# DadosSemiotica: coleta e processamento de análises semióticas de texto escrito*

**Ana Cristina Fricke Matte[1], Rubens Takiguti Ribeiro[1], William Daniel Colen de Moura Silva[2], Hugo Leonardo Canalli[1]**

[1]Laboratório SEMIOTEC e Grupo Texto Livre – Universidade Federal de Minas Gerais (UFMG)
31.270-970 – Belo Horizonte – MG – Brasil

[2]Departamento de Ciência da Computação – Intituto de Matemática e Estatística da Universidade de São Paulo (IME/USP)
055080-090 – São Paulo – SP – Brasil

{acris,rubs,hugleo}@textolivre.org, colen@ime.usp.br

***Abstract.*** *One of the advantages of free software is the possibility of have professionals from various fields participating in the development of softwares that will be used by them in their own research and projects. The dadossemiotica was originated from a specific need: to do semiotic research on a large scale in order to seek automated solutions for complex analysis and in order to obtain statistical significance with a large number of parameters and variables, as usual in the research analysis of text and discourse. It is a modular program in PHP/MySQL developed with the SIMP framework to gather and manage the data of textual analysis based on the semiotic conception of text .*

***Resumo.*** *Uma das vantagens do software livre é a possibilidade de profissionais das mais diversas áreas participarem do desenvolvimento dos softwares que serão por eles usados em suas pesquisas e projetos. O dadossemiotica nasceu de uma necessidade específica: fazer pesquisa semiótica em larga escala para buscar soluções automáticas e para obter significância estatística em análises complexas com grande número de parâmetros e variáveis, como geralmente são as pesquisas de análise do texto e do discurso. Trata-se de um programa modular em PHP/MySQL desenvolvido no framework SIMP para coleta e gerenciamento de dados de análise textual baseado na concepção semiótica de texto.*

## 1. Introdução

O software livre dadossemiotica foi desenhado tendo em vista agilizar o processo de análise textual pelo semioticista e fornecer um apoio a outros analistas de linguagem verbal. Trata-se de um aplicativo web que permite acelerar a coleta manual de dados sobre textos e sua organização e recuperação para análises estatísticas ou outras. O lançamento da primeira versão estável está previsto para julho de 2012.

Criado dentro da perspectiva teórica da semiótica de linha francesa, o dadossemiotica tem como unidade de sentido o texto [Matte 2012] e como elementos

* Disponível em http://www.sourceforge.net/projects/dadossemiotica Licença GPL v.2. **Apoio FAPEMIG processo PPM-00206-10.**

constitutivos a frase. A análise é feita por categorias: a categoria analítica deve ser clara o bastante para que diferentes analistas, ao trabalhar seus textos na mesma instalação do sistema, auxiliem na criação de um banco de análises semióticas e linguísticas que possa ser utilizado como referência para cruzamento com outras categorias analíticas.

O dadossemiotica trabalha com dois tipos de análises: as análises manuais e as análises automáticas, feitas por módulos especiais. Esta modularidade traduz a escalabilidade do programa e a possibilidade de adequá-lo a diferentes áreas de pesquisa. Para realizar estes tipos de análise, o software separa três grupos de usuários: o analista, que é o principal ator das análises manuais, o gerente, que organiza o material e as perspectivas de análise, acionando os módulos de preprocessamento (análises automáticas), e o administrador do sistema.

Ao separar a função do gerente e do analista, o programa permite personalizar o banco de dados com categorias de interesse de um determinado grupo de pesquisadores, cujas pesquisas normalmente seriam realizadas individualmente e sem qualquer tipo de cooperação ou possibilidade de cruzamento de dados. O gerente define as categorias e insere os textos de análise. Cada analista poderá realizar sua pesquisa individualmente mas, ao seguir as instruções na utilização de cada categoria, alimenta um banco de dados cujos resultados cruzados poderão ser obtidos pelo gerente para utilização pela comunidade. Por outro lado, a instalação também pode, caso se deseje, ser interdisciplinar, congregando análises de diversos campos do conhecimento sem que isso gere confusões indesejadas, pois os projetos restringem as categorias àquelas que interessam a cada analista.

