par(1,'A representação de grandes dicionários de língua natural, principalmente nos casos em que se trabalha com vários milhões (ou dezenas de milhões) de palavras, é um interessante problema computacional a ser tratado dentro da área de Processamento de Língua Natural (PLN)').

par('2a','Autômatos finitos (AFs), largamente usados na construção de compiladores, são excelentes estruturas para representação desses dicionários').

par('2b','permitindo acesso direto às palavras e seus possíveis atributos (gênero, número, grau, etc)').

par('3','Um dicionário contendo mais de 430.000 palavras da língua portuguesa sem atributos, cuja representação em formato texto ocupa mais de 4.5Mb, pode ser convertido em um AF compactado de apenas 218Kb, conforme será apresentado (o processo de conversão para esse caso levou 4:17Seg, rodando em um Pentium166/Linux)').

par('4a','Esse trabalho tem por objetivo investigar métodos eficientes de construção, representação e minimização de AFs de grande porte (dezenas de milhões de estados)').

par('4b','a fim de propor um conjunto de estruturas e algoritmos para criação de um sistema eficiente e versátil de representação de grandes léxicos de língua natural').

par('5a','Outro objetivo é o de investigar métodos de representação para os atributos das entradas léxicas').

par('5b','de forma a permitir a criação de um formato otimizado capaz de ser inserido diretamente no AF, com o menor impacto possível no tamanho final desse último').

par(6,'Será também desenvolvido um protótipo do sistema, para efeito de testes e confirmação de eficiência do sistema').

Figura 5 – Texto-fonte segmentado em arquivo

```

dominates(sol,probl).
dominates(met,probl).
dominates(res,probl).
dominates(res,sol).
dominates(res,met).
dominates(sit,aspectofsit).
dominates(aspectofsit,sit).

satisfaction-precedes(probl,sol).
satisfaction-precedes(probl,met).
satisfaction-precedes(probl,res).
satisfaction-precedes(sol,res).
satisfaction-precedes(met,res).

supports(met,sol).
supports(prop,sit).
supports(prop,aspectofsit).
supports(sit,sol).
supports(sit,met).
supports(sit,probl).
supports(sit,res).
supports(aspectofsit,sol).
supports(aspectofsit,met).
supports(aspectofsit,probl).
supports(aspectofsit,res).
supports(prop,sol).
supports(prop,met).
supports(prop,probl).
supports(prop,res).

generates(probl,res).
generates(sol,res).
generates(met,res).

```

Figura 6 – Base intencional em arquivo

3.5. Dados de Saída do DMSumm

Os dados de saída do DMSumm são armazenados em arquivos. Os seguintes arquivos são gerados após o processamento do sistema:

- *bc_condensada.pro*: após a seleção de conteúdo, este arquivo contém a base de conhecimento reduzida segundo as heurísticas selecionadas;
- *pts.sum*, *planos.sum*, *prea.sum*: após o planejamento textual, estes arquivos contêm, respectivamente, um histórico da geração dos planos de texto, os planos de texto para serem exibidos na interface de exibição dos planos (vide Figura 13) e os planos de texto para serem realizados lingüisticamente;
- *sumários.sum*: após a realização lingüística, este arquivo contém os sumários referentes aos planos do arquivo *prea.sum*.

3.6. Processamento

O sistema possui interfaces específicas para cada um dos processos de sumarização, isto é, a seleção de conteúdo, o planejamento textual e a realização lingüística, que podem ser selecionados por meio dos botões na parte superior da interface, conforme pode ser visto na Figura 7.

A Figura 7 mostra a interface para o processo de seleção de conteúdo. Nesta interface, o usuário deve selecionar a BC, as heurísticas² para podar essa base e a ordem de aplicação das heurísticas. Na figura, foram selecionadas:

- a BC da Figura 3 selecionando-se a opção *outra*. Ao selecionar esta opção, o sistema mostra uma janela de abertura de arquivo, onde se deve selecionar o arquivo desejado. As BCs já indicadas na interface são exemplos prontos que foram utilizados para a modelagem inicial do sistema;
- as heurísticas *excluir detalhes de entidades* e *excluir cenário contextual*;
- a ordem de aplicação das heurísticas, de acordo com as caixas de texto ao lado das heurísticas habilitadas. Neste caso, a heurística para *excluir detalhes de entidades* será aplicada primeiro.

