Universidade Federal de Santa Catarina
Departamento de Informática e Estatística

IMPLEMENTAÇÃO DE UM SERVIDOR DE MEMÓRIAS DE TRADUÇÃO

Lucas Augusto Deters

trabalho de conclusão do
Curso de Bacharel em Sistemas de Informação

orientador:
José Eduardo De Lucca

Florianópolis, Dezembro de 2008

Lucas Augusto Deters

IMPLEMENTAÇÃO DE UM SERVIDOR DE MEMÓRIAS DE TRADUÇÃO

Trabalho de conclusão de curso
apresentado como parte das atividades
para obtenção do título de Bacharel em
Sistemas de Informação pela
Universidade Federal de Santa Catarina.

professor orientador:
José Eduardo De Lucca

Florianópolis, 2008

Autoria: Lucas Augusto Deters

Título: Implementação de um Servidor de Memórias de Tradução

Trabalho de conclusão de curso apresentado como parte das atividades para obtenção do título de Bacharel em Sistemas de Informação pela Universidade Federal de Santa Catarina.

Os componentes da banca de avaliação, abaixo listados, consideram este trabalho aprovado.

	Nome	Titulação	Assinatura	Instituição
1				
2				
3				

Data da Aprovação: _____ de _____ de _____.

Dedico este trabalho
à minha família.

AGRADECIMENTOS

À comunidade de Software Livre por seus ideais e pela inestimável contribuição de seu trabalho para a humanidade.

Estarei eu falando de coisas irrealizáveis?

Alberto Santos Dumont

RESUMO

O presente trabalho de conclusão de curso visa o desenvolvimento de um servidor de memórias de traduções, dando ênfase à resolução do problema de armazenamento de grandes bases de dados, e à recuperação eficiente utilizando busca aproximada. Durante seu desenvolvimento, é proposto o uso de uma ferramenta de busca full text acoplada a um banco de dados tradicional como forma de possibilitar consultas eficientes sobre as memórias de tradução. Também é desenvolvido um Web Service que possibilita a utilização deste servidor a partir da internet. O desenvolvimento deste servidor vem de encontro com a ausência de soluções robustas de servidores de memória de tradução baseados em software livre, contribuindo desta forma para sanar esta deficiência da área.

Palavras-Chave: Localização de software; Memórias de Tradução; Servidor de Memórias de Tradução; Busca Full Text.

ABSTRACT

This study aims at the development of a remote translation memories with the emphasis on solving the problem of storing large databases and efficient recovery of these when using fuzzy search. During its development, it proposes an architecture based on the use of a full text search engine coupled to a database, and making use of web services to allow customers to use this server through the Internet. The development of this server is aimed to help minimize the absence of robust solutions for servers translation memory based on free software, contributing to minimize the deficiences in the area.

Keywords: Software Localization; Translation Memory; Translation Memory Server; Full Text Search.

LISTA DE FIGURAS

LISTA DE SIGLAS E ABREVIATURAS

CAT	Computer Aided Tools
CPU	Central Processing Unit
ETL	Extract, Transform, and Load
FOLT	Forum Open Language Tools
FTS	Full Text Search
GPL	GNU General Public License
ISO	International Organization for Standardization
JDBC	Java Database Connectivity
LISA	Localisation Industry Standards Association
LiSoG	Linux Solutions Group
MT	Memória de Tradução
ODBC	Open Database Connectivity
ONU	Organização das Nações Unidas
PL/pgSQL	Procedural Language/PostgreSQL Structured Query Language
RFC	Request for Comments
SGML	Standard Generalized Markup Language
TMOSS	Translation Machine Open Source Software
TM Server	Translation Memory Server

TMS	Translation Memory Server
TM	Translation Memory
TMX	Translation Memory Exchange
TTX	TRADOS TagEditor TRADOStag XML
UTF-8	8-bit UCS/Unicode Transformation Format
W3C	World Wide Web Consortium
XLIFF	XML Localization Interchange File Format
XML	eXtensible Markup Language

SUMÁRIO

1 Introdução

1.1 Tema

A ampliação do mercado de software ocorrido em paralelo à disseminação dos computadores pessoais ocorrido na década de 80 exigiu das empresas criadoras de software um novo patamar de velocidade e qualidade de seus produtos. A internacionalização do software passou a ser um fator crítico de aceitação, e ponto chave para o domínio de novos mercados.

O modelo de tradução de software tal qual era realizado na época, até então realizado por departamentos de tradução dentro da própria empresa, se mostrou ineficiente para atender às novas demandas, e as produtoras de softwares foram obrigadas a buscar um novo modelo de internacionalização para seus softwares. Neste contexto, surgiram as primeiras empresas especializadas em localização de software. (ALFARO, 1998)

A crescente complexidade dos projetos de software e a velocidade no desenvolvimento de novas versões obrigou esta indústria especializada a desenvolver ferramentas de auxílio à tradução que permitissem o aumento da produtividade e a manutenção da uniformidade de produção entre as dezenas de tradutores envolvidos em um projeto de grande porte, e para manter as traduções entre as várias versões de um mesmo produto. Foi neste contexto de redução de prazos e melhoria da qualidade que surgiram diversas ferramentas de auxílio à tradução. (RIECHE, 2004)

Hoje a tecnologia de tradução baseia-se principalmente em duas tecnologias

(JUNIOR , 2004):

- tradução por máquina: também conhecida como tradução automática, em seu princípio sem a necessidade de auxílio humano.
- tradução assistida por computador: tem como objetivo auxiliar o tradutor humano em seu trabalho, facilitando e agilizando o processo.

As expectativas sobre a tradução por máquina sempre foram muito altas, porém logo se percebeu que os bancos de dados de lógica aplicados eram insuficientes para armazenar tamanha quantidade de informação referente ao sentido das frases e ocorreu um certo descrédito na área de lingüística computacional. A tradução por máquina ainda não atingiu uma maturidade suficiente para não precisar de auxílio humano. (PRUDÊNCIO, 2006)

Já a tradução assistida por computador vem ganhando cada vez mais importância nos processos de tradução, uma vez que esta representa uma redução significativa de esforços, e contribui para diminuir o tempo gasto nas traduções.

Memória de Tradução é uma técnica que consiste em armazenar as traduções de sentenças já realizadas, para que estas traduções sejam reaproveitadas totalmente ou parcialmente quando uma sentença semelhante vier a ser traduzida. Cabe ao tradutor aceitar uma das traduções sugeridas, ou mesmo simplesmente corrigir a tradução sugerida.

O principal componente de um sistema de tradução é seu repositório de memórias de tradução. Ele é o responsável por armazenar cada segmento de texto, juntamente com sua tradução, e possibilitar que as ferramentas de tradução tenham acesso a essas traduções. Quando o tradutor utiliza uma ferramenta com suporte a memórias de tradução, a ferramenta se conecta a este servidor e solicita a pesquisa de unidades de traduções semelhantes ao segmento de texto que está atualmente sendo editado pelo usuário. A ferramenta deve ser capaz de localizar segmentos iguais ou similares,

retornando os mais parecidos, que serão apresentados para o usuário.

O presente trabalho de conclusão de curso visa o desenvolvimento de um servidor de memórias de traduções, dando ênfase à resolução do problema de armazenamento de grandes bases de dado, e à recuperação eficiente utilizando algoritmos de busca aproximada.

No capítulo 2 é apresentada uma revisão bibliográfica dos conceitos envolvidos na construção do protótipo. O capítulo 3 apresenta um breve levantamento das propostas de servidores de código aberto existentes. Os capítulos 4 e 5 tratam da especificação e do desenvolvimento do protótipo deste servidor de memórias de tradução, apresentando também os resultados obtidos.

1.2 Delimitação do Tema

O desenvolvimento deste servidor deve envolver várias atividades, que vão desde a concepção de uma base de dados para o armazenamento das memórias de tradução, escolha de ferramentas e técnicas de indexação e de busca, utilização de bancos de dados distribuídos, definição de uma metodologia para ranking dos melhores resultados, até a disponibilização das funcionalidades na forma de serviços.

Não é a intenção deste trabalho desenvolver um sistema de memórias de tradução completo, com interface de edição, nem tampouco o desenvolvimento de outros componentes especializados, ou de gerência do processo de tradução.

O foco deste trabalho será dado na questão do armazenamento e recuperação eficientes de memórias de tradução, bem como na disponibilização destas na forma de serviços.

1.3 Objetivo Geral

Implementação de um protótipo funcional de um servidor de memória de tradução, que poderá ser acessado remotamente utilizando web services.

1.4 Objetivos Específicos

- Compreensão dos mecanismos envolvidos em um servidor de memória de tradução;
- Estudo de técnicas de armazenamento, indexação e busca *full-text;*
- Estudo de ferramentas que implementem busca *full-text;*
- Prototipação de um sistema de busca para um servidor de memórias de tradução;
- Disponibilização de buscas sobre este sistema, na forma de serviços.

1.5 Motivação

Quando usadas localmente, as memórias de tradução tornam-se ferramentas especialmente úteis para os tradutores. Os tradutores constróem suas próprias bases de memórias de tradução e se beneficiam destas em novos projetos de tradução.

Muito embora esta forma de utilização de memória de tradução já represente um ganho significativo em desempenho, este ganho poderia ser otimizado se houvesse a possibilidade dos tradutores consultarem mais de uma base de memórias de tradução, ou se as memórias de tradução fossem trabalhadas na forma de serviços, nos quais os repositórios de memórias de traduções pudessem ser consultados, sem a necessidade de envio de grandes arquivos contendo estas memórias.

A grande maioria dos provedores de serviços de tradução se vale de tradutores autônomos trabalhando de forma remota. Para estas empresas, que trabalham com prazos muito curtos, seria interessante que as memórias de tradução da empresa estivessem disponíveis de forma automática para seus tradutores, sem a necessidade de transmissão de grandes memórias de tradução através da Internet. As memórias de tradução atingiram tal patamar de importância que muitas empresas já as tratam como um ativo intelectual da empresa e desejam proteger este patrimônio centralizando suas bases de tradução.

Além das vantagens para empresas tradicionais, a disponibilização de memórias de tradução sob a forma de serviços também pode viabilizar a criação de grandes bases públicas de tradução, seguindo tendências de colaboração recentemente incorporadas à nossa sociedade, tais como os wikis e os softwares livres.

Sistemas de Memória de Tradução foram desenvolvidos para uso comercial por mais de 20 anos. Seu uso é extremamente difundido nas empresas de localização. Conforme ALFARO (1998), o mercado de tradução é dominado por um pequeno número de fornecedores de softwares comerciais. Apesar dos avanços ocorridos nos últimos anos, estes softwares ainda são em sua maioria incompatíveis entre si, e não permitem que dados sejam facilmente intercambiáveis. Isso implica que o processo de criação de documentos em várias línguas acaba envolvendo a utilização de um grande número de ferramentas, o que resulta em tradutores sendo selecionados mais devido à sua experiência com determinadas ferramentas do que devido à seu domínio em uma área. Além disso, o custo das soluções comerciais existentes é desproporcionalmente alto devido ao seu domínio de mercado. (FOLT, 2008)

No âmbito de software livre, há algumas opções de softwares de auxílio à tradução, porém a maioria possui implementações bastante limitadas de memórias de tradução, que permitem apenas a consulta de memórias de tradução locais.

1.6 Problema

O grande desafio deste projeto é estruturar um sistema de busca eficiente, capaz de realizar pesquisas textuais não exatas em grandes bases de dados multilíngües. Ao contrário dos dicionários, que armazenam traduções de palavras isoladas, as memórias de tradução devem armazenar traduções para sentenças completas (uma sentença pode ser uma frase completa ou um trecho da mesma). Cada sentença deverá ser armazenada individualmente, juntamente com sua tradução, o que torna o problema do armazenamento e recuperação de dados bem mais complexo que a simples pesquisa por um termo isolado num dicionário.

Uma parte da complexidade é a implementação de uma técnica de busca capaz de localizar sentenças contendo forma e conteúdo semelhante, bem como classificar essas sentenças conforme um ranking de similaridade destas em relação à sentença que está sendo pesquisada.

Alguns dos principais desafios técnicos:

- os algoritmos relacionados à busca full-text, tais como a quebra de sentenças em *tokens*, o *stemming* e as *stopwords* são exclusivos para cada língua; não existe ainda uma disponibilidade farta de algoritmos publicados. Além disso, as ferramentas full-text devem ser flexíveis o suficiente para que estes algoritmos possam ser utilizados durante a criação dos índices;
- um servidor de memórias de tradução estabelecido em uma arquitetura centralizada tende a acumular um volume muito grande de dados, sendo necessário garantir um bom tempo de resposta;
- Como realizar um ranking de similaridade em grandes bases sem perder performance? qual deve ser o parâmetro para medir a similaridade entre duas sentenças?

- Como disponibilizar os serviços e garantir o acesso concorrente ao servidor?

1.7 Metodologia

A metodologia adotada para este projeto é baseada no desenvolvimento de um protótipo, sendo composta pelas seguintes etapas:

1. Pesquisa, revisão e referêncial bibliográfico;

2. Formulação de um modelo conceitual;

2.1. Descrição do ambiente da aplicação / Especificação de requisitos;

2.2. Identificação e definição de componentes, funções do sistema e modelagem;

3. Desenvolvimento do protótipo;

4. Testes para verificação e validação do protótipo;

5. Conclusão da documentação.

2 Revisão Bibliográfica

2.1 Ferramentas computacionais de auxílio à tradução

Com o advento da computação, a tradução de textos deixou de ser uma tarefa puramente artesanal. A computação passou a ser vista como uma forma de baratear custos, melhorar a qualidade e reduzir prazos.

Com a invenção do computador, percebeu-se o potencial que a máquina tinha para automatizar este processo. A princípio, os estudos baseavam-se na tradução de palavras isoladas e posterior interpretação e finalização manual da tradução, em um processo significativamente lento.

Hoje a tecnologia de tradução baseia-se principalmente em duas tecnologias (JUNIOR, 2004):

- tradução por máquina: também conhecida como tradução automática, em seu princípio sem a necessidade de auxílio humano.
- tradução assistida por computador: tem como objetivo auxiliar o tradutor humano em seu trabalho, facilitando e agilizando o processo.

As expectativas sobre a tradução por máquina sempre foram muito altas, porém logo se percebeu que os bancos de dados de lógica aplicados eram insuficientes para armazenar tamanha quantidade de informação referente ao sentido das frases e ocorreu um certo descrédito na área de lingüística computacional. Conforme

ALFARO (1998), a tradução por máquina ainda não atingiu uma maturidade suficiente para não precisar de auxílio humano.

Já a tradução assistida por computador vem ganhando cada vez mais importância nos processos de tradução, uma vez que esta representa uma redução significativa de esforços, contribuindo para diminuir o tempo gasto nas traduções.

2.2 Memórias de Tradução

As memórias de tradução fazem parte deste segundo grupo, e têm como fundamento uma premissa básica: sentenças semelhantes possuem traduções semelhantes. As memórias de tradução tentam reutilizar as traduções realizadas pelo tradutor sempre que uma frase ou trecho semelhante já tenha sido traduzido. Cabe ao tradutor aceitar uma das traduções sugeridas ou mesmo corrigir a tradução sugerida.

As memórias de tradução são ideais para aplicação em textos técnicos, em especial para manuais e interfaces de software. Não é à toa que a indústria de localização de software impulsionou o uso das ferramentas de memória de tradução. Alguns fatores que contribuíram para esta aceitação das memórias de tradução na indústria de localização são:

- interfaces de software normalmente seguem um mesmo padrão, possibilitando o reaproveitamento de traduções mesmo entre softwares distintos;
- novas versões de software são publicadas em ciclos muito pequenos – com as memórias de tradução, as mensagens que se mantiverem inalteradas entre duas versões de um mesmo software ou de um manual não precisarão ser novamente traduzidas, o que proporciona uma economia de tempo e dinheiro;

A maioria das ferramentas de auxílio à tradução já possui embutido alguma forma de suporte a memórias de tradução, quase sempre se limitando a buscar

trechos de texto em pequenas memórias de traduções disponíveis na forma de arquivos. Alguns sistemas recuperam apenas correspondências exatas no banco de dados, enquanto outros mais sofisticados são capazes de recuperar segmentos similares aos fornecidos, e que são apresentados para o tradutor com as diferenças em destaque. Desta forma, frases nas quais apenas algumas palavras forem modificadas não precisam ser retraduzidas do zero; o tradutor utiliza a tradução sugerida a partir da memória de tradução, realizando as adequações necessárias.

Além do uso em conjunto com ferramentas de auxílio à tradução, as memórias de tradução também são freqüentemente utilizadas em conjunto com sistemas de gerenciamento de terminologias, dicionários multilíngües, bem como para auxiliar na tradução automática.

2.3 Memórias de tradução distribuídas

Dentre as ferramentas proprietárias, muitas ainda se utilizam de formatos proprietários para armazenar e transferir memórias de tradução, o que dificulta muito o intercâmbio de memórias de tradução entre tradutores.

A utilização de memórias de tradução distribuídas permite que diversos tradutores trabalhando em um mesmo projeto possam aumentar muito a consistência através do acesso instantâneo ao trabalho dos demais (O'BRIEN apud GOW, 2003). Para obter resultados similares sem as memórias de tradução, seria necessário muito trabalho de pós-edição das traduções.

Um sistema de memórias de tradução distribuída pode seguir duas arquiteturas (SIMÕES, ALMEIDA & GUINOVART, 2004):

- arquitetura cliente/servidor: As memórias de tradução ficam armazenadas em um servidor central. Isso possibilita que entidades disponibilizem suas

memórias de tradução de forma simples, sem a necessidade do envio de grandes arquivos através da internet.

- arquitetura peer-to-peer: possibilita que tradutores colaborem dinamicamente, sem a necessidade de haver um servidor centralizado. Esta arquitetura é especialmente útil em departamentos de tradução, com tradutores compartilhando automaticamente as traduções que estão realizando.

Em ambas as arquiteturas, se faz necessário um sistema eficiente de armazenamento e recuperação das memórias de tradução.

2.4 Locale

Conforme RIECHE et al. (2004), Locale é um local que reúne características culturais e lingüísticas específicas. Anteriormente entendido como local geográfico, o sentido assumiu um sentido técnico, em que representa uma combinação específica de idioma, região e conjunto de caracteres. Um exemplo clássico é o francês do Canadá, que constitui um 'locale' diferente do francês da França.

2.5 Tags para identificação de línguas ou locales

Em sistemas de informação é freqüente a necessidade de identificar línguas/locales. Como forma de padronização, a identificação de línguas devem ser realizada através de uma tag, conforme a sintaxe em ABNF definida pela RFC4646 em [1]:

```
Language-Tag  = langtag
                / privateuse            ; private use tag
                / grandfathered         ; grandfathered registrations

   langtag       = (language
                    ["-" script]
```

1 http://tools.ietf.org/html/rfc4646

```
                  ["-" region]
                  *("-" variant)
                  *("-" extension)
                  ["-" privateuse])

language      = (2*3ALPHA [ extlang ]) ; shortest ISO 639 code
              / 4ALPHA                 ; reserved for future use
              / 5*8ALPHA               ; registered language subtag

extlang       = *3("-" 3ALPHA)         ; reserved for future use

script        = 4ALPHA                 ; ISO 15924 code

region        = 2ALPHA                 ; ISO 3166 code
              / 3DIGIT                 ; UN M.49 code

variant       = 5*8alphanum            ; registered variants
              / (DIGIT 3alphanum)

extension     = singleton 1*("-" (2*8alphanum))

singleton     = %x41-57 / %x59-5A / %x61-77 / %x79-7A / DIGIT
              ; "a"-"w" / "y"-"z" / "A"-"W" / "Y"-"Z" / "0"-"9"
              ; Single letters: x/X is reserved for private use

privateuse    = ("x"/"X") 1*("-" (1*8alphanum))

grandfathered = 1*3ALPHA 1*2("-" (2*8alphanum))
                ; grandfathered registration
                ; Note: i is the only singleton
                ; that starts a grandfathered tag

alphanum      = (ALPHA / DIGIT)       ; letters and numbers
```

Uma tag de linguagem (Language-tag) é uma seqüência normalmente se constituída de:

- identificação da língua (obrigatório): Os códigos para identificação de línguas são definidos pelas normas ISO 639-1 e ISO 639-2. Esta relação é mantida pela Library of Congress, entidade responsável pelo seu registro. Uma relação atualizada de códigos de línguas está disponível em [2];
- identificação do formato de escrita (opcional): A identificação de formatos de escrita é normatizada pela ISO 15924 e sua manutenção está a encargo do Unicode Consortium. A tabela atualizada pode ser obtida através do endereço

2 http://www.loc.gov/standards/iso639-2/php/English_list.php

[3];

- identificação da região (opcional): Os códigos para identificação de regiões são normatizados pela ISO 3166 e através do Padrão de Códigos de Áreas UN M.49. A lista de regiões da norma ISO 3166 é mantida pela ISO 3166 Maintenance Agency e está disponível através do endereço [4]. A tabela de códigos de áreas do padrão UN M.49 é mantida pela ONU e está disponível em [5];
- identificação da variante da língua (opcional);
- identificação de uma extensão (opcional);
- identificação de uso privado (opcional).

Os exemplos a seguir foram retirados da especificação da RFC 4646 e ilustram algumas tags de identificação de linguagem válidas:

- de (Alemão)
- fr (Francês)
- ja (Japonês)
- sr-Cyrl (Sérbio escrito com formato de escrita Cirílico)
- sr-Latn (Sérbio escrito usando o formato de escrita Latino)
- zh-Hant (Chinês escrito usando o formato de escrita Chinês Trandicional)
- zh-Hans (Chinês escrito usando o formato de escrita Chinês Simplificado)
- zh-Hans-CN (Chinês escrito usando o formato de escrita Chinês Simplificado da China continental).
- Sr-Latn-CS (Sérbio escrito usando o formato de escrita Latino como usado na Sérbia e Montenegro).

3 http://unicode.org/iso15924/iso15924-codes.html
4 http://www.iso.org/iso/country_codes/iso_3166_code_lists.htm
5 http://unstats.un.org/unsd/methods/m49/m49.htm

- Sl-IT-nedis (Eslovênio como usado na Itália, dialeto Nadiza)

A RFC 4646 estabelece ainda que todas as subtags presentes no Tag de Identificação de Linguagens devem ser escritas em caixa baixa, exceto o código que identifica o formato de escrita, cuja primeira letra deverá estar em caixa alta e as demais em caixa baixa, e o código de identificação da região, que deve estar totalmente em caixa alta. Estas regras se espelham no uso do caso nos padrões subjacentes.

Não existe nenhuma autoridade responsável pelo registro e manutenção de tags de identificação de linguagens. Na prática, qualquer tag que siga as regras definidas por esta regulamentação será válido.

2.6 Segmentos e Unidades de Tradução

As traduções são armazenadas no banco de dados sob a forma de segmentos. Um segmento normalmente corresponde à uma sentença ou a um parágrafo em uma língua.

Um segmento é composto por (FOLT, 2008):

- Conteúdo (essencialmente texto)
- Identificação da Língua
- Metadados (dados de criação/modificação, comentários, etc)

A associação entre segmentos equivalentes, cada qual em uma língua, forma uma unidades de tradução.

2.7 Documento

No contexto de tradução, um documento é qualquer coleção completa e coerente

de textos traduzíveis combinados em um arquivo. (FOLT, 2008)

2.8 Match

Um match é um resultado de uma busca aproximada na memória de tradução.

O nível de similaridade em buscas fuzzy é chamado match quality. A similaridade geralmente é medida através da distância de edição de Levenshtein, que se baseia na quantia de substituições, inserções e deleções necessários para transformar uma frase em outra (RAYA, 2004). Por exemplo, para transformar a sentença "computador" na sentença "computação", será necessário manipular 3 dos 10 caracteres originais, o que corresponde a uma similaridade de 70% entre as duas sentenças. Algumas ferramentas possuem métodos próprios para definir a similaridade entre duas sentenças.

2.9 Busca Full Text

Full Text Search é um termo usado para se referir às tecnologias que permitem recuperação eficiente de documentos relevantes de acordo com uma requisição de busca. Percorrer todos os documentos de uma base para verificar se estes correspondem à requisição não é viável para grandes conjuntos de dados, e como conseqüência métodos mais eficientes se tornam necessários. (WITTEN, MOFFAT & BELL, 1999)

Quando o volume de dados a buscar é potencialmente grande ou a quantia de requisições de busca a executar é substancial, o problema da busca full text é dividido em duas tarefas: indexação e busca.

A principal técnica de indexação utilizada nos atuais sistemas de FTS é o índice

invertido, uma vez que este é o melhor método de indexação de grandes coleções de texto. (ZOBEL, MOFFAT & RAMAMOHANARAO, 1998)

2.10 Índices Invertidos

O nome "índice invertido" deriva de seu modo de funcionamento: documentos normalmente são armazenados como listas de palavras. Entretanto, índices invertidos invertem essa lógica porque armazenam, para cada palavra, a lista dos documentos em que a palavra ocorre. (Salerma, 2006)

Há diversas variações de índices invertidos. Implementações mais básicas simplesmente armazenam a lista de documentos em que cada palavra ocorre. O suporte a buscas mais avançadas, como busca por proximidade e busca de frases (que considera a ordem das palavras no texto) exigem índices com informações adicionais, tais como a ordem e freqüência de cada palavra nos documentos.

A freqüência de cada palavra nos documentos também pode ser útil para otimizar o tempo das consultas, uma vez que pesquisando por palavras mais raras delimita muito o escopo de pesquisa das demais palavras. Alguns sistemas implementam duas listas invertidas: uma simples e rápida, que apenas armazena a lista de documentos de cada palavra, e outra mais lenta, mas que armazena também a posição das palavras nos documentos.

Normalmente, antes de indexar os documentos, é necessário que eles sejam pré-processados. No pré-processamento, eles são convertidos para tokens (em um processo chamado de *lexing*) e posteriormente transformados em tokens mais genéricos através de um processo de *stemming.*Também é comum que palavras muito comuns sejam desprezadas, para diminuir o volume dos índices (essas palavras são chamadas *stopwords).* (Salerma, 2006)

2.11 Lexing

O processo de *Lexing* se refere a obter uma lista de tokens a partir dos caracteres de um texto. É comum que seja determinado um tamanho máximo para cada token, geralmente algo em torno de 32 caracteres, e que sejam eliminados tokens contendo um excesso de caracteres numéricos para evitar um crescimento desordenado dos índices sem um benefício relevante. (Salerma, 2006)

Apesar de aparentemente simples, o lexing pode se tornar mais difícil em linguagens sem uma delimitação explícita de palavras, e este problema acaba repercutindo no projeto de indexadores multilíngües. É necessário utilizar lógicas específicas para cada língua para identificar apropriadamente os limites das palavras.

2.12 Stemming

Stemming é o processo de reduzir todas as palavras ao mesmo *radical*, por meio da retirada dos afixos (sufixos e prefixos) da palavra, permanecendo apenas a raiz (SPARK-JONES & WILLET, 1997)

Por exemplo, as palavras “apresentar”, “apresentador”, “apresentação”, “apresentando”, “apresente”, “apresentador” e “apresentará” podem ser indexadas através de sua raiz “apresent”.

A aplicação do stemming possibilita uma redução significativa na quantia de palavras a indexar no índice invertido, produzindo índices menores e mais rápidos. O resultado é um aumento significativo na quantia de registros retornados pelas buscas, porém também pode resultar em uma redução da acurácia do sistema.

O maior obstáculo à implementação do stemming em sistemas multilíngües é a disponibilidade de algoritmos de stemming específicos para cada língua e o próprio

suporte dos sistemas de indexação aos algoritmos existentes.

2.13 Stopwords

Stopwords são termos freqüentes em um texto e que não carregam nenhuma informação de maior relevância. As *stopwords* tipicamente são compostas por palavras das seguintes classes gramaticais: artigos, preposições, conjunções ("e"), pronomes e advérbios.Alguns exemplos de stopwords na língua portuguesa: "o", "a", "os", "as", "de", "da", "este", "e", "ou" ...

Uma lista de stopwords contém uma lista de palavras a serem ignoradas ao indexar uma lista de documentos. A remoção de stopwords tem como objetivo principal eliminar termos que não são representativos ao documento. Esta operação também pode ser considerada como uma técnica de compressão de textos, pois a eliminação de stopwords reduz o número de palavras a serem analisadas no documento e também o número de palavras a serem armazenadas em uma base de dados. (Salerma, 2006)

Em sistemas de memória de tradução a remoção de stopwords deve ser tratada com cuidado, pois elas influenciam no significado e por conseqüência nas traduções das sentenças.

2.14 Tipos de consulta em FTS

Podem haver muitas formas de consulta em sistemas de FTS. Os seguintes tipos são os mais comuns em sistemas baseados em índices invertidos (Salerma, 2006):

- Consulta normal;
- Consulta Booleana;

- Consulta de Frases;
- Consulta de proximidade.

2.14.1 Consulta normal

É qualquer tipo de busca que não indique explicitamente nenhum tipo de consulta especializada.

A semântica para a busca de um termo isolado é simples: retornar todos os documentos que contenham o termo informado.

Já a semântica para consultas com vários termos varia conforme a implementação de cada ferramenta. Algumas retornam apenas documentos que contenham todos os termos. Outras consideram documentos com apenas alguns dos termos, porém retornam primeiro os documentos que possuam o maior número de palavras.

Esta modalidade de consulta não é adequada para utilização em sistemas de memória de tradução, pelo fato de não considerar a ordem de ocorrência das palavras nos segmentos de texto.

2.14.2 Consulta Booleana

Utiliza operadores lógicos (AND, OR e NOT) para determinar o conjunto de palavras necessários nos documentos retornados. Em alguns casos, são utilizados parênteses para agrupar os termos de busca.

Da mesma forma que a consulta normal, a consulta booleana também ignora a ordem de ocorrência das palavras nos segmentos de texto, não sendo adequada para a maioria das formas de consulta realizadas nos servidores de memória de tradução.

2.14.3 Consulta de Frases

Buscas de frases consideram a ordem das palavras informadas para retornar os resultados.

Este tipo de busca é mais pesado porque exige que os índices armazenem a posição das palavras em cada documentos, sendo necessário implementar a conferência da ordem das palavras durante a busca.

Nas memórias de tradução é muito freqüente a existência de frases muito semelhantes entre si, porém com apenas uma ou duas palavras diferentes. Caso a ferramenta exija que todas as palavras informadas ocorram nos resultados, frases muito semelhantes entre si, mas com discordância em apenas algumas poucas palavras, deixarão de ser retornadas para o tradutor.

2.14.4 Consultas de Proximidade

As consultas de proximidade recuperam documentos que contenham dois ou mais termos, desde que a distância entre estes termos não exceda a um determinado número de palavras.

Este tipo de busca também exige que os índices armazenem a posição das palavras nos documentos, podendo exigir um processamento maior do que os algoritmos de consulta de frases.

2.14.5 Consultas através de Caracteres Coringa (WildCards)

A consulta com caracteres coringa é uma forma de consulta inexata. Há duas variantes (Salerma, 2006):

Coringas de palavra inteira: o curinga representa palavras inteiras. Este tipo de consulta pode ser implementada eficientemente de várias formas, utilizando consultas de frases, ou consultas por proximidade.

Coringas dentro de palavras: coringas representam partes de uma palavra. Esse tipo de busca compromete bastante o desempenho das consultas, e normalmente não é implementada pelas ferramentas de *FTS*.

A consulta através de caracteres coringa pode ser interessante desde que se saiba previamente quais as palavras que podem ser substituídas, o que não é o caso das memórias de tradução.

2.15 Ranking de resultados

A fórmula de cálculo de ranking depende do tipo de consultas que é executado e do suporte fornecido por cada sistema de busca. As consultas geralmente são ordenadas em função da maior quantia de termos presentes em cada documento. Algumas ferramentas costumam levar em consideração fatores estatísticos tais como a raridade da palavra no corpo de texto, sendo que palavras mais raras são mais representativas para o usuário do que as palavras mais comuns. A proximidade das palavras informadas dentro de cada documento também pode ser levada em consideração para a definição do ranking, mesmo que o tipo de consulta não exija uma determinada ordem ou distribuição das palavras.

