## Resumo

Este paper documenta os esforços empregados visando o desenvolvimento de uma solução de software capaz de auxiliar na extração de dados do Diário Oficial da União (D.O.U.) e na criação de registros de associação entre órgãos, pessoas, publicações e termos para posterior extração de redes sociais.

## Introdução

A mineração de dados públicos abertos para extração de redes sociais que já foi desenvolvida por alguns autores. Alguns trabalhos lançaram mão de matérias veiculadas na imprensa enquanto outros mineraram documentos de processos jurídicos. (Krebs 2002, Rodriguez 2004, Sageman 2007) . Outros trabalhos usaram o D.O.U. como fonte porem de forma não automatizada. Parece, entretanto, não existir iniciativa de se processar massivamente o D.O.U. com o objetivo de se processar um volume consideravelmente grande de dados e, de forma sistemática, extrair informações de relacionamentos para posterior extração de redes sociais. Esse paper documenta como foi desenvolvida a solução de software capaz de operacionalizar essa extração e detecção de relacionamentos. Essa solução contribuirá para a base empírica de alguns projetos sobre Segurança e Defesa Cibernética em desenvolvimento no grupo de pesquisa em Segurança da Informação do Programa de Pós Graduação em Ciência da Informação da Universidade de Brasília.

## Visão geral da solução

A solução proposta teve os seguintes requisitos funcionais:

* + - Identificar, dentro das paginas do D.O.U. , os limites entre as publicações.
    - Identificar entidades (órgãos e pessoas) dentro das publicações.
    - Registrar relacionamentos entre entidades e publicações armazenando a data de sua ocorrência.

Assim, a tarefa foi dividida em três grandes atividades: Download das páginas do DOU, Identificação das entidades e publicações e identificação e registro dos relacionamentos. A Figura 1 mostra a interação entre os componentes da solução.

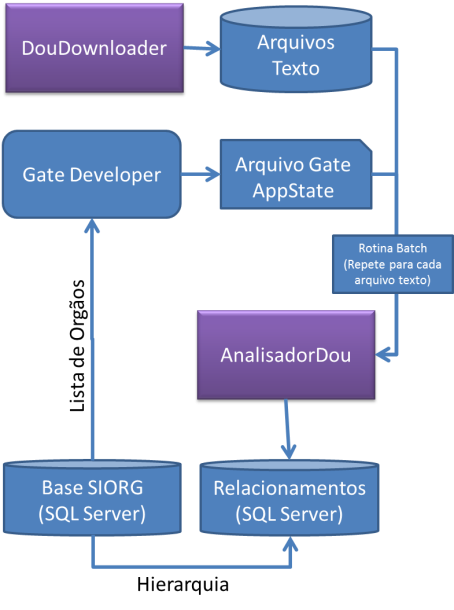


Figura 1 – Visao geral da solução.

## Download das páginas do DOU

Um componente de software, batizado DouDownloader, foi desenvolvido especialmente para buscar as páginas das edições do DOU e gravá-las em formato texto. Embora trate-se de uma tarefa aparentemente simples, há aqui uma grande dificuldade em lidar com dois pontos principais: O formato PDF e a diagramação do DOU.

O formato pdf (portable document format) é um padrão para armazenamento de informações cujo funcionamento é pouco intuitivo. De 1993 a 2011 foram publicadas 11 versoes do padrão, que apesar de ter se tornado aberto em 2008, e contar com uma política de uso livre (royalty free) por parte de seu proprietário (ADOBE 2013), seu funcionamento baseado em objetos complexos, operadores, tokens e posicionamento absoluto de elementos faz com que sua transformação em texto seja algo complexo.

Assim, optou-se por fazer uso de uma biblioteca de terceiros chamada iText. O iText é uma biblioteca que permite a manipulação de arquivos PDF usando java (ITEXT 2013). Ela possui classes que são capazes de extrair texto puro, ignorando imagens, gráficos e tabelas, de uma determinada região (coordenadas x,y) do documento PDF. É a biblioteca para esse fim mais indicada no stack overflow, principal portal de colaboração entre desenvolvedores de software.

