4. SQLLOMining - um sistema de mineração de LOs

A extração de objetos de aprendizado de um texto através do aprendizado de máquina pode ser alcançada seguindo um processo que será desenvolvido e concretizado em um sistema que denominamos de SQLLOMining.

O nome escolhido faz referência a linguagem de programação utilizada Structured Query Language - SQL, o objeto de interesse LO e a técnica utilizada, mineração de textos. O Sistema SQLLOMining tem como objetivo prover uma ferramenta para a criação de ALOs para o projeto PGL a partir de textos existentes, utilizando técnicas de mineração de textos e aprendizado de maquina.

Este processo é composto de várias etapas. A primeira delas diz respeito à definição exata do que se procura, através da especificação de classes ou tipos de ALOs que serão o alvo do aprendizado de máquina. É necessária a criação de um Corpus de exemplos e amostras que deverá ser desenvolvido, a princípio, manualmente. Este Corpus nos permitirá não só a geração de um modelo inicial para o processo de aprendizado como também a classificação, com posterior avaliação, permitindo que possamos levantar os dados estatísticos que nos servirão de métricas do experimento feito.

Os arquivos que farão parte do Corpus deverão ser importados e associados a um tipo já especificado. Estes arquivos serão gravados no banco de dados e fragmentados em sentenças, que serão etiquetadas com o tipo definido para o arquivo.

A seguir apresentamos uma figura ilustrativa desta etapa inicial:

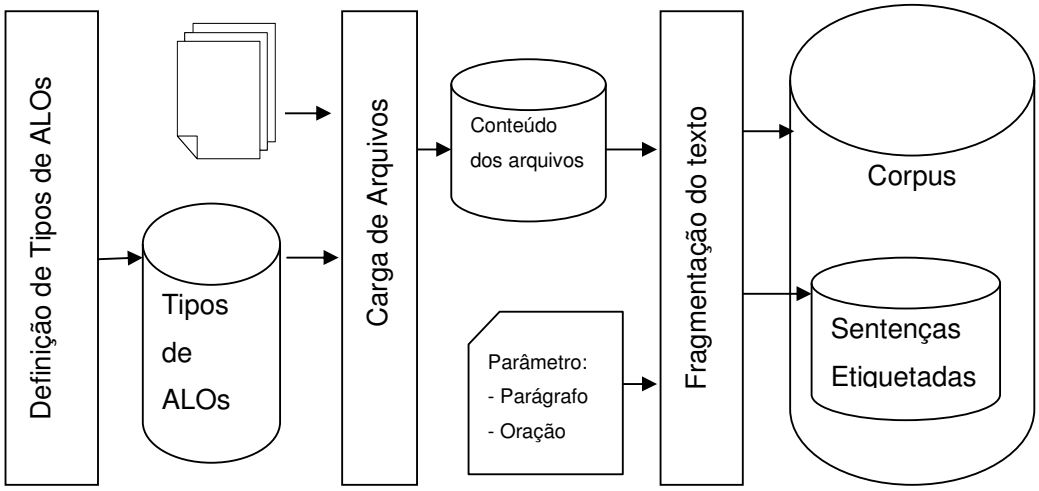

Figura 5 - Geração do Corpus Inicial

Na segunda etapa é feito um pré-processamento destes exemplos resultando na separação das sentenças do Corpus em n-gramas para utilização na análise e geração de input para o algoritmo de aprendizado. Poderemos então desenvolver o modelo propriamente dito, utilizando-nos de diversos recursos incluídos na ferramenta.

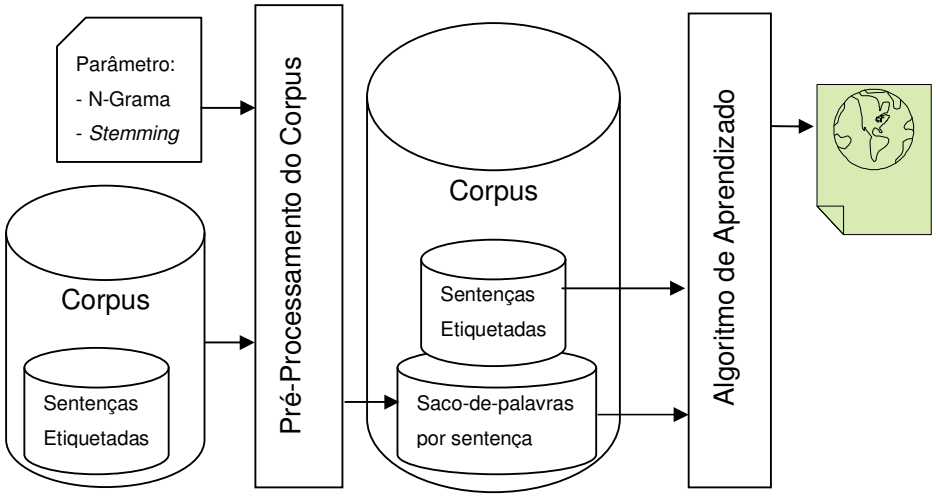

Figura 6 - Geração do Modelo Inicial

Por fim, uma vez gerado o modelo, poderemos utilizá-lo para classificação de qualquer texto apresentado pelo usuário, identificando claramente para ele as sentenças de seu interesse.

Para tanto, o usuário importará o novo arquivo a ser classificado, o qual sofrerá o mesmo processamento dos arquivos anteriores, e então será incluído no Corpus.