Após a recuperação dos melhores resultados através de uma ferramenta de FTS, ainda é necessário aplicar uma métrica para informar ao tradutor um valor indicando a similaridade entre o resultado obtido e o pesquisado.

2.16 Web Services e XML

Conforme a W3C, web services proporcionam uma forma padronizada de interoperabilidade entre diferentes aplicações de software. Web services são caracterizados por sua grande interoperabilidade e extensibilidade, bem como por suas descrições processáveis por máquinas graças ao uso da XML. Eles podem ser combinados de um modo fracamente acoplado, de modo a suportar operações complexas. Programas provendo serviços simples podem interagir uns com os outros de modo a oferecer serviços sofisticados de valor adicionado. (W3C, 2002)

XML, ou Linguagem de Marcação Extensível, é um formato de texto simples e muito flexível derivada do SGML (W3C, 2008). Seu propósito primário é ajudar sistemas de informações a compartilhar estruturas de dados, particularmente através da Internet.

2.17 Translation Memory Exchange (TMX)

O TMX (Translation Memory eXchange, ou Intercâmbio de Memória de Tradução) é um padrão aberto de arquivo proposto pela LISA (Localisation Industry Standards Association), principal referência da área da industria da localização para o intercâmbio de dados entre aplicativos de tradução assistida por computador. (PRUDÊNCIO, 2006)

Baseado em XML, o padrão define uma estrutura de documento multilíngüe que guarda em seu corpo vários pares frase-traduções, que podem ser usados por uma ferramenta CAT para povoar uma memoria de tradução durante a sua execução e também para fazer persistência destes mesmos dados. (PRUDÊNCIO, 2006)

