Referências bibliográficas

BARUQUE, L. B. ; MELO, R. N. . **Reference Model for e-Learning Governance.** In: 22nd ICDE (International Concil for Open and Distance Education), 2006. Proceedings 22nd ICDE 2006, 2006.

BARUQUE, C. B; MELO, R. N.. **Desenvolvimento de Bibliotecas Digitais de Learning Objects Utilizando Técnicas de Data Warehousing e Data Mining**, Tese de Doutorado, PUC-Rio, 2005.

BLUM, A, MITCHELL T. - **Combining labeled and unlabeled data with co-training**. ACM Press, New York, NY, 1998.

BORMAN, S. - **The Expectation Maximization Algorithm - A short tutorial**. October 2006. http://www.seanborman.com/publications/EM_algorithm.pdf - Acesso em Julho de 2007

CHAKRABARTI, S. - **Mining the Web: Discovering Knowledge from Hypertext Data**, Morgan Kaufmann, 2002

CISCO (1999) - **Cisco Systems Reusable Information Object Strategy - Version 3.0** - http://www.cisco.com/warp/public/779/ibs/solutions/learning/whitepapers/el_cisco_rio.pdf - Acesso em Julho de 2007.

DEMPSTER, P., LAIRD, N. M., and RUBIN, D. B.- **Maximum likelihood from incomplete data via the em algorithm**. Journal of the Royal Statistical Society: Series B, 39(1):1–38, November 1977.

GENNARI, J. H., MUSEN, M. A., FERGERSON, W., GROSSO, W. E., CRUBÉZY, M., ERIKSSON, H., NOY, N. F., and TU, S. W. -**The evolution of Protégé: an environment for knowledge-based systems development.** International Journal of Human-Computer Studies, 58(1):89–123, 2003.

GLEISER, M. - **A dança do universo: dos mitos de criação ao Big-Bang** – São Paulo: Companhia das Letras, 2006

GOMES, G. R. R., MELO R. N., SIQUEIRA, S. W. M., BRAZ, M. H. L. B. . **Integrated Searches over Digital Libraries and E-learning Systems.** In: WCCSETE 2006 Congresso Mundial de Educação em Engenharia, Tecnologia e Ciência da Computação, 2006, Itanhaém, Santos, 2006.

HAWKING, S. - **Uma Breve História do Tempo – Do Big-Bang aos Buracos Negros**. Editora Rocco, 1988.

J2EEBrasil - **O site da comunidade J2EE no Brasil** - www.j2eebrasil.com.br- Acesso em Julho de 2007.

LEAL, S, MELO R. N**.- Uma arquitetura para Integração de Repositórios de Objetos de Aprendizagem baseada em Mediadores e Serviços Web**. Dissertação (Mestrado em Informática) - Pontifícia Universidade Católica do Rio de Janeiro, 2006.

LEARNING TECHNOLOGY STANDARDS COMMITTEE **– Draft standard for learning object metadata**. Technical Report IEEE P1484.12.1/D6.4, IEEE March 2002.

LOVINS, J. B. - **Development of a stemming algorithm**. Mechanical Translation and Computacional Linguistics 11 (1–2), 22–31, 1968.

LUBELL, K. (May 2006) – **SQL Stemming algorithm**. http://www.tartarus.org/martin/PorterStemmer/tsql.txt - Acesso em Julho de 2007

MATSUBARA, E. T.; MONARD, M. C. **- O algoritmo de aprendizado semi-supervisionado co-training e sua aplicação na rotulação de documentos.** Dissertação (Mestrado em Informática) - Universidade de São Paulo, USP, Brasil, 2004.

MCCALLUM, A., E NIGAM, K. – **A comparison of event models for naive bayes text classification.** In AAAI-98 Workshop on Learning for Text Categorization. Tech. rep. WS-98-05, AAAI Press. 1998.

MCLACHLAN, G. J., & KRISHNAN, T. -**The EM Algorithm and Extensions**. John Wiley and Sons, New York 1997.

MELO, R. N. ; BARUQUE, L.B.: **A Database Approach to Partnership In Global Learning.** Proc. I PGL Database Research Conference.(2003), Disponível em: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-70/paper5.pdf

MELO, R. N. ; BARUQUE, C. B. ; BARUQUE, L. B.. **Applying Governance in e-learning: a risk-based approach.** In: IADIS - INTERNATIONAL ASSOCIATION FOR DEVELOPMENT OF THE INFOMATION, 2005, Portugal 2005.

MITCHELL, T. M. - **Machine Learning**. Boston, USA: McGraw-Hill, 1997.

NIGAM, K., MCCALLUM, A.; THRUN, S.; MITCHELL, T. **- Text classification from labeled and unlabeled documents using EM**. Machine Learning, 39(2/3): 103–134, 2000.

OCHI, L.S., DIAS, C.R., SOARES, S.S.F. – **Clusterização em Mineração de Dados.** Disponível em: http://bibliotecadigital.sbc.org.br/?module=Public&action=PublicationObjects&Subjects=154&publicationobjectid=9. Acesso em Julho de 2007.

PAPOULIS, A. **Probability, Random Variables, and Stochastic Processes.** 2nd ed. New York: McGraw-Hill, 1984.

PEREIRA, L. A. M., PORTO, F. A. M., MELO, R. N. **Objetos de Aprendizado Reutilizáveis (RLOs):** conceitos, padronização, uso e armazenamento. Monografia em Ciência da Computação N° 10/03, Departamento de Informática, PUC-Rio, 2003.

PORTER, M. - **An algorithm for suffixing stripping**. Program 14(3), 130–137, 1980.

SILVA, D. S.; MELO, R. N.; SIQUEIRA, S. W. M.; BRAZ, M. H. L. B. **Uma Linguagem para Especificação de Seqüências de Objetos de Aprendizagem.** In: 3ª CONFERÊNCIA DO PGL CONSOLIDANDO EXPERIÊNCIAS EM E.LEARNING, 2005, São Paulo, Proceedings, 2005.

SOUZA, W. B. - **Tutorial - Struts Framework** - http://www.j2eebrasil.com.br/jsp/artigos/artigo.jsp?idArtigo=0011

SPARK-JONES, K., WILLET, P.- **Readings in Information Retrieval**. San Francisco:Morgan Kaufmann, 1997.

STEINBRUCH, D., SCHWABE D., MILIDIU R.- **Um estudo de algoritmos para classificação automática multirótulo de textos utilizando Naive-Bayes**, Dissertação (Mestrado em Informática) - Pontifícia Universidade Católica do Rio de Janeiro, 2006

ULLRICH, C. - **The learning-resource-type is dead, long live the learning-resourcetype!** - Learning Objects and Learning Designs, 1(1):7-15.2005. Disponível em: http://www.ags.uni-sb.de/~cullrich/publications/Ullrich-LearningResource-LOLD-2005.pdf. Acesso em Julho de 2007.