A diagramação do DOU foi tratada usando a funcionalidade de extração de texto por região do iText. Para tanto basta implementar uma nova estratégia de extração de texto e usá-la no construtor da classe FilteredTextRenderListener, responsável por extrair as paginas internamente na biblioteca. A Listagem 1 exibe o trecho de código customizado utilizado, o qual o leitor mais familiarizado com a linguagem java não terá dificuldades em assimilar.

**public** **static** String parsePdf(String pdf, Rectangle rect) **throws** IOException

{

//pdf = url do portal onde está a página do DOU a ser processada

PdfReader reader = **new** PdfReader(pdf);

//Rectangle com a região(coluna) de onde o texto será extraido. C

RenderFilter filter = **new** RegionTextRenderFilter(rect);

TextExtractionStrategy strategy;

StringBuilder sb = **new** StringBuilder();

CustomLocationTextExtractionStrategy txstr = **new** CustomLocationTextExtractionStrategy();

strategy = **new** FilteredTextRenderListener(txstr, filter);

String texto = PdfTextExtractor.*getTextFromPage*(reader, 0, strategy);

//0 pois cada url tras apenas 1 pag.

sb.append(texto);

reader.close();

**return** sb.toString();

}

Listagem 1 – Código para download do pdf e transformação em texto.

O problema da diagramação surge de uma característica da publicação oficial ilustrada na Figura 2. Nota-se que nesse exemplo a informação é apresentada em 3 colunas. Assim o software faz 3 leituras por página, uma para cada coluna.

A diagramação da Figura 2 verifica-se a mais comum. A titulo de ilustração, levantou-se a quantidade exata de páginas que não obedecem esse padrão na edição do DOU de 28/05/2013. 79 de 440 páginas não seguem o padrão de três colunas. (17%). Entretanto, a existência de outros padrões em algumas páginas, dentre os quais o exemplificado na figura 3, provoca inconsistências na extração das informações.

O efeito prático dessa diferença de diagramação é que o componente de software, por não ter condições de tratar essa diferença, extrai os textos de forma desordenada, podendo causar ruídos na posterior criação de relacionamentos e extração de redes sociais.