TMX é composto por duas partes principais: um cabeçalho de meta informação e

um corpo. Esta segunda parte é a principal destes documentos, composta por pequenas entradas de unidades de tradução. A Figura 1 apresenta um exemplo de uma memória de tradução em formato TMX.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tmx version="1.4">
  <header creationdate="20050208T175531Z"
          creationtool="TMXEditor"
          creationtoolversion="5.0.3"
          datatype="unknown"
          o-tmf="unknown"
          segtype="block"
          srclang="*all*"
          adminlang="en"/>
  <body>
    <tu changedate="20050208T175613Z">
      <tuv xml:lang="en"><seg>Introduction</seg></tuv>
      <tuv xml:lang="es"><seg>Introducción</seg></tuv>
      <tuv xml:lang="zh-cn"><seg>简介</seg></tuv>
    </tu>
    <tu changedate="20050208T175648Z" >
      <tuv xml:lang="en"><seg>Conclusion</seg></tuv>
      <tuv xml:lang="es"><seg>Conclusión</seg></tuv>
      <tuv xml:lang="zh-cn"><seg>总结</seg></tuv>
    </tu>
  </body>
</tmx>
```

Figura 1: Exemplo de memória de tradução em formato TMX (RAYA, 2004)

O formato TMX deve se tornar padrão para o tratamento de memórias de tradução, e sua implementação deverá ser realizada futuramente nos módulos de importação e exportação de memórias de tradução do servidor.

3 Trabalhos Relacionados

A grande maioria dos softwares de tradução com suporte à memórias de tradução implementa o sistema de busca de forma local, diretamente na interface de tradução. Este tipo de solução possui algumas limitações:

- limitações na implementação das memórias de tradução;
- dificuldades para envio de arquivos contendo memórias de tradução;
- ciclos de reaproveitamento das memórias de tradução se tornam muito longos (dois tradutores no mesmo projeto não conseguem aproveitar as traduções de outros sem transferir a memória de tradução manualmente).

Dentre os softwares de código aberto, ainda não existe nenhum sistema de código aberto que permita a iniciativa com esta finalidade. Ainda são poucas as iniciativas existentes para criação de sistemas de memória de tradução distribuídos no mundo do código aberto.

3.1 TMOSS

Translation Machine Open Source Software (TMOSS) é uma especificação para um Sistema de Memória de Tradução (TMS) livre, cujo objetivo principal é prover uma alternativa aos TMSs comerciais. A especificação do TMOSS foi desenvolvida pelo grupo de trabalho Forum Open Language Tools (FOLT), que agrupa companhias e organizações dos setor de tradução e documentação.

Os seguintes módulos foram propostos na especificação do grupo TMOSS (FOLT,

2008):

- Memória de Tradução: repositório contendo as unidades de tradução.
- Segmentador e Matcher de segmentos: responsável por dividir o texto traduzível em segmentos, bem como buscar segmentos idênticos ou similares no repositório de memórias de tradução;
- Interfaces para o usuário: interfaces para os processos de tradução e administração do TMS;
- Filtros: permitem o pré-processamento e pós-processamento de documentos a serem traduzidos;
- Interfaces (importação/exportação): possibilitam a importação/exportação em diversos formatos, tais como TTX, TMX, XLIFF, formatos para terminologia, metadado, e integração com ferramentas externas de controle de qualidade.

A especificação produzida pelo grupo TMOSS pode ser utilizada como um arquitetura de referência para a implementação de ferramentas de tradução livres.

3.2 TinyTM

TinyTM é uma implementação básica de um servidor de memórias de tradução, open source e gratuita, distribuída em [6].

Os seguintes componentes estão descritos na sua arquitetura:

- **Protocolo:** é o ponto principal do TinyTM, e situa-se entre o cliente e o servidor.
- **Servidor:** composto por um banco de dados, uma camada de API PL/SQL

6 http://www.sphinxsearch.com

sobre o banco de dados, uma interface gráfica web para administração e uma interface xml que encobre as chamadas às funções PL/pgSQL.

- **Cliente:** corresponde aos diversos possíveis clientes que podem se utilizar do servidor de memórias de tradução. Isso inclui um sistema especialmente desenvolvido sobre o TinyTM, e outros sistemas livres e comerciais.

Sua implementação é baseada no sistema de índices FTS do banco PostgreSQL, e os principais métodos do servidor foram implementados através de funções em PL/pgSQL.

Está prevista a criação de uma camada de comunicação baseada em XML, porém esta ainda não foi implementada. Para o momento foi desenvolvido um cliente simplificado na forma de um plugin para o Microsoft Word utilizando os recursos de macros em Visual Basic, sendo que a comunicação com o servidor ocorre por meio de uma conexão ODBC.

O modelo de dados utilizado pela ferramenta TinyTM pode ser visualizado na Figura 2.

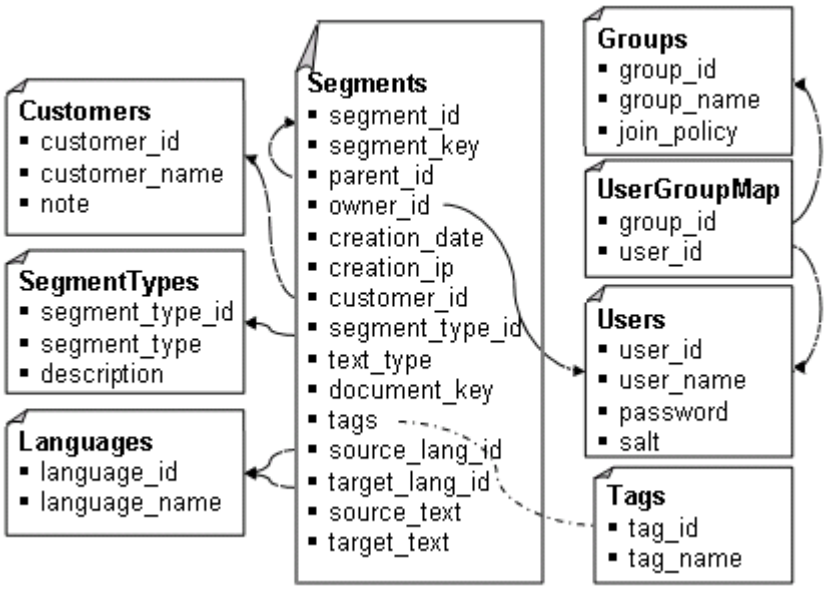

Figura 2: Modelo de dados do software TinyTM

Através do modelo de dados é possível perceber que o software TinyTM é restrito a pares, o que pode representar uma desvantagem caso seja necessário traduzir uma mesma sentença para mais que uma língua, e dificulta ainda a recuperação de traduções se considerarmos que uma tradução pode ser realizada no sentido inverso.

4 Requisitos e Funcionalidades do Protótipo

4.1 Ambiente da Aplicação e Requisitos

4.1.1 Visão Geral

A idéia inicial para o desenvolvimento deste software originou-se no uso da plataforma Rosetta, da empresa Canonical Inc. Esta ferramenta é utilizada para centralizar os esforços dos times de tradução de softwares open source, principalmente os relacionados ao sistema operacional Ubuntu/Linux. Em sua interface, este software apresenta uma listagem de projetos disponíveis para tradução. Qualquer usuário interessado em ajudar na tradução de algum destes projetos se registra junto ao site, e informa quais as línguas com as quais deseja trabalhar. Ao acessar um projeto, a interface do software apresenta um conjunto de mensagens que podem ser traduzidas. Além disso, a ferramenta apresenta sugestões de tradução, baseando-se nas traduções já existentes em sua base de dados, como pode ser visualizado na Figura 3.

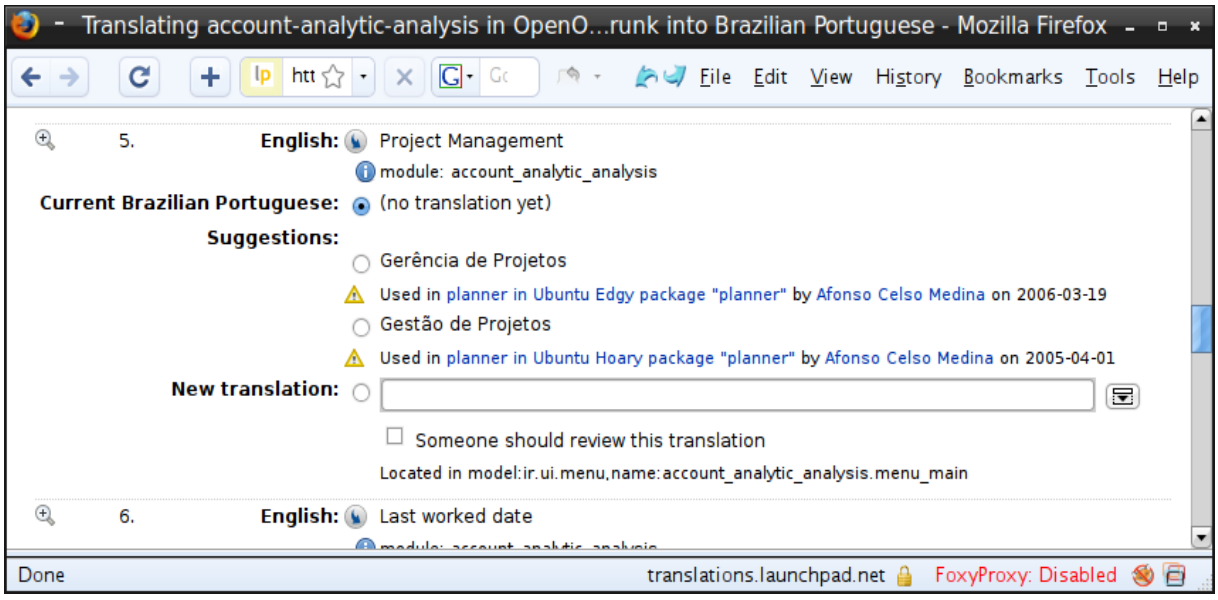

Figura 3: Uso de memórias de tradução na ferramenta Rosetta

Apesar do uso da plataforma Rosetta ser destinado a projetos open source, a ferramenta em si não é open source, e portanto não é possível contribuir em seu desenvolvimento. À época do uso do Rosetta, o sistema de sugestões era bastante precário, com constantes travamentos e lentidão. Desta situação surgiu a motivação para o desenvolvimento de um sistema de memórias de tradução open source.

O desenvolvimento de um sistema de memórias de tradução pode ser bastante complexo. A especificação TMOSS (FOLT, 2008) sugere que a implementação de um sistema de memórias de tradução seja modular e composta das seguintes fases no seu desenvolvimento:

- **Fase 1:** um repositório de memórias de tradução com matching; segmentador de textos; implementação de filtros para leitura e exportação de documentos; interface básica de tradução; interface de administração; e um módulo de análise e estatística.
- **Fase 2:** desenvolvimento de funções de acesso e plugins para aplicativos
- **Fase 3:** desenvolvimento de módulos complementares, como gerência do fluxo de tradução.

É com base nesta proposta que está sendo desenvolvido este servidor de memórias de tradução, que deve englobar a criação de um repositório de memórias de tradução, um sistema de matching e segmentador de textos (que permite a quebra de frases em sentenças menores), além de disponibilizar os métodos do servidor na forma de serviços para ferramentas de tradução.

4.1.2 Ambiente da aplicação

O servidor de memórias de tradução deverá funcionar em um ambiente centralizado, sendo as consultas realizadas de forma remota através de web services.

No futuro, o servidor poderá se comunicar com outros servidores de MT, o que poderá resultar em mais do que uma resposta para determinada frase ou segmento, sendo necessário a escolha de um método para definir as melhores respostas. Esta escolha pode ser feita de várias formas diferentes (SIMÕES, GUINOVART & ALMEIDA, 2004):

- **baseada na resposta mais rápida:** significa que se a tradução já existir na cópia local, o sistema nem chega a comunicar com outros sistemas. Caso contrário, o primeiro servidor que responder será o escolhido;
- **baseada na classificação dos servidores:** a aplicação cliente define uma classificação para cada servidor e escolhe, das respostas que recebeu, a que veio do servidor com uma melhor classificação.
- **baseada na classificação das traduções:** de acordo com uma proposta para a inclusão de uma medida de qualidade por tradução, o cliente poderia ser capaz de escolher a melhor tradução.

4.1.3 Requisitos

Uma série de requisitos devem ser atendidos para garantir o sucesso do desenvolvimento do servidor:

- Alta escalabilidade: o desenvolvimento do servidor deve ser modular, com vários pontos de flexibilidade que permitam escalar o serviço;
- Resposta imediata: o servidor não deve manter o cliente aguardando por traduções. É desejado que as respostas ocorram no intervalo máximo de 1 segundo. Além disso, o servidor deve suportar diversos pedidos simultâneos.
- Aderência aos padrões: isso inclui o suporte ao padrão Unicode e aderência a padrões da área de localização.
- Suporte a vários formatos de escrita: todos os módulos do sistema devem promover amplo suporte a formatos alternativos de escrita
- Não deve ser dada preferência a uma língua específica: O sistema deve funcionar sem depender de uma língua específica
- As unidades de tradução não devem ser baseadas em pares: O sistema deve permitir relacionar mais de duas traduções em uma mesma unidade de tradução;
- O sistema deve permitir traduções multidirecionais entre as sentenças de uma unidade de tradução; não deve haver uma distinção entre a frase original e a traduzida durante a busca por traduções.
- O sistema deve prever o armazenamento de informações contextuais que possam ser úteis para os tradutores, tais como referências à posição da sentença na estrutura de um texto, ou explicações adicionais sobre o sentido da sentença.

4.2 Funcionalidades e Componentes da Aplicação

4.2.1 Principais funcionalidades do servidores de TM

As seguintes funcionalidades podem estar presentes em um sistema de memória de tradução:

- **Recuperação:** consiste em recuperar unidades de tradução a partir de uma das seguintes formas de correspondência:
  a) **Correspondência exata:** ocorre quando o segmento armazenado é exatamente igual ao segmento pesquisado.
  b) **Correspondência em contexto:** ocorre quando o segmento armazenado corresponde ao mesmo segmento pesquisado, inclusive quanto ao arquivo de origem e localização no texto.
  c) **Correspondência aproximada:** quando o segmento armazenado não é exatamente igual ao pesquisado. Alguns sistemas atribuem uma escala para representar a similaridade das sentenças.
  d) **Concordância:** consiste em recuperar segmentos que contenham um grupo de palavras enviadas como parâmetro, sem considerar uma ordem específica destas palavras nos segmentos.
- **Atualização:** permitir que os tradutores adicionem ou atualizem segmentos da memória de tradução
- **Tradução automática:** consiste em recuperar e substituir automaticamente correspondências exatas a partir de traduções armazenadas
- **Importação:** permite transferir unidades de tradução para a memória de tradução, a partir de documentos em formato texto ou em formatos específicos.

- **Análise:** consiste em uma série de passos, e tem como objetivo extrair unidades de tradução e terminologias a partir de traduções presentes em documentos não estruturados.
- **Exportação:** consiste em gerar arquivos estruturados a partir das unidades de tradução presentes na memória de tradução.

Devido a restrições de tempo, a implementação deste protótipo terá enfoque no desenvolvimento da funcionalidade de recuperação de traduções a partir dos vários modos de correspondência descritos acima.

4.2.2 Identificação dos Componentes da Aplicação

Os seguintes componentes serão construídos para obter as funcionalidades propostas para o sistema:

- um sistema de carga de traduções: responsável por realizar a carga de novas traduções para o servidor;
- um repositório de memórias de tradução que irá armazenar as traduções, seus meta dados e o sistema de índices que auxiliará durante as pesquisas na base;
- um sistema de busca eficiente, capaz de realizar a busca com base em uma sentença, calculando um ranking de similaridade;
- uma camada de serviço, que servirá de interface para o serviço de busca;
- um cliente capaz de se conectar a esta camada de serviço.

5 Desenvolvimento do Protótipo

5.1 Definição de estratégias de indexação

As características dos sistemas de memórias de tradução exigem a formulação de uma estratégia especial de indexação e busca.

A arquitetura proposta para o desenvolvimento deste servidor de memórias de tradução é baseada em ferramentas de busca full text e está ilustrada na Figura 4.

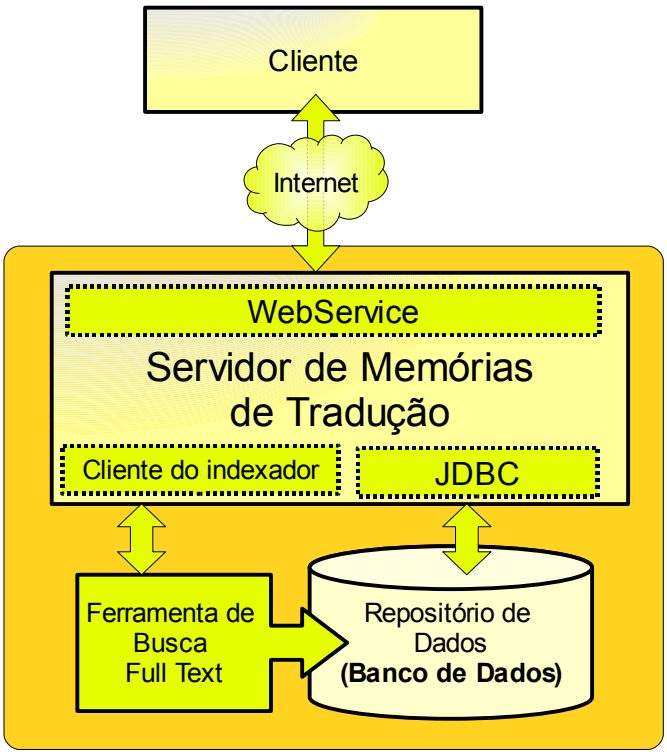

Figura 4: Arquitetura do Servidor de Memórias de Tradução

Segundo essa proposta, é possível identificar 4 componentes:

- repositório de dados;
- ferramenta de busca full text;

- servidor de memórias de tradução;
- cliente;

Esta proposta prevê que os dados fiquem armazenados em um banco de dados, que atua simplesmente como um repositório destas memórias de tradução. Neste banco de dados, apenas os campos de chave estrangeira e os campos de chave primária precisarão ser indexados, uma vez que as consultas sobre os conteúdos das sentenças não serão realizadas por meio do banco de dados.

Integrado ao repositório, deve ser acoplado um sistema de busca full text, que indexa o conteúdo do repositório e permite consultas baseadas em similaridade sobre os dados previamente indexados. Seu papel é fornecer um meio eficiente de busca sobre o conteúdo que esteja em formato texto. A busca aproximada sobre conteúdos de texto deve ser realizada com operadores de consulta apropriados para buscas aproximadas.

Cabe aos métodos desenvolvidos internamente ao servidor de memórias de tradução fazer um refinamento nos resultados do indexador full text. O servidor de memórias de tradução deve ainda ser capaz de se comunicar diretamente com o repositório de dados que contém as memórias de tradução, uma vez que a obtenção das traduções para sentenças específicas pode ser realizada diretamente através de relacionamentos simples de banco de dados. Os métodos podem ser disponibilizados para os clientes através de web services.

O cliente é o responsável por requisitar os métodos disponibilizados pelo servidor e pode ser desde uma ferramenta de auxílio à tradução até um tradutor automático que faça uso da memória de tradução.

5.2 Ferramenta de Busca Full Text

A ferramenta de busca full text será a responsável por indexar as memórias de tradução presentes no repositório de dados e por fornecer meios de pesquisar sobre estes dados com operadores específicos para buscas não exatas, que normalmente não estão disponíveis em SGBDs tradicionais.

A ferramenta Sphinx foi escolhida para utilização no projeto do servidor de memórias de tradução por ser uma ferramenta de alto desempenho, pelo seu suporte à diversos operadores de busca não exatas e principalmente pela sua licença de distribuição GPL.

No site do projeto[7] é possível obter uma descrição mais ampla das características desta ferramenta:

- alta velocidade de indexação (acima de 10MB/s em CPUs modernas);
- Alta velocidade de busca (média das requisições é de 0.1 s em coleções de 2 a 4 GB)
- Alta escalabilidade (mais de 100GB de texto, e mais de 100 milhões de documentos em uma única CPU)
- provê boa relevância através da combinação de ranking por proximidade em frases e ranking estatístico;
- permite buscas distribuídas;
- permite gerar extratos de textos;
- permite buscas de dentro do MySQL através de uma engine de armazenamento plugável;
- suporta consultas booleanas, de frases, e de proximidade de palavras;

7 http://www.sphinxsearch.com

- suporta múltiplos campos full text por documento;
- suporta múltiplos atributos adicionais por documento (exemplo: grupos, timestamps, etc);
- suporta stopwords;
- suporta tanto codificações de byte único, quanto UTF-8;
- suporta nativamente stemming em inglês e russo, e stemming de mais de 15 línguas através da biblioteca Libstemmer[8], também distribuída sob a licensa GPL;
- pode acessar dados no MySQL e no PostgreSQL nativamente.

A ferramenta Sphinx suporta ainda dois modos de indexação: rebuild e re-merge. O rebuild é a reconstrução total do índice. O re-merge consiste em manter um índice principal (main) que contém a massa crítica de dados antigos, e manter um índice secundário (delta) que contém apenas os novos registros, e que graças a seu tamanho menor pode ser indexado com uma freqüência mais alta. De tempos em tempos se torna necessário mesclar o índice "delta" ao índice principal, para que o índice "delta" sempre permaneça pequeno. Esse processo de "merging" é facilmente automatizável, e pode ser realizado sem impedir a realização de buscas concorrentemente.

A configuração do Sphinx é realizada por meio do arquivo **sphinx.conf**, que é responsável por identificar as fontes de dados e os índices que serão criados.

Uma fonte de dados pode ser desde um arquivo XML até uma consulta realizada sobre um banco de dados. As fontes de dados são as responsáveis por identificar quais os dados a serem indexados em cada índice da ferramenta Sphinx.

A identificação dos índices a serem mantidos pela ferramenta Sphinx também é realizada através do arquivo **Sphinx.conf**. Para cada índice a ser criado, é necessário

8 Disponível para download em http://snowball.tartarus.org/

identificar sua fonte de dados, o local em que o índice será mantido, o charset utilizado na fonte de dados, a identificação do algoritmo de stemming adequado à língua utilizada no conteúdo oriundo da fonte de dados e uma tabela contendo os símbolos que não devem ser ignorados durante a indexação para aquela língua. Símbolos que não estiverem presentes nesta tabela serão tratados como delimitadores de palavra e serão ignorados durante a indexação e as buscas.

A ferramenta Sphinx fornece suporte a stemming para 15 línguas com o auxílio da biblioteca open source LibStemmer[9]. Para cada língua suportada pela biblioteca libstemmer foi criado um índice principal e um índice secundário, bem como suas respectivas fontes de dados. Também foram criados dois índices (um principal e um secundário) a serem utilizados para as línguas ainda não suportadas por parte da ferramenta, sendo que as buscas com matching parciais nestas linguagens podem ficar mais limitadas. No Anexo I é possível visualizar a configuração de índices da ferramenta Sphinx para este servidor de memórias de tradução.

A criação de índices específicos para cada linguagem resulta em uma série de vantagens: ao realizar consultas por uma determinada língua, só será necessário consultar os índices específicos daquela língua e estes índices terão um volume de dados limitado; os índices menos utilizados não precisarão estar em memória todo o tempo; o tamanho do índice a ser atualizado é menor, agilizando as atualizações do índice; a qualquer tempo será possível reconfigurar a distribuição dos dados nos índices, bastando para isso reconfigurar os índices e suas fontes de dados para que executem uma distribuição diferente dos dados.

É possível ainda montar uma arquitetura distribuída, utilizando o próprio suporte já fornecido pela ferramenta Sphinx, de modo que as consultas e atualizações estejam distribuídas em várias máquinas.

---

9 Disponível para download em http://snowball.tartarus.org/download.php

5.3 Repositório de Dados

O repositório de traduções é um banco de dados na ferramenta PostgreSQL que contém as tabelas que irão armazenar as memórias de tradução.

A ilustra o modelo de dados desenvolvida para o servidor.

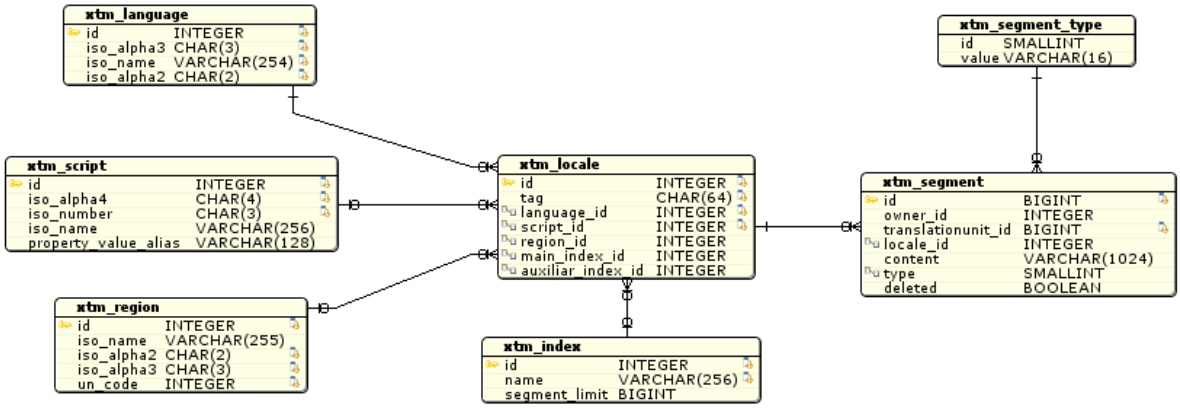

Figura 5: Modelo de dados do repositório

5.3.1 Tabela xtm_locale

A tabela xtm_locale é responsável por armazenar as diversas variações lingüísticas suportadas pelo sistema. No contexto deste servidor de memórias de tradução, um locale é definido como uma combinação única entre uma língua, um formato de escrita, e uma região, esendo que sua tag de identificação se baseia no formato descrito pela RFC 4646.

Na Figura 6 é possível verobservar uma amostra dos dados presentes dnesta tabela.

id [PK] intege	tag character(6	language_id integer	script_id integer	region_id integer	main_index integer	auxiliar_inc integer
81	en_AS	1815		4	4	19
83	en_BE	1815		21	4	19
85	en_BZ	1815		22	4	19
87	en_Dsrt	1815	25		4	19
89	en_GB	1815		230	4	19
91	en_HK	1815		96	4	19
93	en_IN	1815		99	4	19
95	en_MH	1815		135	4	19
97	en_MT	1815		134	4	19
99	en_NZ	1815		156	4	19
101	en_PK	1815		165	4	19
103	en_Shaw	1815	101		4	19
105	en_UM	1815		231	4	19
107	en_US	1815		232	4	19
109	en_ZA	1815		202	4	19
4279	eng	1815			4	19
10187	en	1815			4	19

Figura 6: Amostra de dados da tabela xtm_locale

O atributo id atua como identificador na tabela.

O atributo tag armazena o tag de representação de linguagem descrito pela RFC 4646.

Os atributos language_id, script_id e region_id identificam respectivamente a língua, o formato de escrita e a região identificadas pela tag. Estes atributos são úteis para identificar a compatibilidade entre diferentes tags de identificação de linguagem.

Os atributos main_index_id e auxiliar_index_id são utilizados para identificar quais os índices adequados para a linguagem atual na ferramenta Sphinx. A pré-configuração dos índices é realizada no arquivo de configuração do Sphinx.

5.3.2 Tabela xtm_language

A tabela xtm_language é reponsável por armazenar as línguas suportadas pelo sistema. Esta tabela foi pré-populada a partir da tabela de línguas disponível em [10].

10 http://www.loc.gov/standards/iso639-2/php/English_list.php

Uma amostra dos dados presentes nesta tabela pode ser visualizada na Figura 7.

	id [PK] intege	iso_alpha3 character(3	iso_name character v	iso_alpha2 character(2
31	1475	cym	Welsh	cy
32	1493	dan	Danish	da
33	1548	deu	German	de
34	1600	div	Dhivehi	dv
35	1756	dzo	Dzongkha	dz
36	1789	ell	Modern Gree	el
37	1815	eng	English	en
38	1827	epo	Esperanto	eo
39	1848	est	Estonian	et
40	1862	eus	Basque	eu
41	1866	ewe	Ewe	ee
42	1884	fao	Faroese	fo
43	1887	fas	Persian	fa
44	1900	fij	Fijian	fj
45	1902	fin	Finnish	fi
46	1927	fra	French	fr
47	1938	fry	Western Fris	fy
48	1950	ful	Fulah	ff

Figura 7: Amostra de dados da tabela xtm_language

O atributo *iso_alpha3* representa o código alfabético de três posições para representação de linguagens, conforme descrito pela norma ISO 639-2. O atributo iso_alpha2 representa o código alfabético de duas posições normatizado pela ISO 639-1. O atributo *iso_name* representa o nome da língua em inglês.

O atributo id é um código seqüencial que serve como identificação única de uma língua no contexto deste sistema.

5.3.3 Tabela xtm_script

A tabela xtm_script é reponsável por armazenar uma relação de formatos de escrita. Esta tabela foi pré-populada a partir da tabela de scripts disponibilizada pelo Unicode Consortium em [11].

11 http://unicode.org/iso15924/iso15924-codes.html

A Figura 8 apresenta uma amostra dos dados contidos nesta tabela.

	id [PK] intege	iso_alpha4 character(4	iso_numbe character(3	iso_name character v	property_v character v
52	52	Kana	411	Katakana	Katakana
53	53	Khar	305	Kharoshthi	Kharoshthi
54	54	Khmr	355	Khmer	Khmer
55	55	Knda	345	Kannada	Kannada
56	56	Kore	287	Korean (alia:	
57	57	Kthi	317	Kaithi	
58	58	Lana	351	Lanna, Tai T	
59	59	Laoo	356	Lao	Lao
60	60	Latf	217	Latin (Fraktu	
61	61	Latg	216	Latin (Gaelic	
62	62	Latn	215	Latin	Latin
63	63	Lepc	335	Lepcha (Rón	Lepcha
64	64	Limb	336	Limbu	Limbu
65	65	Lina	400	Linear A	
66	66	Linb	401	Linear B	Linear_B
67	67	Lyci	202	Lycian	Lycian
68	68	Lydi	116	Lydian	Lydian

Figura 8: Amostra de dados da tabela xtm_script

O atributo *iso_alpha4* representa o código alfabético de quatro posições para representação de formatos de escrita, conforme descrito pela norma ISO 15924. O atributo iso_number é um código numérico de três posições. O campo iso_name é o nome padronizado do formato de escrita em inglês. O valor property_value_alias é um valor informativo que relaciona o formato de escrita descrito pela norma ISO 15924 com os formatos de escrita definidos no padrão Unicode[12].

O atributo id é um código seqüencial que serve como identificação única de um formato de escrita no contexto deste sistema.

12 http://www.unicode.org/standard/standard.html

5.3.4 Tabela xtm_region

A tabela xtm_region armazena uma relação de regiões que é válida para especificar tags identificação de linguagens. A tabela foi populada a partir da relação de regiões disponibilizada em [13].

A Figura 9 apresenta exemplos de dados presentes na tabela de regiões.

	id [PK] serial	iso_name character v	iso_alpha2 character(2	iso_alpha3 character(3	un_code integer
29	29	Bouvet Islan	BV	BVT	74
30	30	Brazil	BR	BRA	76
31	31	British India	IO	IOT	86
32	32	Brunei Darus	BN	BRN	96
33	33	Bulgaria	BG	BGR	100
34	34	Burkina Fasc	BF	BFA	854
35	35	Burundi	BI	BDI	108
36	36	Cambodia	KH	KHM	116
37	37	Cameroon	CM	CMR	120
38	38	Canada	CA	CAN	124
39	39	Cape Verde	CV	CPV	132
40	40	Cayman Isla	KY	CYM	136
41	41	Central Afric	CF	CAF	140
42	42	Chad	TD	TCD	148
43	43	Chile	CL	CHL	152
44	44	China	CN	CHN	156
45	45	Christmas Is	CX	CXR	162

Figura 9: Amostra de dados da tabela xtm_region

O atributo iso_name representa o nome do país/região em inglês. O atributo iso_alpha2 representa o código alfabético de duas posições descrito pela norma ISO 3166-1. O atributo iso_alpha3 representa o código alfabético de três posições descrito pela norma ISO 3166-1. O código un_code representa o código da região segundo o padrão UN M.49.

O atributo id é um código seqüencial que serve como identificação única de uma região no contexto deste sistema.

13 http://www.iso.org/iso/country_codes/iso_3166_code_lists.htm

5.3.5 Tabela xtm_index

A tabela xtm_index armazena o nome dos índices configurados na ferramenta Sphinx, de modo a permitir que os segmentos de textos de uma determinada linguagem sejam indexados com índices apropriados à linguagem em questão.

A Figura 10 apresenta uma amostra dos dados desta tabela.

	id [PK] integer	name character varying(256)	segment_limit bigint
1	1	index_default	0
2	2	index_da	0
3	3	index_nl	0
4	4	index_en	2033104
5	5	index_fi	0
6	6	index_fr	0
7	7	index_de	0
8	8	index_hu	0
9	9	index_it	0
10	10	index_no	0
11	11	index_pt	3121316
12	12	index_ro	0
13	13	index_ru	103
14	14	index_es	0
15	15	index_sv	0
16	16	index_tr	0
17	17	index_da_delta	
18	18	index_nl_delta	

Figura 10: Amostra de dados da tabela xtm_index

A configuração de índices específicos para cada linguagem viabiliza a aplicação de processos que envolvem a estrutura da língua, tais como a tokenização de frases e o stemming de palavras, resultando em índices menores e mais eficientes. Por fim, acaba contribuindo para uma melhoria geral nos tempos de resposta do servidor, uma vez que nem todos os índices precisam estar carregados em memória sem estarem sendo utilizados, e as consultas podem ser executadas apenas nos índices à qual a

linguagem está vinculada.

O atributo *name* identifica o nome do índice existente na ferramenta Sphinx. O atributo segment_limit identifica o maior segmento indexado para o índice em questão. Todos os segmentos associados a um índice, e cujo identificador seja maior do que o valor segment_limit de seu índice é um segmento novo e precisará ser atualizado no índice. O atributo id serve como atributo identificador.

5.3.6 Tabela xtm_segment

A tabela xtm_segment é a principal tabela do banco de dados, e é a responsável por armazenar cada segmento das memórias de tradução.

O atributo id é utilizado como identificador único da sentença. O principal campo da tabela é o campo content, que armazena o conteúdo do segmento. O campo locale_id identifica a linguagem relacionada ao segmento. O atributo owner_id identifica o usuário responsável pelo segmento. O campo type identifica o tipo de segmento relacionado à sentença atual.

A relação entre dois segmentos é realizada através do campo translationunit_id, que agrupa diversos segmentos de texto em uma única unidade de transação.

5.4 Desenvolvimento do Servidor

O desenvolvimento do servidor de memórias de tradução foi realizado na plataforma Java[14]. Os principais motivos para a escolha desta plataforma são a sua maturidade, suporte ao padrão Unicode, interoperabilidade e a vasta gama de bibliotecas disponíveis para a linguagem.

14 http://java.com

A Figura 11 apresenta o diagrama de classes desenvolvido para o protótipo.

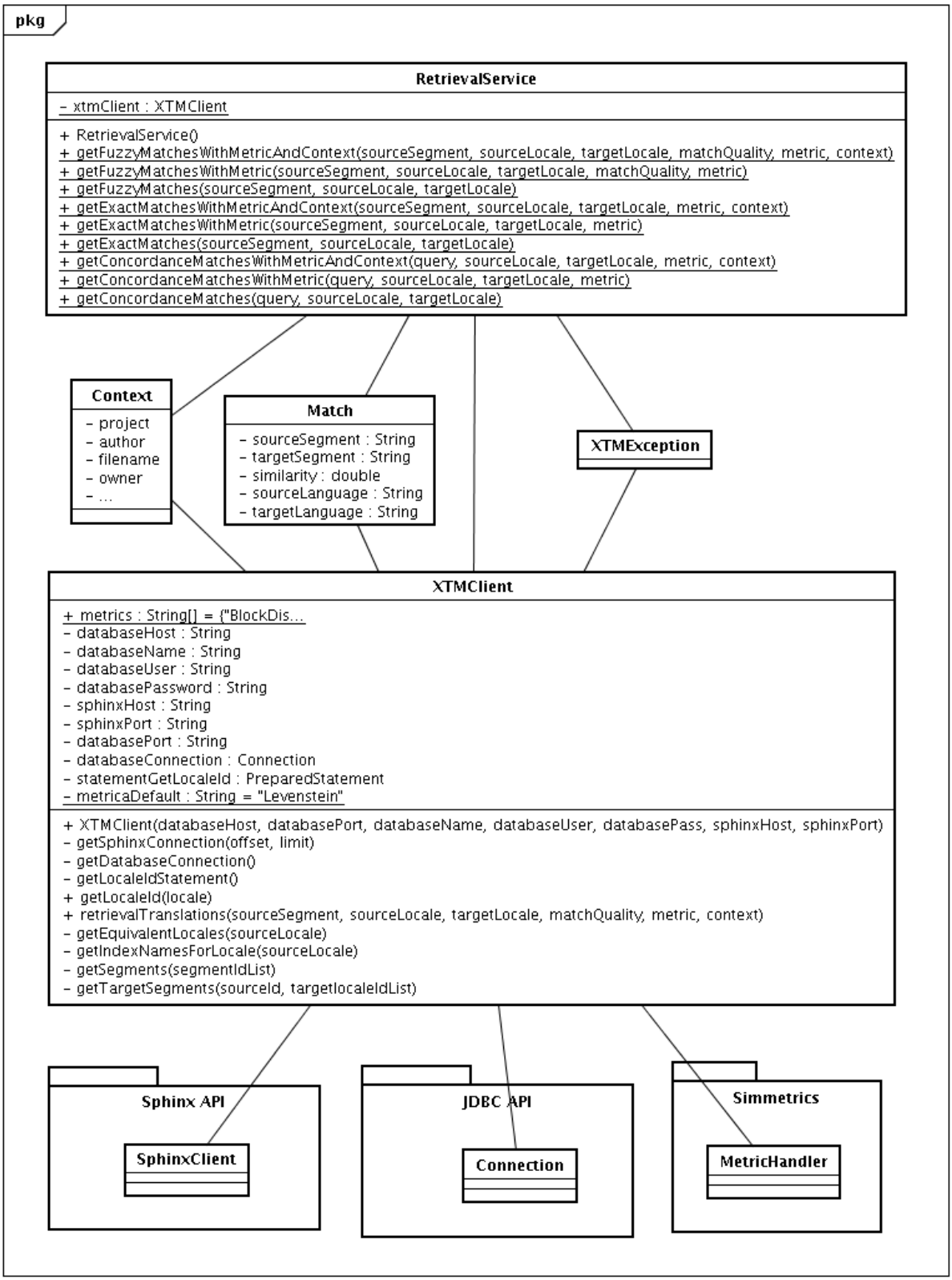

Figura 11: Diagrama de Classes do Protótipo

5.5 Recuperação de Traduções

Consiste em recuperar unidades de tradução a partir de uma das seguintes formas de correspondência: correspondência aproximada, correspondência exata, correspondência em contexto e concordância.

Os modos de correspondência diferem entre si apenas no que se refere ao tipo de consulta que será necessário enviar à ferramenta full text. O procedimento genérico executado para recuperar traduções está ilustrado na Figura 12.

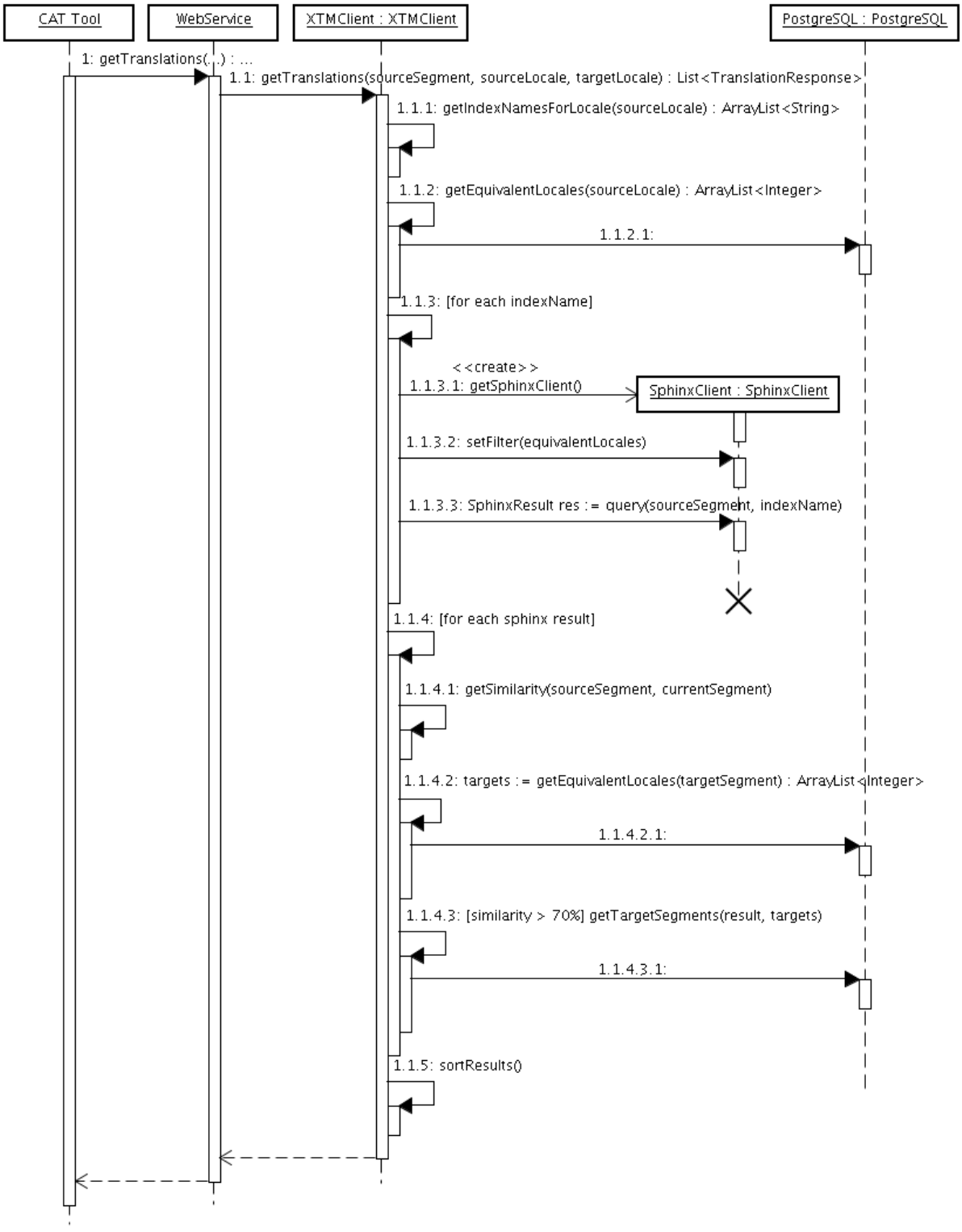

Figura 12: Seqüência genérica de métodos executados para recuperar traduções

O processo inicia com a ferramenta de auxílio à tradução realizando uma chamada ao método getTranslations[1], disponibilizado através de um web service. O web service repassa o pedido à classe XTMClient [1.1], responsável por processar a

requisição.

O primeiro procedimento executado é descobrir quais locales são compatíveis com o locale de origem [1.1.1]. Uma busca por sentenças no locale 'pt', deverá ter como resultado sentenças em qualquer um dos locales baseados na língua portuguesa, dentre os quais os locales 'pt_BR', 'pt_PT', 'por', dentre outras variações válidas.

O próximo procedimento [1.1.2] consiste em verificar quais os índicesda ferramenta Sphinx que são responsáveis por indexar estes locales. Essa informação fica armazenada junto ao cadastro de cada locale no banco de dados. Cada locale pode estar vinculado a um índice específico, ou podemos ter vários locales sendo indexados pelo mesmo índice. A definição de índices na ferramenta Sphinx depende das características de cada língua a ser indexada. Uma mesma língua escrita com diferentes formatos de escrita, por exemplo, pode exigir um índice para cada um de seus formatos de escrita. A configuração atual dos índices na ferramenta Sphinx pode ser visualizada no Anexo I.

Sabendo quais índices consultar na ferramenta Sphinx, é preciso buscar junto a estes índices [1.1.3] pela sentença enviada como parâmetro. O tipo de consulta a ser enviada à ferramenta de indexação dependerá do modo de correspondência desejada: correspondência aproximada, correspondência exata, correspondência em contexto ou concordância. Cada um destes modos de correspondência será detalhado nos capítulos posteriores.

Pelo fato de um mesmo índice poder estar associado a mais que um locale, se faz necessário aplicar um filtro na busca, de modo a retornar apenas as sentenças dos locales desejados [1.1.3.2].

A ferramenta irá retornar um ranking da sentenças mais similares à sentença original, porém ainda é necessário calcular a similaridade entre cada uma das sentenças retornadas pelo FTS em relação à sentença original [1.1.4.1]. O algoritmo

utilizado para medir a similaridade entre duas sentenças foi a distância de edição de Levenshtein. Foi utilizada a implementação da métrica de distância de edição de Levenshtein disponível na biblioteca simmetrics. Esta biblioteca implementa várias outras métricas. A métrica utilizada para calcular a similaridade entre duas sentenças foi a de distância de edição de Levenshtein, por meio da biblioteca Simmetrics. A biblioteca simmetrics é open source, e está disponível em [15].

Uma vez de posse das sentenças mais similares à sentença original, e antes de buscar as traduções para estas sentenças diretamente via relacionamentos do banco de dados, é necessário saber quais os locales compatíveis com o locale de destino especificado [1.1.4.2].

A obtenção das traduções é realizada diretamente em uma simples consulta SQL ao repositório de traduções, informando a unidade de tradução e os locales compatíveis ao locale de destino.

5.5.1 Correspondência aproximada

A recuperação de traduções com *correspondência aproximada* permite recuperar traduções de sentenças que não sejam exatamente iguais à sentença enviada como parâmetro. Este tipo de correspondência segue os procedimentos básicos descritos anteriormente, porém se diferencia dos demais modos de correspondência através do uso de operadores apropriados para busca aproximada na ferramenta Sphinx.

A ferramenta Sphinx possui dois operadores que implementam correspondência aproximada considerando a ordem das palavras na busca: o *operador de busca por proximidade*, e o *operador de matching de quorum.*

O *operador de busca por proximidade* permite buscar por uma sentença indicando

15 http://sourceforge.net/projects/simmetrics/

um número máximo de palavras adicionais presentes entre a seqüência de palavras da frase fornecida como parâmetro. A busca pela frase “hello world” com 1 palavra adicional poderia ter como resultados válidos as frases “Hello my world.”, e “Then I said hello to world”.

Já o *operador de matching de quorum* permite a busca por sentenças indicando o número mínimo de ocorrências de palavras na sentença armazenada, dentre as palavras da sentença original. A busca pela frase “the world is a wonderful place” com o parâmetro mínimo de 3 palavras, irá retornar qualquer frase que contenha três palavras dentre as contidas na frase.

O operador escolhido para utilização na correspondência aproximada foi o operador de matching de quorum, devido à sua tolerância à ausência de algumas das palavras pesquisadas no interior dos segmentos.

Uma vez realizada a consulta com o operador apropriado à ferramenta Sphinx, os procedimentos restantes não se modificam em relação aos demais modos de consulta.

5.5.2 Correspondência Exata

O pedido de correspondência exata irá retornar para o cliente apenas traduções de segmentos iguais ao segmento enviado como parâmetro.

O procedimento para recuperação de correspondências exatas é muito parecido com um o procedimento realizado para as demais formas de recuperação. O grande diferencial se dá no momento de calcular as medidas de similaridade, pois apenas traduções de sentenças em que este valor seja igual a 100% serão retornadas para o cliente.

Entretanto, é possível otimizar o desempenho deste tipo de requisições através do uso de um *operador de consulta de frases* na ferramenta Sphinx. Uma *consulta de*

frases é mais pesada do que uma *consulta normal* porque ela deve levar em consideração a ordem das palavras presentes na sentença, porém será mais rápida do que uma *consultas de proximidade* que se faz necessária em recuperações de traduções com correspondência aproximada.

5.5.3 Correspondência em Contexto

O pedido de correspondência em contexto é similar a uma correspondência exata, porém com com algumas restrições de contexto. Além de retornar apenas traduções de segmentos iguais aos enviados como parâmetro, será necessário que o contexto da tradução seja o mesmo da sentença enviada; o contexto pode definido como alguma informação que identifique o arquivo de origem, a localização da sentença, ou o projeto ao qual a sentença pertença originalmente.

A implementação do procedimento de recuperação de sentenças em contexto é semelhante ao procedimento executado para recuperar sentenças exatas, com exceção de que se faz necessário um filtro para delimitar o escopo da busca ao contexto indicado pelo cliente.

5.5.4 Concordância

A concordância consiste em recuperar segmentos que contenham um grupo de palavras enviadas como parâmetro, sem considerar uma ordem específica destas palavras nos segmentos.

O procedimento geral adotado será o mesmo das demais sentenças, com diferencial apenas no tipo de consulta a ser enviado à ferramenta de indexação: é possível utilizar uma *consulta normal*, que não considera a ordem das palavras nas sentenças, mas que tem significativas vantagens de desempenho em relação aos

demais tipos de consulta. Os pedidos de concordância podem ser utilizados tanto por meio do operador booleano 'OR' quanto pelo operador booleano 'AND'.

5.6 Importação

A importação de traduções pode ocorrer a partir de memórias de tradução presentes em arquivos em formato texto ou em formatos específicos próprios para intercâmbio de memórias de tradução, tal como o formato TMX.

O padrão TMX define uma estrutura de documentos multilíngüe que guarda em seu corpo vários pares frase-traduções, que podem ser usados por uma ferramenta CAT para povoar uma memória de tradução durante a sua execução e também para fazer persistência destes mesmos dados (PRUDÊNCIO, 2006).

O método de importação pode ser desenvolvido à parte do servidor, e pode fazer uso de consecutivas chamadas ao método de importação de dados disponibilizado pelo servidor de memórias de tradução. Outra possibilidade é que o módulo de importação tenha acesso direto ao repositório de memórias de tradução. No protótipo atual a importação de dados foi realizada como auxílio de uma ferramenta de ETL.

5.7 Desenvolvimento do Web Service

Um web service foi desenvolvido para permitir que computadores remotos tenham acesso aos métodos implementados pelo servidor de memórias de tradução.

O seguintes métodos foram disponibilizados pelo serviço RetrievalService:

- **getFuzzyMatches**(sourceSegment, sourceLocale, targetLocale): realiza a correspondência aproximada para recuperar o matchings. A métrica padrão utilizada é a distância de edição de Levenshtein, e a similaridade mínima é de

75%.

- **getFuzzyMatchesWithMetric**(sourceSegment, sourceLocale, targetLocale, metric, matchQuality): idêntico ao método anterior, porém é possível especificar a qualidade dos mathes, e a métrica a ser utilizada pelo servidor. As seguintes métricas estão disponíveis na versão atual da biblioteca SimMetrics utilizada: BlockDistance, ChapmanLengthDeviation, ChapmanMeanLength, CosineSimilarity, DiceSimilarity, EuclideanDistance, JaccardSimilarity, Jaro, JaroWinkler, Levenshtein, MatchingCoefficient, MongeElkan, ChapmanMatchingSoundex, NeedlemanWunch, OverlapCoefficient, QGramsDistance, SmithWaterman, SmithWatermanGotohWindowedAffine, SmithWatermanGotoh e Soundex.
- **getExactMatches**(sourceSegment, sourceLocale, targetLocale): realiza a correspondência exata para recuperar o matchings. A métrica padrão utilizada é a distância de edição de Levenshtein, e a similaridade mínima é de 100%. Uma similaridade de 100% obtida por algumas métricas (Soundex, por exemplo) não garante necessariamente que as strings comparadas serão exatamente iguais entre si.
- **getExactMatchesWithMetric**(sourceSegment, sourceLocale, targetLocale, metric): idêntico ao método anterior, porém é possível especificar a métrica a ser utilizada pelo servidor.
- **getConcordanceMatches**(query, sourceLocale, targetLocale): realiza a concordância para recuperar os matchings. O método retornará apenas matches cuja sentença contenha todas as palavras presentes no parâmetro 'query'. O método retorna os melhores matchings, mas não garante uma qualidade mínima. A métrica de similaridade utilizada por padrão para qualificar os matches será a distância de edição de Levenshtein.
- **getConcordanceMatchesWithMetric**(sourceSegment, sourceLocale,

targetLocale, metric): idêntico ao método anterior, porém é possível especificar a métrica a ser utilizada pelo servidor para qualificar os matches.

A parametrização das informaces de contexto da tradução ainda está sendo desenvolvida, e será suportada pelos seguintes métodos do web service:

- **getFuzzyMatchesWithMetricAndContext**(sourceSegment, sourceLocale, targetLocale, metric, matchQuality, context): idêntico ao método getFuzzyMatchesWithMetric, porém aceita também o contexto da busca.
- **getExactMatchesWithMetricAndContext**(sourceSegment, sourceLocale, targetLocale, metric, context): idêntico ao método getExactMatchesWithMetric, porém aceita também o contexto da busca.
- **getConcordanceMatchesWithMetricAndContext**(sourceSegment, sourceLocale, targetLocale, metric, context): idêntico ao método getConcordanceMatchesWithMetric, porém aceita também o contexto da busca.

5.8 Desenvolvimento de um cliente para o Web Service

Para poder testar a funcionalidade do web service foi desenvolvida uma página simples que fornece acesso a seus métodos.

A Figura 13 apresenta o cliente em ação, apresentando os resultados da busca aproximada retornados pelo método getFuzzyMatches. No *frame* de resultados, é possível observar o segmento localizado, a qualidade do match e a tradução referente ao segmento localizado na língua de destino informada.

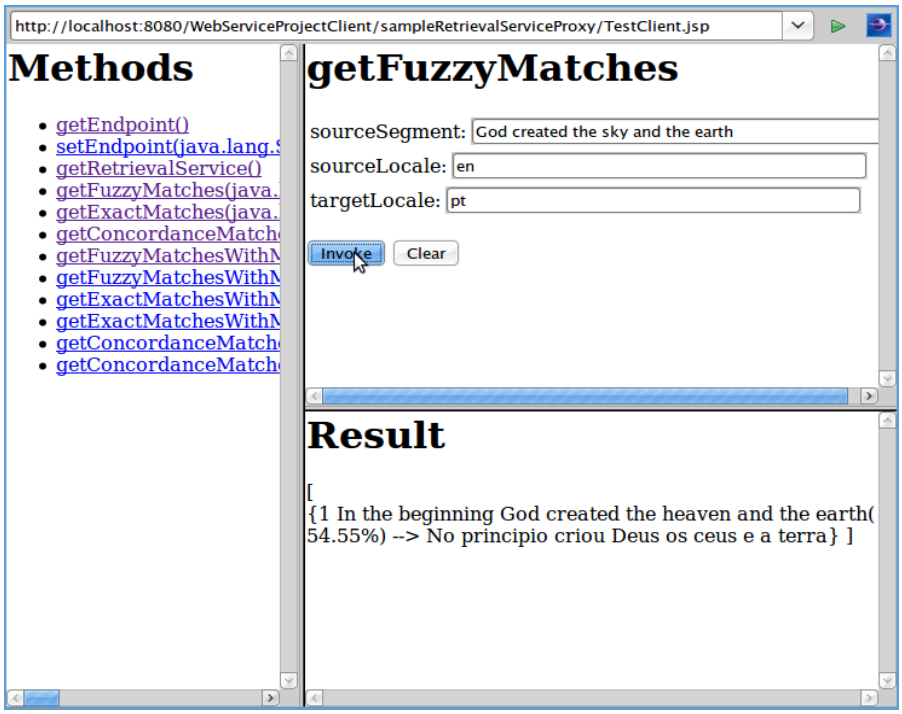

Figura 13: Cliente executando chamada ao método getFuzzyMatches

Na Figura 14 é possível observar o resultado de uma recuperação com correspondência exata, através do método getExactMatches.

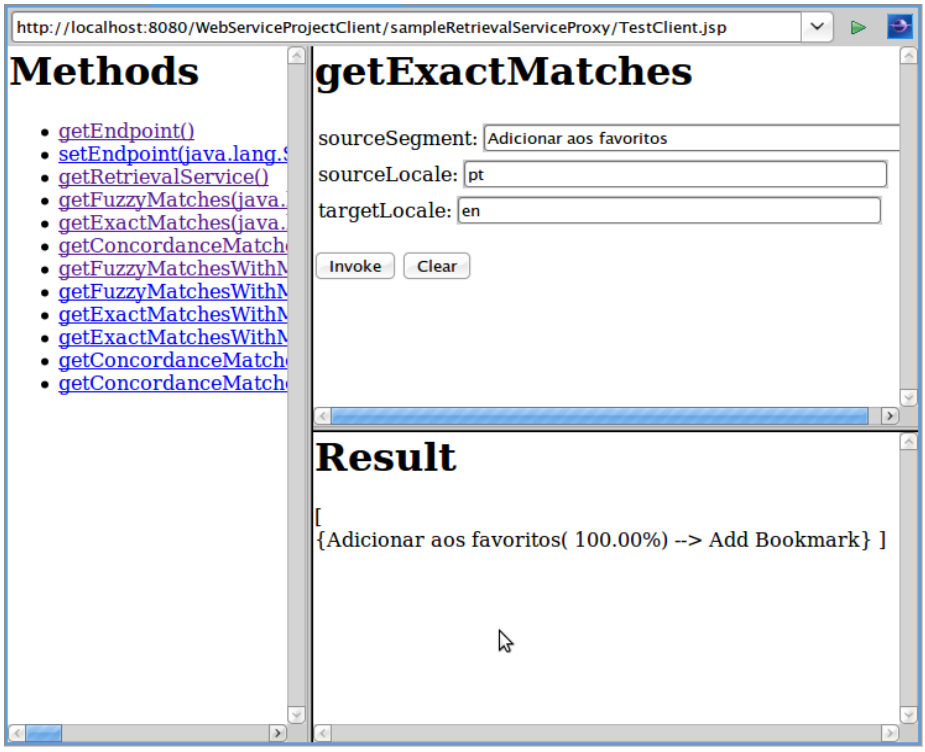

Figura 14: Cliente executando chamada ao método getExactMatches

Na Figura 15 é apresentada uma chamada ao método getFuzzyMatchesWithMetrics, que permite especificar a qualidade do *match* e também a métrica a ser utilizada. Neste exemplo, utilizando a métrica "Levenshtein" a qualidade do *match* calculada ficou em 66,67%. A mesma consulta utilziando a métrica "Soundex" retorna a qualidade do match com o valor 100%, conforme observado na Figura 16. Neste exemplo também é possível observar que a métrica Soundex informou uma similaridade de 100% entre o sourceSegment e o segmento retornado, apesar de ambos não serem exatamente iguais.

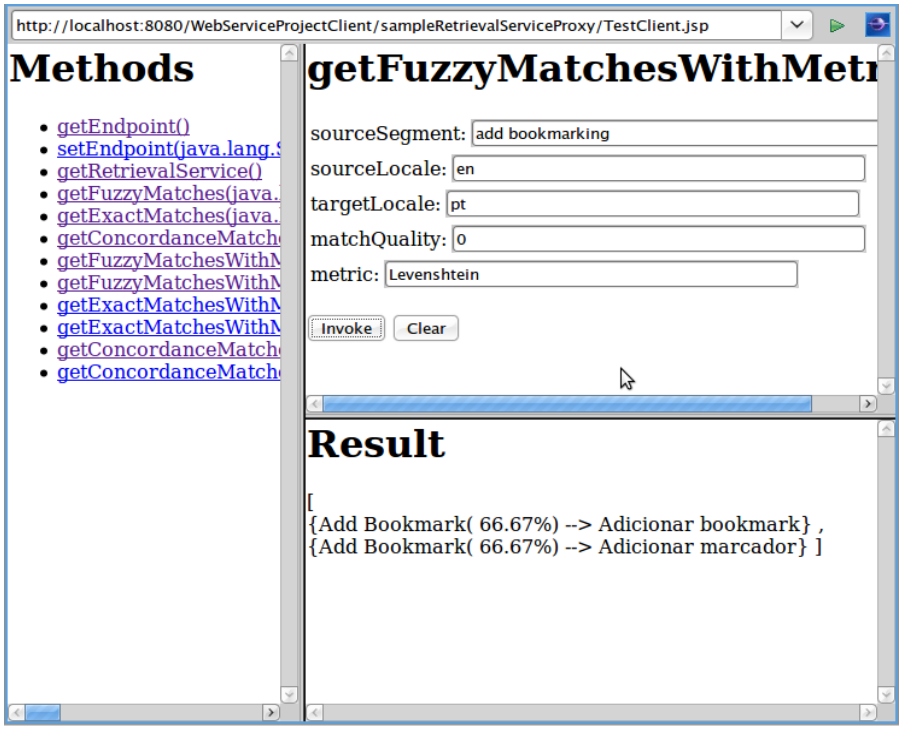

Figura 15: Chamada ao método getFuzzyMatchesWithMetrics, utilizando a métrica "Levenshtein"

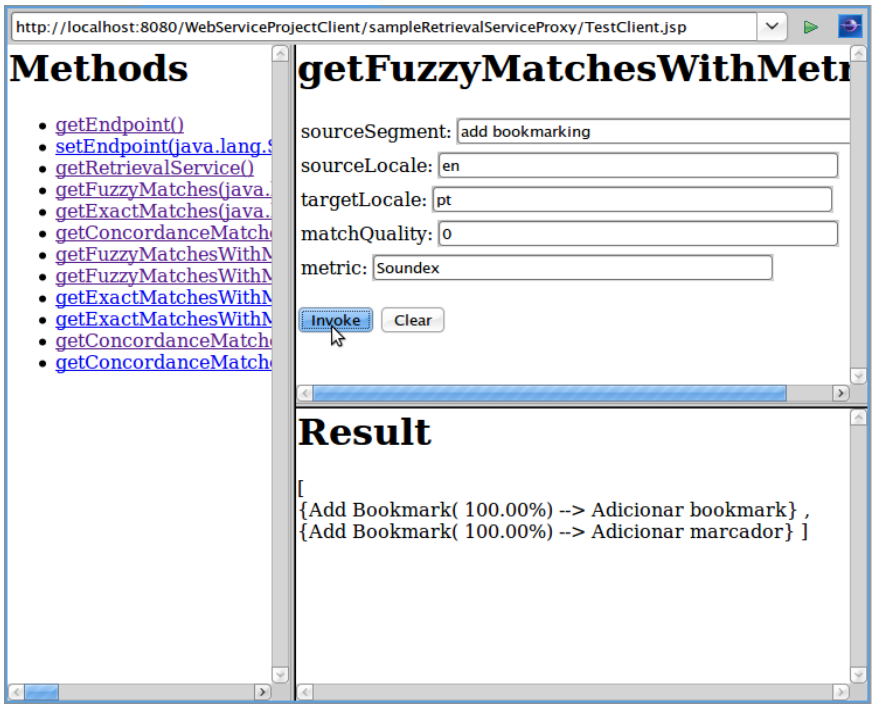

Figura 16: Chamada ao método getFuzzyMatchesWithMetrics, utilizando a métrica "Soundex"

A Figura 17 apresenta uma chamada ao método getConcordanceMatchesWithMetrics, responsável pela recuperação de concordâncias. A qualidade dos *matches* retornados é calculada conforme a métrica fornecida.

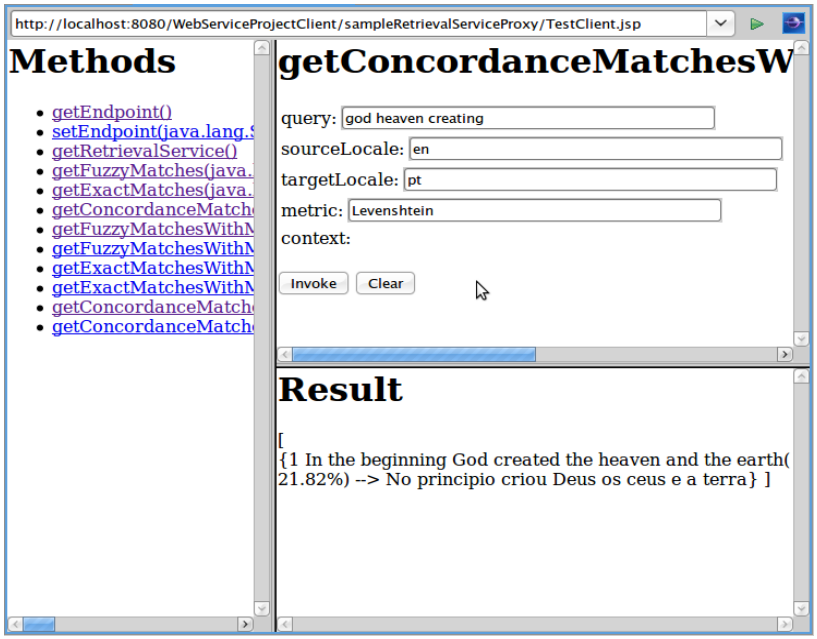

Figura 17: Cliente executando chamada ao método getConcordanceMatchesWithMetrics

5.9 Resultados Obtidos

O protótipo produzido a partir deste trabalho já é funcional, permitindo consultas aproximadas em tempos relativamente baixos. Nos testes preliminares, o tempo de médio de resposta do sistema para recuperar traduções a partir de correspondência aproximada com sentenças da língua inglesa é de 0.6 segundos em uma base contendo aproximadamente 1,6 milhões de registros em inglês. Os pedidos de concordância realizados sobre esta base demoram cerca de 0.16 segundos.

A qualidade dos resultados nestes testes preliminares também se mostrou satisfatória. Graças às medidas de similaridade calculadas pelo servidor, apenas frases realmente relevantes são retornadas para os usuários, sendo possível determinar um coeficiente mínimo de similaridade para os resultados.

O sistema de stemming em várias línguas também se apresentou muito satisfatório. Testes realizados com variações morfológicas nas palavras contidas em sentenças em russo, por exemplo, funcionaram sem problemas, trazendo os resultados esperados.

A arquitetura do sistema também se mostrou satisfatória, pois prevê vários pontos de flexibilidade que permitem a escalabilidade do sistema. É possível ampliar o armazenamento da base de dados através da adição de novos repositórios de dados que podem estar espalhados em várias máquinas. A ferramenta Sphinx também é capaz de realizar o processamento distribuído, através do recurso de índices remotos, e em último caso também é possível distribuir os serviços do servidor de memória de tradução em várias máquinas. Com esta arquitetura, seria possível suportar todas as traduções de grandes portais de tradução, tal como portais de tradução de software livre, ou outro sistema comercial equivalente.

O servidor foi projetado tentando se aderir ao máximo aos padrões vigentes na

indústria de localização. Isso vai desde os padrões de representação de linguagens, até o suporte a formatos de escrita alternativos por todos os componentes do sistema. A estrutura do sistema também não foi baseada em uma língua específica, para permitir a recuperação de traduções em qualquer par de linguagens, e sem uma direção específica (segmentos equivalentes no sentido das línguas A $\rightarrow$ B são automaticamente válidos para o sentido B $\rightarrow$ A).

O servidor também está sendo finalizado para permitir o armazenamento de informações de contexto, que permitem identificar a origem das traduções, qual o tradutor, de qual projeto e arquivo a tradução teve origem, dentre outras informações úteis para a implementação da recuperação de informações em contexto.

6 Considerações Finais

As memórias de tradução já atingiram o patamar de softwares essenciais para as empresas de tradução e localização.

No âmbito do desenvolvimento de software livre, porém, ainda se observa uma grande precariedade na utilização de ferramentas de auxílio à tradução, especialmente de ferramentas mais avançadas, tais como as memórias de tradução. A grande maioria dos softwares livres que implementam memórias de tradução o fazem de forma precária, e permitem apenas o trabalho com pequenas memórias de tradução armazenadas localmente.

O desenvolvimento de um servidor de memórias de tradução de código aberto vem de encontro à necessidade de ferramentas mais eficientes para gerenciamento de memórias de tradução em projetos de software livre. Os resultados obtidos até o momento não deixam dúvida quanto à viabilidade da utilização deste servidor para a disponibilização de serviços baseados em memórias de tradução, beneficiando tanto a comunidade de software livre quanto empresas produtoras de software comercial que desejem utilizar este servidor em seus processos de tradução de software.

6.1 Trabalhos Futuros

Durante a pesquisa e o desenvolvimento deste servidor de memórias de tradução, foi possível identificar uma série de aprimoramentos possíveis para o sistema:

- melhorias na implementação dos métodos no servidor;

- autenticação de usuários e grupos de acesso;
- estatísticas;
- segurança e auditoria;
- integração com outros servidores de memórias de tradução;
- testes com outras ferramentas de indexação (Lucene, e ferramentas mais apropriadas para línguas orientais);
- melhoria no suporte a formatos de escrita que não permitam stemming;
- utilização dos conceitos de memória de tradução de 2 geração;
- integração de n-gram parsers para línguas orientais;
- utilização de uma arquitetura peer to peer.

Referências Bibliográficas

ALFARO, C. **Descobrindo, Compreendendo e Analisando a Tradução Automática.** Rio de Janeiro, 1998.

FOLT (Forum Open Language Tools), LiSoG (Linux Solutions Group). **Translation Memory Open Source System**. Stuttgart-Möhringen, 2007.

INTERNATIONAL SOCIETY FOR STANDARDIZATION. **ISO 15924:2004 information and documentation - codes for the representation of names of scripts**. Disponível em http://www.iso.org/iso/catalogue_detail?csnumber=29546.

INTERNATIONAL SOCIETY FOR STANDARDIZATION. **ISO 3166-1:2006 codes for the representation of names of countries and their subdivisions - part 1: country codes**. Disponível em http://www.iso.org/iso/catalogue_detail?csnumber=39719.

INTERNATIONAL SOCIETY FOR STANDARDIZATION. **ISO 639-1:2002 codes for the representation of names of languages - part 1: alpha-2 code**. Disponível em http://www.iso.org/iso/catalogue_detail?csnumber=22109.

INTERNATIONAL SOCIETY FOR STANDARDIZATION. **ISO 639-2:1998 codes for the representation of names of languages - part 2: alpha-3 code**. http://www.iso.org/iso/catalogue_detail?csnumber=4767.

JUNIOR, Helion Cardozo. **Ferramentas para apoio à Geração de Código Internacionalizável**. Florianópolis, SC, 2004.

LISA: The Localization Industry Standards Association. **Translation Memory eXchange**. Disponivel em http://www.lisa.org/standards/tmx/. Acesso em 15 de novembro de 2007.

O'BRIEN, Sharon. **Practical Experience of Computer-Aided Translation Tools in the Software Localization Industry**. 1998. In: GOW, Francie. **Metrics for Evaluating Translation Memory Software**. University of Canada, 2003. p. 14.

PRUDÊNCIO, Achilles Colombo. **Computer Aided Translation**. Universidade Federal de Santa Catarina, 2006.

RAYA, Rodolfo. **XML in Localisation: reuse translations with TM and TMX**. LISA, 2004. Disponível em http://www.lisa.org/globalizationinsider/2007/05/xml_in_localisa.html.

RIECHE, Adriana Ceschin; MARTINS, Marcia do Amaral Peixoto. **Memória de tradução: auxílio ou empecilho?** Rio de Janeiro, 2004. 177p. Dissertação de Mestrado - Departamento de Letras, Pontifícia Universidade Católica do Rio de Janeiro. Disponível em http://www.maxwell.lambda.ele.puc-rio.br/cgi-bin/db2www/PRG_0490.D2W/INPUT?CdLinPrg=pt

SALERMA, Osku. **Design of a Full Text Search index for a database management system** . Universidade de Helsinki, 2006.

SIMÕES, Alberto. GUINOVART, Xavier Gómes. ALMEIDA, José João. **Distributed translation memories implementation using webservices**. Sociedade Espanola para el Procesamiento del Lenguaj, 2004.

SIMÕES, Alberto. GUINOVART, Xavier Gómes. ALMEIDA, José João. **Memórias de tradução distribuídas.** Universidade do Minho/Universidade do Migo, 2004.

SPARK-JONES, WILLETT. **Readings in Information Retrieval (context)**. ACM, 1997

THE INTERNET SOCIETY. **RFC4646 – tags for identifying languages**. rfc.net/rfc4646.html

UNICODE CONSORTIUM. **The unicode character code charts by script**. Disponível em http://unicode.org/charts/.

UNITED NATIONS STATISTICAL DIVISION. **Standard country or area codes for statistical use (UN M.49)**. Disponível em http://unstats.un.org/unsd/methods/m49/m49.htm

W3C. **Web Service Activity Statement**, 2002. Disponível em http://www.w3.org/2002/ws/Activity.

W3C. **Extensible Markup Language (XML) 1.0 (Fifth Edition)**, 2008. Disponível em http://www.w3.org/TR/xml/.

WITTEN, Ian. MOFFAT, Alistar. BELL, Timothy C. **Managing Gigabytes**. Academic Press, 1999.

ZOBEL, Justin. MOFFAT, Alistair. RAMAMOHANARAO, Kotagiri. **Inverted files versus signature files for text indexing**. ACM Transactions on Database Systems, 1998.

ANEXO I: sphinx.conf