ULLRICH C. - **Description of an instructional ontology and its application in web services for education**. In Proceedings of Workshop on Applications of Semantic Web Technologies for E-learning, SW-EL'04, pages 17-23, Hiroshima, Japan, 2004. Disponível em: http://www.ags.uni-sb.de/~cullrich/publications/Ullrich-InstructionalOntology-SWEL-2004.pdf. Acesso em Julho de 2007.

VERBERT, K. and DUVAL, E. **-Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models**.2002. Disponível em: http://www.cs.kuleuven.ac.be/~hmdb/publications/files/pdfversion/41315.pdf. Acesso em Julho de 2007.

WILEY, D. A. - **Learning Object Design And Sequencing Theory**. Unpublished doctoral dissertation, Brigham Young University, 2000. Disponível em: http://opencontent.org//docs/dissertation.pdf. Acesso em Julho de 2007.

W3C - **World Wide Web Consortium**, http://www.w3.org/. Acesso em Julho de 2007.

Strutus - http://struts.apache.org/api/index.html. Acesso em Julho de 2007.

Apêndice I

Funções para Algoritmo Porter


```
/****** Object: Table [dbo].[tblPorterStemming] ******/
CREATE TABLE [dbo].[tblPorterStemming]
( [Step] [int] NOT NULL ,
[Ordering] [int] NOT NULL ,
[phrase1] [nvarchar] (15) NOT NULL ,
[phrase2] [nvarchar] (15) NULL
) ON [PRIMARY]
```

```
GO


CREATE FUNCTION [dbo].[fnPorterCVCpattern]
( @Word nvarchar(4000) )
RETURNS nvarchar(4000)
AS
BEGIN
--local variables
      DECLARE @Ret nvarchar(4000), @i int

--checking each character to see if it is a consonent or a vowel.
also inputs the information in const_vowel
      SELECT @i = 1, @Ret = ''''

      WHILE @i <= LEN(@Word)
      BEGIN
            IF CHARINDEX(SUBSTRING(@Word,@i,1), ''aeiou'') > 0
            BEGIN
                  SELECT @Ret = @Ret + ''v''
            END

-- if y is not the first character, only then check the previous
character
            ELSE IF SUBSTRING(@Word,@i,1) = ''y'' AND @i > 1
            BEGIN

--check to see if previous character is a consonent
                  IF CHARINDEX(SUBSTRING(@Word,@i-1,1), ''aeiou'')
                  = 0
                        SELECT @Ret = @Ret + ''v''
                  ELSE
                        SELECT @Ret = @Ret + ''c''
            END
            Else
            BEGIN
                  SELECT @Ret = @Ret + ''c''
            END
            SELECT @i = @i + 1
      END

      RETURN @Ret
END
'
END


GO
CREATE FUNCTION [dbo].[fnPorterStep]
( @step int, @InWord nvarchar(4000) )
RETURNS nvarchar(4000)
AS

BEGIN
      DECLARE @Ret nvarchar(255)
      DECLARE @Phrase1 NVARCHAR(15), @Phrase2 NVARCHAR(15)
      DECLARE @CursorName CURSOR

-- DO some initial cleanup
      SELECT @Ret = @InWord
```

```
-- Create Cursor for Porter Step
      SET @CursorName = CURSOR FOR
      SELECT phrase1, phrase2
      FROM tblPorterStemming
      WHERE Step = @step
      AND RIGHT(@Ret,LEN(Phrase1)) = Phrase1
      ORDER BY Ordering

      OPEN @CursorName
-- Do Step 1

      FETCH NEXT FROM @CursorName INTO @Phrase1, @Phrase2

      WHILE @@FETCH_STATUS = 0
      BEGIN
            --
            IF RIGHT(@Ret ,LEN(@Phrase1)) = @Phrase1
            BEGIN
                  SELECT @Ret = LEFT(@Ret, LEN(@Ret) -
                  LEN(@Phrase1)) + @Phrase2
                  BREAK
            END

            FETCH NEXT FROM @CursorName INTO @Phrase1, @Phrase2
      END

-- Free Resources

      CLOSE @CursorName

      DEALLOCATE @CursorName

      return @Ret

END
'
END

GO


CREATE FUNCTION [dbo].[fnPorterCountPosV]
( @Word nvarchar(4000) )
RETURNS tinyint
AS

BEGIN
--A \consonant\ in a word is a letter other than A, E, I, O or U,
and other

--declaring local variables
      DECLARE @pattern nvarchar(4000), @ret tinyint, @i int, @flag
      tinyint

--initializing
      SELECT @ret = 1, @flag = 0, @i = 1

      If Len(@Word) > 0
      BEGIN
--find out the PosV
            SELECT @pattern = dbo.fnPorterCVCpattern(@Word)
```

```
--se a palavra comeca com duas ou mais vogais, posV esta na
primeira consoante;
            IF SUBSTRING(@pattern,1,2) = ''vv''
            BEGIN
                  set @i = 3

                  WHILE @i <= LEN(@pattern)
                  BEGIN
                        IF SUBSTRING(@pattern,@i,1) = ''c''
                        BEGIN
                              IF (SUBSTRING(@pattern,@i-1,1) =
                              ''v'')
                                    BREAK
                        END
                        SELECT @ret = @ret + 1
                        SELECT @i = @i + 1
                  END
            END

--se comeca com vogal-consoante, posV esta na segunda vogal;
            IF SUBSTRING(@pattern,1,2) = ''vc''
            BEGIN
                  set @i = 3

                  WHILE @i <= LEN(@pattern)
                  BEGIN
                        IF SUBSTRING(@pattern,@i,1) = ''v''
                              BREAK
                        SELECT @ret = @ret + 1
                        SELECT @i = @i + 1
                  END
            END

--se comeca com consoante, posV esta na primeira vogal ou na
terceira letra que esta mais a direita.

            IF SUBSTRING(@pattern,1,1) = ''c''
            BEGIN
                  set @i = 1

                  WHILE @i <= 3
                  BEGIN
                        IF SUBSTRING(@pattern,@i,1) = ''v''
                              BREAK
                        SELECT @ret = @ret + 1
                        SELECT @i = @i + 1
                  END
            END
      END
      RETURN @ret
END
END
GO


CREATE FUNCTION [dbo].[fnPorterCountPos2]
( @Word nvarchar(4000) )
RETURNS tinyint
AS
```

```
BEGIN
--A \consonant\ in a word is a letter other than A, E, I, O or U,
and other

--pos2 está no fim de uma sequencia de consoantes que segue a
segunda sequencia de vogais;

--declaring local variables
      DECLARE @pattern nvarchar(4000), @ret tinyint, @i int, @flag
tinyint

--initializing
      SELECT @ret = 1, @flag = 0, @i = 1

      If Len(@Word) > 0
      BEGIN
--find out the Pos2
            SELECT @pattern = dbo.fnPorterCVCpattern(@Word)

            WHILE @i <= LEN(@pattern)
            BEGIN
                  IF SUBSTRING(@pattern,@i,1) = ''v''
                  BEGIN
                        IF (SUBSTRING(@pattern,@i-1,1) = ''c'' or
                        @i = 1)
                              SELECT @flag = @flag + 1

                        IF @flag = 3
                              BREAK
                  END
                  SELECT @ret = @ret + 1
                  SELECT @i = @i + 1
            END
      END
      RETURN @ret - 1
END
'
END

GO

CREATE FUNCTION [dbo].[fnPorterAlgorithm] ( @InWord nvarchar(4000)
)
RETURNS nvarchar(255)
AS

BEGIN

      DECLARE @Ret nvarchar(4000), @Temp nvarchar(4000), @Pos2
      int, @PosV int
-- DO some initial cleanup

      SELECT @Ret = LOWER(ISNULL(RTRIM(LTRIM(@InWord)),''''))

-- only strings greater than 2 are stemmed
      IF LEN(@Ret) > 2
      BEGIN
            select @PosV = dbo.fnPorterCountPosV(@Ret)
            select @Pos2 = dbo.fnPorterCountPos2(@Ret)
            SELECT @Ret = dbo.fnPorterStep(1,@Ret)
            SELECT @Temp = dbo.fnPorterStep(2,@Ret)
```

```
            IF LEN(@Temp) < @Pos2
                  SELECT @Ret = SUBSTRING(@Ret,1,@Pos2)
            ELSE
                  SELECT @Ret = @Temp
            IF @Ret = LOWER(ISNULL(RTRIM(LTRIM(@InWord)),''''))
            BEGIN
                  SELECT @Temp = dbo.fnPorterStep(3,@Ret)
                  IF LEN(@Temp) < @PosV
                        SELECT @Ret = SUBSTRING(@Ret,1,@PosV)
                  ELSE
                        SELECT @Ret = @Temp
            END
      END
--End of Porter''s algorithm.........returning the word

      RETURN @Ret
END
'
END
```

