E-Dictor: novas perspectivas na codificação e edição de corpora de textos históricos

Maria Clara Paixão de Sousa (USP)

Fábio Natanel Kepler (USP/PG)

Pablo Picasso Feliciano de Faria (UNICAMP/PG)

Resumo:
Neste artigo apresentamos o E-Dictor, uma ferramenta concebida para auxiliar a edição eletrônica em XML de textos antigos para fins de análise lingüística automática. A versão preliminar da ferramenta (Paixão de Souza & Kepler, 2007) surgiu de demandas observadas na construção do Corpus Anotado do Português Tycho Brahe (CTB) e em atividades de consórcio entre a equipe deste corpus e a equipe do projeto PROHPOR-UFBA. A experiência com o processo de edição de textos no CTB, em que, além de filologia e linguística, cada editor tinha que aprender a manipular a linguagem XML, tornou flagrante a necessidade de se facilitar a aplicação do sistema e, assim, ampliar seu uso para diferentes grupos de editores. Quanto à confiabilidade, esta experiência inicial nos mostrou que a codificação em XML com intervenção direta sobre o documento é demasiadamente sujeita a falhas e demanda extensa e incessante revisão da codificação. No entanto, as ferramentas disponíveis (na internet) para este fim não supriam as necessidades do CTB. Portanto, ampliar o alcance da anotação XML e torná-la mais amigável e confiável foi a motivação primeira do desenvolvimento de uma ferramenta de anotação específica para textos históricos, com uma interface que medie a relação entre o editor (usuário) e a codificação XML. Além disso, a ferramenta une em um só ambiente tanto a edição do texto quanto a correção de etiquetas morfológicas aplicadas às palavras. Os resultados preliminares do uso do E-Dictor mostram um ganho de pelo menos 50% no tempo total do processo de edição (transcrição-edição-revisão).

Palavras-chave: corpora de textos – filologia – processamento eletrônico – análise linguística

Abstract:
This paper introduces E-Dictor, an auxiliary editing tool for codifying ancient texts in XML, for further linguistic analyses. The preliminary version of the tool (Paixão de Sousa & Kepler, 2007) came from observed needs in the building of Tycho Brahe Parsed Corpus of Historical Portuguese (CTB) and from consortium activities between CTB and PROHPOR-UFBA projects teams. The process of text edition on CTB demanded each human editor to learn not only those aspects concerning linguistics and philology, but also how to use and manipulate the XML standard. The experience in this process exposed the need for an easier way to apply the XML structure over the texts if we wanted to extend its use to different groups of editors. In terms of reliability, we concluded that the direct intervention of the editor on the XML document (and code) leads to an undesirable amount of errors which are difficult to revise and fix (due to the mixing of two layers, text content and XML code). On searching for other tools, we

concluded that none of the available tools found could attend our needs. Thus, extending the use of XML annotation and turning it into a more friendly and reliable activity were the two primary motivations for the development of an annotation tool for ancient texts, providing an interface that mediates the relation between the human editor and XML code. Furthermore, the tool is intended to bring the activities of editing the text and of revising morphological annotation into the same environment. The preliminary results of its use show a decrease of at least 50% on the editing process overall time (transcription-edition-revision).

Keywords: text corpora – philology – eletronic processing – linguistic analysis

1. Introdução

Neste artigo apresentamos uma ferramenta auxiliar para a edição eletrônica de textos antigos para fins de análise lingüística automática, o E-Dictor. A versão preliminar da ferramenta (PAIXÃO DE SOUSA & KEPLER, 2007) surgiu de demandas observadas na construção do Corpus Anotado do Português Tycho Brahe (CTB) e em atividades de consórcio entre a equipe deste corpus e a equipe do projeto PROHPOR-UFBA. Mais recentemente, o E-Dictor começou a ser testado pela equipe de edição de textos ligada ao projeto Brasiliana Digital (BBD), e, no futuro próximo, será incorporada por outros grupos de pesquisa dedicados à edição filológica de textos portugueses em meio digital em âmbito nacional. O desenvolvimento do E-Dictor se integra, ainda, às preocupações do grupo temático “Protección del patrimonio literario através de formatos digitales”, da Associação Internacional de Literatura Comparada (ICLA, 2009), que pesquisa programas computacionais aplicáveis à digitalizações de obras raras em geral.

Aqui apresentamos as modificações e avanços obtidos na versão atual (1.0 *beta, build* 001), e discutimos os desafios a serem enfrentados nas próximas etapas de desenvolvimento.

1.1 Motivações

A iniciativa de desenvolvimento do E-Dictor surgiu no âmbito da construção do Corpus Anotado do Português Tycho Brahe. Atualmente composto por 52 textos em português escritos por autores nascidos entre 1380 e 1845, com mais de dois milhões de palavras, o Corpus começou a ser construído em 1998, quando representou uma das principais frentes de trabalho num projeto de pesquisa dedicado à investigação da história da Língua Portuguesa (Galves e Galves, 1998). Esse projeto se consolidou como uma das diversas iniciativas responsáveis pela retomada da perspectiva histórica sobre a língua no Brasil nas últimas duas décadas, a partir da renovação da relevância teórica dos estudos da mudança lingüística em diferentes quadros teóricos (Mattos e Silva, 1988; Kato & Roberts, 1993; Castilho, 1998).

O crescimento do interesse pelo olhar diacrônico trouxe como conseqüência a intensificação do trabalho com textos antigos no país (Megale & Cambraia, 1999) - e, para algumas das pesquisas realizadas a partir do final da década de 1990, passou a conferir centralidade também para a Lingüística de Corpus, dando vazão ao surgimento de um campo de confluência entre duas áreas de estudo aparentemente díspares - a Filologia e a Ciência da Computação.

Para entendermos os desafios que se colocam para o trabalho com o texto antigo no meio digital, precisamos notar que os requerimentos do processamento computacional nem sempre estão em consonância com os requerimentos filológicos. Centralmente, os objetivos de uma edição com fins de processamento computacional não correspondem plenamente aos objetivos de uma edição filológica tradicional para fins de análise linguística.

Os estudos linguísticos baseados em textos antigos dependem, antes de tudo, da garantia da fidelidade às formas originais dos textos – este é de fato o pilar de sustentação que qualquer estudo lingüístico, em qualquer quadro teórico, deve pressupor. Naturalmente, a forma mais fiel de se reproduzir um texto antigo é o fac-simile - e as tecnologias digitais têm facilitado imensamente a obtenção de cópias dessa natureza, por meio de fotografias ou escanerização. Os fac-similes digitais, entretanto, não são usados como fonte central nas pesquisas lingüísticas, uma vez que nelas é necessário trabalhar o texto como seqüências de caracteres, não como imagens. No caso de textos impressos, uma opção possível seria lançar mão dos recursos automáticos de reconhecimento de caracteres, ou OCR (*Optical Character Recognition*), que transformam imagens em textos. Para os textos mais antigos, entretanto, as tecnologias atuais de reconhecimento têm se mostrado inadequadas: em um estudo recente (Paixão de Souza, 2009a), mostramos que um software de OCR de ponta apresenta um índice de acertos na faixa de apenas 57% a 77% em impressos portugueses do século XVII, por exemplo. Já para os documentos manuscritos, simplesmente não há tecnologia de reconhecimento disponível. Assim, as pesquisas linguísticas precisam lançar mão do trabalho de transcrição dos documentos a serem estudados.

A "transcrição", naturalmente, é por sua vez também uma forma de interferência no texto, que pode ser mais ou menos intensa. No primeiro degrau da escala de fidelidade ao original estão as transcrições ditas "conservadoras", que sofrem o menor grau possível de interferência editorial: as chamadas "Edições Diplomáticas". Grande parte das pesquisas atuais, entretanto, opta pelas edições "Semi-Diplomáticas", nas quais um grau ligeiramente maior de interferência é considerado aceitável - basicamente, a modernização tipográfica ou grafemática, e o desenvolvimento ou "abertura" das abreviaturas dos textos originais. Um texto editado semi-diplomaticamente, portanto, guarda a maior parte das características linguísticas interessantes do original - lexicais, morfológicas, sintáticas - mas tem sua leitura ligeiramente facilitada no que remete aos aspectos grafemáticos ou paleográficos. Daí se depreende o objetivo central de uma edição filológica: tornar o texto acessível para o leitor especializado de hoje, com a máxima preservação de suas características originais.

Na construção de um arquivo de textos digitais, entretanto, esse objetivo precisa ser integrado com requerimentos impostos pela vertente computacional e lingüística dos estudos: a necessidade de quantidade, agilidade e automação no trabalho estatístico de seleção de dados. Notemos, sobretudo, que as características gráficas e grafemáticas dos textos mais antigos (preservadas nas transcrições conservadoras) dificultam o processamento automático posterior (como a anotação morfológica). Assim, para cumprir o objetivo de processamento automático, o texto original precisa ser preparado, ou editado, com um grau de interferência mais elevado do que aquele aceitável numa edição semi-diplomática. É neste ponto, portanto, que os objetivos da

preparação para o processamento automático entram em conflito com os objetivos da edição filológica.