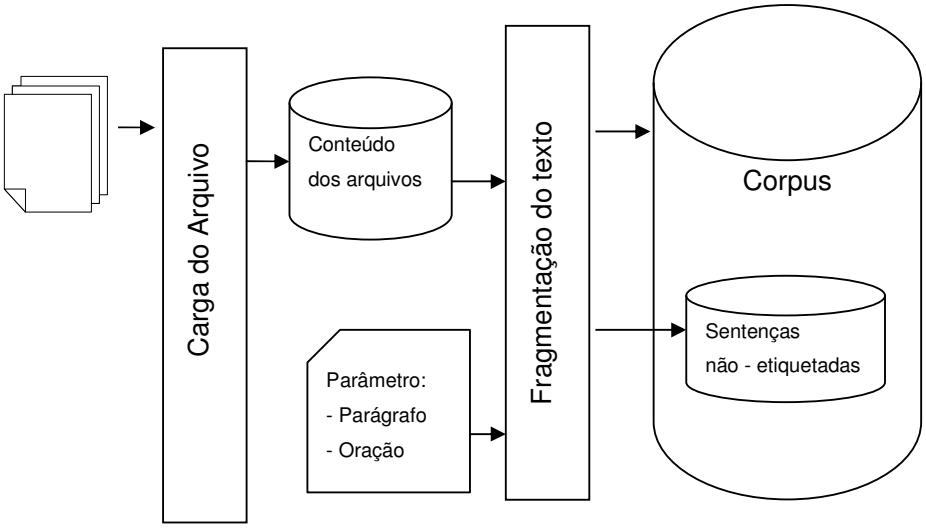

Figura 7 - Carga do Arquivo a ser classificado

Utilizando-se das sentenças não etiquetadas deste arquivo e do modelo inicial, o usuário poderá executar um aprendizado semi-supervisionado que classificará as sentenças não etiquetadas, aprimorando o modelo.

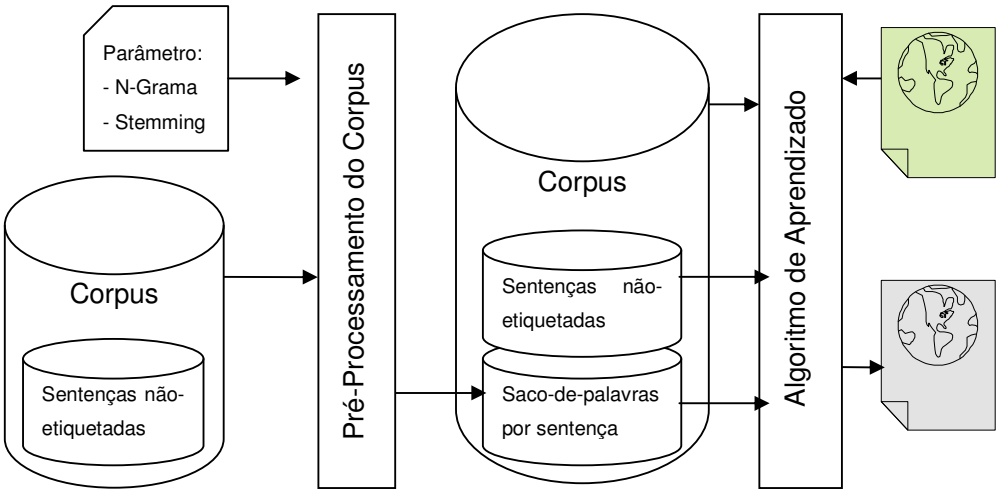

Figura 8 - Aprendizado Semi-Supervisionado

Finalmente, será possível através de telas de consulta selecionar dentre as sentenças pertencentes ao Corpus aquelas que forem de interesse do usuário.

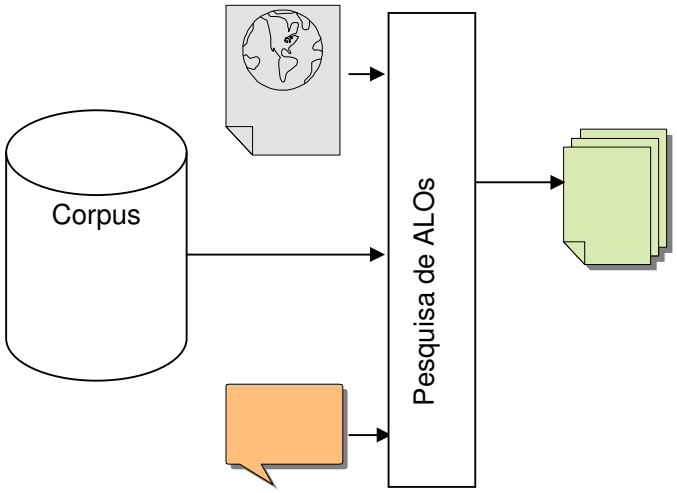

Figura 9 - Pesquisa de ALOs

4.1.Especificação do processo de extração de Objetos de Aprendizado

A seguir definimos mais claramente cada uma destas etapas e como elas foram disponibilizadas na ferramenta SQLLOMining.

4.1.1.Geração do Corpus Inicial

A palavra corpus significa corpo em latin. No contexto de Processamento de Linguagem Natural, corpus se refere a um conjunto de textos utilizados para experimentação e validação de modelos (Mitchell, 1997).

O Corpus inicial é gerado para apoiar a criação do modelo inicial. Faz parte do Corpus os tipos, ou classes, que o modelo irá utilizar, e arquivos contendo os exemplos. O Corpus inicial é gerado a partir destes exemplos, por isso todos os arquivos carregados para este Corpus deverão ser associados pelo usuário a um tipo de tal forma que todas as sentenças destes arquivos herdarão este tipo como etiqueta gerando um grupo de sentenças etiquetadas.

É possível utilizar diversos arquivos, cada um contendo exemplos de um dos tipos e diversos tipos. A definição dos tipos e destes arquivos permitem a

geração de experimentos diferentes, que dependerão do conteúdo deste Corpus Inicial. Os tipos definidos serão incluídos na tabela *classe* e os arquivos serão armazenados nas tabelas *Arquivostxt* e *ArquivosTxtConteudos.* Após a fragmentação do texto as tabelas *corpus* e *sentenca* serão preenchidas.

4.1.1.1.Definição dos tipos de ALOs

Disponibiliza-se para o usuário uma tela de cadastro de tipos. Nesta tela será informado apenas a Descrição ou nome do tipo e depois basta clicar no botão “Gravar”. A mesma tela pode ser utilizada para selecionar um dos tipos já cadastrados e então será possível alterar sua descrição ou exclui-lo. Para isso, depois de selecioná-lo basta clicar no botão “Limpar”.

Os tipos serão nada mais do que as classes que serão utilizadas no aprendizado de máquina. Poderemos trabalhar diversos experimentos variando a definição dos tipos. Desta forma poderemos investigar a combinação que melhor modelará o domínio de interesse. Lembramos que o modelo Bayesiano considera que cada texto está associado a apenas uma classe e por esta razão, a definição das classes é de extrema importância para o êxito da classificação.