Assim, o dadossemiotica busca na cultura das comunidades de software livre dois de seus conceitos fundamentais: o compartilhamento de informação como forma de crescimento da comunidade, no caso pela possibilidade de colaboração entre pesquisas e criação de bancos de dados de análise abertos, e o código aberto, como forma de incorporar no processo de desenvolvimento do software a própria comunidade ao qual se destina.

Neste artigo buscamos descrever o programa, seu funcionamento, o framework SIMP no qual foi desenvolvido e os módulos de preprocessamento morfológico e de chat, a fim de indicar as possibilidades de uso do programa para engenharia reversa e produção de conhecimento computacionalmente válido. No primeiro tópico apresentamos o programa do ponto de vista de seu principal usuário: o analista.

## 2. A interface gráfica do analista – exemplo de linguística

Neste tópico apresentamos um caso de uso para exemplificar e apresentar a interface principal do programa: a interface do analista. O dadossemiotica não se restringe, como mencionado, a análises semióticas. Como um exemplo de análise semiótica exigiria um preâmbulo teórico que não cabe no escopo do presente artigo, vamos usar um exemplo de análise sintática, especificamente identificação de vocativos.

Para este exemplo, o gerente inseriu textos que são logs (registros) do programa Konversation. A rede acessada é a Freenode[1], com amostras do canal de suporte a software livre #aco. Os logs de chat apresentam cada “fala” separada por quebra de linha e, no caso exclusivo do chat, é essa a definição de “frase”, o que também resolve para uma delimitação automática de sentenças o problema da falta de pontuação,

1 Freenode: http://www.freenode.net

extremamente comum em amostras de chat.

As categorias cadastradas pelo gerente para este exemplo foram:

- Vocativo-texto: trata-se do texto correspondente ao vocativo, excluídos os espaços e a pontuação que o precede ou sucede; pode ser diretamente selecionado e arrastado até a caixa de texto correspondente;
- Vocativo: trata-se de verificar o tipo de vocativo e sua posição na frase – geral (g), nick ativo no chat (n) ou nome próprio ou nick não ativo (p) – e verifica se o vocativo for precedido ou sucedido por texto (marcado por um traço no código).

Se houver mais de um vocativo, repete-se a letra do campo *vocativo* e os textos selecionados para o campo *vocativo-texto*, são separados por [espaço traço espaço].

A Ilustração 1 mostra a tela inicial para o analista.


**Ilustração 1: Tela de recepção para o analista.**

Cada projeto é definido como um conjunto de textos a serem analisados para um conjunto de categorias de análise. Ao criar o projeto, o analista já deve ter essas categorias e textos definidos e inseridos no sistema pelo gerente.

Para inserir os textos – ou depois visualizar o projeto – basta escolher na lista de projetos. O ícone do livro serve para abrir o projeto, seu nome pode ser editado no ícone de escrita (Ilustração 2). A escolha das categorias e dos textos é feita nesse momento. Nosso exemplo terá dois textos e as duas categorias relativas aos vocativos.


**Ilustração 2: Lista de projetos: é possível, a qualquer momento, editar o nome do projeto e incluir/excluir textos e categorias.**

Ao iniciar cada lote de análises, o analista pode escolher até 4 categorias para

analisar simultaneamente. A possibilidade de fazer essa escolha permite uma visualização mais adequada a cada tipo de análise a ser feito. No caso do nosso exemplo, analisar o vocativo ao mesmo tempo em que seleciona o próprio texto do vocativo agiliza a coleta de dados; note que na tela de escolha somente aparecem as categorias que foram cadastradas para este projeto.

A tela de análise apresenta todas as frases do texto dispostas numa coluna e seguida de tantos campos para análise quantos forem escolhidos pelo analista para visualização (Ilustração 3).