Nos primeiros anos da construção do CTB, a edição dos textos (que incluia a modernização das grafias e a normalização dos aspectos grafemáticos) tornava-os adequados para o processamento automático, mas ocasionava a perda de características do texto original importantes para o estudo histórico da língua. Ao finalizarem-se os primeiros cinco anos de desenvolvimento do CTB, a tensão resultante desse duplo vetor - filologia e computação - deu origem ao projeto "Memórias do Texto" (Paixão de Souza, 2004), que pretendia reestruturar o Corpus com base no desenvolvimento de anotações XML (*eXtensible Markup Language*). A idéia central do projeto era aproveitar as características centrais desse tipo de codificação: a abertura para uma grande variedade de manipulações das informações codificadas, por exemplo, através de transformações utilizando a tecnologia XSLT (W3C, 1999), que permitem gerar “versões” (de fato, transformações) do documento XML (que podem exibir a lista de palavras, o texto original ou o editado, converter para PDF, etc.).

Como resultado, concebeu-se e implementou-se um sistema de anotação de edição em XML que permitia resguardar as informações filológicas fundamentais dos textos ao mesmo tempo em que os tornava passíveis de tratamento computacional em grande escala (Paixão de Souza, 2006). Esse sistema de anotação de edição foi aplicado a 48 textos portugueses escritos entre os séculos 16 e 19 (2.279.455 palavras), metade dos quais recebeu também anotação automática para classes de palavras, tendo servido de base para diversas teses, dissertações e outros trabalhos sobre a morfologia e a sintaxe do português clássico. A partir do ano de 2006, o sistema foi experimentado por outros grupos de pesquisa interessados na produção de corpora do português antigo e clássico (notadamente, o Programa para a História da Língua Portuguesa, PROHPOR -UFBa). É possível dizer que o sistema de 2006 chegou a atender aos objetivos inicialmente colocados pelo projeto - mas falhava nos quesitos confiabilidade e (sobretudo) facilidade de uso.

Quanto à facilidade de uso, notamos que, embora o XML tenha uma definição bastante simples, a marcação direta com XML nos arquivos era desafiante para alguns, e trabalhosa para todos. No sistema original até 2006, cada texto era editado sob forma de uma anotação em XML sobre a transcrição do texto; a anotação codificava os itens originais e as interferências do editor, permitindo que cada uma dessas categorias fosse mais tarde selecionada isoladamente do arquivo (via XSLT), atendendo portanto o objetivo de facilitar o processamento ao mesmo tempo em que se preservam as informações históricas. A anotação era aplicada aos textos manualmente, no processador Emacs. Em seguida, a versão modernizada dos textos gerada por XSLT passava para a anotação morfológica (POS) automática; o resultado dessa anotação POS, por sua vez, era corrigido semi-manualmente, também no processador Emacs, no modo "lex". O processo de ampliação do uso do sistema de anotação do CTB tornou flagrante a necessidade de se facilitar a aplicação do sistema e ampliar seu uso para diferentes grupos de editores - evitando que, além de filologia e linguística, cada editor tivesse que aprender a manipular a linguagem XML. Quanto à confiabilidade, os primeiros anos de experiência nos mostraram que a codificação em XML com intervenção direta sobre o documento é demasiadamente sujeita a falhas e demanda extensa e incessante revisão da codificação.

Ficou claro, portanto, que era necessário encontrar uma outra alternativa para a codificação eletrônica de textos, uma que tornasse a tarefa mais amigável, confiável e produtiva.

1.2 Justificativas

Como vimos, a necessidade de uma ferramenta de anotação específica para textos antigos surgiu da nossa avaliação de que era preciso tornar o sistema concebido para a codificação do CTB mais amigável e confiável, ao mesmo tempo em que preservasse as vantagens de uma codificação especificamente voltada para edições filológicas. Diante disso, o primeiro passo foi pesquisar e levantar ferramentas e/ou tecnologias já disponíveis, que pudessem satisfazer esse duplo requerimento. Este levantamento, no entanto, mostrou que não havia uma ferramenta que atendesse satisfatoriamente às nossas necessidades específicas.

De todo modo, vale a pena citar algumas das opções existentes, numa espécie de quadro do "estado da arte", em termos de codificação eletrônica de textos em XML, visto que tais ferramentas podem ser suficientemente adequadas para as necessidades específicas a outros projetos.

1.2.1 O estado da arte

Multext

Coordenado por Jean Véronis (Université de Provence), o Multext é resumidamente descrito[1] como:

"[...] uma série de projetos cujas metas são o desenvolvimento de padrões e especificações para a codificação e processamento de corpora linguísticos, e o desenvolvimento de ferramentas, corpora e recursos linguísticos que encarnem estes padrões. O Multext tem desenvolvido ferramentas, corpora e recursos linguísticos para uma grande variedade de línguas, incluindo Bambara, Búlgaro, Catalão, Tcheco, Holandês, Inglês, Estoniano, Francês, Alemão, Húngaro, Italiano, Quicongo, Occitano, Romeno, Esloveno, Espanhol, Sueco e Suaíli. Todos os resultados do Multext são liberados e disponíveis publicamente para propósitos não-comerciais e não-militares."

Nas definições de seus padrões, o Multext também levou em consideração os esforços de outros grandes projetos, a saber, o EAGLES[2] e o *Text Encoding Initiative* (TEI[3]). Portanto, o Multext prevê a codificação do textos em XML. Para isso, desenvolveu o editor MtScript, que permite a utilização de vários sistemas de escrita, tais como o Latino, o Arábico, o Grego, o Chinês, etc., para transcrever o texto.

Embora não tenhamos conseguido testar a ferramenta, para termos uma idéia exata de seu funcionamento, em função de problemas com sua instalação, o que concluímos, a partir da documentação oferecida na página do projeto, é que este editor parece mediar o contato do usuário com a linguagem XML, gerando a estrutura subjacente a partir da formatação do texto através da interface do editor. Um ponto forte deste projeto é que este já desenvolveu algumas ferramentas para processar o texto editado em XML, entre elas, ferramentas de segmentação de texto, de processamento

morfo-lexical (etiquetagem, disambiguação, etc.), entre outras. Este aspecto é de especial importância, pois poupa as equipes responsáveis por corpora do desenvolvimento de tais ferramentas de processamento.

CLaRK[4]

Esta ferramenta foi desenvolvida em linguagem Java, o que permite que seja executada em vários sistemas operacionais (como Windows, Mac e Linux). Ela consiste em um editor de texto em Unicode (padrão que permite a codificação de caracteres de praticamente qualquer língua), com facilidades para edição em XML, como checagem de validade da codificação (de acordo com restrições que podem ser definidas pelo usuário), aplicação de sistemas de cores para diferenciar código XML do conteúdo textual, entre outras. Além disso, esta ferramenta oferece uma série de outras funcionalidades voltadas para o processamento do texto, para extrair informações, como listas de concordância, extração de informações diversas, com base no que foi codificado, buscas por expressões regulares e por sequência, importação de documentos XML e em formato RTF, estatísticas de frequência, tokenizador, entre outras funcionalidades. Pode-se dizer que esta ferramenta é um "super-editor de texto", que além das funcionalidades específicas do editor, ainda agrega diversas funcionalidades importantes para a Linguística de Corpus.

Editores de texto (Emacs, Kate, EditPlus, etc.)

Um arquivo XML não é nada mais que um arquivo de texto simples. O que o identifica de modo especial é o seu conteúdo, ou seja, as marcações que contém. Portanto, um arquivo XML pode, a priori, ser editado em qualquer editor de texto desde que se mantenha seu caráter de texto simples. Por exemplo, poderíamos utilizar o Notepad do Windows e editores semelhantes no Mac ou no Linux. Entretanto, em função das particularidades de um arquivo XML, há diversos editores que agregam funções especiais para facilitar sua edição.

Nessa linha, dentre as opções disponíveis, em nossa experiência no Projeto Tycho Brahe, utilizamos o Emacs, o EditPlus e o Kate[5]. Todos estes oferecem opções como a distinção de cores, que permite separar visualmente o que é código XML do que é conteúdo textual, auto-identação das estruturas e busca de expressões, com opção de substituição. Além disso, oferecem suporte à codificação em UTF-8, um padrão que permite a codificação de praticamente todos os tipos de caracteres. Estes editores, particularmente, são gratuitos e bastante úteis para o que se propõem.

Avaliação das ferramentas

Em termos gerais, a opção que mais se aproxima das necessidades do CTB é o conjunto de ferramentas do Multext, visto que oferece uma interface para evitar o contato direto com a codificação XML, além de ferramentas para processar o documento XML gerado. Quanto aos editores em geral, incluindo a ferramenta CLaRK, a principal limitação reside em dois pontos: não há verificação da estrutura XML, quanto à má formação, isto é, se a linguagem XML foi usada corretamente (exceto para o CLaRK); não há como visualizar apenas o conteúdo textual codificado, para verificar

sua correção. Para suprir ambas as limitações, é preciso recorrer ao uso de um navegador web (por exemplo, Firefox ou Internet Explorer) e de transformações XSLT, tanto para exibir o código XML gerado e descobrir erros de má formação, quanto para exibir transformações do XML, para exibição apenas do conteúdo textual.