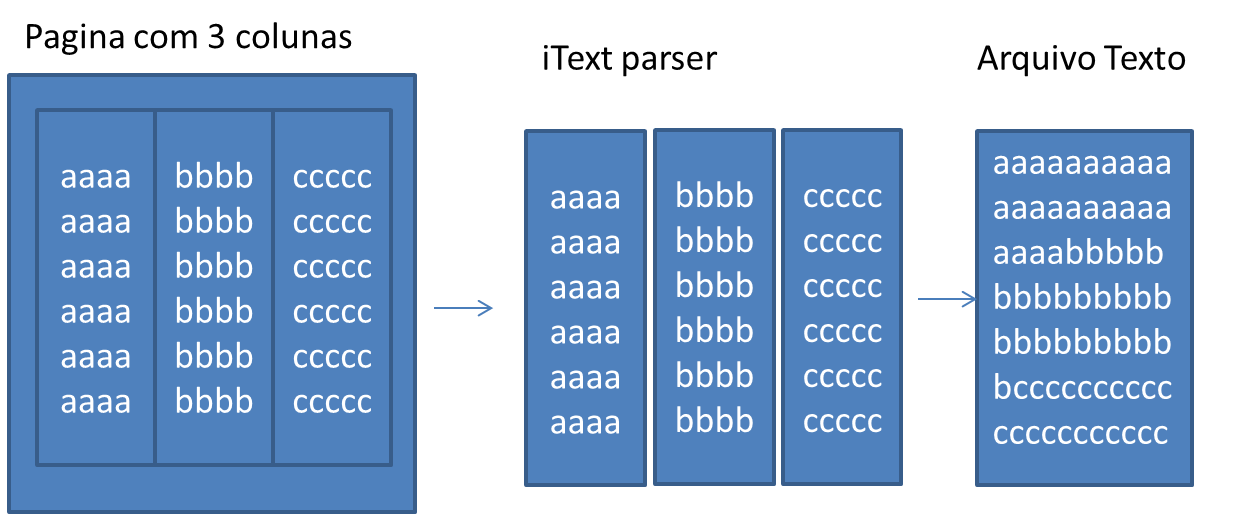


Figura 2 - Extração de texto de pagina com 3 colunas

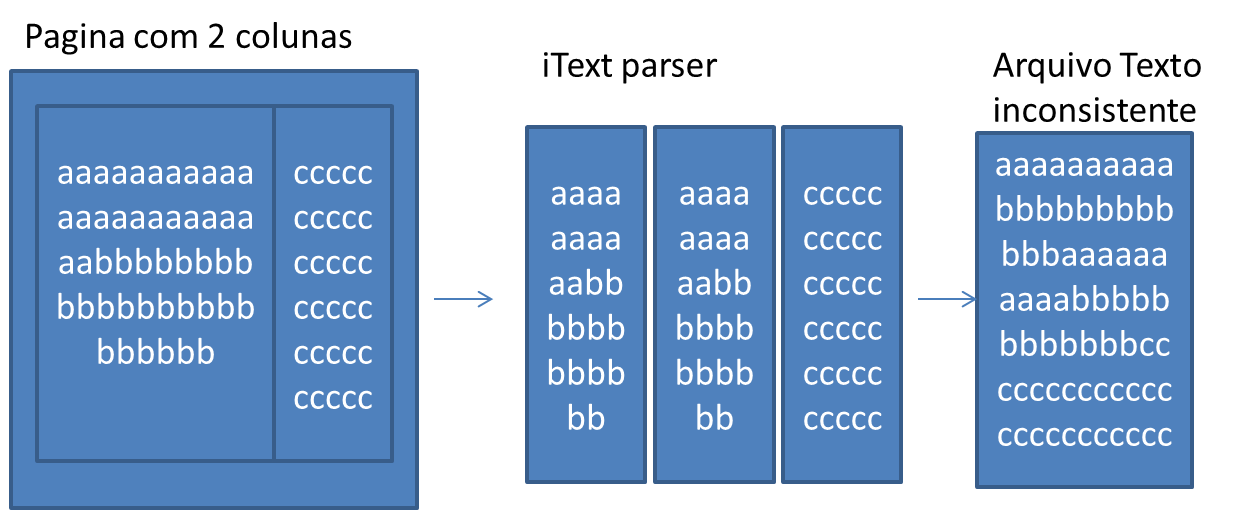


Figura 3 – Extração de texto de pagina com 2 colunas. Extração inconsistente.

Para facilitar a identificação dos limites entre as publicações, é preciso inserir marcadores de quebras duplas de linha. Uma quebra de linha maior do que as quebras comuns pode significar uma fronteira entre o fim de uma publicação e o inicio de outra. Embora essa identificação não seja feita no modulo DouDownloader, é nesse módulo que temos a chance de inserir marcadores desses espaços entre linhas.

Para tanto, é necessário modificar a estratégia de extração de texto usada pelo iText. Essa é uma possibilidade prevista pela biblioteca, cuja documentação orienta que seja criada uma classe que implementa a interface TextExtractionStrategy e que faça a extração customizada. A injeção desse código é feita nas linhas 5 e 6 da listagem 1, e a customização pode ser vista na listagem 2 abaixo, onde se reimplementa o método responsável por transformar o objeto TextChunk em texto String, inserindo o marcador “@@DOUBLE\_NW@@” onde se detecta quebra de linha maior que o padrão.