```
# Abstract Data Source for Localhost postgres


source empty_source
{
        type                                    = pgsql
        sql_host                                = localhost
        sql_user                                = xtm
        sql_pass                                = xtm
        sql_db                                  = xtm
        sql_port                                = 5432
        sql_ranged_throttle     = 0
        sql_query_info          = SELECT id, owner_id, translationunit_id, locale_id, content, type,
deleted FROM xtm_segment WHERE 1 = 2 AND deleted = false
        sql_query                = SELECT id, owner_id, translationunit_id, locale_id, content, type,
deleted FROM xtm_segment WHERE 1 = 2 AND deleted = false
}


source source1
{
        type                                            = pgsql
        sql_host                        = localhost
        sql_user                        = xtm
        sql_pass                        = xtm
        sql_db                                  = xtm
        sql_port                        = 5432
        sql_ranged_throttle      = 0
        sql_query_info           = SELECT id, owner_id, translationunit_id, locale_id, content, type,
deleted FROM xtm_segment WHERE id=$id
        sql_attr_uint = owner_id
        sql_attr_uint = locale_id
}

# Source for danish language
source xtm_da : source1
{
        sql_query_pre = \
                update xtm_index x\
                set segment_limit = (\
                        select coalesce(max(s.id),0)\
                        from xtm_segment s\
                        inner join xtm_locale l\
                          on l.id = s.locale_id\
                        inner join xtm_index i\
                          on i.id = l.main_index_id\
                        where i.name = x.name\
                )\
                where name = 'index_da'

        sql_query                               = \
               select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_da'\
                  and s.id <= i.segment_limit
}

source xtm_da_delta : source1
{
        sql_query                               = \
               select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
```

```
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_da'\
                  and s.id > i.segment_limit
}


# Source for dutch language
source xtm_nl : source1
{
        sql_query_pre = \
                update xtm_index x\
                set segment_limit = (\
                        select coalesce(max(s.id),0)\
                        from xtm_segment s\
                        inner join xtm_locale l\
                          on l.id = s.locale_id\
                        inner join xtm_index i\
                          on i.id = l.main_index_id\
                        where i.name = x.name\
                )\
                where name = 'index_nl'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_nl'\
                  and s.id <= i.segment_limit
}

source xtm_nl_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_nl'\
                  and s.id > i.segment_limit
}

# Sources for english language
source xtm_en : source1
{
        sql_query_pre = \
                update xtm_index x\
                set segment_limit = (\
                        select coalesce(max(s.id),0)\
                        from xtm_segment s\
                        inner join xtm_locale l\
                          on l.id = s.locale_id\
                        inner join xtm_index i\
                          on i.id = l.main_index_id\
                        where i.name = x.name\
                )\
                where name = 'index_en'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
```

```
                  on i.id = l.main_index_id\
                where i.name = 'index_en'\
                  and s.id <= i.segment_limit
}

source xtm_en_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_en'\
                   and s.id > i.segment_limit
}




# Source for finish language
source xtm_fi : source1
{
         sql_query_pre = \
                 update xtm_index x\
                 set segment_limit = (\
                         select coalesce(max(s.id),0)\
                         from xtm_segment s\
                         inner join xtm_locale l\
                           on l.id = s.locale_id\
                         inner join xtm_index i\
                           on i.id = l.main_index_id\
                         where i.name = x.name\
                 )\
                 where name = 'index_fi'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_fi'\
                   and s.id <= i.segment_limit
}

source xtm_fi_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_fi'\
                   and s.id > i.segment_limit
}

# Source for frensh language
source xtm_fr : source1
{
         sql_query_pre = \
                 update xtm_index x\
                 set segment_limit = (\
                         select coalesce(max(s.id),0)\
                         from xtm_segment s\
```

```
                        inner join xtm_locale l\
                          on l.id = s.locale_id\
                        inner join xtm_index i\
                          on i.id = l.main_index_id\
                        where i.name = x.name\
                )\
                 where name = 'index_fr'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_fr'\
                   and s.id <= i.segment_limit
}

source xtm_fr_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_fr'\
                   and s.id > i.segment_limit
}

# Source for deutsch language
source xtm_de : source1
{
         sql_query_pre = \
                 update xtm_index x\
                 set segment_limit = (\
                         select coalesce(max(s.id),0)\
                         from xtm_segment s\
                         inner join xtm_locale l\
                           on l.id = s.locale_id\
                         inner join xtm_index i\
                           on i.id = l.main_index_id\
                         where i.name = x.name\
                 )\
                 where name = 'index_de'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_de'\
                   and s.id <= i.segment_limit
}

source xtm_de_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_de'\
```

```
                  and s.id > i.segment_limit
}


# Source for hungarian language
source xtm_hu : source1
{
         sql_query_pre = \
                 update xtm_index x\
                 set segment_limit = (\
                         select coalesce(max(s.id),0)\
                         from xtm_segment s\
                         inner join xtm_locale l\
                           on l.id = s.locale_id\
                         inner join xtm_index i\
                           on i.id = l.main_index_id\
                         where i.name = x.name\
                 )\
                 where name = 'index_hu'

         sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_hu'\
                   and s.id <= i.segment_limit
}

source xtm_hu_delta : source1
{
         sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_hu'\
                   and s.id > i.segment_limit
}

# Source for italian language
source xtm_it : source1
{
         sql_query_pre = \
                 update xtm_index x\
                 set segment_limit = (\
                         select coalesce(max(s.id),0)\
                         from xtm_segment s\
                         inner join xtm_locale l\
                           on l.id = s.locale_id\
                         inner join xtm_index i\
                           on i.id = l.main_index_id\
                         where i.name = x.name\
                 )\
                 where name = 'index_it'

         sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_it'\
                   and s.id <= i.segment_limit
```

```
}

source xtm_it_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_it'\
                   and s.id > i.segment_limit
}

# Source for norwegian language
source xtm_no : source1
{
         sql_query_pre = \
                 update xtm_index x\
                 set segment_limit = (\
                         select coalesce(max(s.id),0)\
                         from xtm_segment s\
                         inner join xtm_locale l\
                           on l.id = s.locale_id\
                         inner join xtm_index i\
                           on i.id = l.main_index_id\
                         where i.name = x.name\
                 )\
                 where name = 'index_no'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_no'\
                   and s.id <= i.segment_limit
}

source xtm_no_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_no'\
                   and s.id > i.segment_limit
}

# Source for portuguese language
source xtm_pt : source1
{
         sql_query_pre = \
                 update xtm_index x\
                 set segment_limit = (\
                         select coalesce(max(s.id),0)\
                         from xtm_segment s\
                         inner join xtm_locale l\
                           on l.id = s.locale_id\
                         inner join xtm_index i\
                           on i.id = l.main_index_id\
                         where i.name = x.name\
                 )\
```

```
                where name = 'index_pt'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_pt'\
                   and s.id <= i.segment_limit
}

source xtm_pt_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_pt'\
                   and s.id > i.segment_limit
}

# Source for romanian language
source xtm_ro : source1
{
         sql_query_pre = \
                 update xtm_index x\
                 set segment_limit = (\
                         select coalesce(max(s.id),0)\
                         from xtm_segment s\
                         inner join xtm_locale l\
                           on l.id = s.locale_id\
                         inner join xtm_index i\
                           on i.id = l.main_index_id\
                         where i.name = x.name\
                 )\
                 where name = 'index_ro'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_ro'\
                   and s.id <= i.segment_limit
}

source xtm_ro_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                 from xtm_segment s\
                 inner join xtm_locale l\
                   on l.id = s.locale_id\
                 inner join xtm_index i\
                   on i.id = l.main_index_id\
                 where i.name = 'index_ro'\
                   and s.id > i.segment_limit
}

# Source for russian language
source xtm_ru : source1
{
```

```
        sql_query_pre = \
                update xtm_index x\
                set segment_limit = (\
                        select coalesce(max(s.id),0)\
                        from xtm_segment s\
                        inner join xtm_locale l\
                          on l.id = s.locale_id\
                        inner join xtm_index i\
                          on i.id = l.main_index_id\
                        where i.name = x.name\
                )\
                where name = 'index_ru'

        sql_query                               = \
               select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_ru'\
                  and s.id <= i.segment_limit
}

source xtm_ru_delta : source1
{
        sql_query                               = \
               select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_ru'\
                  and s.id > i.segment_limit
}

# Source for spanish language
source xtm_es : source1
{
        sql_query_pre = \
                update xtm_index x\
                set segment_limit = (\
                        select coalesce(max(s.id),0)\
                        from xtm_segment s\
                        inner join xtm_locale l\
                          on l.id = s.locale_id\
                        inner join xtm_index i\
                          on i.id = l.main_index_id\
                        where i.name = x.name\
                )\
                where name = 'index_es'

        sql_query                               = \
               select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_es'\
                  and s.id <= i.segment_limit
}

source xtm_es_delta : source1
{
        sql_query                               = \
               select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
```

```
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_es'\
                  and s.id > i.segment_limit
}

# Source for sweden language
source xtm_sv : source1
{
        sql_query_pre = \
                update xtm_index x\
                set segment_limit = (\
                        select coalesce(max(s.id),0)\
                        from xtm_segment s\
                        inner join xtm_locale l\
                          on l.id = s.locale_id\
                        inner join xtm_index i\
                          on i.id = l.main_index_id\
                        where i.name = x.name\
                )\
                where name = 'index_sv'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_sv'\
                  and s.id <= i.segment_limit
}

source xtm_sv_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_sv'\
                  and s.id > i.segment_limit
}

# Source for turkish language
source xtm_tr : source1
{
        sql_query_pre = \
                update xtm_index x\
                set segment_limit = (\
                        select coalesce(max(s.id),0)\
                        from xtm_segment s\
                        inner join xtm_locale l\
                          on l.id = s.locale_id\
                        inner join xtm_index i\
                          on i.id = l.main_index_id\
                        where i.name = x.name\
                )\
                where name = 'index_tr'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
```

```
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_tr'\
                  and s.id <= i.segment_limit
}

source xtm_tr_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_tr'\
                  and s.id > i.segment_limit
}

# Source for non suported libstemmer languages  - will not be stemmed

source xtm_default : source1
{
        sql_query_pre = \
                update xtm_index x\
                set segment_limit = (\
                        select coalesce(max(s.id),0)\
                        from xtm_segment s\
                        inner join xtm_locale l\
                          on l.id = s.locale_id\
                        inner join xtm_index i\
                          on i.id = l.main_index_id\
                        where i.name = x.name\
                )\
                where name = 'index_default'

        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_default'\
                  and s.id <= i.segment_limit

        sql_attr_uint = owner_id
        sql_attr_uint = locale_id


}

source xtm_default_delta : source1
{
        sql_query                               = \
                select s.id, s.owner_id, s.translationunit_id, s.locale_id, s.content, s.type, s.deleted \
                from xtm_segment s\
                inner join xtm_locale l\
                  on l.id = s.locale_id\
                inner join xtm_index i\
                  on i.id = l.main_index_id\
                where i.name = 'index_default'\
                  and s.id > i.segment_limit
}
```

```
#               #
# Index section #
#               #



index latin_charset
{
        source                  = empty_source
        path                    = /var/data/latin_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+FF10..U+FF19->0..9, U+FF21..U+FF3A->a..z, U+FF41..U+FF5A->a..z, 0..9,
A..Z->a..z, a..z, \
                U+00C0->a, U+00C1->a, U+00C2->a, U+00C3->a, U+00C4->a, U+00C5->a, U+00E0->a, U+00E1->a,
U+00E2->a, U+00E3->a, U+00E4->a, U+00E5->a, U+0100->a, U+0101->a, U+0102->a, U+0103->a, U+010300->a,
U+0104->a, U+0105->a, U+01CD->a, U+01CE->a, U+01DE->a, U+01DF->a, U+01E0->a, U+01E1->a, U+01FA->a, U+01FB-
>a, U+0200->a, U+0201->a, U+0202->a, U+0203->a, U+0226->a, U+0227->a, U+023A->a, U+0250->a, U+04D0->a,
U+04D1->a, U+1D2C->a, U+1D43->a, U+1D44->a, U+1D8F->a, U+1E00->a, U+1E01->a, U+1E9A->a, U+1EA0->a, U+1EA1-
>a, U+1EA2->a, U+1EA3->a, U+1EA4->a, U+1EA5->a, U+1EA6->a, U+1EA7->a, U+1EA8->a, U+1EA9->a, U+1EAA->a,
U+1EAB->a, U+1EAC->a, U+1EAD->a, U+1EAE->a, U+1EAF->a, U+1EB0->a, U+1EB1->a, U+1EB2->a, U+1EB3->a, U+1EB4-
>a, U+1EB5->a, U+1EB6->a, U+1EB7->a, U+2090->a, U+2C65->a, \
                U+0180->b, U+0181->b, U+0182->b, U+0183->b, U+0243->b, U+0253->b, U+0299->b, U+16D2->b,
U+1D03->b, U+1D2E->b, U+1D2F->b, U+1D47->b, U+1D6C->b, U+1D80->b, U+1E02->b, U+1E03->b, U+1E04->b, U+1E05-
>b, U+1E06->b, U+1E07->b, \
                U+00C7->c, U+00E7->c, U+0106->c, U+0107->c, U+0108->c, U+0109->c, U+010A->c, U+010B->c,
U+010C->c, U+010D->c, U+0187->c, U+0188->c, U+023B->c, U+023C->c, U+0255->c, U+0297->c, U+1D9C->c, U+1D9D-
>c, U+1E08->c, U+1E09->c, U+212D->c, U+2184->c, \
                U+010E->d, U+010F->d, U+0110->d, U+0111->d, U+0189->d, U+018A->d, U+018B->d, U+018C->d,
U+01C5->d, U+01F2->d, U+0221->d, U+0256->d, U+0257->d, U+1D05->d, U+1D30->d, U+1D48->d, U+1D6D->d, U+1D81-
>d, U+1D91->d, U+1E0A->d, U+1E0B->d, U+1E0C->d, U+1E0D->d, U+1E0E->d, U+1E0F->d, U+1E10->d, U+1E11->d,
U+1E12->d, U+1E13->d, \
                U+00C8->e, U+00C9->e, U+00CA->e, U+00CB->e, U+00E8->e, U+00E9->e, U+00EA->e, U+00EB->e,
U+0112->e, U+0113->e, U+0114->e, U+0115->e, U+0116->e, U+0117->e, U+0118->e, U+0119->e, U+011A->e, U+011B-
>e, U+018E->e, U+0190->e, U+01DD->e, U+0204->e, U+0205->e, U+0206->e, U+0207->e, U+0228->e, U+0229->e,
U+0246->e, U+0247->e, U+0258->e, U+025B->e, U+025C->e, U+025D->e, U+025E->e, U+029A->e, U+1D07->e, U+1D08-
>e, U+1D31->e, U+1D32->e, U+1D49->e, U+1D4B->e, U+1D4C->e, U+1D92->e, U+1D93->e, U+1D94->e, U+1D9F->e,
U+1E14->e, U+1E15->e, U+1E16->e, U+1E17->e, U+1E18->e, U+1E19->e, U+1E1A->e, U+1E1B->e, U+1E1C->e, U+1E1D-
>e, U+1EB8->e, U+1EB9->e, U+1EBA->e, U+1EBB->e, U+1EBC->e, U+1EBD->e, U+1EBE->e, U+1EBF->e, U+1EC0->e,
U+1EC1->e, U+1EC2->e, U+1EC3->e, U+1EC4->e, U+1EC5->e, U+1EC6->e, U+1EC7->e, U+2091->e, \
                U+0191->f, U+0192->f, U+1D6E->f, U+1D82->f, U+1DA0->f, U+1E1E->f, U+1E1F->f, \
                U+011C->g, U+011D->g, U+011E->g, U+011F->g, U+0120->g, U+0121->g, U+0122->g, U+0123->g,
U+0193->g, U+01E4->g, U+01E5->g, U+01E6->g, U+01E7->g, U+01F4->g, U+01F5->g, U+0260->g, U+0261->g, U+0262-
>g, U+029B->g, U+1D33->g, U+1D4D->g, U+1D77->g, U+1D79->g, U+1D83->g, U+1DA2->g, U+1E20->g, U+1E21->g, \
                U+0124->h, U+0125->h, U+0126->h, U+0127->h, U+021E->h, U+021F->h, U+0265->h, U+0266->h,
U+029C->h, U+02AE->h, U+02AF->h, U+02B0->h, U+02B1->h, U+1D34->h, U+1DA3->h, U+1E22->h, U+1E23->h, U+1E24-
>h, U+1E25->h, U+1E26->h, U+1E27->h, U+1E28->h, U+1E29->h, U+1E2A->h, U+1E2B->h, U+1E96->h, U+210C->h,
U+2C67->h, U+2C68->h, U+2C75->h, U+2C76->h, \
                U+00CC->i, U+00CD->i, U+00CE->i, U+00CF->i, U+00EC->i, U+00ED->i, U+00EE->i, U+00EF->i,
U+010309->i, U+0128->i, U+0129->i, U+012A->i, U+012B->i, U+012C->i, U+012D->i, U+012E->i, U+012F->i,
U+0130->i, U+0131->i, U+0197->i, U+01CF->i, U+01D0->i, U+0208->i, U+0209->i, U+020A->i, U+020B->i, U+0268-
>i, U+026A->i, U+040D->i, U+0418->i, U+0419->i, U+0438->i, U+0439->i, U+0456->i, U+1D09->i, U+1D35->i,
U+1D4E->i, U+1D62->i, U+1D7B->i, U+1D96->i, U+1DA4->i, U+1DA6->i, U+1DA7->i, U+1E2C->i, U+1E2D->i, U+1E2E-
>i, U+1E2F->i, U+1EC8->i, U+1EC9->i, U+1ECA->i, U+1ECB->i, U+2071->i, U+2111->i, \
                U+0134->j, U+0135->j, U+01C8->j, U+01CB->j, U+01F0->j, U+0237->j, U+0248->j, U+0249->j,
U+025F->j, U+0284->j, U+029D->j, U+02B2->j, U+1D0A->j, U+1D36->j, U+1DA1->j, U+1DA8->j, \
                U+0136->k, U+0137->k, U+0198->k, U+0199->k, U+01E8->k, U+01E9->k, U+029E->k, U+1D0B->k,
U+1D37->k, U+1D4F->k, U+1D84->k, U+1E30->k, U+1E31->k, U+1E32->k, U+1E33->k, U+1E34->k, U+1E35->k, U+2C69-
>k, U+2C6A->k, \
                U+0139->l, U+013A->l, U+013B->l, U+013C->l, U+013D->l, U+013E->l, U+013F->l, U+0140->l,
U+0141->l, U+0142->l, U+019A->l, U+01C8->l, U+0234->l, U+023D->l, U+026B->l, U+026C->l, U+026D->l, U+029F-
>l, U+02E1->l, U+1D0C->l, U+1D38->l, U+1D85->l, U+1DA9->l, U+1DAA->l, U+1DAB->l, U+1E36->l, U+1E37->l,
U+1E38->l, U+1E39->l, U+1E3A->l, U+1E3B->l, U+1E3C->l, U+1E3D->l, U+2C60->l, U+2C61->l, U+2C62->l, \
                U+019C->m, U+026F->m, U+0270->m, U+0271->m, U+1D0D->m, U+1D1F->m, U+1D39->m, U+1D50->m,
U+1D5A->m, U+1D6F->m, U+1D86->m, U+1DAC->m, U+1DAD->m, U+1E3E->m, U+1E3F->m, U+1E40->m, U+1E41->m, U+1E42-
>m, U+1E43->m, \
```