Conclui-se, portanto, que, com relação às necessidades de edição do Projeto Tycho Brahe, em especial a modernização de grafia e normalização de aspectos grafemáticos, nenhuma das opções disponíveis atualmente é satisfatória, embora possam ser de grande utilidade noutros contextos.

1.2.2 Solução encontrada

A partir das motivações descritas no início desta seção e da avaliação das ferramentas disponíveis tal como resumida logo acima, começamos a planejar uma ferramenta de anotação específica para textos antigos, que pudesse absorver as vantagens do sistema de anotação em XML para as edições filológicas, mas que possibilitasse uma interface amigável e confiável.

A questão da facilitação da interface para o trabalho de edição dos textos se uniu a um segundo objetivo na nossa busca por uma ferramenta adequada para o trabalho no Corpus: idealizávamos que esta ferramenta possibilitasse a integração entre o sistema de edição e o sistema de correção da anotação morfológica. O processo original de codificação, como mencionado na seção 1.1, funcionava em módulos separados, sendo cada tarefa desempenhada num ambiente de processamento diferente, gerando um desperdício operacional. A idéia agora era desenvolver um sistema no qual o fluxo do processamento passasse do que é apresentado na Figura 1 para o que apresentamos na Figura 2:

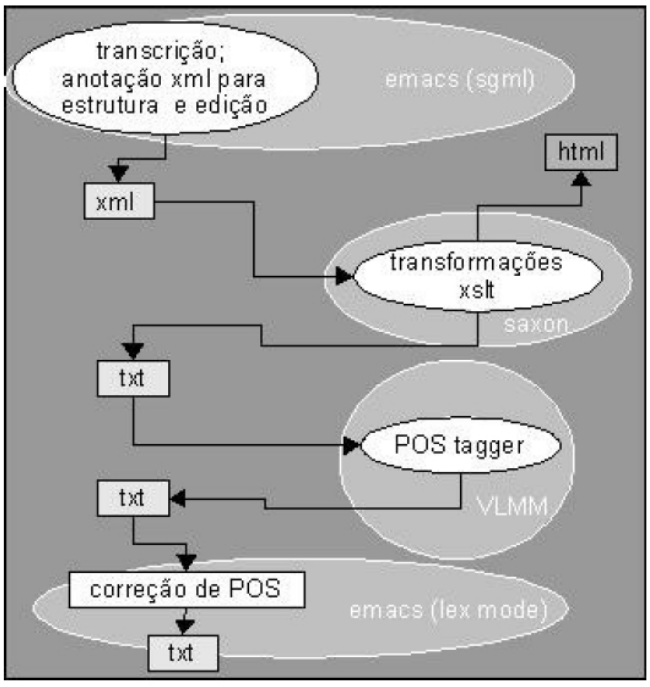

Figura 1. Fluxo do processamento eletrônico de textos no CTB (Paixão de Sousa, 2007)

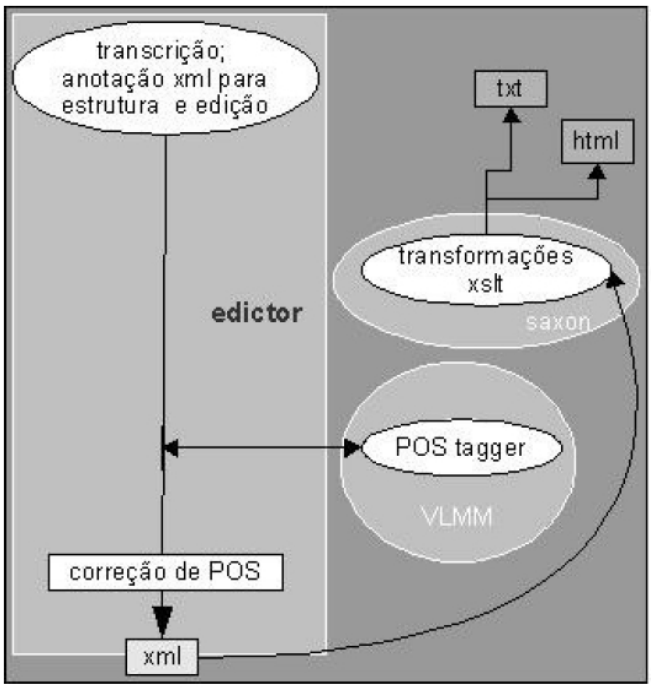

Figura 2. Fluxo previsto (E-Dictor) do processamento eletrônico de textos no CTB (Paixão de Sousa, 2007)

Assim, num primeiro ciclo de desenvolvimento, chegamos à primeira versão (0.3.3) do E-Dictor (Kepler & Paixão de Sousa, 2007), que em seguida passou por outras evoluções importantes até chegar à versão 1.0 *beta (build 001)*[6], apresentada neste artigo.

2. E-Dictor: características

O E-Dictor, através de sua interface, visa a evitar o contato direto entre o editor (usuário) e a estrutura XML subjacente. Para isso, sua interface prima pela exibição do conteúdo textual, deixando as marcas de estrutura em segundo plano, embora visíveis (estas são importantes, afinal o editor precisará ter acesso às quebras de linha, página, marcas de fim de sentença, parágrafo, etc.). Nas seções a seguir, apresentamos com algum detalhe suas características gerais. Vale ressaltar que utilizamos os termos "editor" e "usuário" como sinônimos.

2.1 Características computacionais gerais

Por razões de portabilidade, poder de expressão e acesso à documentação, escolhemos a linguagem de programação Python[7], para desenvolver o E-Dictor. Esta linguagem, inclusive, é utilizada em várias outras aplicações linguísticas, como as do projeto *Natural Language Toolkit*[8] e em Bird et al (2009).

O ambiente preferido para o desenvolvimento foi o ambiente Linux, utilizando a plataforma Eclipse[9], para gerenciar o projeto da ferramenta. Para adminstração do projeto, utilizamos o ambiente Trac[10], que disponibiliza uma série de ferramentas para a gerência de projetos de software, como painel de discussão, página de *downloads*,

estabelecimento de metas de desenvolvimentos, tarefas, listagem de bugs, etc., além de funcionar em conjunto com o Eclipse, através de *plugins*.

Inicialmente, optou-se pelo desenvolvimento de versões do E-Dictor para dois sistemas operacionais: Linux e Windows (XP/Vista). Uma terceira versão em Mac está prevista, mas ainda sem data definida para sair. Quanto à metodologia de desenvolvimento, o processo é incremental, em ciclos, em que versões (com correção, modificação ou inclusão de funcionalidades) são geradas, testadas e disponibilizadas na internet. A ferramenta ainda não chegou na versão 1.0 e deverá passar pelos estágios de versão 1.0 *alpha* e, em seguida, 1.0 *beta*, antes de ser considerada como versão 1.0 *estável*.

2.2 Características gerais da interface

Para atender ao fluxo definido na Figura 2, a interface gráfica do E-Dictor foi definida como mostra a Figura 3:

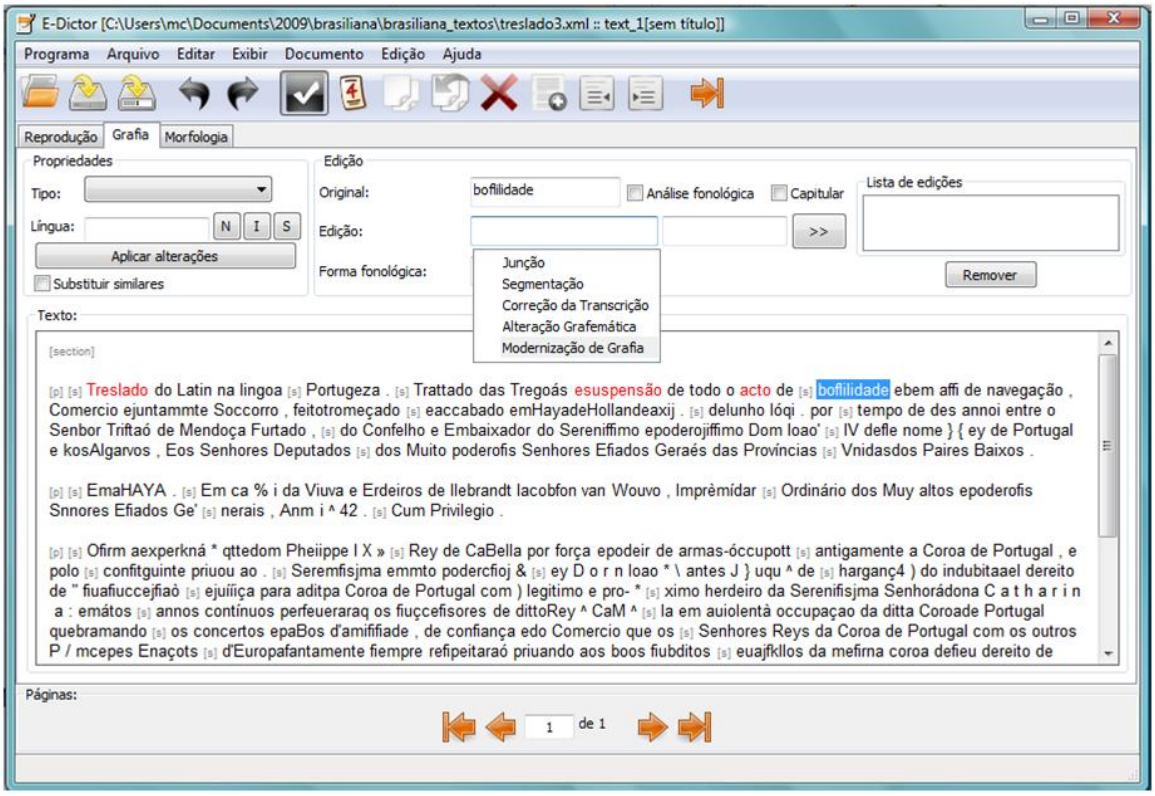

Figura 3. Interface gráfica do E-Dictor

Como vemos acima, temos:

- Menu da aplicação: menus de opções da aplicação, para acesso a todas as suas funcionalidades.
- Barra de ferramentas: acesso às opções mais rotineiras no trabalho de codificação e edição.