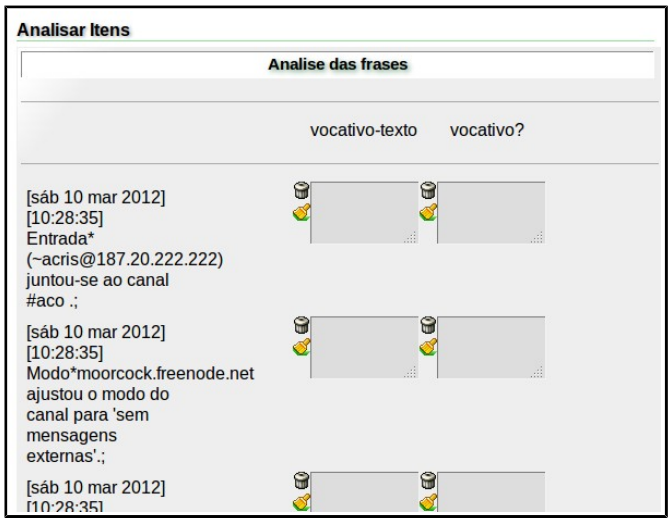


**Ilustração 3: Tela de análise. No exemplo, cada notificação do sistema (do IRC) é uma frase.**

Durante a análise, é possível escolher copiar a última análise digitada, clicando no pincel. O procedimento copia a última análise realizada em todos os campos vazios até o campo escolhido (onde foi feito o clique) e somente numa mesma categoria de análise. É possível selecionar texto e arrastar para o campo de análise ou digitar diretamente a análise desejada[2].

Esta é a parte manual da análise. É importante que seja manual, pois trata-se de coletar dados de análises de texto que não são possíveis, pelo menos até o momento, de realizar automaticamente. No exemplo, a análise destes dois campos em um *corpus* suficientemente grande permitirá, num segundo momento e com um simples parser acoplado ao banco de dados do dadossemiotica, inserir automaticamente todas as tags necessárias em um corpus para treinamento efetivo do CoGrOO[3] para identificação de vocativos. Após esse treinamento, esta análise poderá ser feita automaticamente, pelo módulo de preprocessamento morfológico, e a categoria torna-se desnecessária. No entanto, ainda é possível utilizar seus resultados para comparações de performances, já que todos os resultados de pré-processamento são guardados em tabelas separadas das análises manuais.

A organização dos módulos – da análise manual aos módulos de preprocessamento automático – está intimamente ligada ao framework SIMP, no qual o dadossemiotica foi desenvolvido.

---

2 Outras ferramentas estão sendo estudadas para agilizar a coleta de dados, optamos por apresentar somente as ferramentas já implementadas.

3 Corretor gramatical CoGrOO: http://cogroo.sourceforge.net

## 3. Framework SIMP

O Simp[4] é um framework escrito em linguagem PHP e cujo principal objetivo é prover facilidades e padrões para o desenvolvimento de sistemas de informação modulares em plataforma web. Sua estrutura foi preparada para a criação de sistemas com muitas entidades (tabelas de bancos de dados) e relacionamentos entre elas, que exigem muitas ferramentas básicas de cadastro. Para suprir este objetivo, utiliza alguns padrões consagrados da engenharia de software, que foram adequadamente adaptados para o contexto da web para equilibrar agilidade no desenvolvimento e escalabilidade.

O framework aborda o desenvolvimento ágil em duas frentes principais: (i) a criação de ferramentas básicas de interface com o usuário final e (ii) a manipulação de entidades (estruturas de dados que representam elementos do problema abordado). No primeiro caso, oferece lógicas de negócio genéricas para criação de ferramentas de inserção, alteração, remoção, importação, exibição e listagem de entidades, com pontos de extensão preparados para prover um certo nível de especificidade, quando necessário. No segundo caso, oferece uma arquitetura própria de ORM (Object Relational Mapping), que provê Active Record [Fowler et al 2002], e conjunto de métodos para consulta e persistência de dados de entidades, que tornam a criação de entidades do sistema uma tarefa simples e a manipulação de entidades padronizada e ágil.

O Simp contém um princípio de aplicação, além de padrões e código-fonte genérico. Este pré-sistema possui padrões de interface, algumas entidades básicas e algumas ferramentas que manipulam estas entidades, que garantem o funcionamento do sistema. As principais entidades básicas do framework são: usuários, grupos de usuários, ferramentas, permissões de acesso às ferramentas e configurações básicas da aplicação.

### 3.1. Principais Características

O desenvolvimento no framework Simp se dá com a definição de entidades, que especifica as características das tabelas usadas no BD, semelhante à definição de anotações em frameworks ORM, e com a criação de módulos que possuem um conjunto de ferramentas relacionadas que manipulam tais entidades.