```
                U+00D1->n, U+00F1->n, U+0143->n, U+0144->n, U+0145->n, U+0146->n, U+0147->n, U+0148->n,
U+0149->n, U+019D->n, U+019E->n, U+01CB->n, U+01F8->n, U+01F9->n, U+0220->n, U+0235->n, U+0272->n, U+0273-
>n, U+0274->n, U+1D0E->n, U+1D3A->n, U+1D3B->n, U+1D70->n, U+1D87->n, U+1DAE->n, U+1DAF->n, U+1DB0->n,
U+1E44->n, U+1E45->n, U+1E46->n, U+1E47->n, U+1E48->n, U+1E49->n, U+1E4A->n, U+1E4B->n, U+207F->n, \
                U+00D2->o, U+00D3->o, U+00D4->o, U+00D5->o, U+00D6->o, U+00D8->o, U+00F2->o, U+00F3->o,
U+00F4->o, U+00F5->o, U+00F6->o, U+00F8->o, U+01030F->o, U+014C->o, U+014D->o, U+014E->o, U+014F->o,
U+0150->o, U+0151->o, U+0186->o, U+019F->o, U+01A0->o, U+01A1->o, U+01D1->o, U+01D2->o, U+01EA->o, U+01EB-
>o, U+01EC->o, U+01ED->o, U+01FE->o, U+01FF->o, U+020C->o, U+020D->o, U+020E->o, U+020F->o, U+022A->o,
U+022B->o, U+022C->o, U+022D->o, U+022E->o, U+022F->o, U+0230->o, U+0231->o, U+0254->o, U+0275->o, U+043E-
>o, U+04E6->o, U+04E7->o, U+04E8->o, U+04E9->o, U+04EA->o, U+04EB->o, U+1D0F->o, U+1D10->o, U+1D11->o,
U+1D12->o, U+1D13->o, U+1D16->o, U+1D17->o, U+1D3C->o, U+1D52->o, U+1D53->o, U+1D54->o, U+1D55->o, U+1D97-
>o, U+1DB1->o, U+1E4C->o, U+1E4D->o, U+1E4E->o, U+1E4F->o, U+1E50->o, U+1E51->o, U+1E52->o, U+1E53->o,
U+1ECC->o, U+1ECD->o, U+1ECE->o, U+1ECF->o, U+1ED0->o, U+1ED1->o, U+1ED2->o, U+1ED3->o, U+1ED4->o, U+1ED5-
>o, U+1ED6->o, U+1ED7->o, U+1ED8->o, U+1ED9->o, U+1EDA->o, U+1EDB->o, U+1EDC->o, U+1EDD->o, U+1EDE->o,
U+1EDF->o, U+1EE0->o, U+1EE1->o, U+1EE2->o, U+1EE3->o, U+2092->o, U+2C9E->o, U+2C9F->o, \
                U+01A4->p, U+01A5->p, U+1D18->p, U+1D3E->p, U+1D56->p, U+1D71->p, U+1D7D->p, U+1D88->p,
U+1E54->p, U+1E55->p, U+1E56->p, U+1E57->p, U+2C63->p, \
                U+024A->q, U+024B->q, U+02A0->q, \
                U+0154->r, U+0155->r, U+0156->r, U+0157->r, U+0158->r, U+0159->r, U+0210->r, U+0211->r,
U+0212->r, U+0213->r, U+024C->r, U+024D->r, U+0279->r, U+027A->r, U+027B->r, U+027C->r, U+027D->r, U+027E-
>r, U+027F->r, U+0280->r, U+0281->r, U+02B3->r, U+02B4->r, U+02B5->r, U+02B6->r, U+1D19->r, U+1D1A->r,
U+1D3F->r, U+1D63->r, U+1D72->r, U+1D73->r, U+1D89->r, U+1DCA->r, U+1E58->r, U+1E59->r, U+1E5A->r, U+1E5B-
>r, U+1E5C->r, U+1E5D->r, U+1E5E->r, U+1E5F->r, U+211C->r, U+2C64->r, \
                U+00DF->s, U+015A->s, U+015B->s, U+015C->s, U+015D->s, U+015E->s, U+015F->s, U+0160->s,
U+0161->s, U+017F->s, U+0218->s, U+0219->s, U+023F->s, U+0282->s, U+02E2->s, U+1D74->s, U+1D8A->s, U+1DB3-
>s, U+1E60->s, U+1E61->s, U+1E62->s, U+1E63->s, U+1E64->s, U+1E65->s, U+1E66->s, U+1E67->s, U+1E68->s,
U+1E69->s, U+1E9B->s, \
                U+0162->t, U+0163->t, U+0164->t, U+0165->t, U+0166->t, U+0167->t, U+01AB->t, U+01AC->t,
U+01AD->t, U+01AE->t, U+021A->t, U+021B->t, U+0236->t, U+023E->t, U+0287->t, U+0288->t, U+1D1B->t, U+1D40-
>t, U+1D57->t, U+1D75->t, U+1DB5->t, U+1E6A->t, U+1E6B->t, U+1E6C->t, U+1E6D->t, U+1E6E->t, U+1E6F->t,
U+1E70->t, U+1E71->t, U+1E97->t, U+2C66->t, \
                U+00D9->u, U+00DA->u, U+00DB->u, U+00DC->u, U+00F9->u, U+00FA->u, U+00FB->u, U+00FC->u,
U+010316->u, U+0168->u, U+0169->u, U+016A->u, U+016B->u, U+016C->u, U+016D->u, U+016E->u, U+016F->u,
U+0170->u, U+0171->u, U+0172->u, U+0173->u, U+01AF->u, U+01B0->u, U+01D3->u, U+01D4->u, U+01D5->u, U+01D6-
>u, U+01D7->u, U+01D8->u, U+01D9->u, U+01DA->u, U+01DB->u, U+01DC->u, U+0214->u, U+0215->u, U+0216->u,
U+0217->u, U+0244->u, U+0289->u, U+1D1C->u, U+1D1D->u, U+1D1E->u, U+1D41->u, U+1D58->u, U+1D59->u, U+1D64-
>u, U+1D7E->u, U+1D99->u, U+1DB6->u, U+1DB8->u, U+1E72->u, U+1E73->u, U+1E74->u, U+1E75->u, U+1E76->u,
U+1E77->u, U+1E78->u, U+1E79->u, U+1E7A->u, U+1E7B->u, U+1EE4->u, U+1EE5->u, U+1EE6->u, U+1EE7->u, U+1EE8-
>u, U+1EE9->u, U+1EEA->u, U+1EEB->u, U+1EEC->u, U+1EED->u, U+1EEE->u, U+1EEF->u, U+1EF0->u, U+1EF1->u, \
                U+01B2->v, U+0245->v, U+028B->v, U+028C->v, U+1D20->v, U+1D5B->v, U+1D65->v, U+1D8C->v,
U+1DB9->v, U+1DBA->v, U+1E7C->v, U+1E7D->v, U+1E7E->v, U+1E7F->v, U+2C74->v, \
                U+0174->w, U+0175->w, U+028D->w, U+02B7->w, U+1D21->w, U+1D42->w, U+1E80->w, U+1E81->w,
U+1E82->w, U+1E83->w, U+1E84->w, U+1E85->w, U+1E86->w, U+1E87->w, U+1E88->w, U+1E89->w, U+1E98->w, \
                U+02E3->x, U+1D8D->x, U+1E8A->x, U+1E8B->x, U+1E8C->x, U+1E8D->x, U+2093->x, \
                U+00DD->y, U+00FD->y, U+00FF->y, U+0176->y, U+0177->y, U+0178->y, U+01B3->y, U+01B4->y,
U+0232->y, U+0233->y, U+024E->y, U+024F->y, U+028E->y, U+028F->y, U+02B8->y, U+1E8E->y, U+1E8F->y, U+1E99-
>y, U+1EF2->y, U+1EF3->y, U+1EF4->y, U+1EF5->y, U+1EF6->y, U+1EF7->y, U+1EF8->y, U+1EF9->y, \
                U+0179->z, U+017A->z, U+017B->z, U+017C->z, U+017D->z, U+017E->z, U+01B5->z, U+01B6->z,
U+0224->z, U+0225->z, U+0240->z, U+0290->z, U+0291->z, U+1D22->z, U+1D76->z, U+1D8E->z, U+1DBB->z, U+1DBC-
>z, U+1DBD->z, U+1E90->z, U+1E91->z, U+1E92->z, U+1E93->z, U+1E94->z, U+1E95->z, U+2128->z, U+2C6B->z,
U+2C6C->z, \
                U+00C6->U+00E6, U+01E2->U+00E6, U+01E3->U+00E6, U+01FC->U+00E6, U+01FD->U+00E6, U+1D01-
>U+00E6, U+1D02->U+00E6, U+1D2D->U+00E6, U+1D46->U+00E6, U+00E6
}

index arabic_charset
{
        source                  = empty_source
        path                    = /var/data/arabic_charset
        charset_type              = utf-8
        html_strip              = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+0622->U+0627, U+0623->U+0627, U+0624->U+0648, U+0625->U+0627, U+0626-
>U+064A, U+06C0->U+06D5, U+06C2->U+06C1, U+06D3->U+06D2, U+FB50->U+0671, U+FB51->U+0671, U+FB52->U+067B,
U+FB53->U+067B, U+FB54->U+067B, U+FB56->U+067E, U+FB57->U+067E, U+FB58->U+067E, U+FB5A->U+0680, U+FB5B-
>U+0680, U+FB5C->U+0680, U+FB5E->U+067A, U+FB5F->U+067A, U+FB60->U+067A, U+FB62->U+067F, U+FB63->U+067F,
U+FB64->U+067F, U+FB66->U+0679, U+FB67->U+0679, U+FB68->U+0679, U+FB6A->U+06A4, U+FB6B->U+06A4, U+FB6C-
>U+06A4, U+FB6E->U+06A6, U+FB6F->U+06A6, U+FB70->U+06A6, U+FB72->U+0684, U+FB73->U+0684, U+FB74->U+0684,
```

```
U+FB76->U+0683, U+FB77->U+0683, U+FB78->U+0683, U+FB7A->U+0686, U+FB7B->U+0686, U+FB7C->U+0686, U+FB7E-
>U+0687, U+FB7F->U+0687, U+FB80->U+0687, U+FB82->U+068D, U+FB83->U+068D, U+FB84->U+068C, U+FB85->U+068C,
U+FB86->U+068E, U+FB87->U+068E, U+FB88->U+0688, U+FB89->U+0688, U+FB8A->U+0698, U+FB8B->U+0698, U+FB8C-
>U+0691, U+FB8D->U+0691, U+FB8E->U+06A9, U+FB8F->U+06A9, U+FB90->U+06A9, U+FB92->U+06AF, U+FB93->U+06AF,
U+FB94->U+06AF, U+FB96->U+06B3, U+FB97->U+06B3, U+FB98->U+06B3, U+FB9A->U+06B1, U+FB9B->U+06B1, U+FB9C-
>U+06B1, U+FB9E->U+06BA, U+FB9F->U+06BA, U+FBA0->U+06BB, U+FBA1->U+06BB, U+FBA2->U+06BB, U+FBA4->U+06C0,
U+FBA5->U+06C0, U+FBA6->U+06C1, U+FBA7->U+06C1, U+FBA8->U+06C1, U+FBAA->U+06BE, U+FBAB->U+06BE, U+FBAC-
>U+06BE, U+FBAE->U+06D2, U+FBAF->U+06D2, U+FBB0->U+06D3, U+FBB1->U+06D3, U+FBD3->U+06AD, U+FBD4->U+06AD,
U+FBD5->U+06AD, U+FBD7->U+06C7, U+FBD8->U+06C7, U+FBD9->U+06C6, U+FBDA->U+06C6, U+FBDB->U+06C8, U+FBDC-
>U+06C8, U+FBDD->U+0677, U+FBDE->U+06CB, U+FBDF->U+06CB, U+FBE0->U+06C5, U+FBE1->U+06C5, U+FBE2->U+06C9,
U+FBE3->U+06C9, U+FBE4->U+06D0, U+FBE5->U+06D0, U+FBE6->U+06D0, U+FBE8->U+0649, U+FBFC->U+06CC, U+FBFD-
>U+06CC, U+FBFE->U+06CC, U+0621, U+0627..U+063A, U+0641..U+064A, U+0660..U+0669, U+066E, U+066F,
U+0671..U+06BF, U+06C1, U+06C3..U+06D2, U+06D5, U+06EE..U+06FC, U+06FF, U+0750..U+076D, U+FB55, U+FB59,
U+FB5D, U+FB61, U+FB65, U+FB69, U+FB6D, U+FB71, U+FB75, U+FB79, U+FB7D, U+FB81, U+FB91, U+FB95, U+FB99,
U+FB9D, U+FBA3, U+FBA9, U+FBAD, U+FBD6, U+FBE7, U+FBE9, U+FBFF
}

index armenian_charset
{
        source                  = empty_source
        path                    = /var/data/arabic_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+0531..U+0556->U+0561..U+0586, U+0561..U+0586, U+0587
}


index bengali_charset
{
        source                  = empty_source
        path                    = /var/data/bengali_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+09DC->U+09A1, U+09DD->U+09A2, U+09DF->U+09AF, U+09F0->U+09AC, U+09F1-
>U+09AC, U+0985..U+0990, U+0993..U+09B0, U+09B2, U+09B6..U+09B9, U+09CE, U+09E0, U+09E1, U+09E6..U+09EF
}


