4
Experimentos

4.1
Introdução

Foram realizados experimentos com os dois algoritmos propostos no capítulo anterior em duas bases de documentos, Reuters-21578 e Ohsumed. Primeiramente serão descritas cada uma das bases de documentos. Então, serão apresentados os resultados obtidos nas duas bases.

4.2
Bases de dados

Nessa seção serão descritas as duas bases de documentos onde foram aplicados experimentos com a finalidade de comparar os dois algoritmos propostos no capítulo anterior. As duas bases de documentos são amplamente mencionadas na literatura de máquina de aprendizado.

4.2.1
Reuters-21578

A base de testes Reuters-21578 vem sendo amplamente utilizada, nos últimos dez anos, para testar algoritmos de classificação automática de textos. A base é composta de um conjunto de 21.578 notícias que foram publicadas na rede de notícias Reuters, em 1987, e classificadas de acordo com 135 categorias, a maioria sobre economia e negócios. A base foi originalmente categorizada pelo Carnegie Group, Inc. and Reuters, Ltd. na criação do sistema de classificação automática CONSTRUE [Hayes & Weinstein, 1990] e posteriormente foi coletada e formatada por David Lewis.

Esta coleção possui muitas características importantes que a tornam uma base de dados interessante para os experimentos da classificação automática de textos:

- A base de dados é multirótulo, ou seja, um documento pode estar associado a mais de uma categoria, sendo uma situação muito mais realista.
- A distribuição de documentos pelas 135 categorias não é uniforme.
- Existem muitas relações semânticas implícitas entre as categorias da base (por exemplo, as categorias “wheat” e “grain” estão claramente associadas, pois sempre que um documento pertence à “wheat” também pertence à “grain”).

A base de testes Reuters-21578 certamente é uma base desafiadora para sistemas de classificação automática de textos baseados nas técnicas de aprendizagem de máquina, uma vez que muitas categorias possuem muito poucos exemplos de treino, tornando a construção indutiva de um classificador uma tarefa difícil.

Os documentos realmente utilizados na maioria dos experimentos de classificação automática de textos são apenas 12.902, uma vez que 8.676 documentos não foram considerados na categorização da base.

Com a finalidade de tornar os resultados experimentais comparáveis, partições compostas do conjunto de treinamento e do conjunto de teste foram definidas pelos criadores da base de 12.902 documentos. Salvo algumas exceções, os pesquisadores têm usado a partição ModApté, no qual 9.603 documentos foram selecionados para o conjunto de documentos de treinamento, enquanto 3.299 foram selecionados para compor o conjunto de documentos de teste.

Dentre os 12.902 documentos, alguns documentos não possuem categoria associada, porém, diferentemente dos 8.676 que não foram considerados, tais documentos não estão associados a nenhuma categoria, porque na tarefa de categorização da base não foi encontrada nenhuma categoria que pudesse associá-los.

Dentre as 135 categorias, 20 categorias na partição ModApté não possuem nenhum documento de treino. Portanto, tais categorias nunca foram consideradas em nenhum experimento de classificação automática de textos.

Uma vez que as 115 categorias da partição ModApté possuem pelo menos um documento de treinamento, em princípio, podem ser usadas em qualquer

experimento. Porém, muitos pesquisadores têm preferido executar seus experimentos em diferentes subconjuntos das 115 categorias.

Dentre os subconjuntos, os mais populares são:

- O conjunto das 10 categorias que possuem a maior quantidade de documentos de treinamento, chamado de R(10). Esse conjunto foi utilizado em [Bennett et. al, 2002; McCallum & Nigam, 1998; Nigam et al., 2000; Tong & Koller, 2001].
- O conjunto das 90 categorias que possuem pelo menos um documento de treinamento e um documento de teste, chamado de R(90). Esse conjunto, que está presente na maioria dos experimentos já relatados, foi utilizado em [Chai et al., 2002; Joachims, 1998; Lam & Lai, 2001; Moschitti, 2003a; Sebastiani et al., 2000; Yang & Liu, 1999].
- O conjunto das 115 categorias que possuem pelo menos um documento de treinamento, chamado de R(115). Esse conjunto foi utilizado em [Caropreso et al., 2001; Dumais et al., 1998; Galavotti et al., 2000].

Vale notar que $R(10) \subset R(90) \subset R(115)$.

4.2.2 Ohsumed

A coleção de teste Ohsumed é um subconjunto de 348.566 documentos da base MEDLINE (uma base on-line de textos sobre medicina), compilada por William Hersh e proveniente de 270 jornais médicos por um período de cinco anos (1987- 1991).