Tabela Stemming

Step	Ordering	Phrase1	Phrase2
1	1	-lo	
1	1	-la	
1	1	-los	
1	1	-las	
1	1	-no	
1	1	-se	
1	1	-na	
2	1	ais	al
2	1	eis	al
2	1	ns	m
2	1	res	r
2	1	ções	ção
2	2	ção	
2	5	s	
2	1	adores	ador
2	1	adoras	ador
2	2	ador	
2	2	adora	
2	3	as	
2	4	es	
2	1	abilidade	abil

2	2	abil	
2	3	idade	
2	1	ividade	iv
2	2	iv	
2	1	icidade	ic
2	2	ic	
2	1	os	
2	1	ativamente	ativ
2	2	ativ	
2	1	adamente	ad
2	2	ad	
2	1	osamente	os
2	2	os	
2	1	icamente	ica
2	2	ica	
2	1	icomente	ico
2	2	ico	
2	1	ivamente	iva
2	2	iva	
2	1	ável	
2	1	ível	
2	1	ismo	
2	1	oso	
2	1	osa	
2	1	ista	
2	1	amento	
2	1	imento	
2	1	amente	
2	1	mente	
2	1	inda	
2	1	ivo	
2	1	iva	
2	1	eza	
2	1	ção	
2	1	avel	

2	1	ivel	
2	1	at	
3	1	eremos	
3	1	eríamos	
3	1	íamos	
3	1	emos	
3	1	éssemos	
3	1	éramos	
3	1	amos	
3	1	eis	
3	1	ais	
3	1	ereis	
3	1	eríeis	
3	1	íeis	
3	1	estes	
3	1	esseis	
3	1	éreis	
3	1	íamos	
3	1	em	
3	1	am	
3	1	erão	
3	1	eriam	
3	1	iam	
3	1	eram	
3	1	essem	
3	1	eram	
3	1	erá	
3	1	eria	
3	1	erás	
3	1	erias	
3	1	ereis	
3	1	ia	
3	1	ias	
3	1	este	
3	1	esses	