**public** String getResultantText(TextChunkFilter chunkFilter)

{

**for** (TextChunk chunk : filteredTextChunks)

{

**if** (chunk.sameLine(lastChunk))

{

**float** dist = chunk.distanceFromEndOf(lastChunk);

**if** (dist > chunk.charSpaceWidth / 2.0f

&& !startsWithSpace(chunk.text)

&& !endsWithSpace(lastChunk.text))

{

sb.append(' ');

sb.append(chunktext);

}

**else** //trata-se de uma quebra de linha

{

**if** ((distPerpendicular - as.distPerpendicular) > chunk.NORMAL\_SPACE\_BW\_LINES)

{

sb.append('\n');

sb.append("@@DOUBLE\_NW@@");

}

sb.append('\n');

}

lastChunk = chunk;

}

**return** sb.toString();

}

Listagem 2 – Customização do iText para inclusao de marcação de quebra dupla de linha.

## Analisador DOU – Análise textual

A parte mais desafiadora deste trabalho, contudo, foi desenvolvida ao implementar a identificação das entidades e das publicações nos textos produzidos pelo módulo DouDownloader. Esse trabalho foi feito com o suporte de uma outra biblioteca de terceiros, o GATE – GeneralArchtecture for Text Engeneering. Trata-se de um projeto de código aberto desenvolvido durante mais de 15 anos por esforços de cooperação entre universidades e organizações privadas visando construir uma coleção de soluções para processamento de texto e linguagem humana. (CUNNINGHAM, 2011. p 5).

Dentre as ferramentas oferecidas pelo GATE, duas foram utilizadas neste trabalho: Gazzeter e Tokenizer. O Gazzeter é um componente que recebe uma lista de termos e identifica as ocorrências desses termos dentro do texto analisado, separação silábica em quebras de linha e diferenciação entre maiúsculas e minúsculas. As seguintes listas foram elaboradas: Nomes completos de órgãos, Primeiros nomes de pessoas, Cargos, Palavras-chave de início de publicação.

O Tokenizer é um componente que permite que sejam definidas regras semelhantes a expressões regulares que determinam a identificação de elementos no texto.

As seguintes regras foram desenvolvidas para o uso no tokenizer:

**Nomes de pessoas**: Conjunto de palavras consecutivas iniciadas ou totalmente compostas por letras maiúsculas e dentre as quais pelo menos uma das palavras foi identificada previamente como nome pelo gazzeter.

**Inicio de Publicação**: Conjunto de elementos iniciado por uma palavra chave indicativa de inicio de publicação, precedido de uma quebra dupla de linha e seguido de uma sequencia qualquer de palavras ou números finalizada por outra quebra dupla de linha.

**Assinatura**: Conjunto de elementos composto de um Nome de Pessoa seguido por uma quebra de linha e um Nome de Órgão ou Um Cargo, identificados previamente pela regra de Nomes de Pessoas do Tokenizer ou pelo Gazzeter.

A biblioteca GATE oferece uma interface de desenvolvimento de aplicações chamada Gate Developer. Essa ferramenta permite fazer a orquestração dos componentes Gazzeter e Tokenizer e visualizar os resultados da aplicação sobre textos. O Gate Developer é capaz de exportar a aplicação desenvolvida (Gate AppState, ilustrado na Fig 1) para que ela seja incorporada por código java e utilizada pela biblioteca de forma automática.

O módulo AnalisadorDou foi desenvolvido em java e inclui as bibliotecas do GATE para executar o primeiro passo do desenvolvimento: execução do Gazzeter e do Tokenizer e identificação dos elementos do texto (Pessoas, Órgaos, Inicios e Assinaturas).