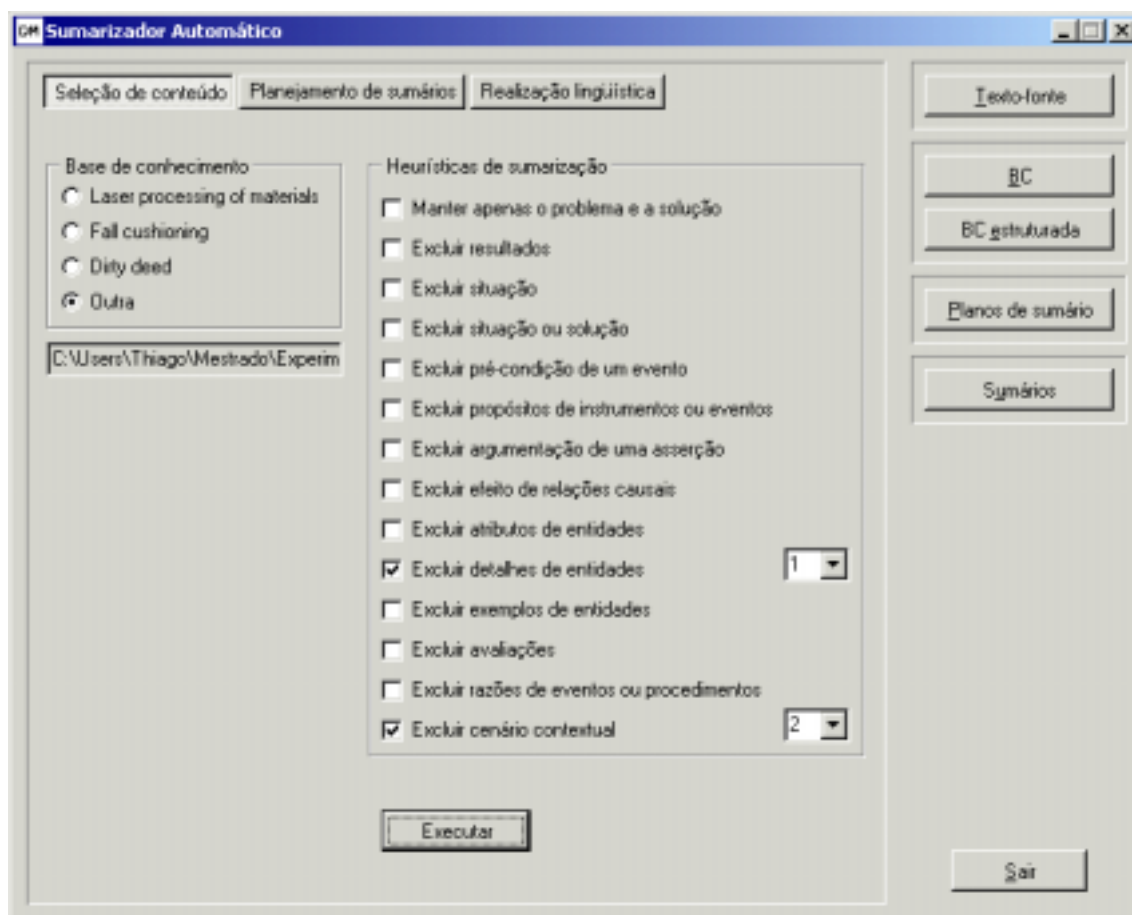


Figura 7 – Interface da seleção de conteúdo

Vale notar que a determinação da ordem em que as heurísticas são aplicadas pode gerar diferentes BCs podadas. Para executar a seleção de conteúdo, basta selecionar o botão *executar*. A BC podada é mostrada na Figura 8.

² Para mais detalhes sobre as heurísticas da seleção de conteúdo, vide Rino e Scott (1994).