A definição de entidades é feita de forma independente de Banco de Dados, oferecendo transparência na criação de instruções em SQL portáveis, tanto para criação da estrutura do Banco de Dados (instruções DDL) quanto para realização de operações básicas de manipulação de dados (DML). Por padrão, o framework suporta os SGBDs MySQL e PostgreSQL e permite a criação de drivers para outros. Neste aspecto, o framework é contrário à dependência de um único BD, como é defendido em [Brampton 2008].

Com a definição de características das entidades, provê algumas validações automáticas, limitando tamanhos de campos, tipos de dados, expressões regulares, além de permitir a inclusão de validações específicas. Além disso, usa as definições para criação dos campos de formulário de forma automática, embora customizável.

Do ponto de vista de criação dos módulos e ferramentas do sistema, o framework possui classes para criação de lógicas básicas de uma entidade com apenas

4 Framework SIMP: http://sourceforge.net/projects/simpframework/, desenvolvido na UFLA e utilizado pela TecnoLivre (http://www.tecnolivre.com.br) desde o ano de 2006.

uma linha de código cada (listagem das entidades, exibição/inserção/alteração/remoção de uma entidade e importação de entidades na forma de CSV ou XML), embora de forma extensível, podendo-se especificar, por exemplo, quais campos devem aparecer nos formulários ou nos quadros de exibição de dados de um registro.

Sua arquitetura ORM provê a abstração das relações 1:1 (um para um) e 1:N (um para muitos) entre as entidades, permitindo, por exemplo, gerar formulários com campos de entidades diferentes, desde que relacionadas, ou realizar consultas envolvendo várias entidades abstraindo os relacionamentos. Este recurso permite realizar consultas sofisticadas abstraindo detalhes referentes à linguagem SQL.

Sob uma visão geral, e para garantir as facilidades citadas, o framework é formado por:

- Algumas classes estruturais, como é o caso da classe objeto, que é uma classe abstrata utilizada como base para todas as entidades do sistema;
- Algumas classes de utilização geral (classes de suporte e de interface);
- Algumas entidades básicas que estendem direta ou indiretamente a classe objeto;
- Módulos básicos, que são considerados cruciais para qualquer sistema de informação Web convencional;
- Alguns temas baseados em folhas de estilos em CSS;
- Alguns scripts em JavaScript de utilização geral que garantem o funcionamento de algumas funcionalidades baseadas em Ajax.

As características apresentadas concernem às especificidades do dadossemiotica; a documentação completa do framework encontra-se no sítio web <http://sourceforge.net/p/simpframework/code-0/2/tree/trunk/doc/>.

### 3.2. Utilização do Simp no dadossemiotica

A aplicação dadossemiotica possui as seguintes entidades:

- Projeto – Associada com textos e categorias de análise, restringindo a análise somente aos textos e categorias de análise selecionados no projeto.
- Texto – Conjunto de frases relacionadas.
- Frase – Unidade de um texto, que recebe análises.
- Categoria de análise – Conjunto de valores cadastrados pelo gerente.
- Análise Manual – Possui uma unidade de texto e associações que permitem a uma análise participar de um projeto, categoria de análise e frase.
- Análise Chat – Composta de uma unidade de texto que registra uma análise de preprocessamento de chat, uma associação com uma frase e um apelido de IRC.
- Identidade de Chat – Associada a uma análise de chat, possui unidades de textos para registrar os apelidos de IRC, identificação do estado e histórico de um apelido de IRC.
- Análise Cogroo – Composta de uma unidade texto, que armazena o XML obtido na análise de preprocessamento morfossintático e uma associação com o texto analisado.

Os analistas, gerentes e administradores são controlados pelas entidades básicas do SIMP que definem permissões para os usuários em termos dos arquivos acessados. São definidas as seguintes permissões principais:

- Analista – Visualizar e selecionar os textos e as categorias de análises para uso no projeto, criar e deletar projetos, visualizar e editar a tela de análise de frases.
- Gerente – Inserir e remover textos, criar e remover categorias de análises, cadastrar e remover um analista, executar análises de preprocessamento, visualizar os projetos e análises de todos os analistas.
- Administrador – Cadastros das páginas e usuários do sistema, cadastro de permissões para todos os níveis de usuários. Remover usuários.