- Área do texto, organizada em três abas: Reprodução, para transcrição do texto-fonte; Grafia, para edição do texto; e, Morfologia, para inclusão/revisão das informações sobre classe de palavras.
- Barra de navegação entre as páginas do documento.

Note que as abas estão ordenadas em concordância com o fluxo do processo de codificação. Assim, o texto deve ser transcrito na aba "Reprodução", para depois ser convertido para XML e editado na aba "Grafia". Após ser convertido para XML, o usuário pode alternar entre as abas "Grafia" e "Morfologia", embora sugere-se que apenas após completar a edição se inicie a inclusão e/ou revisão das etiquetas morfológicas.

Na aba "Reprodução", o texto deve ser transcrito conforme está na fonte. Nesta aba é possível marcar quebras de parágrafo (acrescentando uma linha em branco entre trechos do texto) e quebra de sentença (com quebra de linha, quando não há uma pontuação de final de sentença). Ao converter o texto transcrito para XML, o E-Dictor vai tentar inferir da melhor forma possível (automaticamente) sua estrutura interna (que deverá ser então revisada e corrigida por um editor humano).

Na aba "Grafia", o texto é exibido com marcações de sua estrutura (seção, parágrafos, sentença, cabeçalho e rodapé) e os símbolos (usaremos "símbolo" e "palavra" de modo intercambiável) são todos individualizados (separados por espaço em branco). As edições feitas são exibidas em destaque, para que o usuário possa ver exatamente quais símbolos foram editados, quais não. Para acessar os símbolos basta clicar sobre eles ou navegar para frente ou para trás, no texto, utilizando teclas de atalho.

A aba "Morfologia" tem uma apresentação e funcionamento semelhante à da aba "Grafia", exceto por duas distinções: palavras e etiquetas de classes de palavra são exibidas no formato "palavra/ETIQUETA" e os trechos marcados como não analisáveis (ver seção 2.4) são exibidos em cor cinza e não podem ser acessados. Na Figura 4, abaixo, temos um exemplo de como o conteúdo é exibido nesta aba:

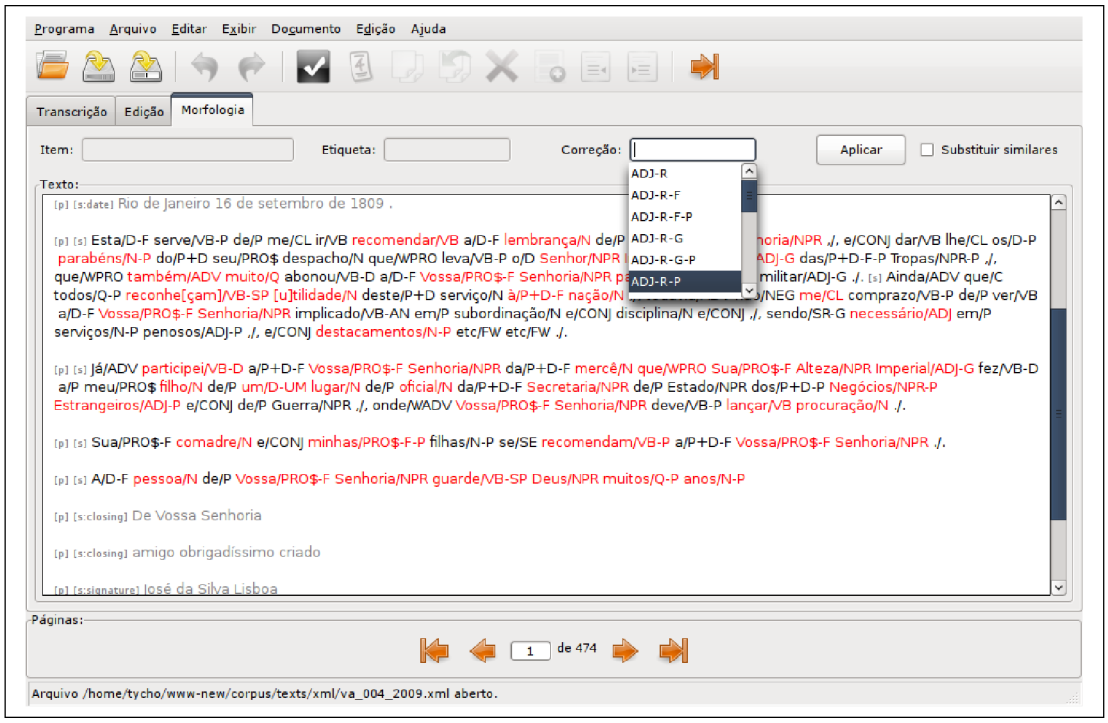

Figura 4. Aba "Morfologia" do E-Dictor.

Um aspecto importante a mencionar é que parte do funcionamento do processo de codificação e das abas mencionadas está sujeito à configuração das informações na janela de preferências da aplicação (ver seção 2.4).

2.3 Estrutura XML

A especificação da estrutura XML para codificação no E-Dictor vai de encontro a dois objetivos principais: (i) ser o mais neutra possível (em relação ao conteúdo textual codificado) e (ii) atender a necessidades linguísticas e filológicas, em outras palavras, é preciso que a preparação de conteúdo para análises lingüísticas seja simples e eficiente, sem que se percam informações relevantes para estudos filológicos.

Em termos gerais, a codificação de informações em XML é muito flexível. Pode-se conceber padrões para diversas finalidades. Um exemplo disso, são os padrões derivados do XML, tais como o HTML (utilizado em páginas web), o DOCX (utilizado nas versões atuais do Microsoft Office), entre outros. Em razão desta flexibilidade, a especificação da estrutura XML deve ser feita de um modo sistemático e muito (pré)refletido, sob pena de se obter uma estrutura ambígua e redundante, que demandará diversas revisões, até ficar satisfatória.

Grosso modo, o XML prevê dois tipos de elementos "estruturais" (à parte o conteúdo codificado): *etiquetas* ("tag") e *propriedades de etiquetas* ("property"). O exemplo a seguir ilustra estes elementos:

```
<etiqueta1 propriedade1=valor1 propriedade2=valor2 ... >conteúdo codificado</etiqueta1>
<etiqueta2 propriedade1=valor1 propriedade2=valor2 ... />
```

O exemplo acima ilustra ainda outros aspectos. Primeiro, note que há valores para as propriedades ("valor1" e "valor2") que podem ser utilizados tanto para codificar parte do conteúdo a ser textual quanto para informações meta-textuais. Note, também, que o elemento "etiqueta1" é "fechado", ou seja, possui um elemento "</etiqueta1>" correspondente, encerrando um "conteúdo codificado". Por fim, note que o elemento "etiqueta2" é "aberto": não possui um correspondente, como a "etiqueta1", sendo marcado com um "/>" ao final da lista de propriedades. Elementos abertos são utilizados quando não há conteúdo a ser codificado para ele.

Com base nesta breve apresentação, podemos ver que um aspecto de suma importância, portanto, é a decisão quanto ao que será codificado como etiqueta (<tag></tag> ou <tag/>) e ao que será codificado como propriedade de uma certa etiqueta (<tag prop=valor>). A diretriz mais assumida é a de que informações estruturais (abstraídas do conteúdo a ser codificado) devem ser etiquetas, enquanto informações sobre as “informações estruturais” (meta-informações) devem ser codificadas como propriedades. Um exemplo para ilustrar:

```
<texto>
    <secao tipo='capitulo'>
        <elemento_de_secao tipo='titulo'>Título da seção</elemento_de_secao>
        <paragrafo>
            Primeira linha do parágrafo<quebra tipo="linha"/>
            que segue na segunda linha...
        </paragrafo>
    </secao>
</texto>
```

Acima, temos vários elementos: um elemento <texto> codificando todo o texto, o elemento <secao> codificando uma seção do texto, o elemento <elemento_de_secao> codificando o título da seção, um elemento <paragrafo>, para codificar os parágrafos do texto e um elemento <quebra>, para marcar quebras de linha no texto. Repare que alguns possuem a propriedade "tipo", que permite especificar subtipos específicos a cada elemento. Assim, podemos ter vários tipos de seção (como "capítulo", "prólogo", etc.) ou quebras (de "linha", "página", etc.). Seguindo esta orientação, ou seja, uma estrutura neutra com opções de especialização, podemos obter um nível interessante de flexibilidade para codificação de diferentes gêneros textuais.