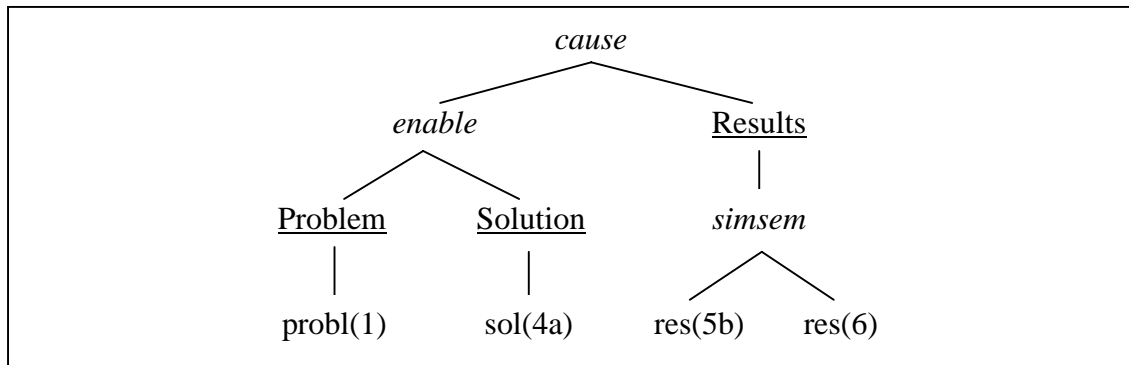


Figura 8 – BC podada

A interface para o planejamento textual é mostrada na Figura 9. Nesta interface, o usuário deve selecionar a BC, o grupo de operadores de plano (OPs), a base intencional, o OC, a PC e as opções sobre o nível (profundidade) e o número de planos a serem gerados. Na figura, foram selecionados:

- a BC podada da Figura 8 selecionando-se a opção *outra* (quando se seleciona uma das BCs já disponíveis na interface para a seleção de conteúdo, a opção *Base condensada* é automaticamente selecionada pelo sistema durante o planejamento);
- o grupo de OPs genéricos;
- a base intencional da Figura 6 selecionando-se a opção *outras* na caixa de texto correspondente (novamente, caso se selecione uma das BCs da interface, a base intencional é selecionada automaticamente);
- o OC *relatar* (utilizando-se a barra de rolagem da caixa de texto correspondente);
- a PC *sol(4a)* (utilizando-se a barra de rolagem da caixa de texto correspondente)³;
- a opção de planejamento *qualquer* para o nível dos planos a se gerar;
- a opção de planejamento *todos* para o número de planos a se gerar.

Os grupos de OPs são conjuntos de operadores de plano já selecionados para uma determinada combinação de OC e PC (esses grupos de OPs foram desenvolvidos para as BCs utilizadas para a modelagem do sistema). Por exemplo, caso se queira *descrever* um *problema*, é possível selecionar o segundo grupo de operadores. A opção escolhida, *operadores de plano genéricos*, indica o grupo irrestrito de operadores, ou seja, que não foi customizado para nenhum tipo de combinação de OC e PC, podendo ser aplicado para quaisquer dados de entrada. Essa opção pode permitir maior variedade na geração de planos, entretanto, pode consumir mais tempo de processamento caso se escolha gerar *todos* os planos. O usuário pode, ainda, escolher dentre os operadores de plano genéricos os operadores que lhe convém para o seu caso particular, montando um arquivo a parte e selecionando-o por meio da opção *outros* na interface.

Para executar o planejamento textual, basta selecionar o botão *executar*. Os planos de texto resultantes, em número de 14, são mostrados de forma hierárquica na Figura 10.

A Figura 11 mostra a interface para a realização lingüística. O usuário deve selecionar o arquivo de planos de texto, o idioma e o arquivo com o texto-fonte segmentado. Na figura, foram selecionados:

- o arquivo de planos de texto referente à Figura 10;
- o idioma *português*;
- o arquivo com o texto-fonte segmentado da Figura 5.

³ A caixa de texto da PC lista todos os segmentos da BC selecionada.

Como se pode notar, a interface oferece as opções de idioma *inglês* e *português*, já que foram desenvolvidos grupos de *templates* para os dois casos. Os botões *procurar* e *padrão* abaixo da caixa de texto que indica o arquivo de planos de texto servem, respectivamente, para selecionar um arquivo de planos de texto qualquer e para selecionar o arquivo de planos de texto que é gerado no diretório onde o DMSumm se encontra. Sempre que se executa algum dos casos-exemplo da interface, o arquivo de planos de texto é gerado no próprio diretório do DMSumm.