3	1	era	
3	1	eras	
3	1	endo	
3	1	ida	
3	1	ido	
3	1	idas	
3	1	idos	

Apêndice II

Procedures SQL para Algoritmo EM

```
/**Object:StoredProcedure [dbo].[ModeloApartirDeExemplos]****/
/**Procedure que estima o modelo a partir dos exemplos ******/

CREATE PROCEDURE [dbo].[ModeloApartirDeExemplos]
@TotaldeSentencasExemplo int,
@SPIij bit,
@SFIjp bit,
@testset int


AS
--Utilização da classificação dos exemplos para o cálculo do
Modelo de Misturas
delete PIij

declare @TotaldeSentencasExemploClasse int

set @TotaldeSentencasExemploClasse = @TotaldeSentencasExemplo /
(select count(*) from classe)

declare @classeid int
declare @linhasatualizadas int

set @classeid = 1
While @classeid <= (select count(*) from classe)
BEGIN
      insert into PIij
      select Top (@TotaldeSentencasExemploClasse) sentencaid,
      s.classeid, 1.0, 1
      from sentenca s
      where classeid = @classeid
      and testset = @testset
      order by sentencaid

      select @linhasatualizadas = @@rowcount

      if @linhasatualizadas < @TotaldeSentencasExemploClasse
            select @TotaldeSentencasExemploClasse = (2*
            @TotaldeSentencasExemploClasse) - @linhasatualizadas
      else
            set @TotaldeSentencasExemploClasse =
            @TotaldeSentencasExemplo / (select count(*) from
            classe)

      set @classeid = @classeid + 1
END

select @TotaldeSentencasExemplo = count(distinct sentencaid) from
PIij where exemplo = 1
```

```
---COMEÇO DA ESTIMATIVA DO MODELO
BEGIN

--Cálculo de PI para cada classe

            delete PIj

            insert into PIj
            select classeid, sum(isnull(PIij,0))/(select count(*)
            from PIij)
            from PIij
            group by classeid


--Cálculo de denominador de FI de cada classe
            delete SPIijXNip

            insert into SPIijXNip
            select classeid, sum(isnull(PIij,0) * isnull(Nip,0))
            from PIij p, Nip n
            where p.sentencaid = n.sentencaid
            group by classeid

--Cálculo de FI de cada classe de cada palavra

            delete FIjp

            if @SFIjp = 1

--Suavização de Laplace

                  BEGIN
                        insert into FIjp
                        select p.classeid, n.pid, (sum((PIij *
                        Nip)) + 1.0)/ (SPIijXNip +(select count(*)
                        from palavra))
                        from PIij p, Nip n, SPIijXNip s
                        where p.classeid = s.classeid
                        and p.sentencaid = n.sentencaid
                        group by p.classeid, n.pid, SPIijXNip

--Inclusão de um valor suavizado para as componentes ainda não
contempladas

                        insert into FIjp
                        select classeid, pid, 1.0 / ((select
                        s.SPIijXNip from SPIijXNip s where
                        s.classeid = i.classeid)+(select count(*)
                        from palavra))
                        from FIjprincipal i
                        where not exists (select * from FIjp f
                        where i.classeid = f.classeid and i.pid =
                        f.pid)
                  END
            else
                  BEGIN
                        insert into FIjp
                        select p.classeid, n.pid, sum(PIij *
                        Nip)/SPIijXNip
                        from PIij p, Nip n, SPIijXNip s
```

```
                        where p.classeid = s.classeid
                        and p.sentencaid = n.sentencaid
                        group by p.classeid, n.pid, SPIijXNip

                        insert into FIjp
                        select classeid, pid, 0
                        from FIjprincipal i
                        where not exists (select * from FIjp f
                        where i.classeid = f.classeid and i.pid =
                        f.pid)
                  END

            declare @Bij table (sentencaid int, classeid int, Bij
            float)

            delete @Bij

            DECLARE CNip CURSOR FOR
            SELECT n.sentencaid, f.classeid, Nip, FIjp
            FROM Nip n inner join FIjp f on n.pid = f.pid
            WHERE sentencaid in (select sentencaID from PIij)
            order by n.sentencaid, f.classeid, Nip

            OPEN CNip;
            DECLARE @B float, @F float
            DECLARE @N int, @I int, @J int, @IA int, @JA int

            FETCH NEXT FROM CNip
            INTO @I, @J, @N, @F
            set @B = 1
            set @IA = @I
            set @JA = @J
            WHILE @@FETCH_STATUS = 0
            BEGIN
                  IF @I <> @IA OR @J <> @JA
                        BEGIN
                              insert into @Bij
                              select @IA, @JA, @B
                              set @IA = @I
                              set @JA = @J
                              set @B = Power(@F, @N)
                        END
                  ELSE
                        BEGIN
                              set @B = @B * Power(@F, @N)
                        END
                  FETCH NEXT FROM CNip
                  INTO @I, @J, @N, @F
            END;

            insert into @Bij
            select @IA, @JA, @B

            CLOSE CNip;
            DEALLOCATE CNip;

--Calculo de PI para cada sentença para cada classe
            delete NPIij

            if @SPIij = 1
--Suavização de Laplace
```

```
                  BEGIN
                        insert into NPIij
                        select distinct sentencaid, P.classeid,
                        case when (p.PIj * f.Bij) = 0 then 1.0
                        else (p.PIj * f.Bij) end, 0
                        from PIj p, @Bij f
                        where sentencaid in (select sentencaID
                        from PIij)
                        and p.classeid = f.classeid
                        order by sentencaid, P.classeid

                        update NPIij set S =
                        (select sum(case when (p.PIj * f.Bij) = 0
                        then 1.0 else (p.PIj * f.Bij) end)
                        from PIj p, @Bij f
                        where p.classeid = f.classeid
                        and f.sentencaid = NPIij.sentencaid )
                        where NPIij.sentencaid in (select
                        sentencaID from PIij)
                  END
            else
                  BEGIN
                        insert into NPIij
                        select distinct sentencaid, P.classeid,
                        (p.PIj * f.Bij), 0
                        from PIj p, @Bij f
                        where sentencaid in (select sentencaID
                        from PIij)
                        and p.classeid = f.classeid
                        order by sentencaid, P.classeid

                        update NPIij set S =
                        (select sum(p.PIj * f.Bij)
                        from PIj p, @Bij f
                        where p.classeid = f.classeid
                        and f.sentencaid = NPIij.sentencaid )
                        where NPIij.sentencaid in (select
                        sentencaID from PIij)
                  END


            update NPIij set PIij = PIij / S
            where sentencaid in (select sentencaID from PIij)

      END

--Cálculo de estatisticas e métricas

            insert estatisticas
            select @TotaldeSentencasExemplo, 0,
            convert(float,count(*)) / @TotaldeSentencasExemplo,
            0, 0, getdate(), @testset
            from NPIij n, PIij p
            Where n.sentencaid = p.sentencaid
            and n.PIij = (select max(PIij) from NPIij n2 where
            round(PIij,11) <> round(1.0/(select count(*) from
            classe),11) and n2.sentencaid = n.sentencaid)
            and round(n.PIij,11) <> round(1.0/(select count(*)
            from classe),11)
            and n.classeid = p.classeid
            and p.PIij = 1