Abaixo segue a tela do sistema associada a esta etapa do processo:

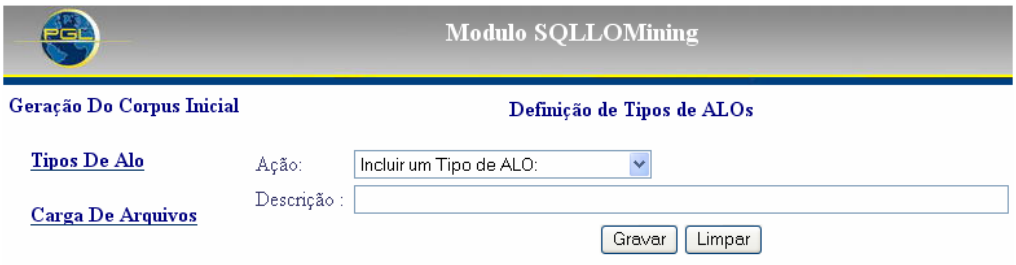

Figura 10 - Tela de Definição de Tipos de ALOs

4.1.1.2.Carga de arquivos

A Segunda tela apresentada abaixo é relativa ao processo de carga de arquivos. Na inclusão de um arquivo o nome dele deverá ser especificado e através do botão “Upload” o seu conteúdo será copiado para o banco de dados como *datatype image*. Desta forma será possível acessar o arquivo completo no

momento da consulta final. Após o preenchimento dos dados basta clicar no botão “Gravar”.

No processo de inclusão de um arquivo no banco, cada linha dele será lida e armazenada em uma linha de tabela. A seguir cada linha será desmembrada em palavras e em sentenças, sendo que as sentenças poderão compreender um parágrafo inteiro ou apenas uma oração. Estas sentenças serão distribuídas igualmente entre *testsets* que poderão depois ser manipulados separadamente, permitindo a utilização da validação cruzada na avaliação de métricas de desempenho do aprendizado de máquina.

Quando o arquivo for associado a um tipo, todas as sentenças que forem extraídas deles serão etiquetadas com o tipo associado. Será possível incluir mais de um arquivo por tipo ou também arquivos sem a associação de um tipo específico. Neste caso as sentenças apenas poderão ser utilizadas para classificação.

Nesta mesma tela será possível selecionar um arquivo já carregado para o banco para alteração ou para exclusão. O processo pode ser feito da mesma forma que na tela de Tipos utilizando os botões “Gravar” e “Limpar”. Abaixo segue a tela do sistema associada a esta etapa do processo:

Figura 11 - Tela Carga de Arquivos

4.1.1.3.Fragmentação do texto

Um ponto muito controverso é a fragmentação do texto importado em sentenças. Para nós existe uma unidade importante que definirá o conjunto de palavras que será analisado. Na realidade é possível encontrar uma variação muito grande em torno disso. Não existe um padrão que defina que um ALO deva possuir sempre apenas uma oração ou um parágrafo. Por isso disponibilizamos as duas alternativas que poderão ser escolhidas pelo usuário.

Na fragmentação por oração, o que definirá a quebra das sentenças é um "ponto final", "ponto de exclamação" ou "ponto de interrogação". Na segunda opção a quebra se dará apenas no parágrafo o que resultará em sentenças com várias orações. Com esta simplificação do problema estaremos perdendo informações valiosas, mas ganhamos em agilidade na obtenção de resultados. Clicando em um dos dois botões disponibilizados: "Fragm. por Oração" e "Fragm. por Paragrafo", a fragmentação do texto será executada.

Na mesma tela também serão disponibilizadas as sentenças resultantes do processo de desmembramento, para que sejam possíveis consultas após a fragmentação. As sentenças criadas serão apresentadas na tela uma a uma, utilizando para isso os botões "Próxima" e "Anterior". Durante esta consulta é possível alterar o tipo de uma sentença específica diferenciando-a das outras constantes no arquivo. Para isto basta selecionar a sentença e clicar no botão "Alterar". O botão "Limpar" exclui a sentença apresentada.

4.1.2.Geração do Modelo Inicial

A geração do Modelo Inicial é uma tarefa necessária e bastante importante no processo de classificação de textos. Este modelo servirá como base para o aprendizado semi-supervisionado e, como foi citado, será criado utilizando-se apenas os exemplos. Faz parte desta etapa do processo o pré-processamento do texto para a criação dos dados de input para o algoritmo e a geração do modelo multinomial propriamente dito.

O pré-processamento preencherá a tabela *Palavra* e, de acordo com os parâmetros informados, atualizará a tabela *Corpus*. Logo após preencherá a

tabela *Nip* que relacionará as palavras existentes em cada uma das sentenças e a quantidade de vezes que elas ocorrem.

Os dados do modelo serão armazenados nas tabelas *Plij*, fator de cada sentença para cada classe, e *Fljp*, fator de cada palavra para cada classe.

4.1.2.1.Pré-processamento do Corpus

O primeiro passo para a geração do modelo inicial é o Pré-processamento do Corpus. Após a extração do texto de arquivos txt para a base de dados, é necessário fazer um tratamento, via rotinas, facilitando o manuseio do conteúdo dos documentos. Estas rotinas permitem a estruturação da informação contida nos textos na forma de tabelas, de onde serão gerados os dados esperados para o input do algoritmo de aprendizado de máquina.

Abaixo relacionamos as técnicas de pré-processamento incluídas no SQLLOMining. Cada uma delas visa dar ênfase a alguma informação que está contida no texto, permitindo desta forma que ela não se perca quando a análise estatística for feita. Estas técnicas poderão ser utilizadas pelo usuário através de parâmetros de execução do pré-processamento, as quais serão disponibilizadas para preenchimento na tela correspondente.

Abaixo segue a tela do sistema associada a esta etapa do processo:

Figura 12 – Tela Pré-Processamento do Corpus

4.1.2.1.1.Saco de Palavras

Após a separação das sentenças é necessária a separação por palavra do texto para que seja possível a criação do saco de palavras (*Bag of Words*). Este modelo é bastante difundido nas aplicações de categorização, especificando quais palavras aparecem em cada sentença ou quantas vezes cada uma delas é utilizada. Este será o input principal para o algoritmo de classificação. No nosso caso estaremos utilizando a quantidade de ocorrência de cada palavra.