## 4. Módulo de preprocessamento morfossintático

O módulo de preprocessamento morfossintático, composto por um servidor de análise e um aplicativo de integração, atua sobre todo e qualquer texto inserido no sistema pelo gerente, de forma automática e em background. A unidade de trabalho do dadossemiotica é o texto produzido de forma natural, que do ponto de vista computacional constitui um tipo desestruturado de dado [Feldman 2007]. A característica desestruturada dos textos naturais inviabiliza que seu conteúdo semântico se enquadre diretamente em modelos tradicionais de dados, como por exemplo bancos de dados relacionais, ou que sejam facilmente interpretados por programas tradicionais.

O papel do módulo de preprocessamento morfossintático no dadossemiotica é automaticamente inferir estruturas a partir dos textos coletados pelo sistema. Tais estruturas, então apresentando um modelo de dados computacionalmente tratável, podem ser indexadas e armazenadas em um banco de dados, viabilizando consultas como “encontrar todos as ocorrências do verbo *haver* em uma frase que tenha vocativo”, por exemplo.

Tendo como base os anotadores desenvolvidos para o corretor gramatical software livre CoGrOO, desenvolvido pelo Centro de Competência em Software Livre (CCSL) para a suíte de escritório LibreOffice[5], o módulo é capaz de segmentar um documento em sentenças, tokens (palavras e símbolos) e sintagmas, assim como classificar, dado o contexto, as palavras de acordo com sua classe gramatical, identificar seu lema e ainda suas características como gênero, número, pessoa e tempo [Kinoshita 2007].

Esta análise envolve resolução de ambiguidades em diversos níveis. Por exemplo nos níveis de segmentação de sentenças e palavras, o software deve saber quando separar ou não uma palavra do símbolo seguinte, como na frase “O Sr. José entregou-me os documentos.”, onde “Sr.”, por se tratar de uma abreviatura constitui uma unidade e deve ser mantida junto com o ponto, formando um token. Por outro lado a sequencia “documentos.” deve ser dividida em dois tokens: a palavra “documentos” e o símbolo delimitador de sentença. Já o clítico “-me” deve ser separado da palavra “entregou”, formando dois tokens.

Na sentença “Quem casa quer casa.” existe ambiguidade morfológica entre as ocorrências da palavra “casa” que só pode ser resolvida efetuando a análise de contexto.

5 CCSL: http://ccsl.ime.usp.br, LibreOffice: http://pt-br.libreoffice.org

O sistema deve anotar a primeira ocorrência como verbo e a segunda como substantivo.

Para o dadossemiotica, além dos anotadores já presentes no CoGrOO, será desenvolvido um anotador capaz de identificar elementos da oração, como por exemplo vocativos. A Ilustração 4 mostra detalhadamente o módulo de processamento morfossintático, que internamente segue o padrão de projeto dutos e filtros, em que cada analisador desempenha uma tarefa específica, e a saída de um é a entrada do seguinte. O documento proveniente do dadossemiotica é submetido a esta sequência de análises, que, no final do processo, fornece anotações indicando os limites das sentenças, dos tokens, das entidades, dos sintagmas e dos termos da oração, assim como etiquetas classificando cada termo, por exemplo, etiquetas que definem a classe gramatical de cada token ou o tipo de sintagma. Tais anotações são então armazenadas em um banco de dados para futura referência.

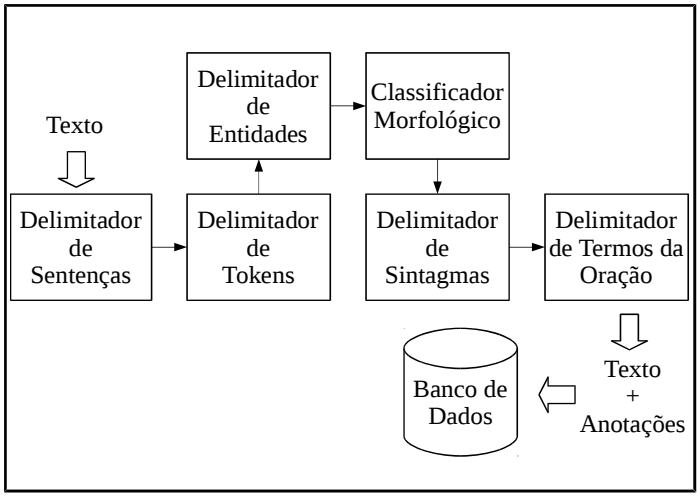


**Ilustração 4: Analisadores do módulo de processamento morfossintático, organizados seguindo o padrão dutos e filtros.**