O próximo passo, portanto, é decidir o que codificar. Um texto possui diversos aspectos passíveis de serem codificados, como os aspectos gráficos (layout), o conteúdo, análises (filológicas e linguísticas) do conteúdo, etc., mas nem todos são relevantes, a depender de suas necessidades. Com relação ao CTB, inicialmente, foi estabelecida uma estrutura capaz de codificar as seguintes informações:

- Metadados: informações diversas acerca do texto-fonte codificado, como dados do(s) autor(es), dados bibliográficos do texto-fonte, informações sobre o estado do processamento etc.
- Elementos do texto em geral (delimitação de seções, parágrafos, sentenças, cabeçalho e rodapé, e símbolo).
- Classe de palavras (etiqueta morfológica) e forma por extenso, para cada símbolo.

- Níveis de edição filológica para cada símbolo (aspectos gráficos, grafemáticos e modernização).
- Comentários do editor (relacionados ao texto em geral, a uma seção, parágrafo ou símbolo específico).

Os elementos do texto possuem uma propriedade que permite criar subtipos específicos. Por exemplo, pode-se codificar diferentes tipos de seções para as partes de um livro (prólogo, prefácio, capítulo, etc.) ou mesmo para codificar um conjunto de cartas ou contos de um dado autor (cada carta, por exemplo, poderia ser uma seção). O mesmo raciocínio pode ser aplicado aos parágrafos, sentenças e até mesmo símbolos (por exemplo, marcar números que iniciam títulos de seções como "numeração").

Os elementos do texto estão em relação de continência. Uma seção contém parágrafos, cabeçalho e rodapé (estes por sua vez contém parágrafos e o número da página). Cada parágrafo deverá conter uma ou mais sentenças. Sentenças contém um ou mais símbolos. Outras informações textuais, como quebra de linha, coluna ou página também podem ser marcadas. Em suma, na versão atual, estas são as informações codificadas. Este sistema não está fechado, podendo ser ampliado no futuro, em decorrência de novas formas de utilização da ferramenta, ainda não previstas.

2.4 Flexibilidade de codificação

Embora desenvolvido dentro de um contexto particular, isto é, o do CTB, um dos principais objetivos para o E-Dictor, já em sua versão 1.0, é a de ser flexível o suficiente para que possa ser útil em outros contextos de construção de corpora de textos.

Para isto, foi desenvolvida uma funcionalidade através da qual o usuário pode configurar "preferências" da aplicação. A janela respectiva é exibida na Figura 5 abaixo:

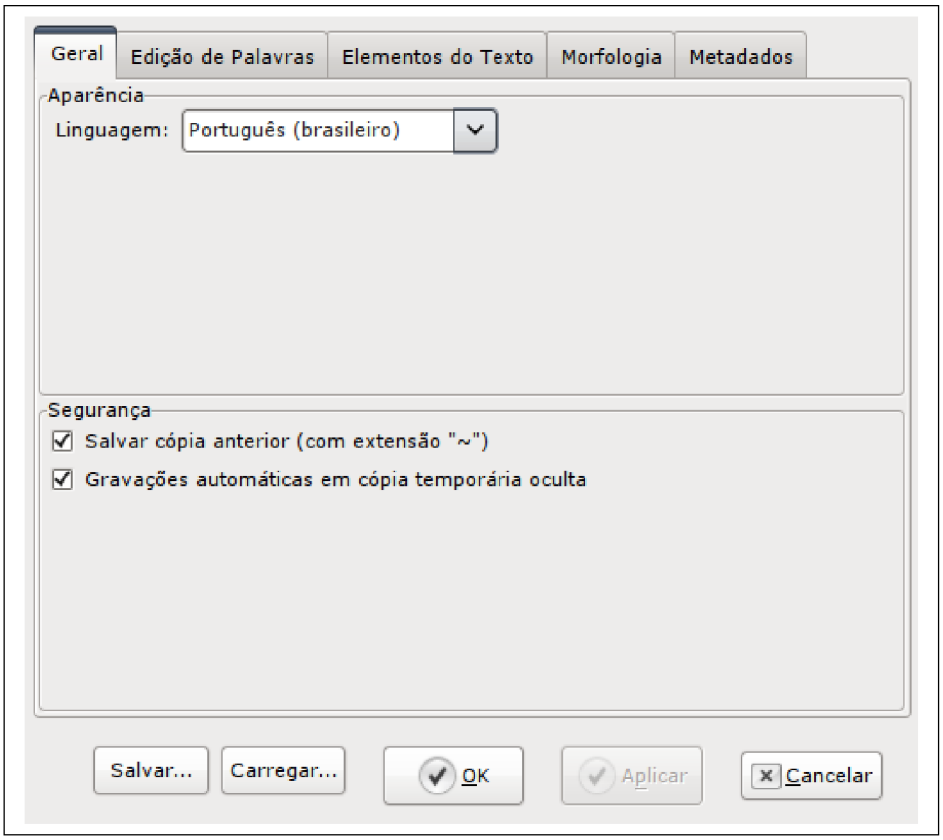

Figura 5. Janela de preferências da aplicação.

Como vermos, na aba 'Geral', o usuário pode configurar alguns aspectos gerais da aplicação, tais como opções de salvamento e o idioma da aplicação (disponível, até o momento, em inglês e português). O que importa ressaltar, no entanto, são as demais abas, que explicamos a seguir.

Edição de Palavras

Se as necessidades de codificação do usuário incluírem a edição do texto, ou seja, interferências sobre a grafia e a grafemática do texto original, é nesta aba que ele poderá configurar os níveis de edição sobre os símbolos (palavras e pontuações) de que precisa. Por padrão, o E-Dictor assume dois níveis básicos de edição, a saber, *junção* e *segmentação*. O primeiro permite a junção de segmentos de palavras eventualmente separadas no original (por quebra de página, por exemplo) e o segundo é utilizado para (des)segmentar uma palavra. Estes são os níveis mínimos necessários à ferramenta. A partir daí, os níveis são cadastrados pelo usuário, com base nas necessidades de edição de seu contexto.

Os níveis de edição devem ser cadastrados e ordenados, conforme a prioridade de aplicação. Este sistema foi pensado para um contexto em que um nível de edição não pode "conter" outro(s). Ou seja, o objetivo do E-Dictor é manter a informação sobre cada nível (inclusive o "nível" original). Assim, o usuário poderá, num segundo momento, acessar o texto em diferentes "versões", cada uma relativa a um dado nível de edição.

Para exemplificar, digamos que dois outros níveis sejam cadastrados: *expansão*, para expandir siglas e contrações, e *modernização*, para modernizar o léxico, de acordo com padrões atuais. Ao se deparar com o termo "Ilmo", dependendo da data em que foi escrito o texto-fonte, a edição de expansão seria "Ilustríssimo" (texto moderno) ou "Illustrissimo" (texto mais antigo). No primeiro caso, a modernização não seria necessária. No segundo caso, entretanto, há que se fazer a modernização para "Ilustríssimo". Ou seja, no caso do texto antigo, mantêm-se a informação sobre cada nível de edição, podendo-se ler o texto original apenas expandido ou expandido e modernizado.

Elementos do texto

Nesta aba é que o usuário pode "especializar" os elementos da estrutura para fins específicos, como já adiantado anteriormente. Esta aba permite criar subtipos para os elementos "Seção", "Parágrafo", "Sentença" e "Palavra". Assim, podemos obter tipos específicos a serem utilizados na codificação de diferentes gêneros textuais. A importância dessa flexibilidade é permitir mais possibilidades de manipulação do arquivo XML gerado, através do uso de outras ferramentas, como o XSLT, entre outros. Por exemplo, tendo acesso aos subtipos, pode-se desenvolver uma transformação XSLT que gere uma versão em HTML do texto, com layout especial, voltado para o gênero em questão. O próprio E-Dictor, em sua próxima versão *beta* (*build 002*) (ver seção 3.2), prevê a disponibilização de uma rotina de exportação para HTML que faça uso dos subtipos definidos (que na próxima versão poderão ser vinculados a informações de estilo, CSS[11]) para gerar uma apresentação mais interessante.

Além disso, dado que certos trechos de um texto podem não ser relevantes para análise morfossintática, nesta aba o usuário pode dizer ao E-Dictor para "ignorar" determinados subtipos de elementos, no momento da revisão da morfologia (aba "Morfologia"). Com isso, tais elementos não poderão ser acessados para inclusão/revisão de etiqueta, embora estejam visíveis.