Constrói-se então uma tabela que conterá cada uma das palavras existentes nos textos analisados, além de uma outra tabela com a relação de palavras por sentença e a quantidade de vezes que aquela palavra aparece naquela sentença (*Nip*). A tabela abaixo ilustra o modelo de saco de palavras:

sentencaid	palavraid	Palavra	Nip
5546	78575	kelvin	1
5546	79164	Ação	1
5546	79258	Atuam	1
5546	80938	Contrários	1
5546	81024	e	2

Tabela 1 – Saco de Palavras Unigrama

4.1.2.1.2.N-Grama

Por se tratar apenas da contagem de vezes que a palavra aparece, diversas características do texto não são capturadas como, por exemplo, a ordem em que elas ocorrem. Isso dificulta a percepção da semântica do texto, pois diversas palavras na linguagem natural dependem do contexto para assumir significados distintos.

Um N-grama é um conjunto de N palavras consecutivas extraídas de uma sentença. O recurso de N-grama tem como objetivo capturar informações gramaticais no texto, aumentando a semântica capturada e amenizando a limitação do processo, devido ao fato de não se levar em conta a ordem em que as palavras são utilizadas nas sentenças.

O modelo de saco de palavras é levemente modificado, pois passamos a contar a quantidade de vezes que o conjunto de n palavras ocorre no texto. Um

caso em que tenhamos “é a” ou “é definido” é mais significativo do que apenas as palavras “é”, “a” e “definido” consideradas separadamente.

A tabela abaixo ilustra o modelo de N-Grama para N = 2. Este parâmetro estará disponível para o usuário escolher no momento da execução do passo de pré-processamento do texto. Caso o usuário queria utilizar o saco de palavras simples basta ele especificar o N-grama = 1.

sentencaid	palavraid	palavra	Nip
5546	78575	0 kelvin	1
5546	79164	ação e	1
5546	79258	atuam em	1
5546	80938	contrários e	1
5546	81024	corpos diferentes	1

Tabela 2 – Saco de Palavras Bigrama

4.1.2.1.3.Stemming

Freqüentemente, estamos observando diversas palavras que na verdade são somente variantes de uma única palavra. Plurais, formas de gerúndio e sufixos de tempo são exemplos de variações sintáticas, que impedem uma perfeita combinação entre uma palavra tratada como atributo e uma palavra do respectivo documento. Este problema pode ser parcialmente solucionado com a substituição de palavras pelos respectivos *stems* delas.

Stem é o conjunto de caracteres resultante de um procedimento de *stemming*. Ele não necessariamente é igual à raiz lingüística, mas servirá como uma denotação mínima não ambígua do termo. *Stemming* consiste em reduzir todas as palavras ao mesmo *stem*, por meio da retirada dos afixos da palavra, permanecendo apenas a sua raiz. O propósito é chegar a um *stem* que captura uma palavra com generalidade suficiente para permitir o sucesso na combinação de caracteres, mas sem perder muito detalhe e precisão. Um exemplo típico de um stem é “conect” que é o stem de “conectar”, “conectado” e “conectando”.

Em (Porter, 1980) é apresentado um dos métodos existentes, e que é o mais utilizado para este processo. A partir das linhas básicas deste método foi desenvolvido por Keith Lubell (Keith, 2006) um algoritmo em Structured Query Language (SQL). Utilizamos este algoritmo com algumas modificações sugeridas em (Matsubara e Monard, 2006) para a adaptação para a língua portuguesa. O

algoritmo desenvolvido numa *function* em SQL, e impresso como apêndice a este trabalho, utilizou três fases distintas onde cada uma delas tratava um tipo de pós-fixo. Estes tratamentos foram listados em uma tabela que foi também anexada para referência. Além disso, foram calculados os pontos de corte das palavras limitando a redução feita nas etapas de tratamento dos pós-fixos.

O usuário será convidado a definir um parâmetro que indicará se deverá ser utilizado *stemming* ou não. Caso ele defina este parâmetro como positivo o sistema executará o algoritmo *stemming* que gerará o *stem* de cada uma das palavras existentes no Corpus, e acrescentará esta informação a esta tabela (*Corpus*). Depois disso será criada a relação normal de palavras e também a relação de palavras por sentença baseadas no *stem* delas.

4.1.2.2. Algoritmo de aprendizado de máquina

Após a geração do Corpus devemos partir para o processo de criação do modelo que permitirá a classificação de textos. O algoritmo de aprendizado de máquina será utilizado para a geração do modelo a partir dos exemplos e para seu posterior aprimoramento através do mecanismo de aprendizado semi-supervisionado utilizando amostras adicionadas ao Corpus. A seguir falaremos sobre cada uma destas etapas do algoritmo de aprendizado de máquina.

4.1.2.2.1. Geração do Modelo Multinomial

Nesta etapa será possível especificar alguns parâmetros para a geração do modelo de misturas multinomiais. Os parâmetros dizem respeito ao *testset* a ser utilizado, além da especificação se será ou não utilizada a suavização de Laplace nos cálculos.

Nesta etapa é importante utilizar apenas sentenças etiquetadas ou seja, exemplos, permitindo a geração de um modelo o mais preciso possível. Como resultado, serão disponibilizadas na tela as medidas Accuracy e Abrangência alcançadas pelo modelo, quando aplicado aos mesmos exemplos que o geraram.

Na figura 13 segue a tela do sistema associada a esta etapa do processo:

Figura 13 – Tela Geração do Modelo Multinomial

4.1.2.3. Aprendizado Semi-Supervisionado

Num segundo passo, ainda desta etapa, será possível executar o aprendizado semi-supervisionado e acompanhar o resultado alcançado através das estatísticas calculadas a cada iteração. Neste passo os parâmetros solicitados serão, além dos mesmos da geração do modelo, a quantidade de amostras a ser adicionada.

O Aprendizado pode ser executado sobre um *testset* ou sobre um arquivo específico como detalhado no próximo item. Desta forma disponibilizaremos uma ferramenta bem poderosa, que permitirá o usuário classificar sentenças não etiquetadas e ao mesmo tempo aprimorar o seu modelo de Misturas Multinomiais.