Os anotadores do CoGrOO são desenvolvidos em Java, e fazem uso do software livre Apache OpenNLP[6], um framework livre para o processamento de linguagens naturais. Todos os anotadores são desenvolvidos com técnicas estatísticas de aprendizado de máquina, como Máxima Entropia [Filho 2002] e Perceptron HMM [Collins 2002].

### 4.1. Integração com o dadossemiotica

Para a integração entre o módulo de preprocessamento morfossintático do dadossemiotica, o primeiro desenvolvido em Java e o segundo em PHP, foi utilizado o Apache UIMA[7], um framework desenvolvido especificamente para a criação de ferramentas que lidam com dados desestruturados. Conforme descrito em [Silva 2010], os analisadores do CoGrOO podem ser usados como extensões do Apache UIMA. Um exemplo de análise pode ser vista na Ilustração 5.

6 Apache OpenNLP: http://opennlp.apache.org
7 Apache UIMA: http://uima.apache.org

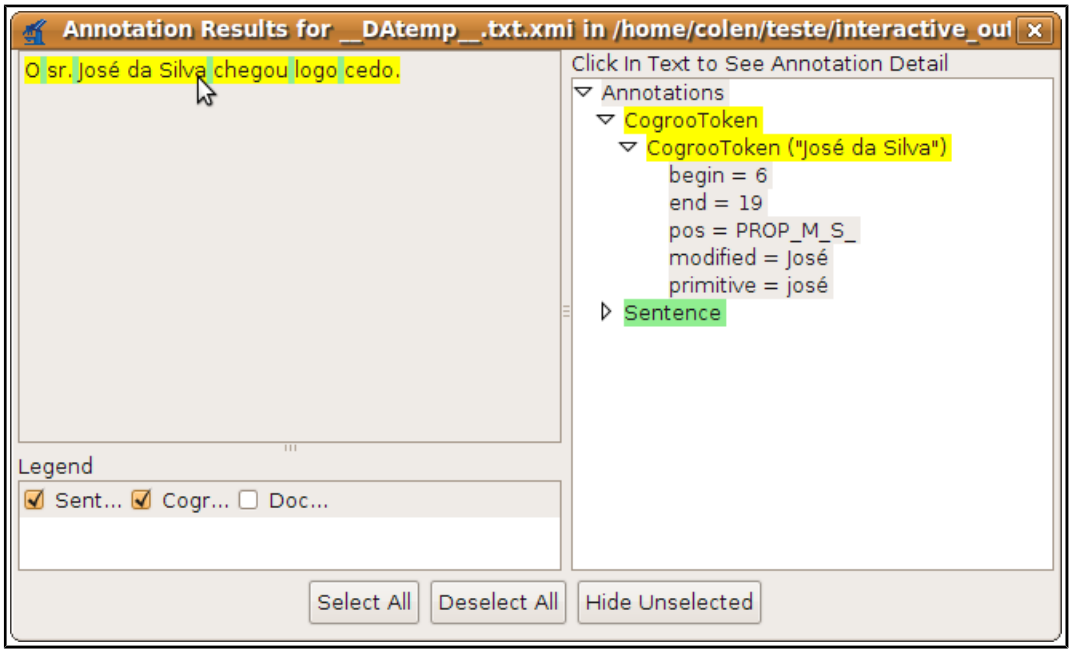


**Ilustração 5: Visualização do texto anotado no Apache UIMA Annotation Viewer**

Fazendo uso destas ferramentas, para o dadossemiotica foi criado um servidor de anotações UIMA seguindo o esquema descrito na Ilustração 4 e instalado em um servidor web.

A comunicação entre o dadossemiotica e o servidor de anotações se faz por requisições HTTP POST em que o sistema envia ao módulo o texto a ser analisado e recebe um XML contendo as anotações como resposta. A solução em teste para o gerenciamento desses resultados é a utilização de Apache Lucene + Solr[8].