Morfologia

Nesta aba, caso o usuário tenha interesse em vincular e/ou corrigir a vinculação de classes de palavras aos itens lexicais do texto, pode-se cadastrar o sistema de classes (ou etiquetas) previsto. Com este sistema definido, o usuário terá a lista de etiquetas disponível na aba "Morfologia", durante a inclusão e/ou revisão das mesmas.

Metadados

Outro aspecto importante nos contextos de construção de corporta de textos é a especificação de metadados sobre os documentos codificados, dados estes que vão desde informações sobre o texto-fonte (autor, editor, ano de publicação, etc.) até informações sobre o processamento eletrônico do texto (estado da edição, nomes dos editores que trabalharam no texto, etc.). Sabe-se que cada projeto tem seu próprio sistema de metadados, contemplando aquilo que lhe parece relevante. Portanto, o E-Dictor prevê o cadastramento de metadados estruturados da seguinte forma: *tipo do*

metadado e *campo de metadado*. Por exemplo, podemos cadastrar o seguinte sistema de metadados:

- Tipo: "Dados do Texto-fonte", contendo os campos de metadados "Título", "Ano de Publicação/Produção", "Gênero" e "Editor".
- Tipo: "Dados do Autor", contendo os campos "Nome", "Ano de Nascimento" e "País de origem".
- Tipo: "Dados da Codificação", contendo os campos "Estado da Edição", "Editores", "Data da última revisão" e "Data de criação".

Com um sistema definido, pode-se proceder à especificação destas informações (valores para os campos de metadados) através da janela de "Metadados", disponível após a criação de um novo documento XML.

2.5 Funcionalidades

Além das características já mencionadas nas sessões precedentes, o E-Dictor, em sua versão apresentada neste artigo, possui outras funcionalidades, não apenas para atender às necessidades da edição de textos, mas também para facilitar esta tarefa. Como estas funcionalidades estão disponíveis através do menu do aplicativo, vamos apresentá-las na ordem em que este está organizado.

Quanto às opções de arquivo, além das opções de abrir e salvar documentos (em TXT e em XML), o E-Dictor:

- Memoriza e oferece atalhos para abertura de arquivos editados recentemente (os 10 últimos);
- Dá a opção de reverter o estado da edição para o estado na última vez que o documento foi salvo (por exemplo, quando o usuário se arrepende de uma série de modificações feitas);
- Tenta importar o conteúdo de arquivos XML em outros formatos, mas sem nenhum tratamento especial. Assim, embora às vezes útil, esta rotina pode gerar um documento muito trabalhoso para ser corrigido, ou seja, às vezes transcrever novamente é mais eficaz.
- Importa um arquivo texto etiquetado (no padrão "palavra/ETIQUETA"), desde que este possa ser emparelhado com o XML, ou seja, é necessário que o texto etiquetado seja exatamente idêntico ao texto codificado.
- Exportar o texto codificado em três diferentes formatos: texto editado (nível máximo de edição), texto etiquetado (ou para ser submetido à análise morfológica) e texto para análise fonológica (que exporta as formas "por extenso" das palavras, quando houver).

Quanto às opções comuns em editores, o E-Dictor oferece opções para *desfazer* e *refazer* operações, opções para *colar*, *copiar* e *recortar*, e opções para *procurar* e *substituir* (em ambas as direções do texto). Quanto à exibição, o E-Dictor permite exibir ou esconder a barra de ferramentas, liberando mais espaço para a exibição e edição do texto.

Em relação à manipulação do documento, há opções para:

- Converter o texto transcrito para XML;
- Informar os valores dos campos de metadados para o documento;
- Editar propriedades do texto (título, ano de produção, autor, ano de nascimento e extensão do texto - parcial ou completo), bem como registrar comentários gerais sobre este (comentários de edição/codificação);
- Inserir cabeçalho ou rodapé, independentes, para cada página;
- Inserir número de página (no cabeçalho ou no rodapé);
- Registrar comentários para os elementos Seção, Parágrafo e Símbolo;
- Ligar e desligar o "modo de edição" de símbolos.

Quando se está no modo de edição de símbolos (para isto basta clicar em um), o E-Dictor habilita operações específicas para estes elementos. São elas:

- Inserir ou remover quebras estruturais de linha, coluna ou página;
- Marcar fim de parágrafo ou sentença;
- Aplicar edições;
- Deslocar o símbolo para frente ou para trás (por conta de erros na transcrição);
- Remover o símbolo;
- Transformar o símbolo em número de página (inserindo uma quebra de página naquele ponto do texto);
- Inserir texto (não estruturado[12]) antes ou depois do símbolo (para trechos que por alguma razão foram omitidos na transcrição).

Além destas, o usuário pode informar o idioma de trechos em língua estrangeira, bem como marcá-los com "negrito", "itálico" e "sublinhado". Em suma, estas são as funcionalidades presentes na versão *beta* (*build 001*) do E-Dictor. Na seção 3.2 comentamos outras funcionalidades que estão sendo pensadas para futuras versões, bem como revisões de funcionalidades atuais.

2.6 Edição

Para encerrar esta seção de apresentação da ferramenta, vamos comentar os detalhes da edição de símbolos (palavras e pontuação) do texto, que é a principal funcionalidade do E-Dictor. A Figura 6, a seguir, mostra a interface disponível (aba "Grafia") para esta tarefa:

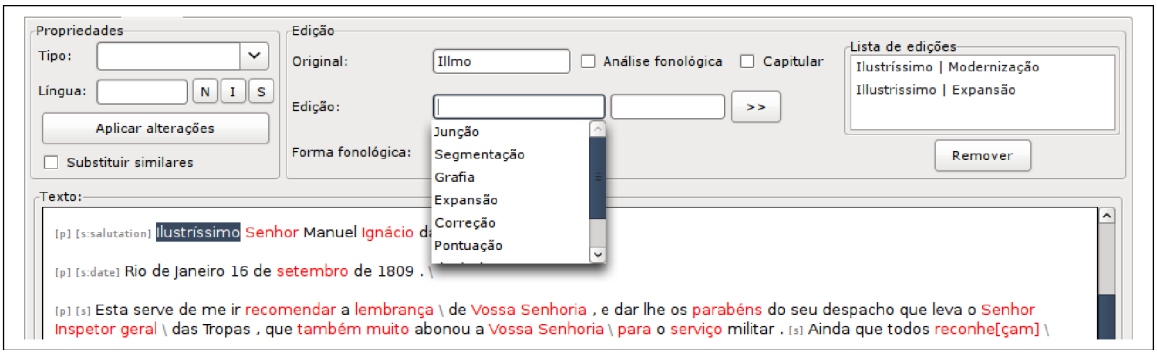

Figura 6. Detalhes da interface de edição de símbolos.

Note que a palavra "Ilustríssimo", ressaltada em fundo azul, é a palavra sendo editada na figura. É sobre ela que podemos fazer uma série de modificações, comentadas a seguir:

- Painel "Propriedades": aqui, podemos especificar o "Tipo" do símbolo (de acordo com as definições de subtipos previstas nas preferências da aplicação), a "Língua" ou idioma (se for estrangeiro), opções de formatação (negrito, itálico e sublinhado) e temos o botão "Aplicar alterações" com a opção para "Substituir similares", comentados mais à frente.
- Painel "Edição": aqui, podemos marcar algumas propriedades do símbolo, bem como inserir os níveis de edição (de acordo com os níveis informados nas preferências da aplicação). Vamos aos seus elementos:
    - O campo "Original" exibe a forma original do símbolo, como transcrita do texto-fonte. Normalmente, a forma original não deve ser alterada, mas se preciso, pode ser feito. Repare que o E-Dictor exibe sempre o nível máximo de edição na área de texto, não o texto original.
    - À frente deste campo, temos as propriedades "Análise fonológica" (que diz ao E-Dictor para exportar a forma original para análise fonológica) e "Capitular" (que informa que no texto original esta palavra inicia com capitular).
    - O campo "Edição" permite escolher um nível de edição, cujo conteúdo será especificado no campo imediatamente à frente. Após informar o conteúdo, é preciso clicar no botão ">>" para incluir o nível de edição na "Lista de Edições".
    - O campo "Forma fonológica": em alguns casos (como números, por exemplo), é preciso especificar a forma a ser exportada para análise fonológica, pois ela nem pode ser a original, nem pode ser a forma editada.
    - A "Lista de Edições": lista as edições incluídas para o símbolo, permitindo sua alteração ou exclusão, através do botão "Remover".
- Botão "Aplicar alterações": efetiva as edições feitas (neste painel e no painel "Edição"). Se o usuário mudar de palavra, antes de clicar neste botão, perderá todas as modificações feitas sobre o símbolo (se houver). Este botão possui a opção "Substituir similares", que repete a operação para os símbolos idênticos, no restante do texto (o que agiliza muito o trabalho de edição).