Na figura 14 segue a tela do sistema associada a esta etapa do processo:

Figura 14 – Tela Aprendizado Semi_supervisionado

4.1.3.Carga do Arquivo a ser Classificado

Outra opção disponibilizada no sistema SQLLOMining é a carga de um arquivo do qual se deseja retirar os ALOs. Este procedimento terá algumas etapas que repetirão o que já vimos nos itens anteriores, mas executados apenas para um arquivo e não para o Corpus completo.

4.1.3.1.Carga e Fragmentação

O primeiro passo é carregar o arquivo para dentro do Corpus. Será possível definir um tipo para ele. A tela relativa a esta etapa segue abaixo.

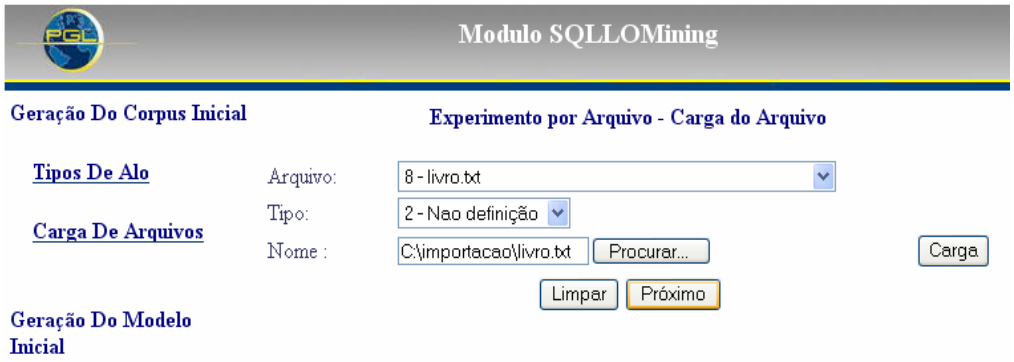

Figura 15 – Tela Experimento por Arquivo - Carga do Arquivo

Logo após, clicando no botão “Próximo” segue-se para o passo seguinte onde se pode escolher como se deseja fragmentar o texto existente neste arquivo. As sentenças resultantes poderão ser consultadas e o tipo de cada uma delas poderá ser alterado.

Figura 16 – Tela Experimento por Arquivo – Fragmentação do Texto

O próximo passo será o pré-processamento onde se especificará o n-grama e se o algoritmo de *stemming* será utilizado, da mesma forma que se fez no modelo. Deve-se sempre utilizar os mesmos parâmetros de forma que o modelo possa contribuir positivamente na classificação do arquivo que está sendo acrescentado ao Corpus.

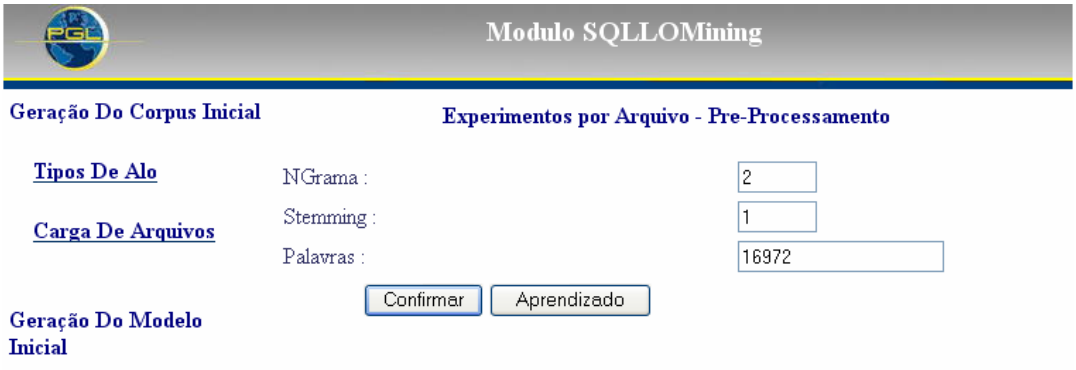

Figura 17 – Tela Experimento por Arquivo – Pré-rocessamento

4.1.3.2. Aprendizado Semi-supervisionado

O último passo será o aprendizado semi-supervisionado, onde o arquivo carregado será classificado e o modelo multinomial será recalculado utilizando-se o algoritmo EM já apresentado. Nesta etapa pode-se escolher se a suavização será utilizada ou não.

Figura 18 - Tela Experimento por Arquivo – Aprendizado Semi-Supervisionado

Como resultado, nesta mesma tela são apresentados os valores de métricas calculados após a classificação. Para cada execução de aprendizado estas métricas são recalculadas e listadas na tabela que se apresenta na tela. Clicando em uma das linhas da tabela principal o valor de *Precision* e *Recall* para cada classe é apresentado na tabela secundária, detalhando melhor os resultados alcançados.

4.1.4. Pesquisa de ALOs

A tela de pesquisa de ALOs fará a busca de acordo com os parâmetros: palavra chave e tipo, definidos pelo usuário. Na tela de resposta será disponibilizado tanto um *link* para o arquivo correspondente, como o trecho em questão. As sentenças selecionadas serão *rankeadas* pelos fatores calculados pelo classificador.

Abaixo segue a tela do sistema associada a esta etapa do processo:

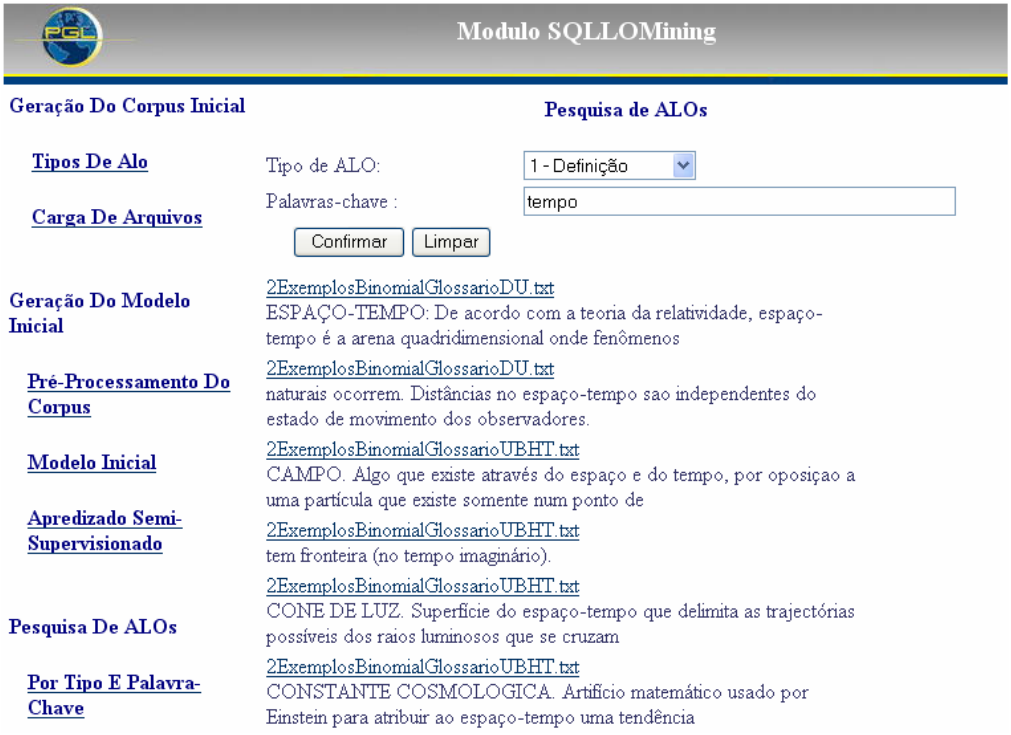

Figura 19 – Tela Pesquisa de ALOs

Acrescentamos também a pesquisa de sentenças classificadas erradas, acrescida de dois parâmetros: o tipo e palavras-chave de interesse. Ela retornará as sentenças classificadas em uma classe específica, que é definida como um dos parâmetros da consulta, e que é diferente da definida na etiquetagem da sentença. E a última pesquisa fornecida seleciona as sentenças pertencentes a um arquivo especificado, e que tenham sido classificadas em um tipo, também especificado pelo usuário. Ela retornará as sentenças que foram identificadas pelo classificador como ALOs, e que são pertencentes a um dos arquivos carregados para classificação. Nesta pesquisa também é possível definir como parâmetro palavras-chave de interesse.

Estas consultas tornaram-se úteis durante os experimentos feitos, pois elas permitiram a avaliação dos resultados alcançados e a análise de casos específicos. Muitas vezes estas análises puderam orientar os experimentos de modelagem, indicando ocorrências especiais na classificação de sentenças que possuíam uma característica singular no texto.

Abaixo segue a tela do sistema associada a esta etapa do processo:

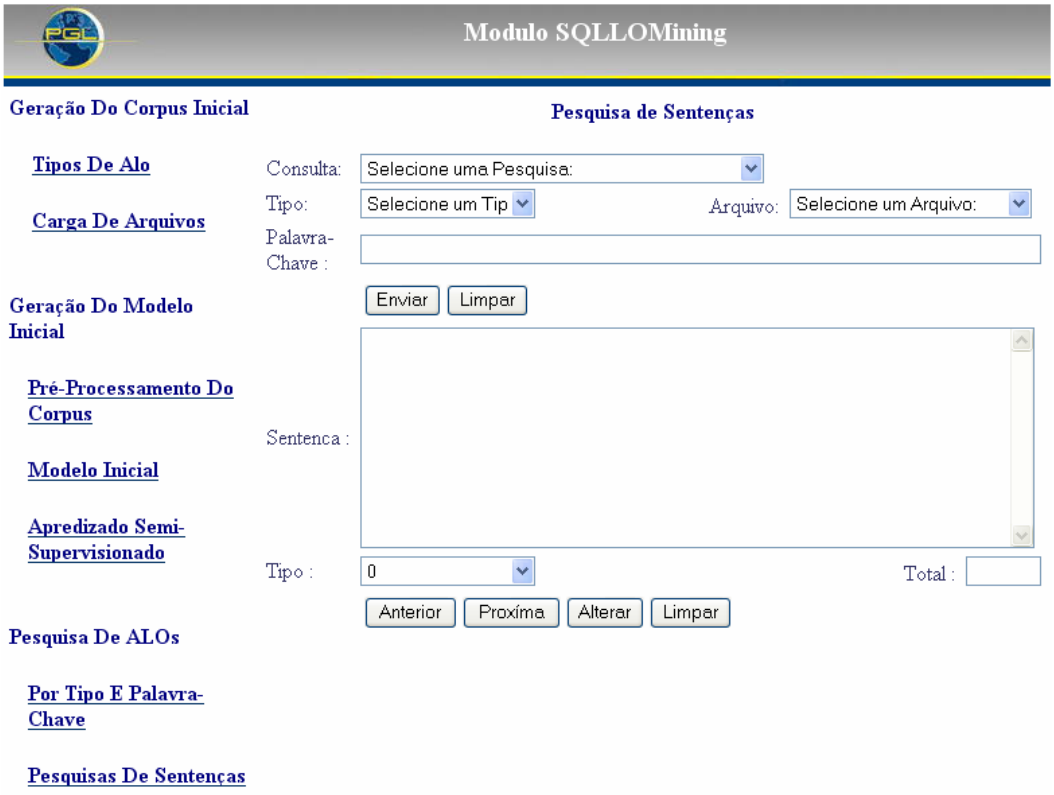

Figura 20 – Tela Pesquisa de Sentenças

4.2. Diagramas de Especificação

A seguir faremos uma especificação formal do sistema desenvolvido com diagramas de classe e de entidade relacionamento. O sistema foi desenvolvido na plataforma Java utilizando o Framework Struts (Struts) e um banco de dados relacional.

Alguns dos motivos para utilização do Strutus Framework que podemos enumerar são: garantia da Apache Group que manterá a manutenção e aprimoramento do framework (correção de *bugs* e novos *releases*); foco de esforços em regras de negócio; separação da camada de negócio da camada de apresentação; criação de aplicações padronizadas, facilitando a manutenção; criação de aplicações internacionalizadas; possibilidade de gerar a saída de acordo com o dispositivo usado (HTML, SHTML, WML, etc). A Struts está disponível sobre a licença "*free-to-use-license*" da Apache Software Foundation.

Todas as rotinas de cálculo e pré-processamento de texto foram feitas em Structured Query Language - SQL. Elas estão disponibilizadas no Apêndice II deste documento.