index cjk_charset
{
        source                  = empty_source
        path                    = /var/data/cjk_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+F900->U+8C48, U+F901->U+66F4, U+F902->U+8ECA, U+F903->U+8CC8, U+F904-
>U+6ED1, U+F905->U+4E32, U+F906->U+53E5, U+F907->U+9F9C, U+F908->U+9F9C, U+F909->U+5951, U+F90A->U+91D1,
U+F90B->U+5587, U+F90C->U+5948, U+F90D->U+61F6, U+F90E->U+7669, U+F90F->U+7F85, U+F910->U+863F, U+F911-
>U+87BA, U+F912->U+88F8, U+F913->U+908F, U+F914->U+6A02, U+F915->U+6D1B, U+F916->U+70D9, U+F917->U+73DE,
U+F918->U+843D, U+F919->U+916A, U+F91A->U+99F1, U+F91B->U+4E82, U+F91C->U+5375, U+F91D->U+6B04, U+F91E-
>U+721B, U+F91F->U+862D, U+F920->U+9E1E, U+F921->U+5D50, U+F922->U+6FEB, U+F923->U+85CD, U+F924->U+8964,
U+F925->U+62C9, U+F926->U+81D8, U+F927->U+881F, U+F928->U+5ECA, U+F929->U+6717, U+F92A->U+6D6A, U+F92B-
>U+72FC, U+F92C->U+90CE, U+F92D->U+4F86, U+F92E->U+51B7, U+F92F->U+52DE, U+F930->U+64C4, U+F931->U+6AD3,
U+F932->U+7210, U+F933->U+76E7, U+F934->U+8001, U+F935->U+8606, U+F936->U+865C, U+F937->U+8DEF, U+F938-
>U+9732, U+F939->U+9B6F, U+F93A->U+9DFA, U+F93B->U+788C, U+F93C->U+797F, U+F93D->U+7DA0, U+F93E->U+83C9,
U+F93F->U+9304, U+F940->U+9E7F, U+F941->U+8AD6, U+F942->U+58DF, U+F943->U+5F04, U+F944->U+7C60, U+F945-
>U+807E, U+F946->U+7262, U+F947->U+78CA, U+F948->U+8CC2, U+F949->U+96F7, U+F94A->U+58D8, U+F94B->U+5C62,
U+F94C->U+6A13, U+F94D->U+6DDA, U+F94E->U+6F0F, U+F94F->U+7D2F, U+F950->U+7E37, U+F951->U+964B, U+F952-
>U+52D2, U+F953->U+808B, U+F954->U+51DC, U+F955->U+51CC, U+F956->U+7A1C, U+F957->U+7DBE, U+F958->U+83F1,
U+F959->U+9675, U+F95A->U+8B80, U+F95B->U+62CF, U+F95C->U+6A02, U+F95D->U+8AFE, U+F95E->U+4E39, U+F95F-
>U+5BE7, U+F960->U+6012, U+F961->U+7387, U+F962->U+7570, U+F963->U+5317, U+F964->U+78FB, U+F965->U+4FBF,
U+F966->U+5FA9, U+F967->U+4E0D, U+F968->U+6CCC, U+F969->U+6578, U+F96A->U+7D22, U+F96B->U+53C3, U+F96C-
>U+585E, U+F96D->U+7701, U+F96E->U+8449, U+F96F->U+8AAA, U+F970->U+6BBA, U+F971->U+8FB0, U+F972->U+6C88,
U+F973->U+62FE, U+F974->U+82E5, U+F975->U+63A0, U+F976->U+7565, U+F977->U+4EAE, U+F978->U+5169, U+F979-
```

>U+51C9, U+F97A->U+6881, U+F97B->U+7CE7, U+F97C->U+826F, U+F97D->U+8AD2, U+F97E->U+91CF, U+F97F->U+52F5,
U+F980->U+5442, U+F981->U+5973, U+F982->U+5EEC, U+F983->U+65C5, U+F984->U+6FFE, U+F985->U+792A, U+F986-
>U+95AD, U+F987->U+9A6A, U+F988->U+9E97, U+F989->U+9ECE, U+F98A->U+529B, U+F98B->U+66C6, U+F98C->U+6B77,\
 U+F98D->U+8F62, U+F98E->U+5E74, U+F98F->U+6190, U+F990->U+6200, U+F991->U+649A, U+F992->U+6F23, U+F993-
>U+7149, U+F994->U+7489, U+F995->U+79CA, U+F996->U+7DF4, U+F997->U+806F, U+F998->U+8F26, U+F999->U+84EE,
U+F99A->U+9023, U+F99B->U+934A, U+F99C->U+5217, U+F99D->U+52A3, U+F99E->U+54BD, U+F99F->U+70C8, U+F9A0-
>U+88C2, U+F9A1->U+8AAA, U+F9A2->U+5EC9, U+F9A3->U+5FF5, U+F9A4->U+637B, U+F9A5->U+6BAE, U+F9A6->U+7C3E,
U+F9A7->U+7375, U+F9A8->U+4EE4, U+F9A9->U+56F9, U+F9AA->U+5BE7, U+F9AB->U+5DBA, U+F9AC->U+601C, U+F9AD-
>U+73B2, U+F9AE->U+7469, U+F9AF->U+7F9A, U+F9B0->U+8046, U+F9B1->U+9234, U+F9B2->U+96F6, U+F9B3->U+9748,
U+F9B4->U+9818, U+F9B5->U+4F8B, U+F9B6->U+79AE, U+F9B7->U+91B4, U+F9B8->U+96B8, U+F9B9->U+60E1, U+F9BA-
>U+4E86, U+F9BB->U+50DA, U+F9BC->U+5BEE, U+F9BD->U+5C3F, U+F9BE->U+6599, U+F9BF->U+6A02, U+F9C0->U+71CE,
U+F9C1->U+7642, U+F9C2->U+84FC, U+F9C3->U+907C, U+F9C4->U+9F8D, U+F9C5->U+6688, U+F9C6->U+962E, U+F9C7-
>U+5289, U+F9C8->U+677B, U+F9C9->U+67F3, U+F9CA->U+6D41, U+F9CB->U+6E9C, U+F9CC->U+7409, U+F9CD->U+7559,
U+F9CE->U+786B, U+F9CF->U+7D10, U+F9D0->U+985E, U+F9D1->U+516D, U+F9D2->U+622E, U+F9D3->U+9678, U+F9D4-
>U+502B, U+F9D5->U+5D19, U+F9D6->U+6DEA, U+F9D7->U+8F2A, U+F9D8->U+5F8B, U+F9D9->U+6144, U+F9DA->U+6817,
U+F9DB->U+7387, U+F9DC->U+9686, U+F9DD->U+5229, U+F9DE->U+540F, U+F9DF->U+5C65, U+F9E0->U+6613, U+F9E1-
>U+674E, U+F9E2->U+68A8, U+F9E3->U+6CE5, U+F9E4->U+7406, U+F9E5->U+75E2, U+F9E6->U+7F79, U+F9E7->U+88CF,
U+F9E8->U+88E1, U+F9E9->U+91CC, U+F9EA->U+96E2, U+F9EB->U+533F, U+F9EC->U+6EBA, U+F9ED->U+541D, U+F9EE-
>U+71D0, U+F9EF->U+7498, U+F9F0->U+85FA, U+F9F1->U+96A3, U+F9F2->U+9C57, U+F9F3->U+9E9F, U+F9F4->U+6797,
U+F9F5->U+6DCB, U+F9F6->U+81E8, U+F9F7->U+7ACB, U+F9F8->U+7B20, U+F9F9->U+7C92, U+F9FA->U+72C0, U+F9FB-
>U+7099, U+F9FC->U+8B58, U+F9FD->U+4EC0, U+F9FE->U+8336, U+F9FF->U+523A, U+FA00->U+5207, U+FA01->U+5EA6,
U+FA02->U+62D3, U+FA03->U+7CD6, U+FA04->U+5B85, U+FA05->U+6D1E, U+FA06->U+66B4, U+FA07->U+8F3B, U+FA08-
>U+884C, U+FA09->U+964D, U+FA0A->U+898B, U+FA0B->U+5ED3, U+FA0C->U+5140, U+FA0D->U+55C0, U+FA10->U+585A,
U+FA12->U+6674, U+FA15->U+51DE, U+FA16->U+732A, U+FA17->U+76CA, U+FA18->U+793C, U+FA19->U+795E, U+FA1A-
>U+7965, U+FA1B->U+798F, U+FA1C->U+9756, U+FA1D->U+7CBE, U+FA1E->U+7FBD, U+FA20->U+8612, U+FA22->U+8AF8,\
 U+FA25->U+9038, U+FA26->U+90FD, U+FA2A->U+98EF, U+FA2B->U+98FC, U+FA2C->U+9928, U+FA2D->U+9DB4, U+FA30-
>U+4FAE, U+FA31->U+50E7, U+FA32->U+514D, U+FA33->U+52C9, U+FA34->U+52E4, U+FA35->U+5351, U+FA36->U+559D,
U+FA37->U+5606, U+FA38->U+5668, U+FA39->U+5840, U+FA3A->U+58A8, U+FA3B->U+5C64, U+FA3C->U+5C6E, U+FA3D-
>U+6094, U+FA3E->U+6168, U+FA3F->U+618E, U+FA40->U+61F2, U+FA41->U+654F, U+FA42->U+65E2, U+FA43->U+6691,
U+FA44->U+6885, U+FA45->U+6D77, U+FA46->U+6E1A, U+FA47->U+6F22, U+FA48->U+716E, U+FA49->U+722B, U+FA4A-
>U+7422, U+FA4B->U+7891, U+FA4C->U+793E, U+FA4D->U+7949, U+FA4E->U+7948, U+FA4F->U+7950, U+FA50->U+7956,
U+FA51->U+795D, U+FA52->U+798D, U+FA53->U+798E, U+FA54->U+7A40, U+FA55->U+7A81, U+FA56->U+7BC0, U+FA57-
>U+7DF4, U+FA58->U+7E09, U+FA59->U+7E41, U+FA5A->U+7F72, U+FA5B->U+8005, U+FA5C->U+81ED, U+FA5D->U+8279,
U+FA5E->U+8279, U+FA5F->U+8457, U+FA60->U+8910, U+FA61->U+8996, U+FA62->U+8B01, U+FA63->U+8B39, U+FA64-
>U+8CD3, U+FA65->U+8D08, U+FA66->U+8FB6, U+FA67->U+9038, U+FA68->U+96E3, U+FA69->U+97FF, U+FA6A->U+983B,
U+FA70->U+4E26, U+FA71->U+51B5, U+FA72->U+5168, U+FA73->U+4F80, U+FA74->U+5145, U+FA75->U+5180, U+FA76-
>U+52C7, U+FA77->U+52FA, U+FA78->U+559D, U+FA79->U+5555, U+FA7A->U+5599, U+FA7B->U+55E2, U+FA7C->U+585A,
U+FA7D->U+58B3, U+FA7E->U+5944, U+FA7F->U+5954, U+FA80->U+5A62, U+FA81->U+5B28, U+FA82->U+5ED2, U+FA83-
>U+5ED9, U+FA84->U+5F69, U+FA85->U+5FAD, U+FA86->U+60D8, U+FA87->U+614E, U+FA88->U+6108, U+FA89->U+618E,
U+FA8A->U+6160, U+FA8B->U+61F2, U+FA8C->U+6234, U+FA8D->U+63C4, U+FA8E->U+641C, U+FA8F->U+6452, U+FA90-
>U+6556, U+FA91->U+6674, U+FA92->U+6717, U+FA93->U+671B, U+FA94->U+6756, U+FA95->U+6B79, U+FA96->U+6BBA,
U+FA97->U+6D41, U+FA98->U+6EDB, U+FA99->U+6ECB, U+FA9A->U+6F22, U+FA9B->U+701E, U+FA9C->U+716E, U+FA9D-
>U+77A7, U+FA9E->U+7235, U+FA9F->U+72AF, U+FAA0->U+732A, U+FAA1->U+7471, U+FAA2->U+7506, U+FAA3->U+753B,
U+FAA4->U+761D, U+FAA5->U+761F, U+FAA6->U+76CA, U+FAA7->U+76DB, U+FAA8->U+76F4, U+FAA9->U+774A, U+FAAA-
>U+7740, U+FAAB->U+78CC, U+FAAC->U+7AB1, U+FAAD->U+7BC0, U+FAAE->U+7C7B, U+FAAF->U+7D5B, U+FAB0->U+7DF4,
U+FAB1->U+7F3E, U+FAB2->U+8005, U+FAB3->U+8352, U+FAB4->U+83EF, U+FAB5->U+8779, U+FAB6->U+8941, U+FAB7-
>U+8986, U+FAB8->U+8996, U+FAB9->U+8ABF, U+FABA->U+8AF8, U+FABB->U+8ACB, U+FABC->U+8B01, U+FABD->U+8AFE,
U+FABE->U+8AED, U+FABF->U+8B39, U+FAC0->U+8B8A, U+FAC1->U+8D08, U+FAC2->U+8F38, U+FAC3->U+9072, U+FAC4-
>U+9199, U+FAC5->U+9276, U+FAC6->U+967C, U+FAC7->U+96E3, U+FAC8->U+9756, U+FAC9->U+97DB, U+FACA->U+97FF,
U+FACB->U+980B, U+FACC->U+983B, U+FACD->U+9B12, U+FACE->U+9F9C, U+FACF->U+2284A, U+FAD0->U+22844,\
 U+FAD1->U+233D5, U+FAD2->U+3B9D, U+FAD3->U+4018, U+FAD4->U+4039, U+FAD5->U+25249, U+FAD6->U+25CD0, U+FAD7-
>U+27ED3, U+FAD8->U+9F43, U+FAD9->U+9F8E, U+2F800->U+4E3D, U+2F801->U+4E38, U+2F802->U+4E41, U+2F803-
>U+20122, U+2F804->U+4F60, U+2F805->U+4FAE, U+2F806->U+4FBB, U+2F807->U+5002, U+2F808->U+507A, U+2F809-
>U+5099, U+2F80A->U+50E7, U+2F80B->U+50CF, U+2F80C->U+349E, U+2F80D->U+2063A, U+2F80E->U+514D, U+2F80F-
>U+5154, U+2F810->U+5164, U+2F811->U+5177, U+2F812->U+2051C, U+2F813->U+34B9, U+2F814->U+5167, U+2F815-
>U+518D, U+2F816->U+2054B, U+2F817->U+5197, U+2F818->U+51A4, U+2F819->U+4ECC, U+2F81A->U+51AC, U+2F81B-
>U+51B5, U+2F81C->U+291DF, U+2F81D->U+51F5, U+2F81E->U+5203, U+2F81F->U+34DF, U+2F820->U+523B, U+2F821-
>U+5246, U+2F822->U+5272, U+2F823->U+5277, U+2F824->U+3515, U+2F825->U+52C7, U+2F826->U+52C9, U+2F827-
>U+52E4, U+2F828->U+52FA, U+2F829->U+5305, U+2F82A->U+5306, U+2F82B->U+5317, U+2F82C->U+5349, U+2F82D-
>U+5351, U+2F82E->U+535A, U+2F82F->U+5373, U+2F830->U+537D, U+2F831->U+537F, U+2F832->U+537F, U+2F833-
>U+537F, U+2F834->U+20A2C, U+2F835->U+7070, U+2F836->U+53CA, U+2F837->U+53DF, U+2F838->U+20B63, U+2F839-
>U+53EB, U+2F83A->U+53F1, U+2F83B->U+5406, U+2F83C->U+549E, U+2F83D->U+5438, U+2F83E->U+5448, U+2F83F-
>U+5468, U+2F840->U+54A2, U+2F841->U+54F6, U+2F842->U+5510, U+2F843->U+5553, U+2F844->U+5563, U+2F845-
>U+5584, U+2F846->U+5584, U+2F847->U+5599, U+2F848->U+55AB, U+2F849->U+55B3, U+2F84A->U+55C2, U+2F84B-
>U+5716, U+2F84C->U+5606, U+2F84D->U+5717, U+2F84E->U+5651, U+2F84F->U+5674, U+2F850->U+5207, U+2F851-
>U+58EE, U+2F852->U+57CE, U+2F853->U+57F4, U+2F854->U+580D, U+2F855->U+578B, U+2F856->U+5832, U+2F857-
>U+5831, U+2F858->U+58AC, U+2F859->U+214E4, U+2F85A->U+58F2, U+2F85B->U+58F7, U+2F85C->U+5906, U+2F85D-
>U+591A, U+2F85E->U+5922, U+2F85F->U+5962, U+2F860->U+216A8, U+2F861->U+216EA, U+2F862->U+59EC, U+2F863-
>U+5A1B, U+2F864->U+5A27, U+2F865->U+59D8, U+2F866->U+5A66, U+2F867->U+36EE, U+2F868->U+36FC, U+2F869-

>U+5B08, U+2F86A->U+5B3E, U+2F86B->U+5B3E, U+2F86C->U+219C8, U+2F86D->U+5BC3, U+2F86E->U+5BD8, U+2F86F-
>U+5BE7, U+2F870->U+5BF3, U+2F871->U+21B18, U+2F872->U+5BFF, U+2F873->U+5C06, U+2F874->U+5F53, U+2F875-
>U+5C22, U+2F876->U+3781, U+2F877->U+5C60, U+2F878->U+5C6E, U+2F879->U+5CC0, U+2F87A->U+5C8D, U+2F87B-
>U+21DE4, U+2F87C->U+5D43, U+2F87D->U+21DE6, U+2F87E->U+5D6E, U+2F87F->U+5D6B, U+2F880->U+5D7C, U+2F881-
>U+5DE1, U+2F882->U+5DE2, U+2F883->U+382F, U+2F884->U+5DFD, U+2F885->U+5E28, U+2F886->U+5E3D, U+2F887-
>U+5E69, U+2F888->U+3862, U+2F889->U+22183, U+2F88A->U+387C, U+2F88B->U+5EB0, U+2F88C->U+5EB3, U+2F88D-
>U+5EB6, U+2F88E->U+5ECA, U+2F88F->U+2A392, U+2F890->U+5EFE, U+2F891->U+22331, U+2F892->U+22331, U+2F893-
>U+8201, U+2F894->U+5F22, U+2F895->U+5F22, U+2F896->U+38C7, U+2F897->U+232B8, U+2F898->U+261DA, U+2F899-
>U+5F62, U+2F89A->U+5F6B, U+2F89B->U+38E3, U+2F89C->U+5F9A, U+2F89D->U+5FCD, U+2F89E->U+5FD7, U+2F89F-
>U+5FF9, U+2F8A0->U+6081, U+2F8A1->U+393A, U+2F8A2->U+391C, U+2F8A3->U+6094, U+2F8A4->U+226D4, U+2F8A5-
>U+60C7, U+2F8A6->U+6148, U+2F8A7->U+614C, U+2F8A8->U+614E, U+2F8A9->U+614C, U+2F8AA->U+617A, U+2F8AB-
>U+618E, U+2F8AC->U+61B2, U+2F8AD->U+61A4, U+2F8AE->U+61AF, U+2F8AF->U+61DE, U+2F8B0->U+61F2, U+2F8B1-
>U+61F6, U+2F8B2->U+6210, U+2F8B3->U+621B, U+2F8B4->U+625D, \
U+2F8B5->U+62B1, U+2F8B6->U+62D4, U+2F8B7->U+6350, U+2F8B8->U+22B0C, U+2F8B9->U+633D, U+2F8BA->U+62FC,
U+2F8BB->U+6368, U+2F8BC->U+6383, U+2F8BD->U+63E4, U+2F8BE->U+22BF1, U+2F8BF->U+6422, U+2F8C0->U+63C5,
U+2F8C1->U+63A9, U+2F8C2->U+3A2E, U+2F8C3->U+6469, U+2F8C4->U+647E, U+2F8C5->U+649D, U+2F8C6->U+6477,
U+2F8C7->U+3A6C, U+2F8C8->U+654F, U+2F8C9->U+656C, U+2F8CA->U+2300A, U+2F8CB->U+65E3, U+2F8CC->U+66F8,
U+2F8CD->U+6649, U+2F8CE->U+3B19, U+2F8CF->U+6691, U+2F8D0->U+3B08, U+2F8D1->U+3AE4, U+2F8D2->U+5192,
U+2F8D3->U+5195, U+2F8D4->U+6700, U+2F8D5->U+669C, U+2F8D6->U+80AD, U+2F8D7->U+43D9, U+2F8D8->U+6717,
U+2F8D9->U+671B, U+2F8DA->U+6721, U+2F8DB->U+675E, U+2F8DC->U+6753, U+2F8DD->U+233C3, U+2F8DE->U+3B49,
U+2F8DF->U+67FA, U+2F8E0->U+6785, U+2F8E1->U+6852, U+2F8E2->U+6885, U+2F8E3->U+2346D, U+2F8E4->U+688E,
U+2F8E5->U+681F, U+2F8E6->U+6914, U+2F8E7->U+3B9D, U+2F8E8->U+6942, U+2F8E9->U+69A3, U+2F8EA->U+69EA,
U+2F8EB->U+6AA8, U+2F8EC->U+236A3, U+2F8ED->U+6ADB, U+2F8EE->U+3C18, U+2F8EF->U+6B21, U+2F8F0->U+238A7,
U+2F8F1->U+6B54, U+2F8F2->U+3C4E, U+2F8F3->U+6B72, U+2F8F4->U+6B9F, U+2F8F5->U+6BBA, U+2F8F6->U+6BBB,
U+2F8F7->U+23A8D, U+2F8F8->U+21D0B, U+2F8F9->U+23AFA, U+2F8FA->U+6C4E, U+2F8FB->U+23CBC, U+2F8FC->U+6CBF,
U+2F8FD->U+6CCD, U+2F8FE->U+6C67, U+2F8FF->U+6D16, U+2F900->U+6D3E, U+2F901->U+6D77, U+2F902->U+6D41,
U+2F903->U+6D69, U+2F904->U+6D78, U+2F905->U+6D85, U+2F906->U+23D1E, U+2F907->U+6D34, U+2F908->U+6E2F,
U+2F909->U+6E6E, U+2F90A->U+3D33, U+2F90B->U+6ECB, U+2F90C->U+6EC7, U+2F90D->U+23ED1, U+2F90E->U+6DF9,
U+2F90F->U+6F6E, U+2F910->U+23F5E, U+2F911->U+23F8E, U+2F912->U+6FC6, U+2F913->U+7039, U+2F914->U+701E,
U+2F915->U+701B, U+2F916->U+3D96, U+2F917->U+704A, U+2F918->U+707D, U+2F919->U+7077, U+2F91A->U+70AD,
U+2F91B->U+20525, U+2F91C->U+7145, U+2F91D->U+24263, U+2F91E->U+719C, U+2F91F->U+243AB, U+2F920->U+7228,
U+2F921->U+7235, U+2F922->U+7250, U+2F923->U+24608, U+2F924->U+7280, U+2F925->U+7295, U+2F926->U+24735,
U+2F927->U+24814, U+2F928->U+737A, U+2F929->U+738B, U+2F92A->U+3EAC, U+2F92B->U+73A5, U+2F92C->U+3EB8,
U+2F92D->U+3EB8, U+2F92E->U+7447, U+2F92F->U+745C, U+2F930->U+7471, U+2F931->U+7485, U+2F932->U+74CA,
U+2F933->U+3F1B, U+2F934->U+7524, U+2F935->U+24C36, U+2F936->U+753E, U+2F937->U+24C92, U+2F938->U+7570,
U+2F939->U+2219F, U+2F93A->U+7610, U+2F93B->U+24FA1, U+2F93C->U+24FB8, U+2F93D->U+25044, U+2F93E->U+3FFC,
U+2F93F->U+4008, U+2F940->U+76F4, U+2F941->U+250F3, U+2F942->U+250F2, U+2F943->U+25119, U+2F944->U+25133,
U+2F945->U+771E, U+2F946->U+771F, U+2F947->U+771F, U+2F948->U+774A, U+2F949->U+4039, U+2F94A->U+778B,
U+2F94B->U+4046, U+2F94C->U+4096, U+2F94D->U+2541D, U+2F94E->U+784E, U+2F94F->U+788C, U+2F950->U+78CC,
U+2F951->U+40E3, U+2F952->U+25626, U+2F953->U+7956, U+2F954->U+2569A, U+2F955->U+256C5, U+2F956->U+798F,
U+2F957->U+79EB, U+2F958->U+412F, U+2F959->U+7A40, U+2F95A->U+7A4A, U+2F95B->U+7A4F, U+2F95C->U+2597C,
U+2F95D->U+25AA7, U+2F95E->U+25AA7, U+2F95F->U+7AEE, \
U+2F960->U+4202, U+2F961->U+25BAB, U+2F962->U+7BC6, U+2F963->U+7BC9, U+2F964->U+4227, U+2F965->U+25C80,
U+2F966->U+7CD2, U+2F967->U+42A0, U+2F968->U+7CE8, U+2F969->U+7CE3, U+2F96A->U+7D00, U+2F96B->U+25F86,
U+2F96C->U+7D63, U+2F96D->U+4301, U+2F96E->U+7DC7, U+2F96F->U+7E02, U+2F970->U+7E45, U+2F971->U+4334,
U+2F972->U+26228, U+2F973->U+26247, U+2F974->U+4359, U+2F975->U+262D9, U+2F976->U+7F7A, U+2F977->U+2633E,
U+2F978->U+7F95, U+2F979->U+7FFA, U+2F97A->U+8005, U+2F97B->U+264DA, U+2F97C->U+26523, U+2F97D->U+8060,
U+2F97E->U+265A8, U+2F97F->U+8070, U+2F980->U+2335F, U+2F981->U+43D5, U+2F982->U+80B2, U+2F983->U+8103,
U+2F984->U+440B, U+2F985->U+813E, U+2F986->U+5AB5, U+2F987->U+267A7, U+2F988->U+267B5, U+2F989->U+23393,
U+2F98A->U+2339C, U+2F98B->U+8201, U+2F98C->U+8204, U+2F98D->U+8F9E, U+2F98E->U+446B, U+2F98F->U+8291,
U+2F990->U+828B, U+2F991->U+829D, U+2F992->U+52B3, U+2F993->U+82B1, U+2F994->U+82B3, U+2F995->U+82BD,
U+2F996->U+82E6, U+2F997->U+26B3C, U+2F998->U+82E5, U+2F999->U+831D, U+2F99A->U+8363, U+2F99B->U+83AD,
U+2F99C->U+8323, U+2F99D->U+83BD, U+2F99E->U+83E7, U+2F99F->U+8457, U+2F9A0->U+8353, U+2F9A1->U+83CA,
U+2F9A2->U+83CC, U+2F9A3->U+83DC, U+2F9A4->U+26C36, U+2F9A5->U+26D6B, U+2F9A6->U+26CD5, U+2F9A7->U+452B,
U+2F9A8->U+84F1, U+2F9A9->U+84F3, U+2F9AA->U+8516, U+2F9AB->U+273CA, U+2F9AC->U+8564, U+2F9AD->U+26F2C,
U+2F9AE->U+455D, U+2F9AF->U+4561, U+2F9B0->U+26FB1, U+2F9B1->U+270D2, U+2F9B2->U+456B, U+2F9B3->U+8650,
U+2F9B4->U+865C, U+2F9B5->U+8667, U+2F9B6->U+8669, U+2F9B7->U+86A9, U+2F9B8->U+8688, U+2F9B9->U+870E,
U+2F9BA->U+86E2, U+2F9BB->U+8779, U+2F9BC->U+8728, U+2F9BD->U+876B, U+2F9BE->U+8786, U+2F9BF->U+45D7,
U+2F9C0->U+87E1, U+2F9C1->U+8801, U+2F9C2->U+45F9, U+2F9C3->U+8860, U+2F9C4->U+8863, U+2F9C5->U+27667,
U+2F9C6->U+88D7, U+2F9C7->U+88DE, U+2F9C8->U+4635, U+2F9C9->U+88FA, U+2F9CA->U+34BB, U+2F9CB->U+278AE,
U+2F9CC->U+27966, U+2F9CD->U+46BE, U+2F9CE->U+46C7, U+2F9CF->U+8AA0, U+2F9D0->U+8AED, U+2F9D1->U+8B8A,
U+2F9D2->U+8C55, U+2F9D3->U+27CA8, U+2F9D4->U+8CAB, U+2F9D5->U+8CC1, U+2F9D6->U+8D1B, U+2F9D7->U+8D77,
U+2F9D8->U+27F2F, U+2F9D9->U+20804, U+2F9DA->U+8DCB, U+2F9DB->U+8DBC, U+2F9DC->U+8DF0, U+2F9DD->U+208DE,
U+2F9DE->U+8ED4, U+2F9DF->U+8F38, U+2F9E0->U+285D2, U+2F9E1->U+285ED, U+2F9E2->U+9094, U+2F9E3->U+90F1,
U+2F9E4->U+9111, U+2F9E5->U+2872E, U+2F9E6->U+911B, U+2F9E7->U+9238, U+2F9E8->U+92D7, U+2F9E9->U+92D8,
U+2F9EA->U+927C, U+2F9EB->U+93F9, U+2F9EC->U+9415, U+2F9ED->U+28BFA, U+2F9EE->U+958B, U+2F9EF->U+4995,
U+2F9F0->U+95B7, U+2F9F1->U+28D77, U+2F9F2->U+49E6, U+2F9F3->U+96C3, U+2F9F4->U+5DB2, U+2F9F5->U+9723,
U+2F9F6->U+29145, U+2F9F7->U+2921A, U+2F9F8->U+4A6E, U+2F9F9->U+4A76, U+2F9FA->U+97E0, U+2F9FB->U+2940A,
U+2F9FC->U+4AB2, U+2F9FD->U+29496, U+2F9FE->U+980B, U+2F9FF->U+980B, U+2FA00->U+9829, U+2FA01->U+295B6,

U+2FA02->U+98E2, U+2FA03->U+4B33, U+2FA04->U+9929, U+2FA05->U+99A7, U+2FA06->U+99C2, U+2FA07->U+99FE,
U+2FA08->U+4BCE, U+2FA09->U+29B30, U+2FA0A->U+9B12, U+2FA0B->U+9C40, U+2FA0C->U+9CFD, U+2FA0D->U+4CCE,
U+2FA0E->U+4CED, U+2FA0F->U+9D67, U+2FA10->U+2A0CE, U+2FA11->U+4CF8, U+2FA12->U+2A105, U+2FA13->U+2A20E,
U+2FA14->U+2A291, U+2FA15->U+9EBB, U+2FA16->U+4D56, U+2FA17->U+9EF9, U+2FA18->U+9EFE, U+2FA19->U+9F05,
U+2FA1A->U+9F0F, U+2FA1B->U+9F16, U+2FA1C->U+9F3B, U+2FA1D->U+2A600, U+2F00->U+4E00, U+2F01->U+4E28,
U+2F02->U+4E36, U+2F03->U+4E3F, U+2F04->U+4E59, U+2F05->U+4E85, U+2F06->U+4E8C, U+2F07->U+4EA0, U+2F08-
>U+4EBA, U+2F09->U+513F, U+2F0A->U+5165, U+2F0B->U+516B, U+2F0C->U+5182, U+2F0D->U+5196, U+2F0E->U+51AB,
U+2F0F->U+51E0, U+2F10->U+51F5, U+2F11->U+5200, U+2F12->U+529B, U+2F13->U+52F9, U+2F14->U+5315, U+2F15-
>U+531A, U+2F16->U+5338, U+2F17->U+5341, U+2F18->U+535C, U+2F19->U+5369, U+2F1A->U+5382, U+2F1B->U+53B6,
U+2F1C->U+53C8, U+2F1D->U+53E3, U+2F1E->U+56D7, U+2F1F->U+571F, U+2F20->U+58EB, U+2F21->U+5902, U+2F22-
>U+590A, U+2F23->U+5915, U+2F24->U+5927, U+2F25->U+5973, U+2F26->U+5B50, U+2F27->U+5B80, U+2F28->U+5BF8,
U+2F29->U+5C0F, U+2F2A->U+5C22, U+2F2B->U+5C38, U+2F2C->U+5C6E, U+2F2D->U+5C71, U+2F2E->U+5DDB, U+2F2F-
>U+5DE5, U+2F30->U+5DF1, U+2F31->U+5DFE, U+2F32->U+5E72, U+2F33->U+5E7A, U+2F34->U+5E7F, U+2F35->U+5EF4,
U+2F36->U+5EFE, U+2F37->U+5F0B,\
 U+2F38->U+5F13, U+2F39->U+5F50, U+2F3A->U+5F61, U+2F3B->U+5F73, U+2F3C->U+5FC3, U+2F3D->U+6208, U+2F3E-
>U+6236, U+2F3F->U+624B, U+2F40->U+652F, U+2F41->U+6534, U+2F42->U+6587, U+2F43->U+6597, U+2F44->U+65A4,
U+2F45->U+65B9, U+2F46->U+65E0, U+2F47->U+65E5, U+2F48->U+66F0, U+2F49->U+6708, U+2F4A->U+6728, U+2F4B-
>U+6B20, U+2F4C->U+6B62, U+2F4D->U+6B79, U+2F4E->U+6BB3, U+2F4F->U+6BCB, U+2F50->U+6BD4, U+2F51->U+6BDB,
U+2F52->U+6C0F, U+2F53->U+6C14, U+2F54->U+6C34, U+2F55->U+706B, U+2F56->U+722A, U+2F57->U+7236, U+2F58-
>U+723B, U+2F59->U+723F, U+2F5A->U+7247, U+2F5B->U+7259, U+2F5C->U+725B, U+2F5D->U+72AC, U+2F5E->U+7384,
U+2F5F->U+7389, U+2F60->U+74DC, U+2F61->U+74E6, U+2F62->U+7518, U+2F63->U+751F, U+2F64->U+7528, U+2F65-
>U+7530, U+2F66->U+758B, U+2F67->U+7592, U+2F68->U+7676, U+2F69->U+767D, U+2F6A->U+76AE, U+2F6B->U+76BF,
U+2F6C->U+76EE, U+2F6D->U+77DB, U+2F6E->U+77E2, U+2F6F->U+77F3, U+2F70->U+793A, U+2F71->U+79B8, U+2F72-
>U+79BE, U+2F73->U+7A74, U+2F74->U+7ACB, U+2F75->U+7AF9, U+2F76->U+7C73, U+2F77->U+7CF8, U+2F78->U+7F36,
U+2F79->U+7F51, U+2F7A->U+7F8A, U+2F7B->U+7FBD, U+2F7C->U+8001, U+2F7D->U+800C, U+2F7E->U+8012, U+2F7F-
>U+8033, U+2F80->U+807F, U+2F81->U+8089, U+2F82->U+81E3, U+2F83->U+81EA, U+2F84->U+81F3, U+2F85->U+81FC,
U+2F86->U+820C, U+2F87->U+821B, U+2F88->U+821F, U+2F89->U+826E, U+2F8A->U+8272, U+2F8B->U+8278, U+2F8C-
>U+864D, U+2F8D->U+866B, U+2F8E->U+8840, U+2F8F->U+884C, U+2F90->U+8863, U+2F91->U+897E, U+2F92->U+898B,
U+2F93->U+89D2, U+2F94->U+8A00, U+2F95->U+8C37, U+2F96->U+8C46, U+2F97->U+8C55, U+2F98->U+8C78, U+2F99-
>U+8C9D, U+2F9A->U+8D64, U+2F9B->U+8D70, U+2F9C->U+8DB3, U+2F9D->U+8EAB, U+2F9E->U+8ECA, U+2F9F->U+8F9B,
U+2FA0->U+8FB0, U+2FA1->U+8FB5, U+2FA2->U+9091, U+2FA3->U+9149, U+2FA4->U+91C6, U+2FA5->U+91CC, U+2FA6-
>U+91D1, U+2FA7->U+9577, U+2FA8->U+9580, U+2FA9->U+961C, U+2FAA->U+96B6, U+2FAB->U+96B9, U+2FAC->U+96E8,
U+2FAD->U+9751, U+2FAE->U+975E, U+2FAF->U+9762, U+2FB0->U+9769, U+2FB1->U+97CB, U+2FB2->U+97ED, U+2FB3-
>U+97F3, U+2FB4->U+9801, U+2FB5->U+98A8, U+2FB6->U+98DB, U+2FB7->U+98DF, U+2FB8->U+9996, U+2FB9->U+9999,
U+2FBA->U+99AC, U+2FBB->U+9AA8, U+2FBC->U+9AD8, U+2FBD->U+9ADF, U+2FBE->U+9B25, U+2FBF->U+9B2F, U+2FC0-
>U+9B32, U+2FC1->U+9B3C, U+2FC2->U+9B5A, U+2FC3->U+9CE5, U+2FC4->U+9E75, U+2FC5->U+9E7F, U+2FC6->U+9EA5,
U+2FC7->U+9EBB, U+2FC8->U+9EC3, U+2FC9->U+9ECD, U+2FCA->U+9ED1, U+2FCB->U+9EF9, U+2FCC->U+9EFD, U+2FCD-
>U+9F0E, U+2FCE->U+9F13, U+2FCF->U+9F20, U+2FD0->U+9F3B, U+2FD1->U+9F4A, U+2FD2->U+9F52, U+2FD3->U+9F8D,
U+2FD4->U+9F9C, U+2FD5->U+9FA0, U+3042->U+3041, U+3044->U+3043, U+3046->U+3045, U+3048->U+3047, U+304A-
>U+3049, U+304C->U+304B, U+304E->U+304D, U+3050->U+304F, U+3052->U+3051, U+3054->U+3053, U+3056->U+3055,
U+3058->U+3057, U+305A->U+3059, U+305C->U+305B, U+305E->U+305D, U+3060->U+305F, U+3062->U+3061, U+3064-
>U+3063, U+3065->U+3063, U+3067->U+3066, U+3069->U+3068, U+3070->U+306F, U+3071->U+306F, U+3073->U+3072,
U+3074->U+3072, U+3076->U+3075, U+3077->U+3075, U+3079->U+3078, U+307A->U+3078, U+307C->U+307B, U+307D-
>U+307B, U+3084->U+3083, U+3086->U+3085, U+3088->U+3087, U+308F->U+308E, U+3094->U+3046, U+3095->U+304B,
U+3096->U+3051, U+30A2->U+30A1, U+30A4->U+30A3, U+30A6->U+30A5, U+30A8->U+30A7,\
 U+30AA->U+30A9, U+30AC->U+30AB, U+30AE->U+30AD, U+30B0->U+30AF, U+30B2->U+30B1, U+30B4->U+30B3, U+30B6-
>U+30B5, U+30B8->U+30B7, U+30BA->U+30B9, U+30BC->U+30BB, U+30BE->U+30BD, U+30C0->U+30BF, U+30C2->U+30C1,
U+30C5->U+30C4, U+30C7->U+30C6, U+30C9->U+30C8, U+30D0->U+30CF, U+30D1->U+30CF, U+30D3->U+30D2, U+30D4-
>U+30D2, U+30D6->U+30D5, U+30D7->U+30D5, U+30D9->U+30D8, U+30DA->U+30D8, U+30DC->U+30DB, U+30DD->U+30DB,
U+30E4->U+30E3, U+30E6->U+30E5, U+30E8->U+30E7, U+30EF->U+30EE, U+30F4->U+30A6, U+30AB->U+30F5, U+30B1-
>U+30F6, U+30F7->U+30EF, U+30F8->U+30F0, U+30F9->U+30F1, U+30FA->U+30F2, U+30AF->U+31F0, U+30B7->U+31F1,
U+30B9->U+31F2, U+30C8->U+31F3, U+30CC->U+31F4, U+30CF->U+31F5, U+30D2->U+31F6, U+30D5->U+31F7, U+30D8-
>U+31F8, U+30DB->U+31F9, U+30E0->U+31FA, U+30E9->U+31FB, U+30EA->U+31FC, U+30EB->U+31FD, U+30EC->U+31FE,
U+30ED->U+31FF, U+FF66->U+30F2, U+FF67->U+30A1, U+FF68->U+30A3, U+FF69->U+30A5, U+FF6A->U+30A7, U+FF6B-
>U+30A9, U+FF6C->U+30E3, U+FF6D->U+30E5, U+FF6E->U+30E7, U+FF6F->U+30C3, U+FF71->U+30A1, U+FF72->U+30A3,
U+FF73->U+30A5, U+FF74->U+30A7, U+FF75->U+30A9, U+FF76->U+30AB, U+FF77->U+30AD, U+FF78->U+30AF, U+FF79-
>U+30B1, U+FF7A->U+30B3, U+FF7B->U+30B5, U+FF7C->U+30B7, U+FF7D->U+30B9, U+FF7E->U+30BB, U+FF7F->U+30BD,
U+FF80->U+30BF, U+FF81->U+30C1, U+FF82->U+30C3, U+FF83->U+30C6, U+FF84->U+30C8, U+FF85->U+30CA, U+FF86-
>U+30CB, U+FF87->U+30CC, U+FF88->U+30CD, U+FF89->U+30CE, U+FF8A->U+30CF, U+FF8B->U+30D2, U+FF8C->U+30D5,
U+FF8D->U+30D8, U+FF8E->U+30DB, U+FF8F->U+30DE, U+FF90->U+30DF, U+FF91->U+30E0, U+FF92->U+30E1, U+FF93-
>U+30E2, U+FF94->U+30E3, U+FF95->U+30E5, U+FF96->U+30E7, U+FF97->U+30E9, U+FF98->U+30EA, U+FF99->U+30EB,
U+FF9A->U+30EC, U+FF9B->U+30ED, U+FF9C->U+30EF, U+FF9D->U+30F3, U+FFA0->U+3164, U+FFA1->U+3131, U+FFA2-
>U+3132, U+FFA3->U+3133, U+FFA4->U+3134, U+FFA5->U+3135, U+FFA6->U+3136, U+FFA7->U+3137, U+FFA8->U+3138,
U+FFA9->U+3139, U+FFAA->U+313A, U+FFAB->U+313B, U+FFAC->U+313C, U+FFAD->U+313D, U+FFAE->U+313E, U+FFAF-
>U+313F, U+FFB0->U+3140, U+FFB1->U+3141, U+FFB2->U+3142, U+FFB3->U+3143, U+FFB4->U+3144, U+FFB5->U+3145,
U+FFB6->U+3146, U+FFB7->U+3147, U+FFB8->U+3148, U+FFB9->U+3149, U+FFBA->U+314A, U+FFBB->U+314B, U+FFBC-
>U+314C, U+FFBD->U+314D, U+FFBE->U+314E, U+FFC2->U+314F, U+FFC3->U+3150, U+FFC4->U+3151, U+FFC5->U+3152,
U+FFC6->U+3153, U+FFC7->U+3154, U+FFCA->U+3155, U+FFCB->U+3156, U+FFCC->U+3157, U+FFCD->U+3158, U+FFCE-
>U+3159, U+FFCF->U+315A, U+FFD2->U+315B, U+FFD3->U+315C, U+FFD4->U+315D, U+FFD5->U+315E, U+FFD6->U+315F,

```
U+FFD7->U+3160, U+FFDA->U+3161, U+FFDB->U+3162, U+FFDC->U+3163, U+3131->U+1100, U+3132->U+1101, U+3133-
>U+11AA, U+3134->U+1102, U+3135->U+11AC, U+3136->U+11AD, U+3137->U+1103, U+3138->U+1104, U+3139->U+1105,
U+313A->U+11B0, U+313B->U+11B1, U+313C->U+11B2, U+313D->U+11B3, U+313E->U+11B4, U+313F->U+11B5, U+3140-
>U+111A, U+3141->U+1106, U+3142->U+1107, U+3143->U+1108, U+3144->U+1121, U+3145->U+1109, U+3146->U+110A,
U+3147->U+110B, U+3148->U+110C, U+3149->U+110D, U+314A->U+110E, U+314B->U+110F, U+314C->U+1110, U+314D-
>U+1111, U+314E->U+1112, U+314F->U+1161, U+3150->U+1162, U+3151->U+1163, U+3152->U+1164, U+3153->U+1165,
U+3154->U+1166, U+3155->U+1167, U+3156->U+1168, U+3157->U+1169, U+3158->U+116A, U+3159->U+116B, U+315A-
>U+116C, U+315B->U+116D, U+315C->U+116E, U+315D->U+116F, U+315E->U+1170, U+315F->U+1171, U+3160->U+1172,
U+3161->U+1173, U+3162->U+1174, U+3163->U+1175, U+3165->U+1114, U+3166->U+1115, U+3167->U+11C7, U+3168-
>U+11C8, U+3169->U+11CC, U+316A->U+11CE, U+316B->U+11D3, U+316C->U+11D7, U+316D->U+11D9, U+316E->U+111C,
U+316F->U+11DD, U+3170->U+11DF, U+3171->U+111D, U+3172->U+111E, U+3173->U+1120, U+3174->U+1122, U+3175-
>U+1123, U+3176->U+1127, U+3177->U+1129, U+3178->U+112B, U+3179->U+112C, U+317A->U+112D, U+317B->U+112E,
U+317C->U+112F, U+317D->U+1132, U+317E->U+1136, U+317F->U+1140, U+3180->U+1147, U+3181->U+114C, U+3182-
>U+11F1, U+3183->U+11F2, U+3184->U+1157, U+3185->U+1158, U+3186->U+1159, U+3187->U+1184, U+3188->U+1185,
U+3189->U+1188, U+318A->U+1191, U+318B->U+1192, U+318C->U+1194, U+318D->U+119E, U+318E->U+11A1, U+A490-
>U+A408, U+A491->U+A1B9, U+4E00..U+9FBB, U+3400..U+4DB5, U+20000..U+2A6D6, U+FA0E, U+FA0F, U+FA11, U+FA13,
U+FA14, U+FA1F, U+FA21, U+FA23, U+FA24, U+FA27, U+FA28, U+FA29, U+3105..U+312C, U+31A0..U+31B7, U+3041,
U+3043, U+3045, U+3047, U+3049, U+304B, U+304D, U+304F, U+3051, U+3053, U+3055, U+3057, U+3059, U+305B,
U+305D, U+305F, U+3061, U+3063, U+3066, U+3068, U+306A..U+306F, U+3072, U+3075, U+3078, U+307B,
U+307E..U+3083, U+3085, U+3087, U+3089..U+308E, U+3090..U+3093, U+30A1, U+30A3, U+30A5, U+30A7, U+30A9,
U+30AD, U+30AF, U+30B3, U+30B5, U+30BB, U+30BD, U+30BF, U+30C1, U+30C3, U+30C4, U+30C6, U+30CA, U+30CB,
U+30CD, U+30CE, U+30DE, U+30DF, U+30E1, U+30E2,\
 U+30E3, U+30E5, U+30E7, U+30EE, U+30F0..U+30F3, U+30F5, U+30F6, U+31F0, U+31F1, U+31F2, U+31F3, U+31F4,
U+31F5, U+31F6, U+31F7, U+31F8, U+31F9, U+31FA, U+31FB, U+31FC, U+31FD, U+31FE, U+31FF, U+AC00..U+D7A3,
U+1100..U+1159, U+1161..U+11A2, U+11A8..U+11F9, U+A000..U+A48C, U+A492..U+A4C6
}


index coptic_charset
{
        source                  = empty_source
        path                    = /var/data/coptic_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+2C80->U+2C81, U+2C81, U+2C82->U+2C83, U+2C83, U+2C84->U+2C85, U+2C85,
U+2C86->U+2C87, U+2C87, U+2C88->U+2C89, U+2C89, U+2C8A->U+2C8B, U+2C8B, U+2C8C->U+2C8D, U+2C8D, U+2C8E-
>U+2C8F, U+2C8F, U+2C90->U+2C91, U+2C91, U+2C92->U+2C93, U+2C93, U+2C94->U+2C95, U+2C95, U+2C96->U+2C97,
U+2C97, U+2C98->U+2C99, U+2C99, U+2C9A->U+2C9B, U+2C9B, U+2C9C->U+2C9D, U+2C9D, U+2C9E->U+2C9F, U+2C9F,
U+2CA0->U+2CA1, U+2CA1, U+2CA2->U+2CA3, U+2CA3, U+2CA4->U+2CA5, U+2CA5, U+2CA6->U+2CA7, U+2CA7, U+2CA8-
>U+2CA9, U+2CA9, U+2CAA->U+2CAB, U+2CAB, U+2CAC->U+2CAD, U+2CAD, U+2CAE->U+2CAF, U+2CAF, U+2CB0->U+2CB1,
U+2CB1, U+2CB2->U+2CB3, U+2CB3, U+2CB4->U+2CB5, U+2CB5, U+2CB6->U+2CB7, U+2CB7, U+2CB8->U+2CB9, U+2CB9,
U+2CBA->U+2CBB, U+2CBB, U+2CBC->U+2CBD, U+2CBD, U+2CBE->U+2CBF, U+2CBF, U+2CC0->U+2CC1, U+2CC1, U+2CC2-
>U+2CC3, U+2CC3, U+2CC4->U+2CC5, U+2CC5, U+2CC6->U+2CC7, U+2CC7, U+2CC8->U+2CC9, U+2CC9, U+2CCA->U+2CCB,
U+2CCB, U+2CCC->U+2CCD, U+2CCD, U+2CCE->U+2CCF, U+2CCF, U+2CD0->U+2CD1, U+2CD1, U+2CD2->U+2CD3, U+2CD3,
U+2CD4->U+2CD5, U+2CD5, U+2CD6->U+2CD7, U+2CD7, U+2CD8->U+2CD9, U+2CD9, U+2CDA->U+2CDB, U+2CDB, U+2CDC-
>U+2CDD, U+2CDD, U+2CDE->U+2CDF, U+2CDF, U+2CE0->U+2CE1, U+2CE1, U+2CE2->U+2CE3, U+2CE3
}

index cyrillic_charset
{
        source                  = empty_source
        path                    = /var/data/cyrillic_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+0400->U+0435, U+0401->U+0435, U+0402->U+0452, U+0452, U+0403->U+0433,
U+0404->U+0454, U+0454, U+0405->U+0455, U+0455, U+0406->U+0456, U+0407->U+0456, U+0457->U+0456, U+0456,
U+0408..U+040B->U+0458..U+045B, U+0458..U+045B, U+040C->U+043A, U+040D->U+0438, U+040E->U+0443, U+040F-
>U+045F, U+045F, U+0450->U+0435, U+0451->U+0435, U+0453->U+0433, U+045C->U+043A, U+045D->U+0438, U+045E-
>U+0443, U+0460->U+0461, U+0461, U+0462->U+0463, U+0463, U+0464->U+0465, U+0465, U+0466->U+0467, U+0467,
U+0468->U+0469, U+0469, U+046A->U+046B, U+046B, U+046C->U+046D, U+046D, U+046E->U+046F, U+046F, U+0470-
>U+0471, U+0471, U+0472->U+0473, U+0473, U+0474->U+0475, U+0476->U+0475, U+0477->U+0475, U+0475, U+0478-
>U+0479, U+0479, U+047A->U+047B, U+047B, U+047C->U+047D, U+047D, U+047E->U+047F, U+047F, U+0480->U+0481,
U+0481, U+048A->U+0438, U+048B->U+0438, U+048C->U+044C, U+048D->U+044C, U+048E->U+0440, U+048F->U+0440,
U+0490->U+0433, U+0491->U+0433, U+0490->U+0433, U+0491->U+0433, U+0492->U+0433, U+0493->U+0433, U+0494-
>U+0433, U+0495->U+0433, U+0496->U+0436, U+0497->U+0436, U+0498->U+0437, U+0499->U+0437, U+049A->U+043A,
```