Para se executar a realização lingüística, basta selecionar o botão *executar*. Os sumários gerados para os quatro primeiros planos de texto da Figura 10 são mostrados na Figura 12.

Figura 9 – Interface do planejamento textual

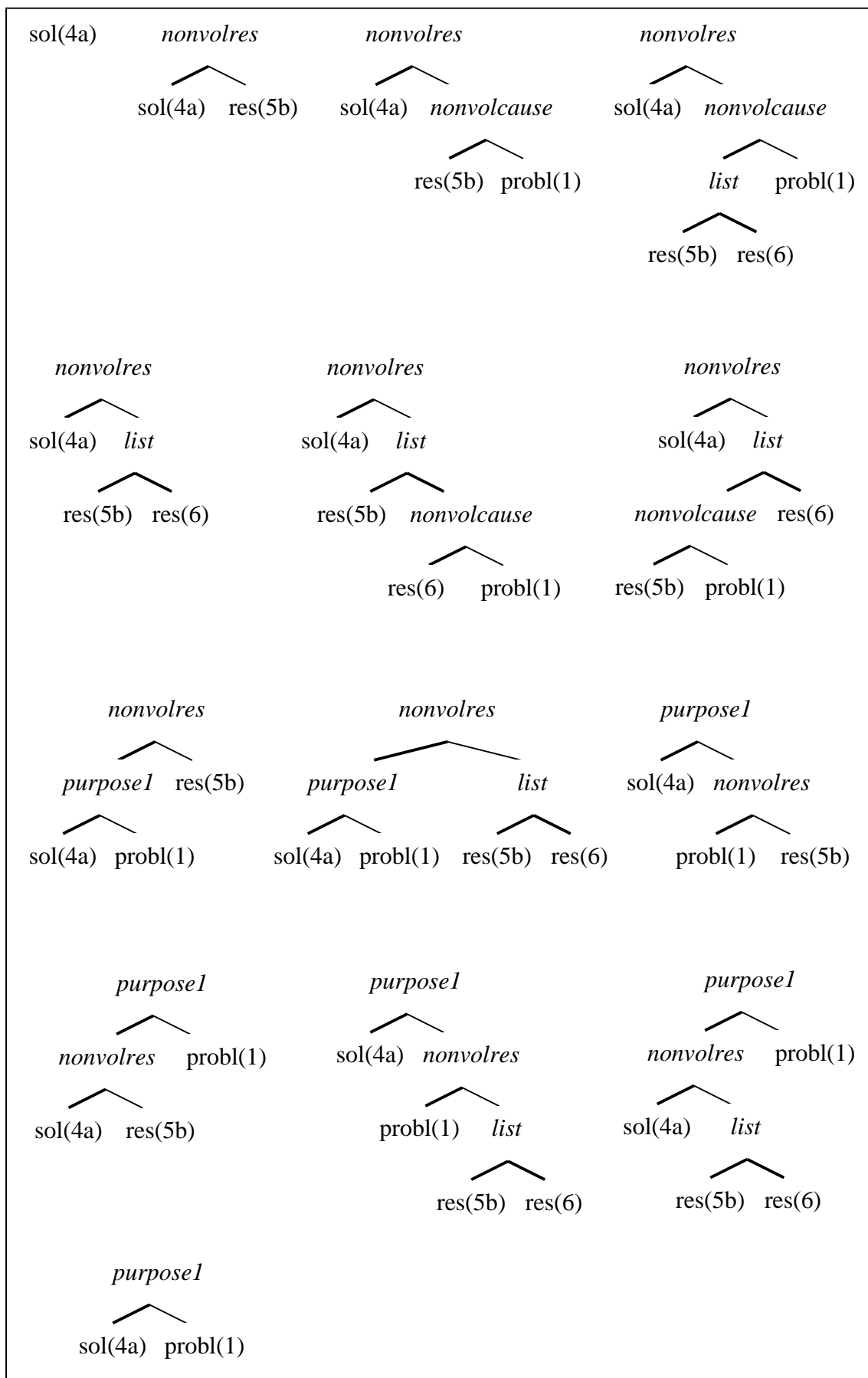


Figura 10 – Planos de texto

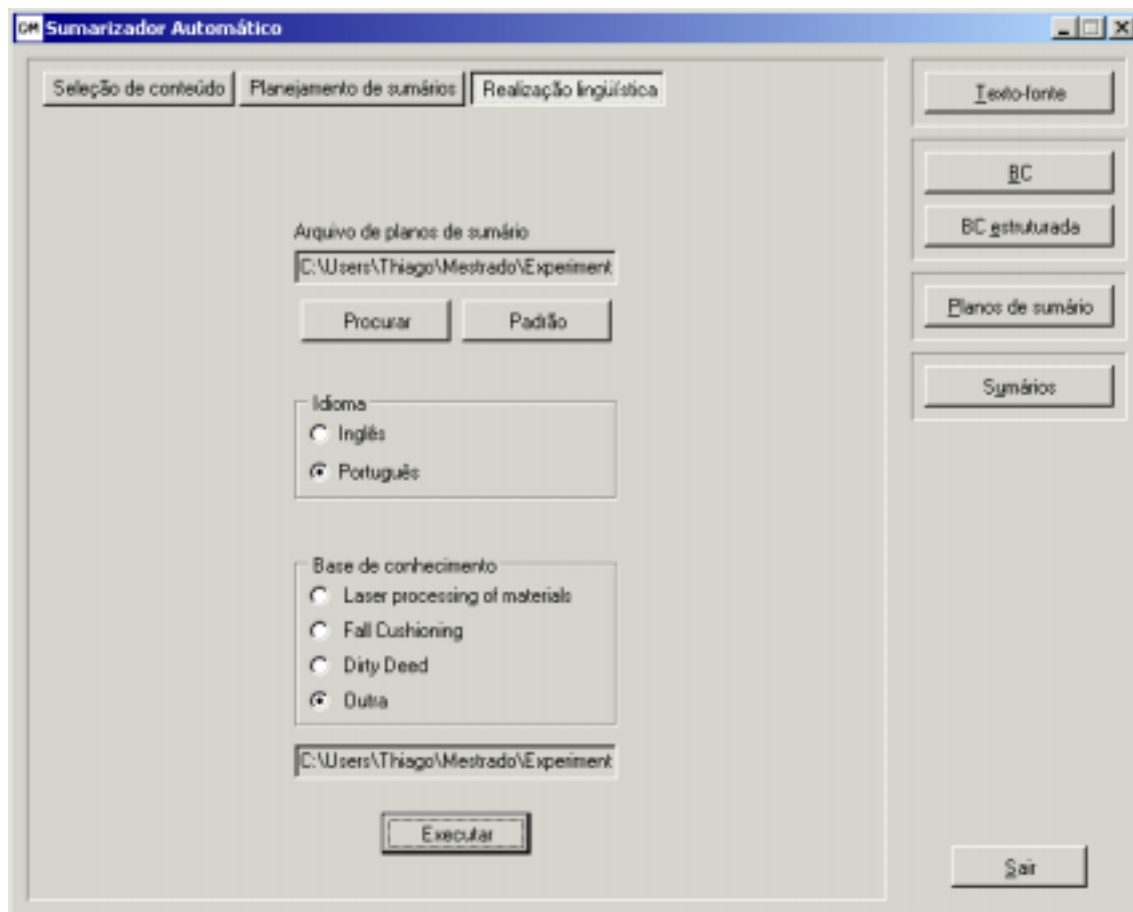


Figura 11 – Interface da realização lingüística

Sumário 1:

Esse trabalho tem por objetivo investigar métodos eficientes de construção, representação e minimização de autômatos finitos de grande porte (dezenas de milhões de estados).

Sumário 2:

Esse trabalho tem por objetivo investigar métodos eficientes de construção, representação e minimização de autômatos finitos de grande porte (dezenas de milhões de estados). Como resultado, será possibilitada a criação de um formato otimizado capaz de ser inserido diretamente no autômato finito, com o menor impacto possível no tamanho final do autômato.