3. Considerações finais

A codificação dos textos antigos em editores comuns já havia tornado evidente a necessidade de uma ferramenta que favorecesse um processo mais eficiente e amigável de edição. A partir dos resultados verificados com o uso do E-Dictor, a importância de uma tal ferramenta já não nos deixa espaço para dúvidas. Entretanto, a ferramenta certamente ainda precisa de melhorias e avanços. As idéias nesse sentido virão a partir da difusão da mesma e de seu uso intenso. Nesta seção final, discutimos alguns planos já em consideração, e apontamos as perspectivas de sua aplicação.

3.1 Resultados preliminares

Os testes da ferramenta foram feitos principalmente no âmbito do CTB, através do trabalho de bolsistas que atuaram ou vêm atuando no projeto. Três aspectos são importantes, na análise dos resultados obtidos com o uso do E-Dictor: são eles o *treinamento*, *trabalho de edição* em si e o trabalho de *revisão*.

Como esperávamos, o uso da ferramenta tornou o processo de treinamento mais curto e simples tanto para quem treina, quanto para quem é treinado. Agora, o que um editor precisa aprender se restringe apenas ao conceito de edição de textos (o que são níveis de edição, o que deve ser codificado, como preencher os metadados, etc.) e ao uso da ferramenta E-Dictor. Em termos gerais, o editor nem precisa tomar conhecimento de que há uma estrutura em XML em jogo e, muito menos, compreender suas particularidades e como a linguagem XML funciona.

O trabalho de edição também registrou uma diminuição em torno de 50% no tempo necessário para codificar e editar um texto. Isto, numa medição razoavelmente informal. Acreditamos que esta queda seja ainda maior, visto que o trabalho de revisão também influencia neste tempo e este, por sua vez, também se tornou mais rápido, pois as possibilidades de erros de codificação (má-formação do XML e/ou uso incorreto das estruturas) foram eliminadas. Em relação à estrutura XML, o E-Dictor tem total controle do que é gerado. Os erros, agora, estão restritos apenas à má compreensão da atividade de transcrição e edição ou são resultantes de falta de atenção. De todo modo, estão restritos ao conteúdo do texto e não à estrutura XML.

Um dos fatores que acreditamos estar na origem desta melhora é o ganho de *legibilidade* que o E-Dictor promoveu. A partir do momento em que o código XML subjacente é omitido, pode-se ler o texto praticamente com a mesma desenvoltura com que se leria o texto sem a codificação. Para ilustrar este ponto, vejamos o texto em XML, como o editor/revisor o via, enquanto trabalhava diretamente sobre ele:

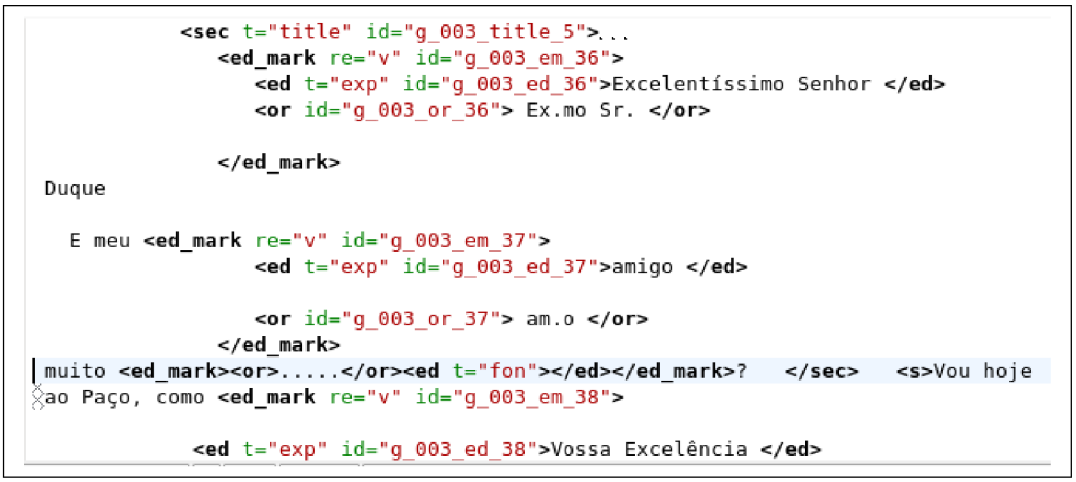

Figura 7. Trecho de codificação em XML.

Note que, em todo este trecho, o conteúdo original editado é apenas "Ex.mo Sr. Duque E meu am.o muito.....? Vou hoje ao Paço, como [V.Exa.]". As dificuldades (e possibilidades de erro/omissão) para o codificador/revisor, diante de material como este são enormes. Agora, não apenas o contato direto com o XML é evitado, como a codificação XML gerada pela ferramenta é mais bem-formada, como vemos a seguir:

Figura 8. Trecho de codificação XML gerada pelo E-Dictor.

3.2 Melhoramentos

Acreditamos que a utilização desta ferramenta em maior escala irá apontar importantes desenvolvimentos, com melhorias e revisões, a serem feitos. A diversidade de contextos de uso também irá contribuir bastante para o desenvolvimento da ferramenta em direção à flexibilidade de usos. Nossa experiência no contexto do CTB e do BRASILIANA já forneceu e continua fornecendo importantes *insights* sobre funcionalidades a acrescentar/melhorar. De fato, algumas já estão sendo desenvolvidas para a versão *beta (build 002)*. Entre elas, podemos citar:

- Novas rotinas de exportação, para substituir as atuais, que gerem versões do documento XML de dois tipos: texto e léxico de edições. A versão "texto" deverá permitir ao usuário escolher o nível máximo de edição a ser exportado, bem como outras opções para omissão/exibição de outras informações; já a versão "léxico de edições" deverá permitir exportar a lista de palavras editadas, com opções de ordenar alfabeticamente e de agrupar itens similares numa mesma linha. Todas as exportações deverão disponibilizar também a opção de exportar em formato texto (TXT) ou hipertexto (HTML).
- Vinculação de folha de estilo (CSS) a subtipos de elementos (nas preferências da aplicação). Estas informações deverão ser utilizadas para formatação da exportação em HTML.
- Pequenas correções/melhorias na interface (nomes de comandos, títulos de abas, etc.).
- Opção para aumentar ou diminuir a fonte de apresentação do texto, tanto na transcrição quanto na edição da grafia ou revisão de etiquetas. Com isto, o usuário pode definir um tamanho que lhe seja visualmente confortável.

Além destas modificações mais iminentes, outras estão previstas para médio e longo prazos, algumas prontamente factíveis, outras dependendo ainda de estudos de viabilidade:

- Dar algumas opções de marcação limitadas já na transcrição, para tornar o trabalho de edição mais eficiente. Por exemplo, permitir que sejam informadas quebras de página e comentários (sobre trechos ilegíveis, por exemplo), que possam ser transferidos automaticamente para a estrutura XML.
- Exportação para o formato PDF.
- Facilitar a correção de OCR, disponibilizando uma visualização do fac-simile, na aba de transcrição do texto.
- Permitir a vinculação entre imagens de OCR e páginas do documento. Esta vinculação poderia se dar em termos de atalhos para arquivos de imagem locais (na própria máquina em que se executa o E-Dictor) ou remotos (disponíveis na internet). Com base nesta funcionalidade, a exportação do documento para HTML/PDF poderia incluir a exibição da imagem.
- Acoplar um analisador morfológico que processe diretamente o documento XML (na atual versão, é preciso exportar o documento, fazer a análise e importar o resultado de volta).
- Desenvolver meios de extrair informações do documento XML (por meio de expressões regulares, etc.), na linha do que faz o aplicativo CLaRK (seção 1.2.1).
- Criação do elemento "chunk", para agrupar segmentos menores que uma sentença (alguns símbolos) ou segmentos menores que um parágrafo (algumas sentenças).
- Ampliar o alcance do E-Dictor para permitir a construção de corpora de textos paralelos (por exemplo, voltados para estudos de tradução).

Finalmente, uma meta de maior vulto, que temos amadurecido a partir de sugestões de

outros colegas, é a incorporação de um léxico de edições a serem sugeridas pelo E-Dictor durante a edição, ou, até, a serem aplicadas automaticamente pelo E-Dictor e depois corrigidas pelo editor (usuário), se necessário.

Há propostas de programas automáticos de reconhecimento da grafemática antiga e de grafias em variação que representam uma segunda solução possível para o tratamento computacional dos textos antigos. Nesse campo, destacam-se, para os textos antigos em português, as propostas de Aluísio (2007), aplicadas segundo a técnica de Candido Jr (2008) ao Corpus fundamental do Dicionário Histórico do Português do Brasil, DHPB (Biderman, 2005).

A intervenção editorial e o desenvolvimento de programas automáticos de reconhecimento da grafemática antiga e de grafias em variação são, a nosso ver, abordagens complementares, uma vez que o campo das vantagens de uma recobre o campo das desvantagens da outra. O método da intervenção editorial, nos moldes do E-Dictor, apresenta duas vantagens principais: a primeira é a flexibilidade dos formatos gerados (tanto formatos passíveis de leitura automática, com aproveitamento para programas de anotação morfológica e sintática, como formatos para leitura humana, especializada ou leiga – sendo a versão modernizada dos textos um sub-produto de interesse para um público mais amplo. A segunda, a garantia da qualidade filológica da edição, mediante seu uso por um editor especializado neste sentido. O sistema apresenta, entretanto, a desvantagem de demandar um investimento de tempo e recursos humanos relativamente grande. Em contraste, o ferramental para buscas com variação de grafia desenvolvido para o DHPB (Candido Jr, 2007) representa uma relativa economia de tempo e recursos humanos.