```
U+049B->U+043A, U+049C->U+043A, U+049D->U+043A, U+049E->U+043A, U+049F->U+043A, U+04A0->U+043A, U+04A1-
>U+043A, U+04A2->U+043D, U+04A3->U+043D, U+04A4->U+043D, U+04A5->U+043D, U+04A6->U+043F, U+04A7->U+043F,
U+04A8->U+04A9, U+04A9, U+04AA->U+0441, U+04AB->U+0441, U+04AC->U+0442, U+04AD->U+0442, U+04AE->U+0443,
U+04AF->U+0443, U+04B0->U+0443, U+04B1->U+0443, U+04B2->U+0445, U+04B3->U+0445, U+04B4->U+04B5, U+04B5,
U+04B6->U+0447, U+04B7->U+0447, U+04B8->U+0447, U+04B9->U+0447, U+04BA->U+04BB, U+04BB, U+04BC->U+04BD,
U+04BE->U+04BD, U+04BF->U+04BD, U+04BD, U+04C0->U+04CF, U+04CF, U+04C1->U+0436, U+04C2->U+0436, U+04C3-
>U+043A, U+04C4->U+043A, U+04C5->U+043B, U+04C6->U+043B, U+04C7->U+043D, U+04C8->U+043D, U+04C9->U+043D,
U+04CA->U+043D, U+04CB->U+0447, U+04CC->U+0447, U+04CD->U+043C, U+04CE->U+043C, U+04D0->U+0430, U+04D1-
>U+0430, U+04D2->U+0430, U+04D3->U+0430, U+04D4->U+00E6, U+04D5->U+00E6, U+04D6->U+0435, U+04D7->U+0435,
U+04D8->U+04D9, U+04DA->U+04D9, U+04DB->U+04D9, U+04D9, U+04DC->U+0436, U+04DD->U+0436, U+04DE->U+0437,
U+04DF->U+0437, U+04E0->U+04E1, U+04E1, U+04E2->U+0438, U+04E3->U+0438, U+04E4->U+0438, U+04E5->U+0438,
U+04E6->U+043E, U+04E7->U+043E, U+04E8->U+043E, U+04E9->U+043E, U+04EA->U+043E, U+04EB->U+043E, U+04EC-
>U+044D, U+04ED->U+044D, U+04EE->U+0443, U+04EF->U+0443, U+04F0->U+0443, U+04F1->U+0443, U+04F2->U+0443,
U+04F3->U+0443, U+04F4->U+0447, U+04F5->U+0447, U+04F6->U+0433, U+04F7->U+0433, U+04F8->U+044B, U+04F9-
>U+044B, U+04FA->U+0433, U+04FB->U+0433, U+04FC->U+0445, U+04FD->U+0445, U+04FE->U+0445, U+04FF->U+0445,
U+0410..U+0418->U+0430..U+0438, U+0419->U+0438, U+0430..U+0438, U+041A..U+042F->U+043A..U+044F,
U+043A..U+044F
}


index czech_charset
{
        source                  = empty_source
        path                    = /var/data/czech_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+0e1->a, U+0c1->a, U+10d->c, U+10c->c, U+10f->d, U+10e->d, U+0e9->e,
U+0c9->e, U+11b->e, U+11a->e, U+0ed->i, U+0cd->i, U+148->n, U+147->n, U+0f3->o, U+0d3->o, U+159->r, U+158-
>r, U+161->s, U+160->s, U+165->t, U+164->t, U+0fa->u, U+0da->u, U+16f->u, U+16e->u, U+0fd->y, U+0dd->y,
U+17e->z, U+17d->z
}


index devanagari_charset
{
        source                  = empty_source
        path                    = /var/data/devanagari_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+0929->U+0928, U+0931->U+0930, U+0934->U+0933, U+0958->U+0915, U+0959-
>U+0916, U+095A->U+0917, U+095B->U+091C, U+095C->U+0921, U+095D->U+0922, U+095E->U+092B, U+095F->U+092F,
U+0904..U+0928, U+092A..U+0930, U+0932, U+0933, U+0935..U+0939, U+0960, U+0961, U+0966..U+096F,
U+097B..U+097F
}

index german_charset
{
        source                  = empty_source
        path                    = /var/data/german_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+C4->U+E4, U+D6->U+F6, U+DC->U+FC, U+DF, U+E4, U+F6, U+FC
}

index georgian_charset
{
        source                  = empty_source
        path                    = /var/data/georgian_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+10FC->U+10DC, U+10D0..U+10FA, U+10A0..U+10C5->U+2D00..U+2D25,
```

```
U+2D00..U+2D25
}

index greek_charset
{
        source                  = empty_source
        path                    = /var/data/greek_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+0386->U+03B1, U+0388->U+03B5, U+0389->U+03B7, U+038A->U+03B9, U+038C-
>U+03BF, U+038E->U+03C5, U+038F->U+03C9, U+0390->U+03B9, U+03AA->U+03B9, U+03AB->U+03C5, U+03AC->U+03B1,
U+03AD->U+03B5, U+03AE->U+03B7, U+03AF->U+03B9, U+03B0->U+03C5, U+03CA->U+03B9, U+03CB->U+03C5, U+03CC-
>U+03BF, U+03CD->U+03C5, U+03CE->U+03C9, U+03D0->U+03B2, U+03D1->U+03B8, U+03D2->U+03C5, U+03D3->U+03C5,
U+03D4->U+03C5, U+03D5->U+03C6, U+03D6->U+03C0, U+03D8->U+03D9, U+03DA->U+03DB, U+03DC->U+03DD, U+03DE-
>U+03DF, U+03E0->U+03E1, U+03E2->U+03E3, U+03E4->U+03E5, U+03E6->U+03E7, U+03E8->U+03E9, U+03EA->U+03EB,
U+03EC->U+03ED, U+03EE->U+03EF, U+03F0->U+03BA, U+03F1->U+03C1, U+03F2->U+03C3, U+03F4->U+03B8, U+03F5-
>U+03B5, U+03F6->U+03B5, U+03F7->U+03F8, U+03F9->U+03C3, U+03FA->U+03FB, U+1F00->U+03B1, U+1F01->U+03B1,
U+1F02->U+03B1, U+1F03->U+03B1, U+1F04->U+03B1, U+1F05->U+03B1, U+1F06->U+03B1, U+1F07->U+03B1, U+1F08-
>U+03B1, U+1F09->U+03B1, U+1F0A->U+03B1, U+1F0B->U+03B1, U+1F0C->U+03B1, U+1F0D->U+03B1, U+1F0E->U+03B1,
U+1F0F->U+03B1, U+1F10->U+03B5, U+1F11->U+03B5, U+1F12->U+03B5, U+1F13->U+03B5, U+1F14->U+03B5, U+1F15-
>U+03B5, U+1F18->U+03B5, U+1F19->U+03B5, U+1F1A->U+03B5, U+1F1B->U+03B5, U+1F1C->U+03B5, U+1F1D->U+03B5,
U+1F20->U+03B7, U+1F21->U+03B7, U+1F22->U+03B7, U+1F23->U+03B7, U+1F24->U+03B7, U+1F25->U+03B7, U+1F26-
>U+03B7, U+1F27->U+03B7, U+1F28->U+03B7, U+1F29->U+03B7, U+1F2A->U+03B7, U+1F2B->U+03B7, U+1F2C->U+03B7,
U+1F2D->U+03B7, U+1F2E->U+03B7, U+1F2F->U+03B7, U+1F30->U+03B9, U+1F31->U+03B9, U+1F32->U+03B9, U+1F33-
>U+03B9, U+1F34->U+03B9, U+1F35->U+03B9, U+1F36->U+03B9, U+1F37->U+03B9, U+1F38->U+03B9, U+1F39->U+03B9,
U+1F3A->U+03B9, U+1F3B->U+03B9, U+1F3C->U+03B9, U+1F3D->U+03B9, U+1F3E->U+03B9, U+1F3F->U+03B9, U+1F40-
>U+03BF, U+1F41->U+03BF, U+1F42->U+03BF, U+1F43->U+03BF, U+1F44->U+03BF, U+1F45->U+03BF, U+1F48->U+03BF,
U+1F49->U+03BF, U+1F4A->U+03BF, U+1F4B->U+03BF, U+1F4C->U+03BF, U+1F4D->U+03BF, U+1F50->U+03C5, U+1F51-
>U+03C5, U+1F52->U+03C5, U+1F53->U+03C5, U+1F54->U+03C5, U+1F55->U+03C5, U+1F56->U+03C5, U+1F57->U+03C5,
U+1F59->U+03C5, U+1F5B->U+03C5, U+1F5D->U+03C5, U+1F5F->U+03C5, U+1F60->U+03C9, U+1F61->U+03C9, U+1F62-
>U+03C9, U+1F63->U+03C9, U+1F64->U+03C9, U+1F65->U+03C9, U+1F66->U+03C9, U+1F67->U+03C9, U+1F68->U+03C9,
U+1F69->U+03C9, U+1F6A->U+03C9, U+1F6B->U+03C9, U+1F6C->U+03C9, U+1F6D->U+03C9, U+1F6E->U+03C9, U+1F6F-
>U+03C9, U+1F70->U+03B1, U+1F71->U+03B1, U+1F72->U+03B5, U+1F73->U+03B5, U+1F74->U+03B7, U+1F75->U+03B7,
U+1F76->U+03B9, U+1F77->U+03B9, U+1F78->U+03BF, U+1F79->U+03BF, U+1F7A->U+03C5, U+1F7B->U+03C5, U+1F7C-
>U+03C9, U+1F7D->U+03C9, U+1F80->U+03B1, U+1F81->U+03B1, U+1F82->U+03B1, U+1F83->U+03B1, U+1F84->U+03B1,
U+1F85->U+03B1, U+1F86->U+03B1, U+1F87->U+03B1, U+1F88->U+03B1, U+1F89->U+03B1, U+1F8A->U+03B1, U+1F8B-
>U+03B1, U+1F8C->U+03B1, U+1F8D->U+03B1, U+1F8E->U+03B1, U+1F8F->U+03B1, U+1F90->U+03B7, U+1F91->U+03B7,
U+1F92->U+03B7, U+1F93->U+03B7, U+1F94->U+03B7, U+1F95->U+03B7, U+1F96->U+03B7, U+1F97->U+03B7, U+1F98-
>U+03B7, U+1F99->U+03B7, U+1F9A->U+03B7, U+1F9B->U+03B7, U+1F9C->U+03B7, U+1F9D->U+03B7, U+1F9E->U+03B7,
U+1F9F->U+03B7, U+1FA0->U+03C9, U+1FA1->U+03C9, U+1FA2->U+03C9, U+1FA3->U+03C9, U+1FA4->U+03C9, U+1FA5-
>U+03C9, U+1FA6->U+03C9, U+1FA7->U+03C9, U+1FA8->U+03C9, U+1FA9->U+03C9, U+1FAA->U+03C9, U+1FAB->U+03C9,
U+1FAC->U+03C9, U+1FAD->U+03C9, U+1FAE->U+03C9, U+1FAF->U+03C9, U+1FB0->U+03B1, U+1FB1->U+03B1, U+1FB2-
>U+03B1, U+1FB3->U+03B1, U+1FB4->U+03B1, U+1FB6->U+03B1, U+1FB7->U+03B1, U+1FB8->U+03B1, U+1FB9->U+03B1,
U+1FBA->U+03B1, U+1FBB->U+03B1, U+1FBC->U+03B1, U+1FC2->U+03B7, U+1FC3->U+03B7, U+1FC4->U+03B7, U+1FC6-
>U+03B7, U+1FC7->U+03B7, U+1FC8->U+03B5, U+1FC9->U+03B5, U+1FCA->U+03B7, U+1FCB->U+03B7, U+1FCC->U+03B7,
U+1FD0->U+03B9, U+1FD1->U+03B9, U+1FD2->U+03B9, U+1FD3->U+03B9, U+1FD6->U+03B9, U+1FD7->U+03B9, U+1FD8-
>U+03B9, U+1FD9->U+03B9, U+1FDA->U+03B9, U+1FDB->U+03B9, U+1FE0->U+03C5, U+1FE1->U+03C5, U+1FE2->U+03C5,
U+1FE3->U+03C5, U+1FE4->U+03C1, U+1FE5->U+03C1, U+1FE6->U+03C5, U+1FE7->U+03C5, U+1FE8->U+03C5, U+1FE9-
>U+03C5, U+1FEA->U+03C5, U+1FEB->U+03C5, U+1FEC->U+03C1, U+1FF2->U+03C9, U+1FF3->U+03C9, U+1FF4->U+03C9,
U+1FF6->U+03C9, U+1FF7->U+03C9, U+1FF8->U+03BF, U+1FF9->U+03BF, U+1FFA->U+03C9, U+1FFB->U+03C9, U+1FFC-
>U+03C9, U+0391..U+03A1->U+03B1..U+03C1, U+03B1..U+03C1, U+03A3..U+03A9->U+03C3..U+03C9, U+03C3..U+03C9,
U+03C2, U+03D9, U+03DB, U+03DD, U+03DF, U+03E1, U+03E3, U+03E5, U+03E7, U+03E9, U+03EB, U+03ED, U+03EF,
U+03F3, U+03F8, U+03FB
}

index gujarati_charset
{
        source                  = empty_source
        path                    = /var/data/gujarati_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+0A85..U+0A8C, U+0A8F, U+0A90, U+0A93..U+0AB0, U+0AB2, U+0AB3,
U+0AB5..U+0AB9, U+0AE0, U+0AE1, U+0AE6..U+0AEF
}
```

```
index gurmukhi_charset
{
        source                  = empty_source
        path                    = /var/data/gurmukhi_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+0A33->U+0A32, U+0A36->U+0A38, U+0A59->U+0A16, U+0A5A->U+0A17, U+0A5B-
>U+0A1C, U+0A5E->U+0A2B, U+0A05..U+0A0A, U+0A0F, U+0A10, U+0A13..U+0A28, U+0A2A..U+0A30, U+0A32, U+0A35,
U+0A38, U+0A39, U+0A5C, U+0A66..U+0A6F
}

index hebrew_charset
{
        source                  = empty_source
        path                    = /var/data/hebrew_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+FB1D->U+05D9, U+FB1F->U+05F2, U+FB20->U+05E2, U+FB21->U+05D0, U+FB22-
>U+05D3, U+FB23->U+05D4, U+FB24->U+05DB, U+FB25->U+05DC, U+FB26->U+05DD, U+FB27->U+05E8, U+FB28->U+05EA,
U+FB2A->U+05E9, U+FB2B->U+05E9, U+FB2C->U+05E9, U+FB2D->U+05E9, U+FB2E->U+05D0, U+FB2F->U+05D0, U+FB30-
>U+05D0, U+FB31->U+05D1, U+FB32->U+05D2, U+FB33->U+05D3, U+FB34->U+05D4, U+FB35->U+05D5, U+FB36->U+05D6,
U+FB38->U+05D8, U+FB39->U+05D9, U+FB3A->U+05DA, U+FB3B->U+05DB, U+FB3C->U+05DC, U+FB3E->U+05DE, U+FB40-
>U+05E0, U+FB41->U+05E1, U+FB43->U+05E3, U+FB44->U+05E4, U+FB46->U+05E6, U+FB47->U+05E7, U+FB48->U+05E8,
U+FB49->U+05E9, U+FB4A->U+05EA, U+FB4B->U+05D5, U+FB4C->U+05D1, U+FB4D->U+05DB, U+FB4E->U+05E4, U+FB4F-
>U+05D0, U+05D0..U+05F2
}

index hungarian_charset
{
        source                  = empty_source
        path                    = /var/data/hungarian_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+00C1->U+00E1, U+00C9->U+00E9, U+00CD->U+00ED, U+00D3->U+00F3, U+00D6-
>U+00F6, U+0150->U+0151, U+00DA->U+00FA, U+00DC->U+00FC, U+0170->U+0171, U+00E1, U+00E9, U+00ED, U+00F3,
U+00F6, U+0151, U+00FA, U+00FC, U+0171, \
                U+00C1->a, U+00C9->e, U+00CD->i, U+00D3->o, U+00D6->o, U+0150->o, U+00DA->u, U+00DC->u,
U+0170->u, U+00E1->a, U+00E9->e, U+00ED->i, U+00F3->o, U+00F6->o, U+0151->o, U+00FA->u, U+00FC->u, U+0171-
>u
}


index kannada_charset
{
        source                  = empty_source
        path                    = /var/data/kannada_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
        charset_table           = U+0C85..U+0C8C, U+0C8E..U+0C90, U+0C92..U+0CA8, U+0CAA..U+0CB3,
U+0CB5..U+0CB9, U+0CE0, U+0CE1, U+0CE6..U+0CEF
}


index limbu_charset
{
        source                  = empty_source
        path                    = /var/data/limbu_charset
        charset_type              = utf-8
        html_strip                = 0
        docinfo          = extern
        mlock                     = 0
```

```
        charset_table           = U+1900..U+191C, U+1930..U+1938, U+1946..U+194F
}

index malayalam_charset
{
        source                  = empty_source
        path                    = /var/data/malayalam_charset
        charset_type              = utf-8
        html_strip              = 0
        docinfo          = extern
        mlock                   = 0
        charset_table           = U+0D05..U+0D0C, U+0D0E..U+0D10, U+0D12..U+0D28, U+0D2A..U+0D39, U+0D60,
U+0D61, U+0D66..U+0D6F
}

index swedish_charset
{
        source                  = empty_source
        path                    = /var/data/swedish_charset
        charset_type              = utf-8
        html_strip              = 0
        docinfo          = extern
        mlock                   = 0
        charset_table           = 0..9, A..Z->a..z, _, -, a..z, U+D6->U+F6, U+C4->U+E4, U+C5->U+E5, U+F6,
U+E4, U+E5
}


index tamil_charset
{
        source                  = empty_source
        path                    = /var/data/tamil_charset
        charset_type              = utf-8
        html_strip              = 0
        docinfo          = extern
        mlock                   = 0
        charset_table           = U+0B94->U+0B92, U+0B85..U+0B8A, U+0B8E..U+0B90, U+0B92, U+0B93, U+0B95,
U+0B99, U+0B9A, U+0B9C, U+0B9E, U+0B9F, U+0BA3, U+0BA4, U+0BA8..U+0BAA, U+0BAE..U+0BB9, U+0BE6..U+0BEF
}


index thai_charset
{
        source                  = empty_source
        path                    = /var/data/thai_charset
        charset_type              = utf-8
        html_strip              = 0
        docinfo          = extern
        mlock                   = 0
        charset_table           = U+0E01..U+0E30, U+0E32, U+0E33, U+0E40..U+0E46, U+0E50..U+0E5B
}

index polish_charset
{
        source                  = empty_source
        path                    = /var/data/polish_charset
        charset_type              = utf-8
        html_strip              = 0
        docinfo          = extern
        mlock                   = 0
        charset_table           = 0..9, A..Z->a..z, a..z, U+0143->U+0144, U+0104->U+0105, U+0106->U+0107,
U+0118->U+0119, U+0141->U+0142, U+00D3->U+00F3, U+015A->U+015B, U+0179->U+017A, U+017B->U+017C, U+0105,
U+0107, U+0119, U+0142, U+00F3, U+015B, U+017A, U+017C, U+0144
}

index slovak_charset
{
        source                  = empty_source
        path                    = /var/data/slovak_charset
```

```
        charset_type            = utf-8
        html_strip              = 0
        docinfo         = extern
        mlock                   = 0
        charset_table          = U+0e1->a, U+0c1->a, U+0e4->a, U+0c4->a, U+10d->c, U+10c->c, U+10f->d,
U+10e->d, U+0e9->e, U+0c9->e, U+0ed->i, U+0cd->i, U+13e->l, U+13d->l, U+13a->l, U+139->l, U+148->n, U+147-
>n, U+0f3->o, U+0d3->o, U+0f4->o, U+0d4->o, U+155->r, U+154->r, U+161->s, U+160->s, U+165->t, U+164->t,
U+0fa->u, U+0da->u, U+0fd->y, U+0dd->y, U+17e->z, U+17d->z
}









# danish
index index_da : latin_charset
{
        source                  = xtm_da
        path                    = /var/data/index_da
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_da
        min_word_len            = 1
        charset_type            = utf-8
        html_strip              = 0
}

index index_da_delta : index_da
{
        source                  = xtm_da_delta
        path                    = /var/data/index_da_delta
}


# dutch (uma variante de alemão)
index index_nl  : german_charset
{
        source                  = xtm_nl
        path                    = /var/data/index_nl
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_nl
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_nl_delta : index_nl
{
        source                  = xtm_nl_delta
        path                    = /var/data/index_nl_delta
}


# english
index index_en  : latin_charset
{
        source                 = xtm_en
        path                   = /var/data/index_en
        docinfo                = extern
        mlock                  = 0
        morphology             = libstemmer_en
        min_word_len           = 1
        charset_type           = utf-8
```

```
        html_strip                              = 0
}

index index_en_delta : index_en
{
        source                  = xtm_en_delta
        path                    = /var/data/index_en_delta
}

# finish
index index_fi : latin_charset
{
        source                  = xtm_fi
        path                    = /var/data/index_fi
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_fi
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_fi_delta : index_fi
{
        source                  = xtm_fi_delta
        path                    = /var/data/index_fi_delta
}

# french
index index_fr : latin_charset
{
        source                  = xtm_fr
        path                    = /var/data/index_fr
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_fr
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_fr_delta : index_fr
{
        source                  = xtm_fr_delta
        path                    = /var/data/index_fr_delta
}

# german
index index_de : german_charset
{
        source                  = xtm_de
        path                    = /var/data/index_de
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_de
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_de_delta : index_de
{
        source                  = xtm_de_delta
        path                    = /var/data/index_de_delta
}


# hungarian
index index_hu : hungarian_charset
```

```
{
        source                  = xtm_hu
        path                    = /var/data/index_hu
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_hu
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_hu_delta : index_hu
{
        source                    = xtm_hu_delta
        path                      = /var/data/index_hu_delta
}

# italian
index index_it : latin_charset
{
        source                  = xtm_it
        path                    = /var/data/index_it
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_it
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_it_delta : index_it
{
        source                    = xtm_it_delta
        path                      = /var/data/index_it_delta
}

# norwegian
index index_no : german_charset
{
        source                  = xtm_no
        path                    = /var/data/index_no
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_no
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_no_delta : index_no
{
        source                    = xtm_no_delta
        path                      = /var/data/index_no_delta
}

# portuguese
index index_pt : latin_charset
{
        source                  = xtm_pt
        path                    = /var/data/index_pt
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_pt
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_pt_delta : index_pt
```

```
{
        source                  = xtm_pt_delta
        path                    = /var/data/index_pt_delta
}

# romanian
index index_ro : latin_charset
{
        source                  = xtm_ro
        path                    = /var/data/index_ro
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_ro
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                          = 0
}

index index_ro_delta : index_ro
{
        source                  = xtm_ro_delta
        path                    = /var/data/index_ro_delta
}

# russian
index index_ru : cyrillic_charset
{
        source                  = xtm_ru
        path                    = /var/data/index_ru
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_ru
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                          = 0
}

index index_ru_delta : index_ru
{
        source                  = xtm_ru_delta
        path                    = /var/data/index_ru_delta
}

# spanish
index index_es : latin_charset
{
        source                  = xtm_es
        path                    = /var/data/index_es
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_es
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                          = 0
}

index index_es_delta : index_es
{
        source                  = xtm_es_delta
        path                    = /var/data/index_es_delta
}

# swedish
index index_sv : swedish_charset
{
        source                  = xtm_sv
        path                    = /var/data/index_sv
        docinfo                 = extern
        mlock                   = 0
```

```
        morphology              = libstemmer_sv
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_sv_delta : index_sv
{
        source                  = xtm_sv_delta
        path                    = /var/data/index_sv_delta
}

# turkish
index index_tr : latin_charset
{
        source                  = xtm_tr
        path                    = /var/data/index_tr
        docinfo                 = extern
        mlock                   = 0
        morphology              = libstemmer_tr
        min_word_len            = 1
        charset_type            = utf-8
        html_strip                              = 0
}

index index_tr_delta : index_tr
{
        source                  = xtm_tr_delta
        path                    = /var/data/index_tr_delta
}


# oher languages
index index_default : latin_charset
{
        source                  = xtm_default
        path                    = /var/data/index_default
        docinfo                 = extern
        mlock                   = 0
        min_word_len            = 1
        charset_type            = utf-8
        html_strip              = 0
}

index index_default_delta : index_default
{
        source                  = xtm_default_delta
        path                    = /var/data/index_default_delta
}

########################################



indexer
{
        mem_limit                       = 100M
}

searchd
{
        port                            = 3312
        log                                     = /var/log/searchd.log
        query_log                       = /var/log/query.log
        read_timeout            = 5
        max_children            = 30
        pid_file                = /var/log/searchd.pid
        max_matches                     = 1000
        seamless_rotate         = 1
```

```
        preopen_indexes           = 0
        unlink_old                        = 1
}


# --eof--
```

ANEXO II: Artigo

IMPLEMENTAÇÃO DE UM SERVIDOR DE MEMÓRIAS DE TRADUÇÃO

Lucas Augusto Deters

deters@inf.ufsc.br

ABSTRACT

This study aims at the development of a remote translation memories with the emphasis on solving the problem of storing large databases and efficient recovery of these through search. During its development, it proposes an architecture based on the use of a full text search engine coupled to a database, and making use of web services to allow customers to use this server through the Internet.

The development of this server is aimed to help minimize the absence of robust solutions for servers translation memory based on free software, contributing to minimize the deficiences in the area.

INTRODUÇÃO

A ampliação do mercado de software ocorrido em paralelo à disseminação dos computadores pessoais ocorrido na década de 80 exigiu das empresas criadoras de software um novo patamar de velocidade e qualidade de seus produtos. A internacionalização do software passou a ser um fator crítico de aceitação, e ponto chave para o domínio de novos mercados.

O modelo de tradução de software tal qual era realizado na época, até então realizado por departamentos de tradução dentro da própria empresa, se mostrou ineficiente para atender às novas demandas, e as produtoras de softwares foram obrigadas a buscar um novo modelo de internacionalização para seus softwares. Neste contexto, surgiram as primeiras empresas especializadas em localização de software.

(Alfaro, 1998)

A crescente complexidade dos projetos de software e a velocidade no desenvolvimento de novas versões obrigou esta indústria especializada a desenvolver ferramentas de auxílio à tradução que permitissem o aumento da produtividade e a manutenção da uniformidade de produção entre as dezenas de tradutores envolvidos em um projeto de grande porte, e para manter as traduções entre as várias versões de um mesmo produto. Foi neste contexto de redução de prazos e melhoria da qualidade que surgiram diversas ferramentas de auxílio à tradução. (Alfaro, 1998)

Memória de Tradução é uma técnica que consiste em armazenar as traduções de sentenças já realizadas, para que estas traduções sejam reaproveitadas totalmente ou parcialmente quando uma sentença semelhante vier a ser traduzida.

Memórias de tradução são ideais para documentos técnicos, e por este motivo são bastante difundidas na indústria de localização de software, em especial para tradução de manuais e interfaces de software. As ferramentas de auxílio à tradução com suporte a memórias de tradução apresentam traduções já realizadas para sentenças similares, e cabe ao tradutor aceitar ou corrigir a tradução fornecida.

No uso tradicional de memórias de tradução, o reaproveitamento das memórias de tradução é restrito devido à dificuldade no compartilhamento das bases de memórias de tradução: sejam dificuldades técnicas (compartilhamento, tamanho da base, atualização desta), bem como questões legais e interesses comerciais (limitação de uso das memórias).

Uma alternativa mais interessante pode ser utilizar um servidor de memórias de tradução, que pode ajudar a resolver os problemas acima.

No mundo do software livre ainda não existe nenhuma opção de servidor de memórias de tradução livre que possa ser utilizado por meio de web services. A

maioria das implementações atuais de memórias de tradução em software livre está atrelada à própria interface de edição e as consultas são realizadas apenas sobre bases de dados locais.

O desenvolvimento de um servidor de memórias de tradução de código aberto vem justamente de encontro a esta necessidade de ferramentas mais eficientes para gerenciamento de memórias de tradução em projetos de software livre.

RECUPERAÇÃO DE TRADUÇÕES

Um servidor de memórias de tradução deve suportar as seguintes formas de recuperação de traduções:

- Correspondência exata: ocorre quando o segmento armazenado é exatamente igual ao segmento pesquisado.
- Correspondência em contexto: ocorre quando o segmento armazenado corresponde ao mesmo segmento pesquisado, inclusive quanto ao arquivo de origem e localização no texto.
- Correspondência aproximada: quando o segmento armazenado não é exatamente igual ao pesquisado. Alguns sistemas atribuem uma escala para representar a similaridade das sentenças.
- Concordância: consiste em recuperar segmentos que contenham um grupo de palavras enviadas como parâmetro, sem considerar uma ordem específica destas palavras nos segmentos.

A implementação destas formas de correspondência não é trivial, uma vez que operadores de bancos de dados tradicionais não são adequados para lidar com similaridade e buscas aproximadas sobre texto.

Os operadores de buscas aproximadas devem prever uma série de situações:

- diferença na ordem de algumas palavras dentro da sentença a ser pesquisada;
- ausência de algumas das palavras dentro da sentença a ser pesquisada;
- existência de palavras a mais dentro da sentença a ser pesquisada;
- substituição de palavras dentro das sentenças a serem pesquisadas;
- mudança na ordem das palavras dentro das sentenças a serem pesquisadas
- variações morfológicas da sentença (tempo, gênero, grau, etc)

Adicionalmente, é preciso considerar que as buscas sobre texto dependem das características de cada língua, e isso pode se tornar um desafio num servidor de memórias de traduções que potencialmente pode aceitar sentenças em qualquer língua.

A Figura 1 ilustra a arquitetura proposta para o servidor de memórias de tradução.

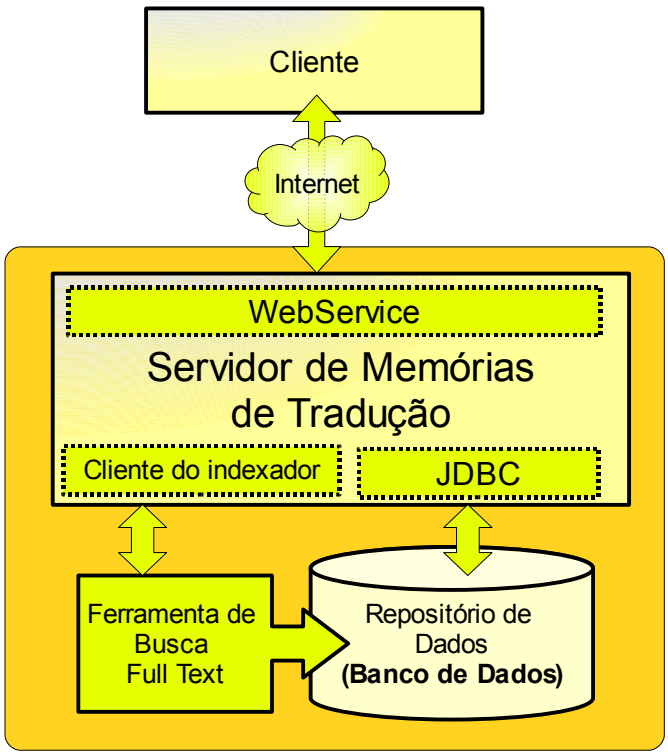

Figura 1: Arquitetura do Servidor de Memórias de Tradução

Esta proposta tenta resolver o problema da busca aproximada através da

utilização de uma ferramenta de busca full text acoplada a um banco de dados tradicional.

Conforme a proposta, o banco de dados fica responsável apenas para servir como repositório de dados, sendo eximida deste a função de realizar busca aproximada sobre o conteúdo das traduções.

Integrado ao repositório, deve ser acoplado o sistema de busca full text, que indexa o conteúdo do repositório e permite consultas baseadas em similaridade sobre os dados previamente indexados. Seu papel é fornecer um meio eficiente de busca sobre o conteúdo que esteja em formato texto. A busca aproximada sobre conteúdos de texto deve ser realizada com operadores de consulta apropriados para buscas aproximadas.

Cabe ao núcleo do servidor de memórias de tradução gerenciar os pedidos de tradução, realizar as consultas utilizando os operadores adequados junto à ferramenta de indexação full text e refinar os resultados obtidos de acordo com os parâmetros de qualidade especificados. Os métodos do servidor podem ser disponibilizados para os clientes através de web services.

O cliente é o responsável por requisitar os métodos disponibilizados pelo servidor e pode ser desde uma ferramenta de auxílio à tradução até um tradutor automático que faça uso da memória de tradução.

IMPLEMENTAÇÃO DE UM PROTÓTIPO

A implementação do protótipo deste servidor foi realizada utilizando a ferramenta de busca full text Sphinx[]. A escolha recaiu sobre esta ferramenta por ser uma ferramenta de alto desempenho, pelo seu suporte à diversos operadores de busca não exatas e principalmente pela sua licença de distribuição GPL.

A ferramenta Sphinx possui uma série de operadores para busca aproximada[]:

- operador de busca por proximidade:
- operador de matching de quorum
- operador de consulta de frases
- operadores de consulta normal (busca booleana)

Para utilização na recuperação de traduções por correspondência aproximada foi utilizado o operador de matching de quorum, devido à sua tolerância à ausência de algumas das palavras pesquisadas no interior dos segmentos.

A recuperação de traduções por correspondência exata foi realizada por meio do *operador de consulta de frases*. Este operador admite um certo grau de tolerância na busca – não fornece uma busca exata - mas exige que a ordem das palavras seja a mesma da sentença fornecida como parâmetro.

A recuperação de sentenças para concordância deve utilizar o modo de consulta normal (busca booleana), que não considera a ordem das palavras nas sentenças, mas que tem significativas vantagens de desempenho em relação aos demais tipos de consulta.

A recuperação em contexto ocorre por meio da utilização de filtros em conjunto com um dos demais modos de recuperação de traduções.

BANCO DE DADOS

O repositório de traduções é um banco de dados que irá armazenar as memórias de tradução. Foi utilizada a ferramenta PostgreSQL[] para gerenciar esta base de dados. A Figura 2 ilustra o modelo de dados desenvolvido:

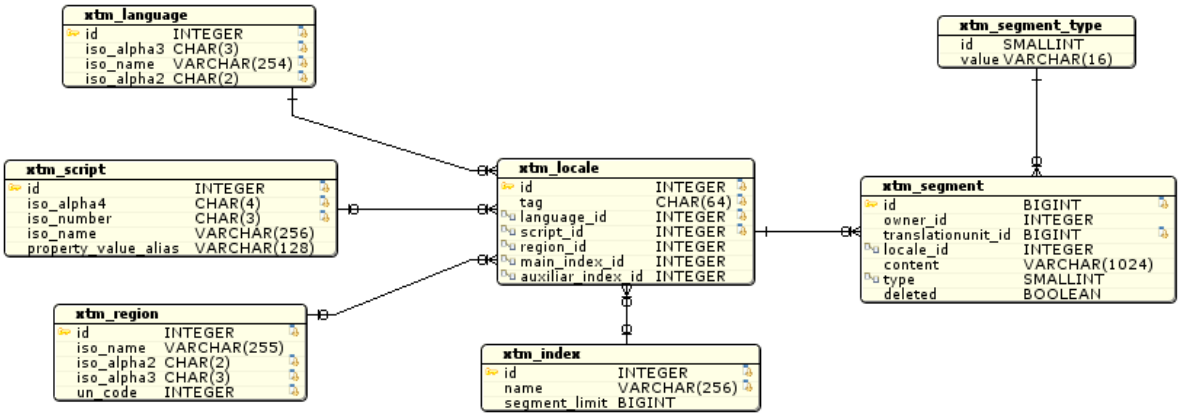

Figura 2: Modelo de dados do repositório

As tabelas de línguas, formatos de escrita, regiões e locales foram preenchidos de acordo com a especificação RFC4646 (tags for identifying languages) e padrões subjacentes (ISO 639-1 e ISO 639-2: codes for the representation of names of languages (Part I & II); ISO 15924: codes for the representation of names of scripts; e ISO 3166-1: codes for the representation of names of countries and their subdivisions).

Os segmentos de texto são armazenados na tabela xtm_segment, e estão vinculados a um locale previamente cadastrado na tabela xtm_locale. Os segmentos são relacionados entre si através do atributo translationunit_id, que agrupa vários segmentos em uma única unidade de tradução.

CONSIDERAÇÕES FINAIS

Nos testes preliminares, o tempo de médio de resposta do sistema para recuperar traduções a partir de correspondência aproximada com sentenças da língua inglesa é de 0.6 segundos em uma base contendo aproximadamente 1,6 milhões de registros em inglês. Os pedidos de concordância realizados sobre esta base demoram cerca de 0.16 segundos.

A qualidade dos resultados nestes testes preliminares também se mostrou satisfatória. Graças às medidas de similaridade calculadas pelo servidor, apenas frases

realmente relevantes são retornadas para os usuários, sendo possível determinar um coeficiente mínimo de similaridade para os resultados.

A arquitetura do sistema também se mostrou satisfatória, pois prevê vários pontos de flexibilidade que permitem a escalabilidade do sistema. É possível ampliar o armazenamento da base de dados através da adição de novos repositórios de dados que podem estar espalhados em várias máquinas. A ferramenta Sphinx também é capaz de realizar o processamento distribuído, através do recurso de índices remotos, e em último caso também é possível distribuir os serviços do servidor de memória de tradução em várias máquinas. Com esta arquitetura, seria possível suportar todas as traduções de grandes portais de tradução, tal como portais de tradução de software livre, ou outro sistema comercial equivalente.

Os resultados obtidos até o momento não deixam dúvida quanto à viabilidade da utilização deste servidor para a disponibilização de serviços baseados em memórias de tradução, beneficiando tanto a comunidade de software livre quanto empresas produtoras de software comercial que desejem utilizar este servidor em seus processos de tradução de software.

REFERÊNCIAS BIBLIOGRÁFICAS

ALFARO, C. **Descobrindo, Compreendendo e Analisando a Tradução Automática.** Rio de Janeiro, 1998.

INTERNATIONAL SOCIETY FOR STANDARDIZATION. **ISO 15924:2004 information and documentation - codes for the representation of names of scripts**. Disponível em http://www.iso.org/iso/catalogue_detail?csnumber=29546.

INTERNATIONAL SOCIETY FOR STANDARDIZATION. **ISO 3166-1:2006 codes for the representation of names of countries and their subdivisions - part 1: country codes**. Disponível em http://www.iso.org/iso/catalogue_detail?csnumber=39719.

INTERNATIONAL SOCIETY FOR STANDARDIZATION. **ISO 639-1:2002 codes for the representation of names of languages - part 1: alpha-2 code**. Disponível em http://www.iso.org/iso/catalogue_detail?csnumber=22109.

INTERNATIONAL SOCIETY FOR STANDARDIZATION. **ISO 639-2:1998 codes for the representation of names of languages - part 2: alpha-3 code**. http://www.iso.org/iso/catalogue_detail?csnumber=4767.

THE INTERNET SOCIETY. **RFC4646 – tags for identifying languages**. rfc.net/rfc4646.html

ANEXO III: RetrievalService.java