```

```
            declare @estatisticaid int
            select @estatisticaid = @@identity

            update estatisticas set abrangencia = (select
            convert(float,count(*)) from NPIij n, PIij p
            Where n.sentencaid = p.sentencaid
            and n.PIij = (select max(PIij) from NPIij n2
            where round(PIij,11) <> round(1.0/(select count(*)
            from classe),11)
            and n2.sentencaid = n.sentencaid)
            and round(n.PIij,11) <> round(1.0/(select count(*)
            from classe),11)
            and n.classeid = p.classeid
            and p.PIij = 1
            and p.classeid = 1) /
            (select convert(float,count(*)) from PIij pp
            Where pp.classeid = 1
            and pp.PIij = 1)
            where abrangencia = 0

--Cálculo F1

update estatisticas set F1 = (2 * Accuracy * Abrangencia) /
(Accuracy + Abrangencia)


            insert metricas
            select p.classeID,
            convert(float,count(*)) as TP,
            (select convert(float,count(*)) from NPIij pp, PIij
            ppp
                              where pp.sentencaid = ppp.sentencaid
                              and pp.classeid = p.classeID
                              and round(pp.PIij,11) <>
                              round(1.0/(select count(*) from
                              classe),11)
                              and pp.PIij = (select max(PIij) from
                              NPIij n3
                              where round(PIij,11) <>
                              round(1.0/(select count(*) from
                              classe),11)
                              and n3.sentencaid = pp.sentencaid)
                              and ppp.PIij = 1
                              and ppp.classeid <> p.classeid),
            @TotaldeSentencasExemplo, 0,
            0, getdate(), 0, (select convert(float,count(*)) from
            PIij pp where pp.classeid = p.classeID and pp.PIij =
            1),
            @estatisticaid
            from NPIij n, PIij p
            Where n.sentencaid = p.sentencaid
            and n.PIij = (select max(PIij) from NPIij n2 where
            round(PIij,11) <> round(1.0/(select count(*) from
            classe),11) and n2.sentencaid = n.sentencaid)
            and round(n.PIij,11) <> round(1.0/(select count(*)
            from classe),11)
            and p.PIij = 1
            and n.classeid = p.classeid
            group by p.classeID
```

```
--Cálculo de Precision, Recall e F1 por Classe

            update metricas set Recall = TP / Amostras, Precision
            = TP / (TP + FP)

            update metricas set F1 = (2 * Recall * Precision) /
            (Recall + Precision)


select exemplos, amostras, round(accuracy, 3) as accuracy,
round(abrangencia, 3) as abrangencia, round(f1, 3) as f1,
convert(char(20), data, 13) data, estatisticaid from estatisticas
order by estatisticaid
GO



/****** Object:  StoredProcedure [dbo].[ChuteInicialAmostras] ***/
/****** Procedure que gera a estimativa para as Amostras ********/

CREATE PROCEDURE [dbo].[ChuteInicialAmostras]
@TotaldeSentencasAmostras int,
@SPIij BIT
AS

declare @TotaldeSentencasExemplo int

--Chute inicial para as amostras a partir do modelo criado
anteriormente
--Chuta apenas para as amostras adicionadas

            declare @Bij table (sentencaid int, classeid int, Bij
            float)

            delete @Bij

            DECLARE CNip CURSOR FOR
            SELECT n.sentencaid, f.classeid, Nip, FIjp
            FROM Nip n Left outer join FIjp f on n.pid = f.pid
            WHERE n.sentencaid in (select sentencaid from PIij p
            where exemplo = 0 and not exists (select * from NPIij
            where sentencaid = p.sentencaid ))
            order by n.sentencaid, f.classeid, Nip

            OPEN CNip;
            DECLARE @B float, @F float
            DECLARE @N int, @I int, @J int, @IA int, @JA int

            FETCH NEXT FROM CNip
            INTO @I, @J, @N, @F
            set @B = 1
            set @IA = @I
            set @JA = @J
            WHILE @@FETCH_STATUS = 0
            BEGIN
                  IF @I <> @IA OR @J <> @JA
                        BEGIN
                              insert into @Bij
                              select @IA, @JA, @B
                              set @IA = @I
```

```
                        set @JA = @J
                        set @B = Power(@F, @N)
                  END
            ELSE
                  BEGIN
                        set @B = @B * Power(@F, @N)
                  END
            FETCH NEXT FROM CNip
            INTO @I, @J, @N, @F
      END;

      insert into @Bij
      select @IA, @JA, @B

      CLOSE CNip;
      DEALLOCATE CNip;

--Cálculo de PI para cada sentença e para cada classe


      if @SPIij = 1

--Suavização de Laplace

            BEGIN
            insert into NPIij
            Select distinct sentencaid, P.classeid, (p.PIj *
            f.Bij) + 1, 0
            from PIj p, @Bij f
            where p.classeid = f.classeid
            order by sentencaid, P.classeid

            update nn set S =
                  (select sum((p.PIj * f.Bij)+ (select
            count(*) from classe)
                  from PIj p, @Bij f
                  where p.classeid = f.classeid
                  and f.sentencaid = nn.sentencaid )
            from @Bij ff, NPIij nn
            where nn.sentencaid = ff.sentencaid

            END
      else
            BEGIN

--sem smooth no PIij

            insert into NPIij
            Select distinct sentencaid, P.classeid, (p.PIj *
            f.Bij), 0
            from PIj p, @Bij f
            where p.classeid = f.classeid
            order by sentencaid, P.classeid

            update nn set S =
            (select sum(p.PIj * f.Bij)
                  from PIj p, @Bij f
                  where p.classeid = f.classeid
                  and f.sentencaid = nn.sentencaid )
            from @Bij ff, NPIij nn
            where nn.sentencaid = ff.sentencaid
```

```
                END

            update nn set PIij = PIij / S
            from @Bij ff, NPIij nn
            where nn.sentencaid = ff.sentencaid
GO


/** Object:  StoredProcedure [dbo].[AprendizadoAmostras] ********/
/** Procedure que executa o algoritmo EM gerando o aprendizado **/

CREATE PROCEDURE [dbo].[AprendizadoAmostras]
@TotaldeSentencasAmostras int,
@SPIij bit,
@SPIj bit,
@SFIjp bit,
@testset int
AS

Declare @TotalSentencasExemplos int
Declare @testsetExemplos smallint

if @testset = 1
      set @testsetExemplos = 2
else
      set @testsetExemplos = 1

--REATUALIZAR A CLASSIFICAÇÃO DOS EXEMPLOS

delete NPIij

insert into NPIij
select distinct sentencaid, classeid, PIij, 0
from PIij
where exemplo = 1

--GUARDAR CLASSIFICAÇÃO DA AMOSTRA
declare @TotaldeSentencasAmostrasClasse int
declare @linhasatualizadas int

set @TotaldeSentencasAmostrasClasse = @TotaldeSentencasAmostras /
(select count(*) from classe)

declare @classeid int
set @classeid = 1
While @classeid <= (select count(*) from classe)
BEGIN
      insert into PIij
      select Top (@TotaldeSentencasAmostrasClasse) sentencaid,
      s.classeid, 1.0, 0
      from sentenca s
      where classeid = @classeid
      and testset = @testset
      and not exists(select * from PIij where sentencaid =
      s.sentencaid)
      order by sentencaid

      select @linhasatualizadas = @@rowcount

      if @linhasatualizadas < @TotaldeSentencasAmostrasClasse
```

```
            select @TotaldeSentencasAmostrasClasse = (2*
            @TotaldeSentencasAmostrasClasse) - @linhasatualizadas
      else
            set @TotaldeSentencasAmostrasClasse =
            @TotaldeSentencasAmostras / (select count(*) from
            classe)

      set @classeid = @classeid + 1
END

-- E-Step - Estimativa para as amostras

exec ChuteInicialAmostras @TotaldeSentencasAmostras, @SPIij

select @TotaldeSentencasAmostras = count(distinct sentencaid) from