Neste caso, a principal desvantagem é a baixa precisão do sistema em textos mais complexos: a ferramenta não suporta variações mais idiossincráticas como, por exemplo, as que caracterizam os textos manuscritos. De fato, o sistema se fundamenta na aplicação em textos escanerizados (submetidos, previamente, à conferência humana), que são normalizados quanto às variações grafemáticas mais imprevisíveis. Nota-se, portanto, a complementariedade entre as duas técnicas: o ferramental automático de reconhecimento de variação apresenta a vantagem da escala, e a técnica de edição semi-automática, a vantagem da qualidade filológica.

No desenvolvimento futuro do E-Dictor, buscaremos caminhos para unir esses campos complementares da maneira mais vantajosa para os diferentes usuários.

3.3 Perspectivas

Como mencionamos mais acima, a nosso ver as perspectivas para a implementação das melhorias já planejadas, bem como para o surgimento de outros avanços, dependem fundamentalmente da difusão e a intensificação do uso do E-Dictor. Assim, será importante aqui apresentarmos resumidamente - para finalizar - o cenário que se abre nesse sentido.

A primeira frente de experimentação e desenvolvimento em torno do E-Dictor é seu ambiente de concepção e uso originais – a construção do CTB, hoje um dos mais conhecidos corpus históricos da língua portuguesa. A essa frente se segue hoje o projeto mais recente e experimental junto à Brasiliana USP - nesse contexto, frente aos resultados obtidos com o reconhecimento automático de caracteres já citados no início do artigo, estamos dando início a um projeto que pretende realizar experimentos

dirigidos ao aumento das taxas de acerto de OCR em impressos dos séculos XVI e XVII (Paixão de Souza, 2009b), com o auxílio do E-Dictor. A ferramenta servirá, a um tempo, como instância de controle, e como instância para a revisão e correção dos resultados finais da manipulação por OCR. Nesse sentido, pensamos que um aspecto importante da ferramenta - a interação entre o modo de transcrição e o modo de edição - poderá ser melhorado (o que, ainda, aproximaria a concretização de nossa contribuição junto ao grupo temático da ICLA). Por fim, a partir do final do ano de 2009, surgiu um cenário de colaboração sistemática inédita entre o grupo de construção do CTB e o grupo experimental da Brasiliana e diferentes sub-grupos do projeto PHPB (Para uma história do Português no Brasil), interessados em experimentar o sistema de edições eletrônicas aqui apresentado (cf. Galves et al 2009). Tendo em vista a larga experiência desse grupo de pesquisas em torno do trabalho de edição filológica de documentos manuscritos, nacionalmente reconhecida, parece-nos que as perspectivas de avanço do uso do E-Dictor na edição filológica de elevada qualidade se tornam mais próximas.

De fato: a meta ideal para o E-Dictor é a de ser capaz de abarcar todo o fluxo de atividades lingüísticas e filológicas sobre um texto qualquer: a transcrição, edição, análise morfossintática e sintática. Esperamos que isso se torne factível, em um futuro próximo, graças à contribuição dos grupos de pesquisa dedicados à construção de corpora de textos antigos que já experimentaram a ferramenta, e dos que vierem a experimentá-la, em um ambiente colaborativo que favoreça avanços dirigidos à flexibilidade e ampliação de uso.

4. Referências Bibliográficas

BBD. Brasiliana Digital. <www.brasiliana.usp.br>

BIRD, Steven, E. Klein & E. Loper (2009). Natural Language Processing with Python. China: O'Reilly.

BRITTO, H. & FINGER, M. (1999). Constructing a Parsed Corpus of Historical Portuguese. <http://www.ime.usp.br/~tycho/participants/britto/britto_finger.htm>

CASTILHO, Ataliba Teixeira de (1998) “Para a história do português brasileiro”. São Paulo:Humanitas. Vol I: Primeiras idéias.

CTB. Corpus Anotado do Português Tycho Brahe. <www.tycho.iel.unicamp.br/~tycho/corpus>

FINGER, M. (2000) . Técnicas de otimização da precisão empregadas no etiquetador Tycho Brahe. <http://www.ime.usp.br/~tycho/participants/finger/propor2000.pdf>

KATO, Mary A. & ROBERTS, Ian. (orgs.) (1993) “Português brasileiro: uma viagem Diacrônica”. Campinas: Editora da Unicamp.

ICLA (2009). International Comparative Literature Association. Research Committee on Comparative Literature in the Digital Age: Protección del patrimonio literario através de formatos digitales . http://ailc-icla.org/?q=node/5

MATTOS E SILVA, Rosa Virgínia. (1988) Fluxo e refluxo: uma retrospectiva da lingüística histórica no Brasil. D.E.L.T.A., 4.1: 85-113. São Paulo.

MEGALE, Heitor & CAMBRAIA, César Nardelli (1999). Filologia Portuguesa no Brasil. D.E.L.T.A, vol. 15, número especial:1:22. São Paulo.

PAIXÃO DE SOUSA, M. C. (2009a). Desafios do processamento de textos antigos: primeiros experimentos na Brasiliana Digital. I Workshop de Linguística Computacional da USP. São paulo, novembro de 2009.

PAIXÃO DE SOUSA(2009b). "Edições Filológicas na Brasiliana Digital". Projeto de pesquisa. Programa Ensinar com Pesquisa, Pró-reitoria de Graduação, Universidade de São Paulo.
<http://lampiao.brasiliana.usp.br/lingua/sites/default/files/projeto_ensinar_com_pesquisa_mcpsousa.pdf>

PAIXÃO DE SOUSA, M.C. (2007). Sistema de Edições Eletrônicas do Corpus Tycho Brahe: Fundamentos, Diretrizes e Procedimentos <http://www.ime.usp.br/~tycho/corpus/manual/prep/index.html>.

PAIXÃO DE SOUSA, M.C. (2006). Hypertext: concepual and methodological frontiers. Comunicação ao Seminário Internacional Literaturas: del Texto al Hipertexto. Faculdade de Filología, Universidade Complutense de Madrid. Madri, 22 de Setembro, 2006.

PAIXÃO DE SOUSA, Maria Clara (2005). Memórias do Texto. Revista Texto Digital, ISSN 1807-9288, ano 2 n.1 2006. <http://www.textodigital.ufsc.br/num02/paixao.htm>

PAIXÃO DE SOUSA, M.C. (2004). Memórias do Texto: Aspectos tecnológicos na construção de um corpus histórico do português. Projeto de pós-doutorado. Unicamp - Fapesp. <http://www.ime.usp.br/~tycho/participants/psousa/memorias/index.html>

TRIPPEL, T. & PAIXÃO DE SOUSA, M. C. (2006). “Metadata and XML standards at work: a corpus repository of Historical Portuguese texts”. Papers from the V International Conference on Language Resources and Evaluation (LREC 2006).

W3C (2009). “Extensible Markup Language”. <http://www.w3.org/XML>

W3C (1999). "Extensible Stylesheet Language Transformation". <http://www.w3.org/TR/xslt>

[1] Tradução livre da apresentação do projeto na internet. URL: http://aune.lpl.univ-aix.fr/projects/multext/. Acessada em 20/01/2010.
[2] Disponível na internet. URL: http://www.ilc.pi.cnr.it/EAGLES/home.html. Acessada em 21/01/2010.
[3] Disponível na internet. URL: http://www-tei.uic.edu/orgs/tei/. Acessada em 21/01/2010.
[4] Disponível na internet. URL: http://www.bultreebank.org/clark/. Acessada em 20/01/2010.
[5] Disponíveis no internet, respectivamente em http://www.gnu.org/software/emacs/, http://www.editplus.com/ e http://kate-editor.org/. Todas as URLs acessadas em 21/10/2010.
[6] As versões beta, assim como a versão que a precedeu (alpha), são versões de testes, consideradas instáveis e ainda sujeitas a erros.
[7] Disponível na internet. URL: http://www.python.org/. Acessada em 21/01/2010.
[8] Disponível na internet. URL: http://www.nltk.org/. Acessada em 21/01/2010.
[9] Disponível na internet. URL: http://www.eclipse.org/. Acessada em 21/01/2010.
[10] Disponível na internet. URL: http://trac.edgewall.org/. Acessada em 21/01/2010.
[11] Disponível na internet. URL: http://www.webstyles-portuguese.info/Style/CSS/. Acessada em 21/01/2010.
[12] O trecho é inserido como parte da sentença a que pertence o símbolo em questão. Portanto, se o trecho for longo, contendo parágrafos e sentenças, sua estrutura deve ser editada após sua inclusão, através das opções de formatação do E-Dictor.