## 5. Módulo de preprocessamento de chat

A análise de chats envolve uma perspectiva interessante para a semiótica. Os registros dos chats, embora sejam textos acabados, são uma espécie de fotografia de uma conversa em curso, com as idas e vindas comuns a conversas coloquiais e um certo "descuido" com a forma padrão da língua. Assim, a análise dos chats pode trazer importantes luzes não só sobre a comunicação na internet, mas também para estudos da construção do sentido como processo.

Os participantes de um chat tem algumas pistas sobre identidade e foco das interações. O módulo de preprocessamento de chat proposto para o dadossemiotica busca automatizar o registro dessas informações para possibilitar o cruzamento destas análises técnicas com as análises semióticas ou linguísticas propriamente ditas. Trata-se de um módulo interno, em PHP/MySQL, a ser chamado pelo gerente. A primeira versão deste módulo, que é apresentada aqui, restringiu a análise a logs gravados pelo programa Konversation[9].

Há basicamente dois tipos de entradas no chat: as notificações do sistema e o fluxo de conversa. As notificações de entrada e saída de um nick no chat e as notificações sobre troca de nick são importantes para a constituição do que chamamos

8 Lucene: http://lucene.apache.org/solr/
9 Konversation: um cliente gráfico de IRC (Internet Relay Chat). http://extragear.kde.org/apps/konversation/

de identidades.

O nick é a forma pela qual um usuário se apresenta no chat do IRC: não há fotos ou qualquer outro tipo de imagem além daquelas feitas com caracteres ASCII, e também não é possível comunicar-se oralmente. A língua escrita, portanto, é o principal código utilizado neste tipo de chat – só não é o único em virtude do uso de *emoticons* e outros desenhos feitos com caracteres ASCII.

O módulo de preprocessamento de chat é um parser opcional que processa cada linha como uma frase independente e, por esse motivo, ao ser acionado pelo gerente, vai rodar antes do módulo de preprocessamento morfossintático e entrará neste diretamente pelo Delimitador de Tokens (ilustração 4). Da frase:

1. guarda a *data* numa tabela específica do chat. Essa informação não aparece para os usuários do chat e portanto não é necessária para o analista, mas pode ser útil para o histórico do chat e está presente em todas as linhas de notificação e conversa nos logs do Konversation.
2. guarda a *frase* (sem a data) na tabela de frases do dadossemiotica. A frase no chat não é concebida como no texto padrão: uma frase será uma "unidade de fala" no chat: uma linha da conversação ou uma linha de notificação. Pode, portanto, corresponder a uma parte ou mesmo um conjunto daquilo que chamamos de frase no texto padrão – o texto delimitado por pontuação.
3. guarda as *palavras* da frase numa tabela específica do chat.
4. guarda o *nick ativo* (autor da frase na conversa ou nick do qual trata a notificação em questão) na tabela do chat.
5. processa e guarda o *estado do nick*: online se está falando ou se é notificação de entrada; offline se é notificação de saída. Faz também a troca de online para offline e de offline para online em caso de troca de um nick para outro.
6. guarda o *histórico do nick*: uma informação por frase. No caso de notificação de entrada é a última informação da frase e vem entre parênteses. Trata-se de informação sobre a identidade da conexão, podendo trazer o IP, nick ao qual está vinculado, programa usado, dentre outras coisas. No caso de notificação de saída também entre parênteses, no final da frase, temos informações menos precisas sobre o motivo da saída. No caso de troca de apelido, marca uma possível identidade entre nicks diferentes (pode ser eventual ou recorrente). As conversas são marcadas como *fala*.

Esse módulo de preprocessamento guarda os dados na tabela *analiseschat* e poderá, inclusive, ser rodado novamente em background em todos os *corpora* marcados como chat, se houver modificações substanciais em sua estrutura. A importância desta característica é permitir a utilização do dadossemiotica como ferramenta de observação de suporte de software livre em chats do IRC, tendo em vista melhorar a performance de chatterbots com base nos resultados obtidos.

## 6. Recuperação de dados

O dadossemiotica possui duas formas de recuperar os dados, com diferentes permissões

para o grupo de gerentes e de analista. É possível recuperar tabelas ou alguns tipos de análise estatística, como gráficos de frequência e ANOVA. As tabelas são obtidas na forma de arquivos csv, prontas para sua utilização no programa R[10], que é o mesmo utilizado para obtenção dessas análises que o dadossemiotica pode prover.