```
/*
Copyright (C) 2008  Lucas Augusto Deters

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA
*/


package br.ufsc.inf.xtm.service;

import br.ufsc.inf.xtm.XTMClient;
import br.ufsc.inf.xtm.XTMException;
import br.ufsc.inf.xtm.serializables.Context;
import br.ufsc.inf.xtm.serializables.Match;

/**
 * Translation Memory Retrieval Web Service
 *
 * Context search is not working yet.
 */
public class RetrievalService {

        private static XTMClient xtmClient;
        public RetrievalService() {
                if (xtmClient == null) {
                        xtmClient = new XTMClient("localhost", "5432", "xtm", "xtm", "xtm", "localhost",
"3312"  );
                }
        }

        /**
         * Return the best fuzzy matches from database. Also do in-context fuzzy matches.
         * @param sourceSegment The source segment.
         * @param sourceLocale The source locale tag, according to RFC4646.
         * @param targetLocale The target locale tag, according to RFC4646.
         * @param matchQuality The match quality.
         * @param metric The metric used to compute matchQuality. Default: Levenshtein.
         * @param context The context restrictions. Default null.
         * @return The best matches.
         * @throws XTMException
         */
        public static Match[] getFuzzyMatchesWithMetricAndContext(String sourceSegment, String
sourceLocale, String targetLocale, Integer matchQuality, String metric, Context context) throws
XTMException {
                return xtmClient.retrievalTranslations(sourceSegment, sourceLocale, targetLocale,
matchQuality, metric, context).toArray(new Match[0]);
        }

        /**
         * Return the best fuzzy matches from database. Also do in-context fuzzy matches.
         * @param sourceSegment The source segment.
         * @param sourceLocale The source locale tag, according to RFC4646.
         * @param targetLocale The target locale tag, according to RFC4646.
         * @param matchQuality The match quality.
         * @param metric The metric used to compute matchQuality. Default: Levenshtein.
         * @return The best matches.
         * @throws XTMException
         */
```

```
        public static Match[] getFuzzyMatchesWithMetric(String sourceSegment, String sourceLocale, String
targetLocale, Integer matchQuality, String metric) throws XTMException {
                return xtmClient.retrievalTranslations(sourceSegment, sourceLocale, targetLocale,
matchQuality, metric, null).toArray(new Match[0]);
        }


        /**
         * Return the best fuzzy matches from database. Also do in-context fuzzy matches.
         * @param sourceSegment The source segment.
         * @param sourceLocale The source locale tag, according to RFC4646.
         * @param targetLocale The target locale tag, according to RFC4646.
         * @return The best matches.
         * @throws XTMException
         */
        public static Match[] getFuzzyMatches(String sourceSegment, String sourceLocale, String
targetLocale) throws XTMException {
                return xtmClient.retrievalTranslations(sourceSegment, sourceLocale, targetLocale, 0.5,
"Levenshtein", null).toArray(new Match[0]);
        }

        /**
         * Return the best fuzzy matches from database. Also do in-context fuzzy matches.
         * @param sourceSegment The source segment.
         * @param sourceLocale The source locale tag, according to RFC4646.
         * @param targetLocale The target locale tag, according to RFC4646.
         * @param metric The metric used to compute matchQuality. Default: Levenshtein.
         * @param context The context restrictions. Default null.
         * @return The best matches.
         * @throws XTMException
         */
        public static Match[] getExactMatchesWithMetricAndContext(String sourceSegment, String
sourceLocale, String targetLocale, String metric, Context context) throws XTMException {
                return xtmClient.retrievalTranslations(sourceSegment, sourceLocale, targetLocale, 1.0,
metric, context).toArray(new Match[0]);
        }


        /**
         * Return the best fuzzy matches from database. Also do in-context fuzzy matches.
         * @param sourceSegment The source segment.
         * @param sourceLocale The source locale tag, according to RFC4646.
         * @param targetLocale The target locale tag, according to RFC4646.
         * @param metric The metric used to compute matchQuality. Default: Levenshtein.
         * @return The best matches.
         * @throws XTMException
         */
        public static Match[] getExactMatchesWithMetric(String sourceSegment, String sourceLocale, String
targetLocale, String metric) throws XTMException {
                return xtmClient.retrievalTranslations(sourceSegment, sourceLocale, targetLocale, 1.0,
metric, null).toArray(new Match[0]);
        }


        /**
         * Return the best fuzzy matches from database. Also do in-context fuzzy matches.
         * @param sourceSegment The source segment.
         * @param sourceLocale The source locale tag, according to RFC4646.
         * @param targetLocale The target locale tag, according to RFC4646.
         * @param metric The metric used to compute matchQuality. Default: Levenshtein.
         * @return The best matches.
         * @throws XTMException
         */
        public static Match[] getExactMatches(String sourceSegment, String sourceLocale, String
targetLocale) throws XTMException {
                return xtmClient.retrievalTranslations(sourceSegment, sourceLocale, targetLocale, 1.0,
"Levenshtein", null).toArray(new Match[0]);
        }
```

```
        /**
         * Return the matches using concordance. Also do in-context concordance.
         * @param sourceSegment The source segment.
         * @param sourceLocale The source locale tag, according to RFC4646.
         * @param targetLocale The target locale tag, according to RFC4646.
         * @param metric The metric used to compute matchQuality. Default: Levenshtein.
         * @param context The context restrictions. Default null.
         * @return The best matches.
         * @throws XTMException
         */
        public static Match[] getConcordanceMatchesWithMetricAndContext(String query, String sourceLocale,
String targetLocale, String metric, Context context) throws XTMException {
                return xtmClient.retrievalTranslations(query, sourceLocale, targetLocale, 0, metric,
context).toArray(new Match[0]);
        }

        /**
         * Return the matches using concordance. Also do in-context concordance.
         * @param sourceSegment The source segment.
         * @param sourceLocale The source locale tag, according to RFC4646.
         * @param targetLocale The target locale tag, according to RFC4646.
         * @param metric The metric used to compute matchQuality. Default: Levenshtein.
         * @return The best matches.
         * @throws XTMException
         */
        public static Match[] getConcordanceMatchesWithMetric(String query, String sourceLocale, String
targetLocale, String metric) throws XTMException {
                return xtmClient.retrievalTranslations(query, sourceLocale, targetLocale, 0, metric,
null).toArray(new Match[0]);
        }

        /**
         * Return the matches using concordance. Also do in-context concordance.
         * @param sourceSegment The source segment.
         * @param sourceLocale The source locale tag, according to RFC4646.
         * @param targetLocale The target locale tag, according to RFC4646.
         * @return The best matches.
         * @throws XTMException
         */
        public static Match[] getConcordanceMatches(String query, String sourceLocale, String targetLocale)
throws XTMException {
                return xtmClient.retrievalTranslations(query, sourceLocale, targetLocale, 0, "Levenshtein",
null).toArray(new Match[0]);
        }

}
```

ANEXO IV: Match.java

```
/*
Copyright (C) 2008  Lucas Augusto Deters

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA
*/
package br.ufsc.inf.xtm.serializables;

import java.io.Serializable;
import java.text.DecimalFormat;

public class Match implements Serializable, Comparable<Match>{
	/**
	 *
	 * Creation date: Sep 19, 2008
	 * Creator:       deters
	 */
	private static final long serialVersionUID = -652264548818027250 1L;


	private String sourceSegment;
	private String targetSegment;
	private double similarity;
	private String sourceLanguage;
	private String targetLanguage;

	public String getSourceLanguage() {
		return sourceLanguage;
	}

	public void setSourceLanguage(String sourceLanguage) {
		this.sourceLanguage = sourceLanguage;
	}

	public String getTargetLanguage() {
		return targetLanguage;
	}

	public void setTargetLanguage(String targetLanguage) {
		this.targetLanguage = targetLanguage;
	}

	public double getSimilarity() {
		return similarity;
	}

	public void setSimilarity(double similarity) {
		this.similarity = similarity;
	}


	public String getSourceSegment() {
		return sourceSegment;
	}


	public void setSourceSegment(String sourceSegment) {
		this.sourceSegment = sourceSegment;
```

```
	}


	public String getTargetSegment() {
		return targetSegment;
	}


	public void setTargetSegment(String targetSegment) {
		this.targetSegment = targetSegment;
	}


	public Match() {
		// TODO Auto-generated constructor stub
	}

	public Match(String sourceSegment, String targetSegment, double similarity, String targetLanguage,
String sourceLanguage) {
		this.sourceSegment = sourceSegment;
		this.targetSegment = targetSegment;
		this.sourceLanguage = sourceLanguage;
		this.targetLanguage = targetLanguage;
		this.similarity = similarity;
	}


	// TODO: drop this method
	public Match(String sourceSegment, String targetSegment, double similarity) {
		this.sourceSegment = sourceSegment;
		this.targetSegment = targetSegment;
		this.sourceLanguage = null;
		this.targetLanguage = null;
		this.similarity = similarity;
	}

	public int compareTo(Match arg0) {
		if (arg0 == null){
			return 1;
		} else {
			return ((Double)this.similarity).compareTo(arg0.getSimilarity()) ;
		}
	}

	public String toString(){

		DecimalFormat nf = new DecimalFormat("###.00");
		return "\n{" + sourceSegment + "( "+nf.format(similarity*100)+"%) --> "+ targetSegment+"}\n
" ;

	}



}
```

ANEXO V: XTMException.java

```
/*
Copyright (C) 2008  Lucas Augusto Deters

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA
*/
/**
 *
 */
package br.ufsc.inf.xtm;

/**
 * @author deters
 *
 */
public class XTMException extends Exception {

	/**
	 *
	 */
	public XTMException() {
		// TODO Auto-generated constructor stub
	}

	/**
	 * @param arg0
	 */
	public XTMException(String arg0) {
		super(arg0);
		// TODO Auto-generated constructor stub
	}

	/**
	 * @param arg0
	 */
	public XTMException(Throwable arg0) {
		super(arg0);
		// TODO Auto-generated constructor stub
	}

	/**
	 * @param arg0
	 * @param arg1
	 */
	public XTMException(String arg0, Throwable arg1) {
		super(arg0, arg1);
		// TODO Auto-generated constructor stub
	}

}
```

ANEXO VI: XTMClient.java

```
/*
Copyright (C) 2008  Lucas Augusto Deters

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA
*/
package br.ufsc.inf.xtm;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;

import org.apache.commons.lang.StringEscapeUtils;
import org.sphx.api.SphinxClient;
import org.sphx.api.SphinxException;
import org.sphx.api.SphinxMatch;
import org.sphx.api.SphinxResult;

import uk.ac.shef.wit.simmetrics.metrichandlers.MetricHandler;
import uk.ac.shef.wit.simmetrics.similaritymetrics.AbstractStringMetric;
import br.ufsc.inf.xtm.serializables.Context;
import br.ufsc.inf.xtm.serializables.Match;

public class XTMClient {

	public static String metrics[] = { "BlockDistance",
			"ChapmanLengthDeviation", "ChapmanMeanLength", "CosineSimilarity",
			"DiceSimilarity", "EuclideanDistance", "JaccardSimilarity", "Jaro",
			"JaroWinkler", "Levenshtein", "MatchingCoefficient", "MongeElkan",
			"ChapmanMatchingSoundex", "NeedlemanWunch", "OverlapCoefficient",
			"QGramsDistance", "SmithWaterman",
			"SmithWatermanGotohWindowedAffine", "SmithWatermanGotoh", "Soundex" };

	private String databaseHost;
	private String databaseName;
	private String databaseUser;
	private String databasePassword;
	private String sphinxHost;
	private String sphinxPort;
	private String databasePort;
	private Connection databaseConnection;

	private static final String SQLGETSEGMENTCONTENT = "select s.id, s.owner_id, s.translationunit_id,
s.locale_id, s.content, s.type, loc.tag from xtm_segment s inner join xtm_locale loc on loc.id =
s.locale_id where s.id = ?";
	private static final String SQLGETLOCALEID = "select id from xtm_locale where tag = ?";
	private static final String SQLGETEQUIVALENTLOCALES = "select l1.id "
			+ "from xtm_locale l "
			+ "inner join xtm_locale l1 "
			+ "  on  l.language_id = l1.language_id "
			+ "  and coalesce(l.script_id,coalesce(l1.script_id,0)) = coalesce(l1.script_id,0)
"
```

```
                        + "  and coalesce(l.region_id,coalesce(l1.region_id,0)) = coalesce(l1.region_id,0)
"
                        + "where l.tag = ? ";
        private static final String SQLGETINDEXES = "select distinct i.name "
                        + "from xtm_locale l "
                        + "inner join xtm_locale l1 "
                        + "  on  l.language_id = l1.language_id "
                        + "  and coalesce(l.script_id,coalesce(l1.script_id,0)) = coalesce(l1.script_id,0)
"
                        + "  and coalesce(l.region_id,coalesce(l1.region_id,0)) = coalesce(l1.region_id,0)
"
                        + "inner join xtm_index i " + "  on  i.id = l.main_index_id "
                        + "where l.tag = ? ";
        private static final String SQLGETTARGETSEGMENTS = "select s2.content "
                        + "from xtm_segment s1 " + "inner join xtm_segment s2 "
                        + "  on s2.translationunit_id = s1.translationunit_id "
                        + "where s1.id = ? " + "  and s2.locale_id = ?";

        private PreparedStatement statementGetLocaleId;

        private static String metricaDefault = "Levenstein";

        public static void main(String[] args) {
                String databaseHost = "localhost";
                String databaseName = "xtm";
                String databaseUser = "xtm";
                String databasePass = "xtm";
                String databasePort = "5432";
                String sphinxHost = "localhost";
                String sphinxPort = "3312";
                XTMClient cliente = new XTMClient(databaseHost, databasePort,
                                databaseName, databaseUser, databasePass, sphinxHost,
                                sphinxPort);
                try {

                        // String teste =
                        // "In the beginning God was creating the heaven and the earth";
                        String teste = "adicionar marcador";
                        String from = "pt";
                        String to = "en";

                        String metric = "Levenshtein";
                        // String metric = "Soundex";

                        System.out.println("translate(" + teste + ", " + from + ", " + to
                                        + "):");
                        long tempoInicial = System.nanoTime();
                        List<Match> responses = cliente.retrievalTranslations(teste, from,
                                        to, 0.5, metric, null);
                        System.out.println("  " + responses + "\n"
                                        + (((System.nanoTime() - tempoInicial)) / 1000000) + "ms");

                } catch (XTMException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

        }

        /**
         * Create a XTMClient
         *
         * @param databaseHost
         * @param databasePort
         * @param databaseName
         * @param databaseUser
         * @param databasePass
         * @param sphinxHost
         * @param sphinxPort
```

```
 * @param sphinxPort2
 */
public XTMClient(String databaseHost, String databasePort,
                String databaseName, String databaseUser, String databasePass,
                String sphinxHost, String sphinxPort) {
        this.databaseHost = databaseHost;
        this.databasePort = databasePort;
        this.databaseName = databaseName;
        this.databaseUser = databaseUser;
        this.databasePassword = databasePass;

        this.sphinxHost = sphinxHost;
        this.sphinxPort = sphinxPort;

        try {
                this.databaseConnection = getDatabaseConnection();
        } catch (ClassNotFoundException e) {
                e.printStackTrace();
                System.exit(1);
        } catch (SQLException e) {
                e.printStackTrace();
                System.exit(1);
        }

}

private SphinxClient getSphinxConnection(int offset, int limit)
                throws XTMException, SphinxException {

        int sortMode = SphinxClient.SPH_SORT_RELEVANCE;
        int rankMode = SphinxClient.SPH_RANK_PROXIMITY_BM25;
        String sortClause = "";

        SphinxClient cl = new SphinxClient();

        try {
                cl.SetServer(sphinxHost, new Integer(sphinxPort).intValue());
        } catch (NumberFormatException e) {
                throw new XTMException("Sphinx port parameter must be an integer.");
        }

        cl.SetWeights(new int[] { 100, 1 });
        cl.SetLimits(offset, limit);
        cl.SetSortMode(sortMode, sortClause);
        cl.SetRankingMode(rankMode);

        return cl;
}

/**
 * @return
 * @throws ClassNotFoundException
 * @throws SQLException
 */
private Connection getDatabaseConnection() throws ClassNotFoundException,
                SQLException {

        Class.forName("org.postgresql.Driver");

        return DriverManager.getConnection("jdbc:postgresql://" + databaseHost
                        + ":" + databasePort + "/" + databaseName, databaseUser,
                        databasePassword);
}

private PreparedStatement getLocaleIdStatement() throws SQLException {
        return databaseConnection.prepareStatement(SQLGETLOCALEID);
}

public int getLocaleId(String locale) throws XTMException {
```

```
        PreparedStatement localeIdStatement = null;

        try {
                localeIdStatement = getLocaleIdStatement();
                statementGetLocaleId.setString(1, locale);
                ResultSet rs = localeIdStatement.executeQuery();

                while (rs.next()) {
                        String name = rs.getString("COF_NAME");
                        float price = rs.getFloat("PRICE");
                        System.out.println(name + "     " + price);
                }

        } catch (SQLException e1) {
                throw new XTMException("Internal error.", e1);
        }

        return 0;
}

private static void info(Object s) {
        System.err.println("Info: " + s);
}

/**
 * Retrieval the translations
 *
 * @param sourceSegment
 *            The source segment or words to make concordance
 * @param sourceLocale
 *            The source locale
 * @param targetLocale
 *            The target locale
 * @param matchQuality
 *            The match quality, from 0.0 to 1.0. If match quality is 0.0,
 *            then will execute concordance search. If match quality is 1.0,
 *            then will execute a exact search.
 * @param metric
 *            The metric to be used. Default: Levenshtein
 * @param context
 *            The context info
 * @return A list of matches
 * @throws XTMException
 */
public List<Match> retrievalTranslations(String sourceSegment,
                String sourceLocale, String targetLocale, double matchQuality,
                String metric, Context context) throws XTMException {

        if (metric == null) {
                metric = metricaDefault;
        }

        AbstractStringMetric metrica = null;

        metrica = MetricHandler.createMetric(metric);
        if (metrica == null) {
                throw new XTMException("Invalid metric.");
        }

        try {

                // get the name of the indexes to query sphinx
                ArrayList<String> indexNames = getIndexNamesForLocale(sourceLocale);

                info("Using index: " + indexNames);

                // get all the equivalent locales to filter
                ArrayList<Integer> equivalentLocales = getEquivalentLocales(sourceLocale);
```

```
                        info("Equivalent locales to search: " + equivalentLocales);

                        // query sphinx
                        ArrayList<SphinxMatch> respostas = new ArrayList<SphinxMatch>();

                        for (Iterator<String> iter = indexNames.iterator(); iter.hasNext();) {

                                String indexname = (String) iter.next();

                                SphinxClient sphinxClient = getSphinxConnection(0, 20);

                                int[] arrayLocales = new int[equivalentLocales.size()];

                                arrayLocales = new int[equivalentLocales.size()];
                                for (int i = 0; i < equivalentLocales.size(); i++) {
                                        arrayLocales[i] = equivalentLocales.get(i).intValue();
                                }

                                sphinxClient.SetFilter("locale_id", arrayLocales, false);

                                StringBuffer q = new StringBuffer();

                                if (matchQuality == 1.0) {

                                        // / phrase query
                                        sphinxClient.SetMatchMode(SphinxClient.SPH_MATCH_EXTENDED2);
                                        q.append(" \""
                                                        + StringEscapeUtils.escapeJava(sourceSegment)
                                                        + "\" ");

                                } else if (matchQuality == 0) {

                                        // / concordance
                                        sphinxClient.SetMatchMode(SphinxClient.SPH_MATCH_ALL);
                                        q.append(StringEscapeUtils.escapeJava(sourceSegment));

                                } else {

                                        // / quorum
                                        sphinxClient.SetMatchMode(SphinxClient.SPH_MATCH_EXTENDED2);
                                        q.append(" \""
                                                        + StringEscapeUtils.escapeJava(sourceSegment)
                                                        + "\"/2 ");

                                }

                                SphinxResult res = null;
                                try {
                                        res = sphinxClient.Query(q.toString(), indexname);
                                } catch (SphinxException e) {
                                        throw new XTMException("Sphinx error on querying", e);
                                }

                                if (res == null) {
                                        throw new XTMException(
                                                        "Sphinx returned null response. Last possible
error: "
                                                                        + sphinxClient.GetLastError());
                                }

                                if (sphinxClient.GetLastWarning() != null
                                                && sphinxClient.GetLastWarning().length() > 0)
                                        System.out.println("WARNING: "
                                                        + sphinxClient.GetLastWarning() + "\n");

                                for (int i = 0; i < res.matches.length; i++) {
                                        respostas.add(res.matches[i]);
                                }
```

```
}

info("Sphinx results: " + respostas);

ArrayList<Long> segmentIds = new ArrayList<Long>();

// print result
for (Iterator<SphinxMatch> iter = respostas.iterator(); iter
                .hasNext();) {
        SphinxMatch element = (SphinxMatch) iter.next();
        segmentIds.add(element.docId);
}

// get the segments in source language
ArrayList<String> sourceSegments = getSegments(segmentIds);
// System.out.println(sourceSegments);

ArrayList<String> topSegments = new ArrayList<String>();
ArrayList<Float> topSegmentsSimilarity = new ArrayList<Float>();
ArrayList<Long> topSegmentsIds = new ArrayList<Long>();

Iterator<Long> iterids = segmentIds.iterator();
iterids.hasNext();

// calculate the levenshtein distance, and mantain only the better
// results
for (Iterator<String> iter = sourceSegments.iterator(); iter
                .hasNext();) {

        String currentSegment = (String) iter.next();
        long currentSegmentId = (Long) iterids.next();
        float similarity = metrica.getSimilarity(sourceSegment,
                        currentSegment);


        info(currentSegment+" ("+similarity+")");

        if (similarity >= matchQuality) {
                topSegments.add(currentSegment);
                topSegmentsIds.add(currentSegmentId);
                topSegmentsSimilarity.add(similarity);
        }
}

Iterator<String> iterTopSegments = topSegments.iterator();
iterTopSegments.hasNext();

Iterator<Float> iterTopSegmentsSimilarity = topSegmentsSimilarity
                .iterator();
iterTopSegmentsSimilarity.hasNext();

ArrayList<Match> responses = new ArrayList<Match>();

for (Iterator<Long> iter = topSegmentsIds.iterator(); iter
                .hasNext();) {

        long segmentId = (Long) iter.next();
        String segment = (String) iterTopSegments.next();
        float similarity = (Float) iterTopSegmentsSimilarity.next();

        ArrayList<String> targetSegments = getTargetSegments(segmentId,
                        getEquivalentLocales(targetLocale));

        for (Iterator<String> iterator = targetSegments.iterator(); iterator
                        .hasNext();) {
                String targetSegment = (String) iterator.next();

                Match response = new Match(segment, targetSegment,
```

```
                                                similarity);

                                responses.add(response);

                        }

                }

                Collections.sort(responses);

                info("Matched segments: " + responses);

                // System.out.println(maxSimilarity+"% "+mostSimilarId+" "+mostSimilar);

                return responses;

        } catch (XTMException e) {
                e.printStackTrace();
                System.exit(1);
        } catch (SphinxException e) {
                e.printStackTrace();
                System.exit(1);
        }

        return null;

}

private ArrayList<Integer> getEquivalentLocales(String sourceLocale)
                throws XTMException {
        ArrayList<Integer> equivalentLocales = new ArrayList<Integer>();

        //
        try {
                PreparedStatement getEquivalentLocalesStatement = databaseConnection
                                .prepareStatement(SQLGETEQUIVALENTLOCALES);
                getEquivalentLocalesStatement.setString(1, sourceLocale);
                ResultSet rs = getEquivalentLocalesStatement.executeQuery();

                while (rs.next()) {
                        int localeid = rs.getInt(1);
                        equivalentLocales.add(localeid);
                }

                getEquivalentLocalesStatement.close();

        } catch (SQLException e1) {
                throw new XTMException("Internal error.", e1);
        }
        return equivalentLocales;
}

private ArrayList<String> getIndexNamesForLocale(String sourceLocale)
                throws XTMException {
        ArrayList<String> indexNames = new ArrayList<String>();

        // Get the names of the indexes to query
        try {
                PreparedStatement indexNamestoLocaleStatement = databaseConnection
                                .prepareStatement(SQLGETINDEXES);
                indexNamestoLocaleStatement.setString(1, sourceLocale);
                ResultSet rs = indexNamestoLocaleStatement.executeQuery();

                while (rs.next()) {
                        String name = rs.getString(1);
                        indexNames.add(name);
                }

                indexNamestoLocaleStatement.close();
```

```
        } catch (SQLException e1) {
                throw new XTMException("Internal error.", e1);
        }
        return indexNames;
}

private ArrayList<String> getSegments(List<Long> segmentIdList)
                throws XTMException {

        ArrayList<String> segmentContentList = new ArrayList<String>();
        PreparedStatement getSegmentContentStatement;
        try {
                getSegmentContentStatement = databaseConnection
                                .prepareStatement(SQLGETSEGMENTCONTENT);
        } catch (SQLException e) {
                throw new XTMException("Internal error.", e);
        }

        for (Iterator<Long> iter = segmentIdList.iterator(); iter.hasNext();) {
                long element = (Long) iter.next();

                //
                try {
                        getSegmentContentStatement.setLong(1, element);
                        ResultSet rs = getSegmentContentStatement.executeQuery();

                        rs.next();

                        String content = rs.getString("content");
                        segmentContentList.add(content);

                        rs.close();

                } catch (SQLException e1) {
                        throw new XTMException("Internal error.", e1);
                }

        }

        try {
                getSegmentContentStatement.close();
        } catch (SQLException e) {
                e.printStackTrace();
        }

        return segmentContentList;
}

private ArrayList<String> getTargetSegments(long sourceId,
                List<Integer> targetlocaleIdList) throws XTMException {

        ArrayList<String> segmentContentList = new ArrayList<String>();
        PreparedStatement getTargetSegmentsStatement;
        try {
                getTargetSegmentsStatement = databaseConnection
                                .prepareStatement(SQLGETTARGETSEGMENTS);
        } catch (SQLException e) {
                throw new XTMException("Internal error.", e);
        }

        for (Iterator<Integer> iter = targetlocaleIdList.iterator(); iter
                        .hasNext();) {
                int locale = (Integer) iter.next();

                //
                try {
                        getTargetSegmentsStatement.setLong(1, sourceId);
                        getTargetSegmentsStatement.setInt(2, locale);
```

```
                                ResultSet rs = getTargetSegmentsStatement.executeQuery();

                                if (rs.next()) {

                                        String content = rs.getString(1);
                                        segmentContentList.add(content);
                                }

                                rs.close();

                        } catch (SQLException e1) {
                                throw new XTMException("Internal error.", e1);
                        }

                }

                try {
                        getTargetSegmentsStatement.close();
                } catch (SQLException e) {
                        e.printStackTrace();
                }

                return segmentContentList;
        }

}
```