4.2.1. Diagrama de Classes

Abaixo segue o diagrama de classes com a estrutura utilizada na programação em java. Esta parte do sistema diz respeito apenas à manipulação de informações básicas, que disponibilizará as telas de cadastro e de consulta descritas anteriormente. A parte maior do sistema foi implementada em SQL utilizando Procedures, que são comentadas mais a frente.

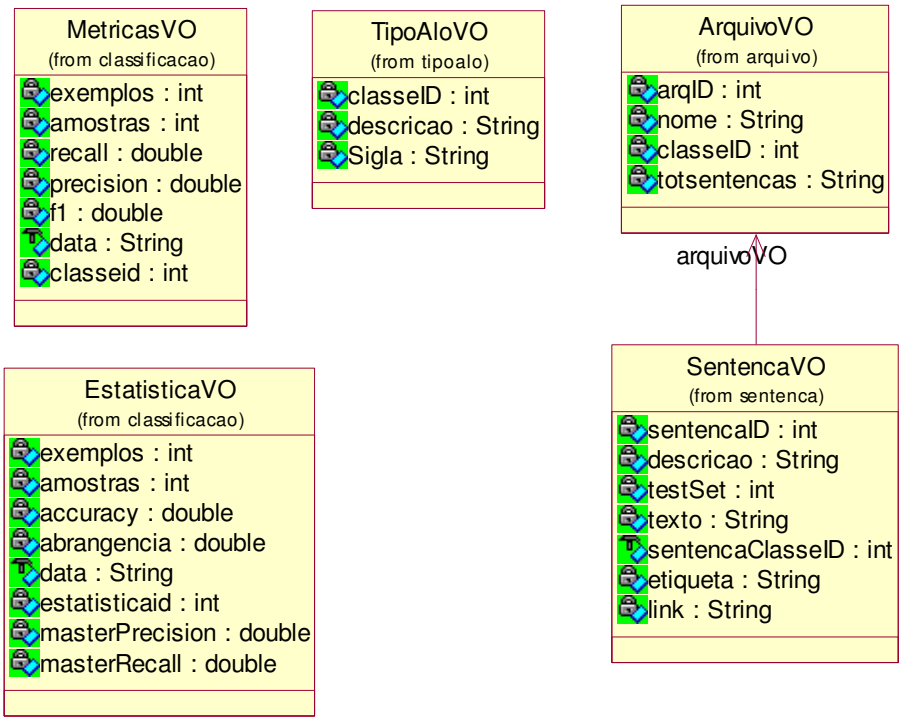

Figura 21 - Diagrama de Classes

Neste digrama apresentamos apenas as classes que representam os tipos de ALOs que o usuário cadastrou, os arquivos que foram carregados pelo usuário e as sentenças que foram geradas a partir da fragmentação destes arquivos. Adicionalmente temos as classes que representam os resultados obtidos e que são retornados para usuário, apresentando as métricas do desempenho obtido na classificação.

4.2.2.Diagramas de Modelagem

O diagrama Entidade Relacionamento da figura abaixo apresenta as entidades principais e os seus relacionamentos.

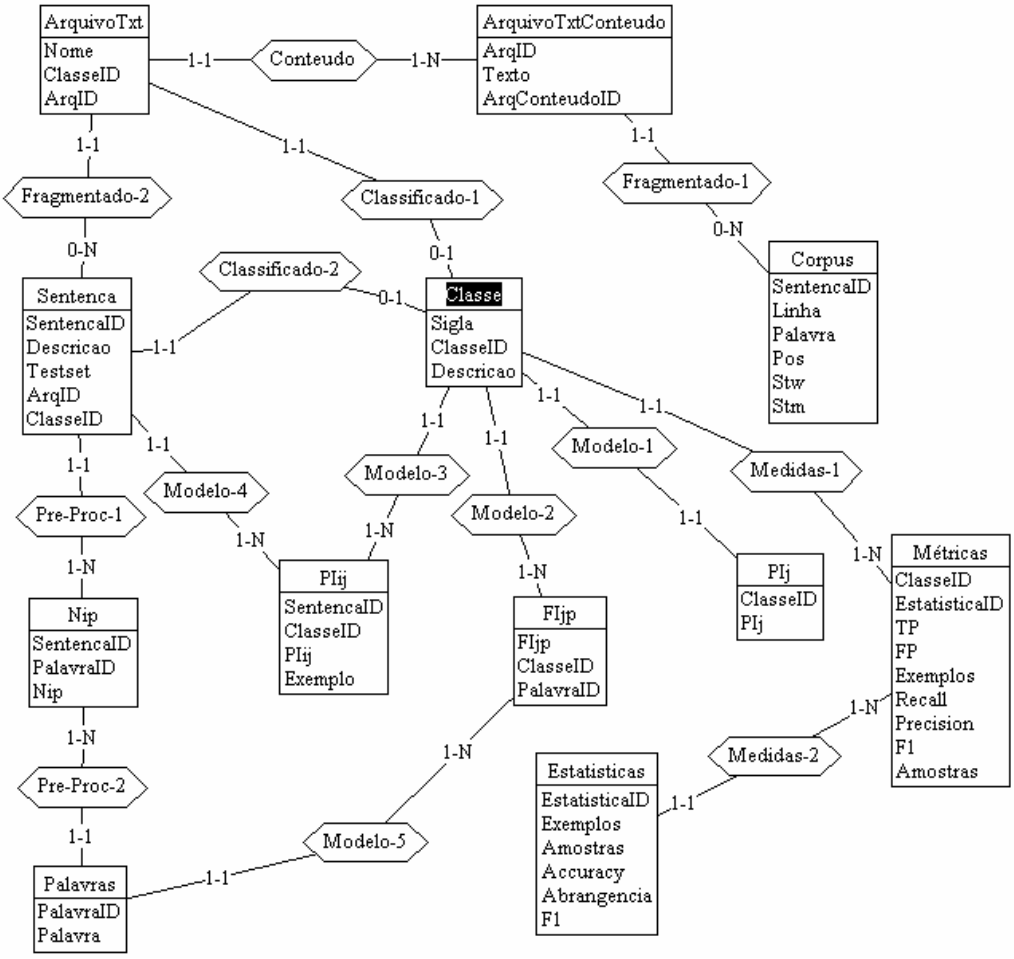

Figura 22- Diagrama ER

No primeiro passo do processo as entidades *Classe*, *ArquivosTxt* e *ArquivosTxtConteudo* são criadas com os dados informados nas interfaces de cadastro disponibilizadas para os usuários. Feito a fragmentação do texto as entidades *Corpus* e *Sentenca* são criadas com as partes dos textos. Na entidade *Corpus* a separação é feita ao nível de palavra e as sentenças são separadas e listadas na entidade *Sentenca*.