Sumário 3:

Esse trabalho tem por objetivo investigar métodos eficientes de construção, representação e minimização de autômatos finitos de grande porte (dezenas de milhões de estados). A representação de grandes dicionários de língua natural, principalmente nos casos em que se trabalha com vários milhões (ou dezenas de milhões) de palavras, é um interessante problema computacional a ser tratado dentro da área de Processamento de Língua Natural. Conseqüentemente, será possibilitada a criação de um formato otimizado capaz de ser inserido diretamente no autômato finito, com o menor impacto possível no tamanho final do autômato.

Sumário 4:

Esse trabalho tem por objetivo investigar métodos eficientes de construção, representação e minimização de autômatos finitos de grande porte (dezenas de milhões de estados). A representação de grandes dicionários de língua natural, principalmente nos casos em que se trabalha com vários milhões (ou dezenas de milhões) de palavras, é um interessante problema computacional a ser tratado dentro da área de Processamento de Língua Natural. Conseqüentemente, será possibilitada a criação de um formato otimizado capaz de ser inserido diretamente no autômato finito, com o menor impacto possível no tamanho final do autômato. Será desenvolvido um protótipo para efeito de testes e confirmação de eficiência do sistema.

Figura 12 – Sumários

Outras facilidades oferecidas pela interface incluem a fácil visualização dos dados do sistema. Neste caso, os botões à direita da interface são utilizados para a visualização dos dados de entrada e de saída do DMSumm. Os dois primeiros botões (*Texto-fonte* e *BC*) e o último botão (*Sumários*) são simples editores de texto que exibem o conteúdo de um arquivo qualquer. Os botões *Planos de sumário* e *BC estruturada* mostram de forma hierárquica os planos produzidos automaticamente e a BC. As Figuras 13 e 14 mostram as interfaces destes últimos.