A segunda parte do processamento consiste em identificar os limites das publicações e associar a elas as Pessoas e Órgaos. Para tanto, aplicou-se o padrão de projeto strategy, que facilita a alternância entre diferentes abordagens na solução de problemas. Isso se deve ao fato de que há duas abordagens para a identificação de publicações: Pode-se assumir que uma publicação começa onde se detecta uma marcação do tipo ‘Inicio’ identificada pelo GATE. Entretanto, o fim dessa publicação pode ser definido pela localização da próxima assinatura ou do próximo inicio de publicação. A primeira estratégia é mais precisa mas falha quando processa algumas publicações que não são assinadas (como acórdãos do TCU). A segunda estratégia apresenta alguns casos em que textos que não pertencem a nenhuma portaria sejam considerados. Assim, o sistema foi concebido de forma a facilitar o intercambio entre essas e outras abordagens que venham a ser formuladas.

Independentemente da estratégia de detecção de limites da publicação que seja adotada, o sistema verifica na lista de entidades (pessoas e órgãos) se a localização delas no texto indica que estão dentro da publicação. Caso isso se verifique, uma ligação entre a publicação e a entidade é criada e precisa ser registrada.

Mais uma vez recorre-se ao padrão de estratégias para o registro dos relacionamentos identificados. Pode-se facilmente adotar uma estratégia de registro em banco de dados ou em texto plano, no formato que se considerar mais adequado. Há uma interface com a qual o modulo principal trabalha e essa interface recebe um objeto chamado Ligacao que contem os atributos do relacionamento como nome da entidade, data da ocorrência, etc...

## Extração de redes sociais

A extração de redes sociais, no formato em que o software se encontra, é feita através de consultas SQL. Por motivos práticos a estratégia de gravação em banco de dados esta sendo adotada. Assim, cabe ao usuário interessado em extrair as redes escrever as consultas conjugando 3 tabelas:

**Tabela de entidades**: Armazena as entidades encontradas no texto. Possui uma coluna indicando o tipo (órgão ou pessoa)

**Tabela de publicações**: Armazena as publicações que contam com uma identificação numérica e o texto integral, o que permite o filtro por assunto.

**Tabela de relacionamentos**: Armazena os relacionamentos entre as entidades e as publicações. Possui a informação de data da ocorrência.

## Considerações finais

A mineração de dados do DOU para extração de redes sociais é uma tarefa que envolve um esforço considerável de desenvolvimento e, como apresentado nesse documento, uma conjugação de diversas tecnologias. Embora trate-se de uma versão inicial, que não contempla todas as exceções e possibilidades da tarefa, a solução desenvolvida mostrou-se capaz de estrair informações coerentes e em grande quantidade, podendo ser considerada uma fonte interessante de informação para estudos acerca da comunicação e interação entre órgãos, pessoas e temas na administração publica brasileira.

O leitor interessado em ter acesso aos códigos fonte utilizados é encorajado a contactar o autor para compartilhamento dos mesmos.

## Referencias

ADOBE, Legal Notes for Developers. 2013. Disponivel em < http://partners.adobe.com/public/developer/support/topic\_legal\_notices.html>

CUNNINGHAM, et al. Text Processing with GATE (Version 6). University of Sheffield Department of Computer Science. Disponivel em <http://gate.ac.uk/sale/tao/tao.pdf>

ITEXT Software Corp. iText PDF Library. 2013. Disponivel em <http://itextpdf.com/terms-of-use/index.php>

RODRIGUEZ, Jose. 2004. The March 11th terrorist network: In its weakness lies its strength. Disponivel em <http://www.ub.edu/epp/wp/11m.PDF> Acessado em 16/05/2013

KREBS, Valdis E. “Mapping Networks of Terrorist Cells,” Connections, Vol. 24, No. 3,pp.43-52, 2002. Disponivel em <http://vlado.fmf.uni-lj.si/pub/networks/doc/Seminar/Krebs.pdf> Acessado em: 19/05/2013

SAGEMAN, Marc. Understanding Terror Networks. UPCC book collections on Project MUSE. University of Pennsylvania Press, 2011 Disponivel parcialmente em <http://books.google.com.br/books?id=iCoYDUv63L8C&dq=> Acessado em 15/05/2013