No segundo passo do processo é feito o Pré-processamento do Corpus. Neste momento as entidades *Palavras* e *Nip* são geradas, a primeira com a lista

de palavras a serem usadas na modelagem, ou seja, com a lista das *features* do modelo e a segunda com o saco-de-palavras por sentença onde são listadas as palavras que ocorrem em cada sentença com a quantidade de vezes que cada uma delas aparecem.

No terceiro passo o algoritmo será utilizado para a geração do modelo. Neste momento os parâmetros estimados do modelo serão calculados e gerados em *Plj* e *Fljp*. E finalmente no último passo faremos o aprendizado semi-supervisionado. Inicialmente as amostras serão estimadas com os valores de *Plij* e depois o modelo será reestimado alterando os valores mencionados anteriormente.

Para estas etapas do algoritmo o foram implementadas três procedures: ModeloAPartirDeExemplos, ChuteInicialAmostras e AprendizadoAmostras, elas estão disponíveis no Apêndice.

É interessante ressaltar que todos os cálculos de somatórios foram facilmente implementados na linguagem SQL, pois podem ser executados em apenas um comando. Como exemplo temos a fórmula (9) apresentada no capítulo 4 que especifica o cálculo para um dos parâmetros do modelo.

Loop j = 0 até a quantidade total de classes

$$\pi_j = \frac{\sum_i \pi_{ij}}{i} \qquad (9)$$

Ao invés da execução de uma repetição é necessário apenas fazer um somatório da forma apresentada a seguir, utilizando a estrutura do banco de dados para a execução do cálculo em apenas um comando.

```
INSERT INTO Plj (classeID, Plj)
SELECT classeID, SUM(Plij) / (SELECT COUNT(*) FROM Plij)
FROM Plij
 GROUP BY classeID
```

A seguir apresentamos o modelo lógico do banco de dados do sistema SQLLOMining.

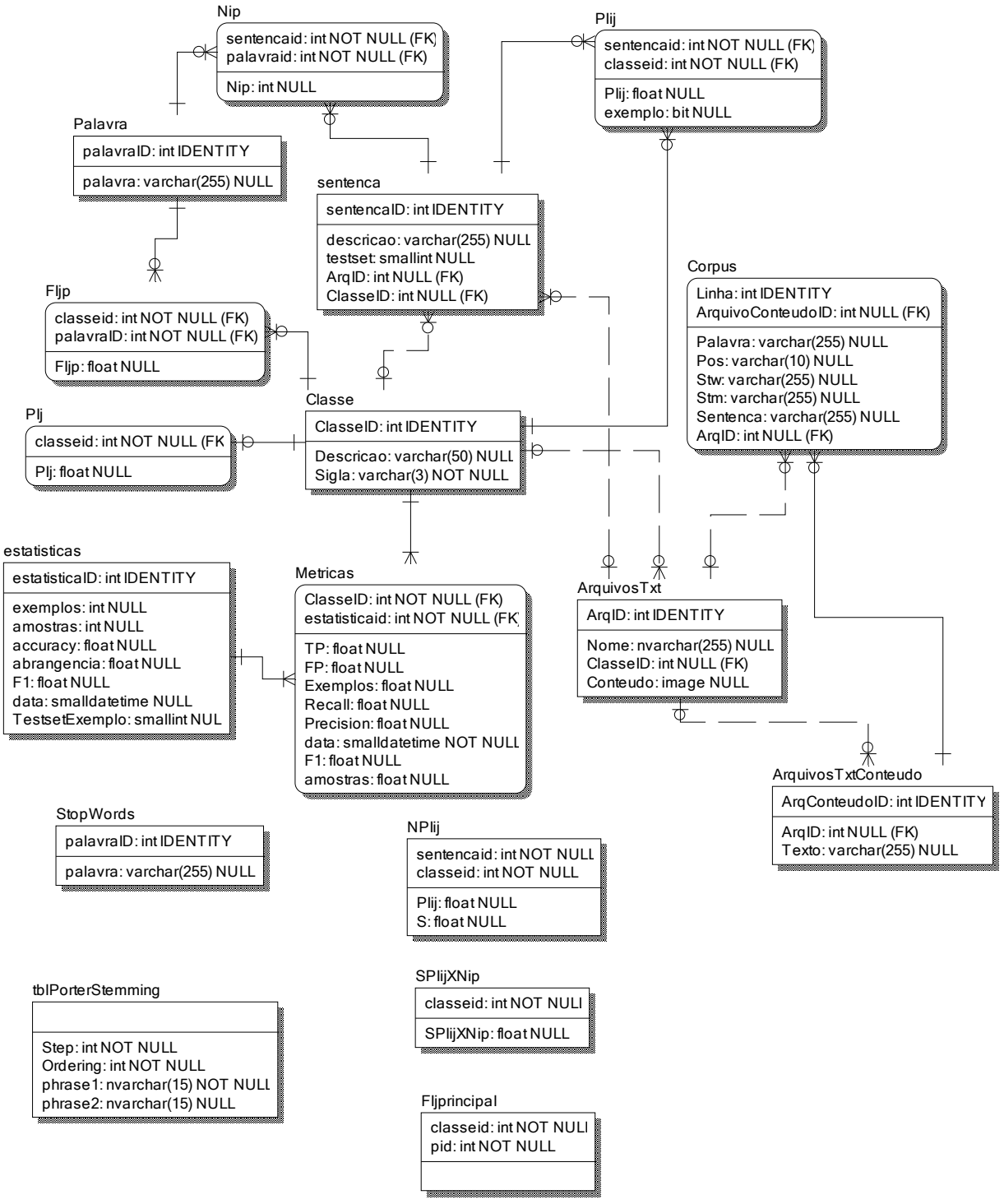

Figura 233- Modelo Lógico

Estão apresentadas neste diagrama todas as entidades mencionadas anteriormente em forma de tabelas já com suas colunas e seus relacionamentos.