O analista só pode realizar coleta de resultados de seus próprios projetos, enquanto o gerente pode obter resultados parciais ou totais de todas as análises registradas no sistema. Garante-se, dessa forma, a privacidade dos dados, muitas vezes sujeitos às normas do Conselho de Ética em Pesquisa[11], sem excluir a possibilidade de análise geral pelo gerente, previamente consentida pelos analistas.

## 7. À guisa de conclusão

Uma das funções possíveis para o dadossemiotica é a engenharia reversa. Por exemplo, a partir das análises em grande escala de conversas no chat, é possível gerar novas regras de identidade entre os nicks. Por exemplo, podemos descobrir qual a probabilidade de um nick ser usado por um visitante casual. É bastante frequente que as pessoas entrem no chat com um nome próprio simples, especialmente aquelas que não utilizam este ambiente com frequência. Nessa situação, o nick utilizado pode mesmo ter sido registrado anteriormente. O próprio IRC pode alterar o nick para Guestxxx (onde xxx é um número qualquer) ou o dono do nick pode, com um comando, recuperar essa identidade. O estudo dessas situações, dentre outras, pode trazer luzes sobre a questão da identidade no chat que, combinada com análises de conteúdo, podem ser extremamente reveladoras sobre a comunicação nesse ambiente.

Um bot que utilize esse conjunto de regras para identificar seu interlocutor pode obter um resultado mais robusto em termos de identificação de threads, estilo do interlocutor e até interesses. Com o uso de adaptatividade [Neto 2000], por exemplo, é possível gerar um script diferente para tratamento de nicks estáveis (com alta frequência e registrados) e de nicks eventuais. Fundamentando a performance do chatterbot no tratamento e verificação de hipóteses sobre o tema do diálogo no lugar de simples relações diretas entre estímulos – da fala do usuário – e respostas – do chatterbot – é possível prever uma melhoria substancial na qualidade dessa interação, com inúmeras aplicações.

O dadossemiotica pode, nesse caso, funcionar como posto de observação de performance de um bot de IRC, com o uso do módulo de preprocessamento de chat, ao mesmo tempo alimentando e adequando tanto o bot quanto o próprio sistema de análise.

O contexto de produção e uso baseado na cultura do software livre e na dinâmica de suas comunidades faz parte da ideia central do programa dadossemiotica: nessa conjuntura social, o dadossemiotica apresenta-se como uma via de mão dupla: da necessidade e dos parâmetros do pesquisador vem a programação, que permite a agilização e padronização das análises a qual, por sua vez, permite automatizar novas funcionalidades analíticas que agreguem novas possibilidades de abordagem dos dados e assim por diante, quase em moto contínuo, dependendo apenas da necessidade e motivação das comunidades envolvidas.

10 The Comprehensive R Archive Network: http://cran.r-project.org/
11 CONEP: http://conselho.saude.gov.br/Web_comissoes/conep/index.html

## Referências


Brampton M., PHP5 CMS Framework Development, 2008

Collins, M., Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms, 2002

Feldman, R. e Sanger, J, The text mining handbook: advanced approaches in analyzing unstructured data, 2007

Filho, A.A.A, Maximização de Entropia em Linguística Computacional para a Língua Portuguesa, 2002

Martin F., Rice D., Foemmel M., Hieatt E., Mee R., Stafford R., Patterns of Enterprise Application Architecture, 2002

Kinoshita, J., Salvador, L.N., Menezes, C.E.D., e Silva, W.D.C.M., CoGrOO - An OpenOffice Grammar Checker, 2007

Matte, A. C. F. (2012), “Epissemiótica: contorno, entorno e turno”, Revista Texto Livre: Linguagem e Tecnologia, v. 5, n. 3, No Prelo.

Neto, João José, Solving complex problems with adaptative automata, Url: <http://lta.poli.usp.br/lta/publicacoes/artigos/2000/neto-jj-00/at_download/file> Acessado em 19 de abril de 2012, 2000

Silva, W. D. C. M., Finger, M., e Menezes, C. E. D., Open Text Annotators Using Apache UIMA, 2010