PIij where exemplo = 0

select @TotalSentencasExemplos = count(distinct sentencaid) from
PIij where exemplo = 1

--COMEÇO DAS ITERAÇÕES DO ALGORITMO EM
declare @iteracoes int
declare @iteracoesf int

select @iteracoes = max(iteracao)+1 from GPIj

select @iteracoesf = @iteracoes + 10

WHILE @iteracoes <= @iteracoesf
      BEGIN
--M-Step Inicio

--Cálculo de PI para cada classe

            delete PIj

            if @SPIj = 1

--Suavização de Laplace

                  insert into PIj
                  select classeid, (sum(isnull(PIij,0)) +
                  1.0)/((select count(distinct sentencaid) from
                  NPIij) + (select count(*) from classe))
                  from NPIij
                  group by classeid
            else
                  insert into PIj
                  select classeid, (sum(isnull(PIij,0)))/(select
                  count(distinct sentencaid) from NPIij)
                  from NPIij
                  group by classeid

--Guarda a convergência

            insert into GPIj (exemplos, iteracao, classeid, PIj)
            select @TotalSentencasExemplos, @iteracoes, * from PIj

            delete SPIijXNip

--Cálculo de FI para cada palavra para cada classe
```

```
            insert into SPIijXNip
            select classeid, sum(isnull(PIij,0) * isnull(Nip,0))
            from NPIij p, Nip n
            where p.sentencaid = n.sentencaid
            group by classeid

            delete FIjp

            if @SFIjp = 1

--Suavização de Laplace
                  BEGIN
                        insert into FIjp
                        select p.classeid, n.pid, (sum((PIij *
                        Nip)) + 1.0)/ (SPIijXNip +(select count(*)
                        from palavra))
                        from NPIij p, Nip n, SPIijXNip s
                        where p.classeid = s.classeid
                        and p.sentencaid = n.sentencaid
                        group by p.classeid, n.pid, SPIijXNip

--Inclusão de um valor suavizado para as palavras ainda não
contempladas

                        insert into FIjp
                        select classeid, pid, 1.0 / ((select
                        s.SPIijXNip from SPIijXNip s where
                        s.classeid = i.classeid)+(select count(*)
                        from palavra))
                        from FIjprincipal i
                        where not exists (select * from FIjp f
                        where i.classeid = f.classeid and i.pid =
                        f.pid)
                  END
            else
                  BEGIN
                        insert into FIjp
                        select p.classeid, n.pid, sum(PIij *
                        Nip)/SPIijXNip
                        from NPIij p, Nip n, SPIijXNip s
                        where p.classeid = s.classeid
                        and p.sentencaid = n.sentencaid
                        group by p.classeid, n.pid, SPIijXNip

                        insert into FIjp
                        select classeid, pid, 0
                        from FIjprincipal i
                        where not exists (select * from FIjp f
                        where i.classeid = f.classeid and i.pid =
                        f.pid)
                  END
--M-Step Fim
--E-Step Inicio

            declare @Bij table (sentencaid int, classeid int, Bij
            float)

            delete @Bij

            DECLARE CNip CURSOR FOR
```

```
            SELECT n.sentencaid, f.classeid, Nip, isnull(FIjp,0)
            FROM Nip n Left outer join FIjp f on n.pid = f.pid
            WHERE sentencaid in (select sentencaid from PIij where
            exemplo = 0)
            order by n.sentencaid, f.classeid, Nip

            OPEN CNip;
            DECLARE @B float, @F float
            DECLARE @N int, @I int, @J int, @IA int, @JA int

            FETCH NEXT FROM CNip
            INTO @I, @J, @N, @F
            set @B = 1
            set @IA = @I
            set @JA = @J
            WHILE @@FETCH_STATUS = 0
            BEGIN
                  IF @I <> @IA OR @J <> @JA
                        BEGIN
                              insert into @Bij
                              select @IA, @JA, @B
                              set @IA = @I
                              set @JA = @J
                              set @B = Power(@F, @N)
                        END
                  ELSE
                        BEGIN
                              set @B = @B * Power(@F, @N)
                        END
                  FETCH NEXT FROM CNip
                  INTO @I, @J, @N, @F
            END;

            insert into @Bij
            select @IA, @JA, @B

            CLOSE CNip;
            DEALLOCATE CNip;

--Recalcula PI para todas as amostras
            delete NPIij
            where sentencaid in (select sentencaid from PIij where
            exemplo = 0)

            if @SPIij = 1
                  BEGIN
                        insert into NPIij
                        select distinct sentencaid, P.classeid,
                        case when (p.PIj * f.Bij) = 0 then 1.0
                        else (p.PIj * f.Bij) end, 0
                        from PIj p, @Bij f
                        where p.classeid = f.classeid
                        order by sentencaid, P.classeid

                        update NPIij set S =
                        (select sum(case when (p.PIj * f.Bij) = 0
                        then 1.0 else (p.PIj * f.Bij) end) --+
                        (select count(*) from classe)
                        from PIj p, @Bij f
                        where p.classeid = f.classeid
                        and f.sentencaid = NPIij.sentencaid)
```

```
                        where sentencaid in (select sentencaid
                        from PIij where exemplo = 0)
                  END
            else
                  BEGIN
                        insert into NPIij
                        select distinct sentencaid, P.classeid,
                        (p.PIj * f.Bij), 0
                        from PIj p, @Bij f
                        where p.classeid = f.classeid
                        order by sentencaid, P.classeid

                        update NPIij set S =
                        (select sum(p.PIj * f.Bij)
                        from PIj p, @Bij f
                        where p.classeid = f.classeid
                        and f.sentencaid = NPIij.sentencaid)
                        where sentencaid in (select sentencaid
                        from PIij where exemplo = 0)
                  END

            update NPIij set PIij = PIij / S
            where sentencaid in (select sentencaid from PIij where
            exemplo = 0)

--E-Step Fim

            set @iteracoes = @iteracoes + 1

      END


--Cálculo de estatisticas e métricas

            insert estatisticas
            select @TotalSentencasExemplos,
            @TotaldeSentencasAmostras,
            convert(float,count(*)) / (
            @TotaldeSentencasAmostras),
            0, 0, getdate(), @testsetExemplos
            from NPIij n, PIij p
            Where n.sentencaid = p.sentencaid
            and n.PIij = (select max(PIij)
                              from NPIij n2
                                    where round(PIij,11) <>
                                    round(1.0/(select
                                    count(*) from
                                    classe),11)
                              and n2.sentencaid =
                              n.sentencaid)
            and round(n.PIij,11) <> round(1.0/(select count(*)
            from classe),11)
            and n.classeid = p.classeid
            and p.PIij = 1
            and p.exemplo = 0

update estatisticas set abrangencia = (select
convert(float,count(*)) from NPIij n, PIij p
            Where n.sentencaid = p.sentencaid
            and n.PIij = (select max(PIij) from NPIij n2