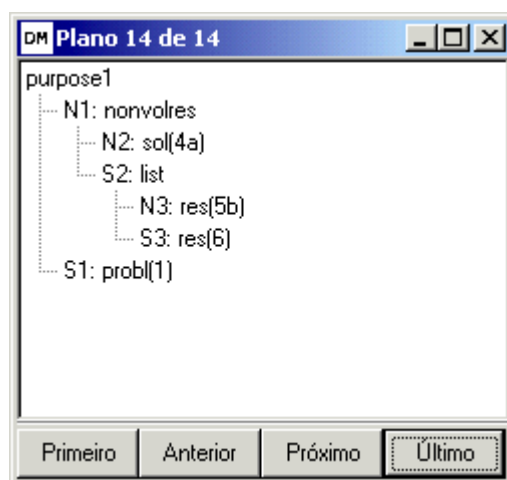


Figura 13 – Interface de exibição dos planos de texto

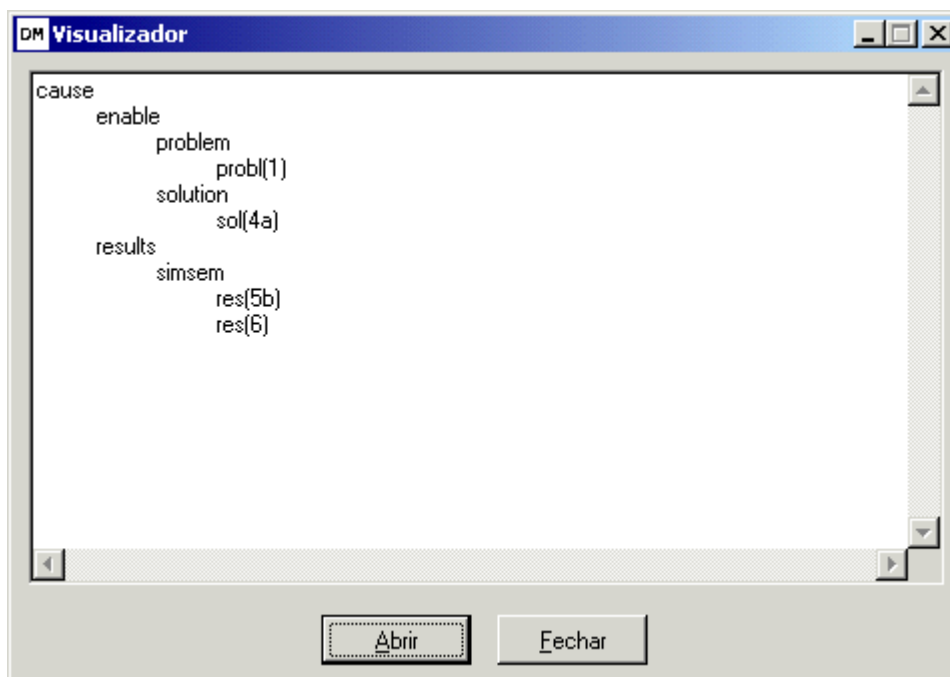


Figura 14 – Interface de exibição da BC

Na Figura 13, pode-se notar que os núcleos e satélites são indicados pelas letras N e S, respectivamente, seguidas de identificadores únicos. Os botões na parte inferior da interface permitem a navegação entre os planos. Clicando-se com o botão direito na interface, abre-se um menu *pop-up* que permite ao usuário abrir um outro arquivo de planos ou fechar a interface.

Durante o processamento de cada passo da sumarização, o botão *executar* é substituído por um *timer* que indica quanto tempo de processamento transcorreu até aquele momento desde a seleção do botão *executar*, sendo este *timer* atualizado a cada ciclo do sistema. Além de marcar o tempo de processamento, o *timer* também indica que o sistema ainda está executando sem problemas aparentes.

4. Conclusões

Este relatório apresentou a implementação do DMSumm, um sumarizador automático de textos com base no modelo de discurso de Rino (1996). O DMSumm, é composto dos processos da geração automática de textos, isto é, a seleção de conteúdo, o planejamento textual e a realização lingüística. Sua limitação se encontra justamente neste ponto: sendo um gerador, a interpretação ainda é feita de forma manual, exigindo que o usuário do sistema seja, portanto, um usuário especialista nos modelos lingüísticos utilizados.

A avaliação do DMSumm, relatada em Pardo (2002a, 2002b), mostrou que o sistema é promissor. Entretanto, muito pode ser feito ainda, destacando-se:

- o aprimoramento do processo de realização lingüística, que, por ser feito com base em *templates*, é muito limitado;
- o acoplamento de um modelo de usuário ao sistema, permitindo a geração de sumários mais dedicados aos interesses do usuário;
- a variedade de línguas naturais abrangidas pelo DMSumm pode ser aumentada, característica esta desejável no ambiente multilingual atual;
- pode-se expandir o modelo discursivo do DMSumm, incorporando-se outros objetivos comunicativos.