```

```
            where round(PIij,11) <> round(1.0/(select count(*)
            from classe),11)
            and n2.sentencaid = n.sentencaid)
            and round(n.PIij,11) <> round(1.0/(select count(*)
            from classe),11)
            and n.classeid = p.classeid
            and p.PIij = 1
            and p.classeid = 1
            and p.exemplo = 0) /
                  (select convert(float,count(*))
                        from PIij pp
                        Where pp.classeid = 1
                        and pp.PIij = 1
                        and pp.exemplo = 0)
            where abrangencia = 0

update estatisticas set F1 = (2 * Accuracy * Abrangencia) /
(Accuracy + Abrangencia)

            declare @estatisticaid int
            select @estatisticaid = @@identity


            insert metricas
            select p.classeID,
            convert(float,count(*)) as TP,
                                          (select
                              convert(float,count(*)) from NPIij
                              pp, PIij ppp
                              where pp.sentencaid = ppp.sentencaid
                              and pp.classeid = p.classeID
                              and round(pp.PIij,11) <>
                              round(1.0/(select count(*) from
                              classe),11)
                              and pp.PIij = (select max(PIij) from
                              NPIij n3
                              where round(PIij,11) <>
                              round(1.0/(select count(*) from
                              classe),11)
                              and n3.sentencaid = pp.sentencaid)
                              and ppp.PIij = 1
                              and ppp.exemplo = 0
                              and ppp.classeid <> p.classeid),
            @TotalSentencasExemplos, 0,
            0, getdate(), 0, (select convert(float,count(*)) from
            PIij pp where pp.classeid = p.classeID and pp.PIij = 1
            and pp.exemplo = 0),
            @estatisticaid
            from NPIij n, PIij p
            Where n.sentencaid = p.sentencaid
            and n.PIij = (select max(PIij) from NPIij n2 where
            round(PIij,11) <> round(1.0/(select count(*) from
            classe),11) and n2.sentencaid = n.sentencaid)
            and round(n.PIij,11) <> round(1.0/(select count(*)
            from classe),11)
            and p.PIij = 1
            and p.exemplo = 0
            and n.classeid = p.classeid
            group by p.classeID

--Cálculo de Precision, Recall e F1 por Classe
```

```
            update metricas set Recall = TP / Amostras, Precision
            = TP / (TP + FP)

            update metricas set F1 = (2 * Recall * Precision) /
            (Recall + Precision)


select exemplos, amostras, round(accuracy, 3) as accuracy,
round(abrangencia, 3) as abrangencia, round(f1, 3) as f1,
convert(char(20), data, 13) data, estatisticaid from estatisticas
order by data desc



/**Object:StoredProcedure [dbo].[spi_GeraModeloParaAlgoritmoEM]**/
/****Geração dos dados do Modelo para o Algoritmo EM ************/

CREATE PROCEDURE [dbo].[spi_GeraModeloParaAlgoritmoEM]
@nGrama int,
@stemming bit,
@arqID int

AS

if @arqID = 0 --Situação onde o modelo está sendo criado
Begin
      delete nip
      delete palavra
end

DECLARE @Stm varchar(255)
DECLARE @Sentenca varchar(255), @Sentencaant varchar(255)
DECLARE @i int, @palavra varchar(255)
DECLARE @tableNGramaSentenca table (sentencaid int, palavra
varchar(255))

if @stemming = 1
BEGIN

--Atualização do stem das palavras do corpus utilizando Porter

update c set stm = dbo.fnPorterAlgorithm (REPLACE( REPLACE(
REPLACE( REPLACE( REPLACE( REPLACE( REPLACE( REPLACE(
REPLACE(REPLACE( convert(varchar(255), palavra) , ',' , '' ), ':',
''), '.', ''), '*', ''), '+', ''), '"', ''), '(', ''), ')', ''),
'[', ''), ']', ''))
      from corpus c, sentenca s
      where c.palavra is not null
      and  c.sentenca = s.descricao
      and c.palavra not like '#%#'
      and (s.arqID = @arqID or @arqID = 0)

      if @ngrama = 1
      BEGIN

--Criação da lista de palavras

            insert into palavra (palavra)
            select distinct c.stm
            from corpus c, sentenca s
```

```
            where charindex ('#', c.stm) = 0
            and  c.sentenca = s.descricao
            and (s.arqID = @arqID or @arqID = 0)
            and not exists (select * from palavra where palavra =
            c.stm)

            delete n
            from nip n, sentenca s
            where n.sentencaid = s.sentencaid
            and s.arqID = @arqID

--Criação do saco de palavras por sentença

            insert into nip
            select s.sentencaid, p.palavraid,  count(*)
            from corpus c, palavra p, sentenca s
            where c.sentenca = s.descricao
            and c.stm = p.palavra
            and (s.arqID = @arqID or @arqID = 0)
            group by s.sentencaid, p.palavraid
      END
      else
      BEGIN

            DECLARE corpus CURSOR FOR
            select Stm, Sentencaid from corpus c, sentenca s
            where c.sentenca = s.descricao
            and (s.arqID = @arqID or @arqID = 0)
            order by arquivoconteudoid, linha

            OPEN corpus;

            FETCH NEXT FROM corpus
            INTO @Stm, @Sentenca
            set @palavra = @stm
            set @sentencaant = @Sentenca
            set @i = 1
            WHILE @@FETCH_STATUS = 0
            BEGIN
                  if @sentenca = @sentencaant and @i < @nGrama
                  BEGIN
                        set @palavra = @palavra + ' ' + @stm
                        set @i = @i + 1
                  END
                  else
                  BEGIN
                  insert into @tableNGramaSentenca values
                  (@sentenca, @palavra)
                        set @palavra = @stm
                        set @i = 1
                  END
                  FETCH NEXT FROM corpus
                  INTO @Stm, @Sentenca
                  set @sentencaant = @Sentenca
            END

            CLOSE corpus;
            DEALLOCATE corpus;

--Criação da lista de palavras
```

```
            insert into palavra (palavra)
            select distinct palavra from @tableNGramaSentenca t
            Where not exists (select * from palavra p where
            p.palavra = t.palavra)

            delete n
            from nip n, sentenca s
            where n.sentencaid = s.sentencaid
            and s.arqID = @arqID

--Criação do saco de palavras por sentença

            insert into nip
            select c.sentencaid, p.palavraid,  count(*)
            from @tableNGramaSentenca c, palavra p
            where c.palavra = p.palavra
            group by c.sentencaid, p.palavraid

      END
END
else

-- Não utilizando o Porter Stemming
BEGIN
      if @ngrama = 1
      BEGIN

--Criação da lista de palavras

            insert into palavra (palavra)
            select distinct REPLACE( REPLACE( REPLACE( REPLACE(
            REPLACE( REPLACE( REPLACE( REPLACE(REPLACE(REPLACE(
            convert(varchar(255),palavra) , ',' , '' ), ':', ''),
            '.', ''), '*', ''), '+', ''), '"', ''), '(', ''), ')',
            '') , '[', ''), ']', '')
            from corpus c, sentenca s
            where charindex ('#', palavra) = 0
            and  c.sentenca = s.descricao
            and (s.arqID = @arqID or @arqID = 0)
            and not exists (select * from palavra where palavra =
            REPLACE( REPLACE( REPLACE( REPLACE( REPLACE( REPLACE(
            REPLACE( REPLACE(REPLACE(REPLACE(
            convert(varchar(255),palavra) , ',' , '' ), ':', ''),
            '.', ''), '*', ''), '+', ''), '"', ''), '(', ''), ')',
            '') , '[', ''), ']', ''))

            delete n
            from nip n, sentenca s
            where n.sentencaid = s.sentencaid
            and s.arqID = @arqID

--Criação do saco de palavras por sentença

            insert into nip
            select s.sentencaid, p.palavraid,  count(*)
            from corpus c, palavra p, sentenca s
            where c.sentenca = s.descricao
            and (s.arqID = @arqID or @arqID = 0)
            and REPLACE( REPLACE( REPLACE( REPLACE( REPLACE(
            REPLACE( REPLACE( REPLACE( REPLACE( REPLACE(
            convert(varchar(255),c.palavra) , ',' , '' ), ':',
```

```
            ''), '.', ''), '*', ''), '+', ''), '"', ''), '(', ''),
            ')', ''), '[', ''), ']', '') = p.palavra
            group by s.sentencaid, p.palavraid
      END
      Else

--Utilizando n-grama

      BEGIN

            DECLARE corpus CURSOR FOR
            select REPLACE(REPLACE (REPLACE(REPLACE
            (REPLACE(REPLACE (REPLACE( REPLACE (REPLACE (REPLACE (
            convert(varchar(255),palavra) , ',' , '' ), ':', ''),
            '.', ''), '*', ''), '+', ''), '"', ''), '(', ''), ')',
            ''), '[', ''), ']', ''),
             Sentencaid from corpus c, sentenca s
            where c.sentenca = s.descricao
            and (s.arqID = @arqID or @arqID = 0)
            order by arquivoconteudoid, linha

            OPEN corpus;

            FETCH NEXT FROM corpus
            INTO @Stm, @Sentenca
            set @palavra = @stm
            set @sentencaant = @Sentenca
            set @i = 1
            WHILE @@FETCH_STATUS = 0
            BEGIN
                  if @sentenca = @sentencaant and @i < @nGrama
                  BEGIN
                        set @palavra = @palavra + ' ' + @stm
                        set @i = @i + 1
                  END
                  else
                  BEGIN
                  insert into @tableNGramaSentenca values
            (@sentenca, @palavra)
                        set @palavra = @stm
                        set @i = 1
                  END
                  FETCH NEXT FROM corpus
                  INTO @Stm, @Sentenca
                  set @sentencaant = @Sentenca
            END

            CLOSE corpus;
            DEALLOCATE corpus;

--Criação da lista de palavras

            insert into palavra (palavra)
            select distinct palavra from @tableNGramaSentenca t
            Where not exists (select * from palavra p where
            p.palavra = t.palavra)

            delete n
            from nip n, sentenca s
            where n.sentencaid = s.sentencaid
            and s.arqID = @arqID
```

```
--Criação do saco de palavras por sentença

            insert into nip
            select c.sentencaid, p.palavraid,  count(*)
            from @tableNGramaSentenca c, palavra p
            where c.palavra = p.palavra
            group by c.sentencaid, p.palavraid

      END
END

delete FIjprincipal

insert into FIjprincipal
select classeid, palavraid from
classe, palavra

select count(*) palavras from palavra
GO


/*** Object:  StoredProcedure [dbo].[spi_CorpusConteudoArquivo]**/
/*** Carga dos arquivos para o corpus separando as palavras******/

CREATE PROCEDURE [dbo].[spi_CorpusConteudoArquivo]
@arqID int
AS

create table #temp (ArquivoConteudoID int, textoinicial
varchar(255), textofinal varchar(255))
insert #temp
select ArqConteudoID, '', convert(varchar(255), texto)
from ArquivosTxtConteudo
where convert(varchar(255), texto) not in ('NULL', '')
and arqID = @arqID

update #temp set textoinicial = substring(textofinal, 0,
charindex(' ', textofinal, 0)),
textofinal = substring(textofinal, charindex(' ', textofinal, 0) +
1, 255)
from #temp

declare @sentencaid int

set @sentencaid = 1

While exists (select * from #temp)
Begin

      insert corpus(ArquivoConteudoID, Palavra, Sentenca, ArqID)
      select ArquivoConteudoID, textoinicial, '#' +
convert(varchar(100), @sentencaid) + '#' + convert(varchar(100),
@arqID), @arqID
      from #temp (nolock)
      where textoinicial not in ('NULL', '')

      delete #temp where textofinal = '' and textoinicial = ''

      delete #temp where textofinal = textoinicial
```

```
      update #temp set textoinicial = substring(textofinal, 0,
charindex(' ', textofinal, 0)),
      textofinal = substring(textofinal, charindex(' ',
textofinal, 0) + 1, 255)
      from #temp

      update #temp set textoinicial = textofinal where charindex('
', textofinal, 0) = 0 and textoinicial = ''

End

drop table #temp
GO


/****** Object:  StoredProcedure [dbo].[spi_ConteudoArquivo] ****/
/****** Importação do arquivo para o banco de dados *************/

CREATE PROCEDURE [dbo].[spi_ConteudoArquivo]
@arqID int,
@nomeArq varchar(255)
AS

DECLARE @Comando varchar(255)

create table #temp_text (texto ntext)

      SET @Comando = 'type "C:\importacao\' + @nomeArq + '"'

      insert #temp_text
      EXEC master.dbo.xp_cmdshell @Comando

      insert into ArquivosTxtConteudo (ArqID, Texto)
      select @arqID, texto
      from #temp_text

      delete #temp_text
GO


/**** Object:StoredProcedure [spi_CorpusConteudoArquivoSentenca]*/
/**** Fragmentação do arquivo em sentenças **********************/

CREATE PROCEDURE [dbo].[spi_CorpusConteudoArquivoSentenca]
@arqID int,
@tipo smallint
AS

DECLARE Corpus_Cursor CURSOR FOR
select at.arqID, linha, palavra, c.arquivoconteudoid, pos
from corpus c, arquivostxtconteudo atc, arquivostxt at
where c.arquivoconteudoid = atc.arqconteudoid
and atc.arqid = at.arqid
and at.arqID = @arqID
order by arquivoConteudoID, linha

DECLARE @linha int, @palavra varchar(255), @arquivoconteudoid int,
@pos varchar(10)

DECLARE @sentenca varchar(255), @sentencaid int
```

```
set @sentencaid = 1

OPEN Corpus_Cursor;

FETCH NEXT FROM Corpus_Cursor
INTO @arqID, @linha, @palavra, @arquivoconteudoid, @pos

SET @sentenca = '#' + convert(varchar(100), @sentencaid) + '#' +
convert(varchar(100), @arqID)

UPDATE corpus set sentenca = @sentenca
WHERE linha = @linha

FETCH NEXT FROM Corpus_Cursor
INTO @arqID, @linha, @palavra, @arquivoconteudoid, @pos

WHILE @@FETCH_STATUS = 0
BEGIN
      UPDATE corpus set sentenca = @sentenca
      WHERE linha = @linha

if (@tipo = 2)
      BEGIN

--Separação na troca de paragrafo, ou seja, quando o textoinicial
= '%.' e não tem mais texto, soma 1 no sentencaid

      if (@palavra like '%.' OR @palavra like '%?' OR @palavra
      like '%!') and not exists (select * from corpus where
      arquivoconteudoid = @arquivoconteudoid and linha > @linha)
            Begin
                  set @sentencaid = @sentencaid + 1
                  set @sentenca = '#' + convert(varchar(100),
                  @sentencaid) + '#' + convert(varchar(100),
                  @arqID)
            End
      END
else
      BEGIN

--Separação no ponto final separando por oração

      if (@palavra like '%.' OR @palavra like '%?' OR @palavra
      like '%!')
            Begin
                  set @sentencaid = @sentencaid + 1
                  set @sentenca = '#' + convert(varchar(100),
                  @sentencaid) + '#' + convert(varchar(100),
                  @arqID)
            End
      END


    FETCH NEXT FROM Corpus_Cursor
      INTO @arqID, @linha, @palavra, @arquivoconteudoid, @pos
END

CLOSE Corpus_Cursor;
DEALLOCATE Corpus_Cursor;

--Recrição das Sentencas
```

```
delete sentenca where arqID = @arqID

insert sentenca (descricao, arqID, classeID)
select distinct sentenca, arqID, (select classeID from ArquivosTxt
where arqID = @arqID)
from corpus c
where not exists (select * from sentenca s where s.descricao =
c.sentenca)
and arqID = @arqID
order by sentenca

declare @testset1 table (sentencaid int)
declare @total int

set @total = (select count(*) from sentenca where arqID = @arqID)

--Separação das sentenças em dois testsets

insert @testset1
select Top (@total/2) sentencaid
from sentenca where arqID = @arqID
order by sentencaid

update sentenca set testset = 1
from sentenca s, @testset1 t1
where s.sentencaid = t1.sentencaid

update sentenca set testset = 2
from sentenca s
where testset is null and arqID = @arqID
GO
```