Referências Bibliográficas

- Baxendale, P.B. (1958). *Machine-made index for technical literature – an experiment*. In: IBM Journal of Research and Development, Vol. 2, pp. 354-365.
- Black, W.J. and Johnson, F.C. (1988). A Practical Evaluation of Two Rule-Based Automatic Abstraction Techniques. In: *Expert Systems for Information Management*, Vol. 1, No. 3. Department of Computation. University of Manchester Institute of Science and Technology.
- Cawsey, A. (1993). Planning Interactive Explanations. In: *Int. Journal of Man-Machine Studies*, Vol. 38, pp. 169-199.
- Feltrim, V.D.; Nunes, M.G.V.; Aluísio, S.M. (2001). *Um corpus de textos científicos em Português para a análise da Estrutura Esquemática*. Série de Relatórios do NILC. NILC-TR-01-4.
- Grosz, B. and Sidner, C. (1986). Attention, Intentions, and the Structure of Discourse. In: *Computational Linguistics*, Vol. 12, No. 3.
- Hovy, E. (1988). *Generating Natural Language under Pragmatic Constraints*. Lawrence Erlbaum Associates Publishers, Hillsdale, New Jersey.
- Jordan, M.P. (1980). Short Texts to Explain Problem-Solution Structures – and Vice Versa. In: *Instructional Science*, Vol. 9, pp. 221-252.
- Jordan, M.P. (1984). Structure, style and word choice in everyday English texts. In: *TESL* 15, Vols. 1 & 2.
- Jordan, M. P. (1992). An Integrated Three-Pronged Analysis of a Fund-Raising Letter. In W. C. Mann and S. A. Thompson (eds), *Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text*, pp. 171-226.
- Knight, K. and Marcu, D. (2000). Statistics-Based Summarization – Step One: Sentence Compression. In: *The 17th National Conference of the American Association for Artificial Intelligence*, Austin, Texas.
- Larocca Neto, J., Santos, A.D., Kaestner, A.A., Freitas, A.A. (2000). Generating Text Summaries through the Relative Importance of Topics. In: *Proceedings of the International Joint Conference IBERAMIA/SBIA*, Atibaia, SP.
- Mani, I. (2001). *Automatic Summarization*. John Benjamins Publishing Co., Amsterdam.
- Mann, W.C. and Thompson, S.A. (1987). *Rhetorical Structure Theory: A Theory of Text Organization*. Technical Report ISI/RS-87-190.
- Marcu, D. (1998a). Improving Summarization through Rhetorical Parsing Tuning. In: *The Sixth Workshop on Very Large Corpora*, pp. 206-215. Montreal, Canada.
- Marcu, D. (1998b). To build text summaries of high quality, nuclearity is not sufficient. In: *The Working Notes of the AAAI-98 Spring Symposium on Intelligent Text Summarization*, pp. 1-8. Stanford, California.
- Martins, C.B.; Pardo, T.A.S.; Espina, A.P.; Rino, L.H.M. (2001). *Introdução à Sumarização Automática*. Relatório Técnico RT-DC 002/2001, Departamento de Computação, Universidade Federal de São Carlos.
- Maybury, M.T. (1992). Communicative Acts for Explanation Generation. In: *Int. Journal of Man-Machine Studies* 37, pp. 135-172.
- McKeown, K.R. (1985). *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.
- Moore, J.D. and Paris, C. (1993). Planning Text for Advisory Dialogues: Capturing Intentional and Rhetorical Information. In: *Computational Linguistics*, Vol. 19, No. 4, pp. 651-694.

- Mushakoji, S. (1993). Constructing 'Identity' and 'Differences' in Original Scientific Texts and Their Summaries: Its Problems and Solutions. In: *Proceedings of Summarizing Text for Intelligent Communication*, B. Endres-Niggemeyer, J. Hobbs and K. Sparck Jones (eds). Dagstuhl, Germany.
- O'Donnell, M. (1997). Variable-Length On-Line Document Generation. In: *Proceedings of the 6th European Workshop on Natural Language Generation*, Gerhard-Mercator University, Duisburg, Germany.
- Pardo, T.A.S e Rino, L.H.M. (2001a). *O Planejamento de Sumários a partir de Operadores de Plano*. Congresso de Pós-Graduação da Universidade Federal de São Carlos - CONGPG.
- Pardo, T.A.S. e Rino, L.H.M. (2001b). A Summary Planner Based on a Three-Level Discourse Model. In *Proceedings of the Natural Language Processing Pacific Rym Symposium - NLPRS*. Tokyo, Japan.
- Pardo, T.A.S. (2002a). *DMSumm: um Gerador Automático de Sumários*. Dissertação de Mestrado. Departamento de Computação, Universidade Federal de São Carlos. São Carlos – SP.
- Pardo, T.A.S. (2002b). *Avaliação do DMSumm*. Relatório Técnico NILC-TR-02-05, São Carlos.
- Rino, L.H.M. and Scott, D. (1994). *Automatic generation of draft summaries: heuristics for content selection*. ITRI Techn. Report ITRI-94-8. University of Brighton, England.
- Rino, L.H.M. (1996). *Modelagem de Discurso para o Tratamento da Concisão e Preservação da Idéia Central na Geração de Textos*. Tese de Doutorado. IFSC-Usp. São Carlos - SP.
- Sparck Jones, K. (1993). What might be in a summary? In: Knorz, Krause and Womser-Hacker (eds.). *Information Retrieval: Von der Modellierung zur Anwendung*, Vol. 93, pp. 9-26, Universitätsverlag Konstanz.
- Winter, E.O. (1976). *Fundamentals of Information Structure*. Hatfield Polytechnic, Hertfordshire, England.