Todos os documentos possuem títulos, porém apenas 233.445 possuem abstracts. Os documentos foram manualmente categorizados em 14.321 categorias definidas no thesaurus Mesh (Medical Subject Heading).

A base Ohsumed é considerada uma base mais difícil uma vez que não existe uma relação bem definida entre as palavras e categorias. No caso da Reuters, o próprio nome da categoria é uma palavra muito freqüente no conjunto de documentos definido pela categoria.

A base de documentos médicos Ohsumed foi utilizada nos trabalhos [Joachims, 1998; Moschitti, 2003a].

4.3 Experimentos Realizados

Os algoritmos propostos na seção anterior foram desenvolvidos na linguagem C# e rodados em uma máquina AMD Semprom 2.2 GHz com 512 MB de memória RAM com Windows XP e Microsoft .NET Framework 2.0. Cada documento foi representado como uma tabela de palavras, definida por um conjunto de tuplas, relacionando cada palavra do documento com sua respectiva freqüência. Além disso, todas as palavras foram transformadas em minúsculas e foram retiradas as "stopwords" definidas no conjunto de 571 "stopwords" da língua inglesa SMARTLIST desenvolvida inicialmente para o sistema SMART [Salton, 1971].

Todos os documentos considerados nos experimentos possuem pelo menos uma categoria associada.

4.3.1 Experimentos com a base Reuters R(10)

No conjunto R(10), primeiramente foi realizado um experimento utilizando a partição ModApté, com 6.490 documentos de treinamento, 2.545 documentos de teste, vocabulário de 21955 palavras e 39 combinações das 10 categorias da base.

As tabelas 3, 4 e 5 apresentam os resultados do algoritmo pseudo-multirótulo. Já as tabelas 6, 7 e 8 apresentam os resultados do algoritmo mulirótulo.

Categoria	Recall	Precision	F1
acq	97,36	97,63	97,49
corn	71,43	72,73	72,07
crude	94,71	90,40	92,51
earn	98,16	98,61	98,39
grain	81,88	99,19	89,71
interest	58,78	87,50	70,32
money-fx	94,41	78,24	85,57
ship	71,91	91,43	80,50
trade	91,45	78,10	84,25
wheat	76,06	77,14	76,60

Tabela 3 – Resultados do algoritmo pseudo-multirótulo na base R(10) para as 10 categorias que compõem a base.

Micro Recall	92,54
Micro Precision	93,58
Micro F1	93,05
Macro Recall	83,61
Macro Precision	87,10
Macro F1	84,74

Tabela 4 – Resultados globais do algoritmo pseudo-multirótulo na base R(10).

Classe	Segundos
Treinamento	3,812
Classificação	7,203

Tabela 5 – Tempo de execução da fase de treinamento e da fase de classificação do algoritmo pseudo-multirótulo na base R(10).

Categoria	Recall	Precision	F1
acq	96,80	97,62	97,21
corn	87,50	62,03	72,59
crude	96,30	88,78	92,39
earn	97,98	98,61	98,29
grain	89,26	97,08	93,01
interest	74,05	86,61	79,84
money-fx	94,41	75,78	84,08
ship	79,78	85,54	82,56
trade	90,60	84,80	87,60
wheat	83,10	62,11	71,08

Tabela 6 – Resultados do algoritmo multirótulo na base R(10) para as 10 categorias que compõem a base.

Micro Recall	94,26
Micro Precision	92,11
Micro F1	93,17
Macro Recall	88,98
Macro Precision	83,90
Macro F1	85,86

Tabela 7 – Resultados globais do algoritmo multirótulo na base R(10).

Classe	Segundos
Treinamento	4,125
Classificação	3,218

Tabela 8 – Tempo de execução da fase de treinamento e da fase de classificação do algoritmo multirótulo na base R(10).

Após o experimento utilizando a partição ModeApté, foi realizado um experimento para verificar como se comportam os dois algoritmos em função da quantidade de documentos de treinamento.

Para isso, para cada um dos algoritmos, foram gerados 10 partições aleatórias, onde 6.490 documentos eram de treinamento e 2.545 documentos eram de teste.

Para cada partição rodou-se o algoritmo 11 vezes, mantendo-se os documentos de teste fixos (2.545 documentos) e variando a quantidade de documentos de treinamento pertencentes ao conjunto de 6.490 documentos, iniciando com 6.490 e dividindo por 2 até a quantidade de documentos de treinamento chegar a 6.

Uma vez gerada a curva para cada uma das 10 partições calculou-se a média das 10 curvas.

As figuras abaixo apresentam os resultados do experimento para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo.

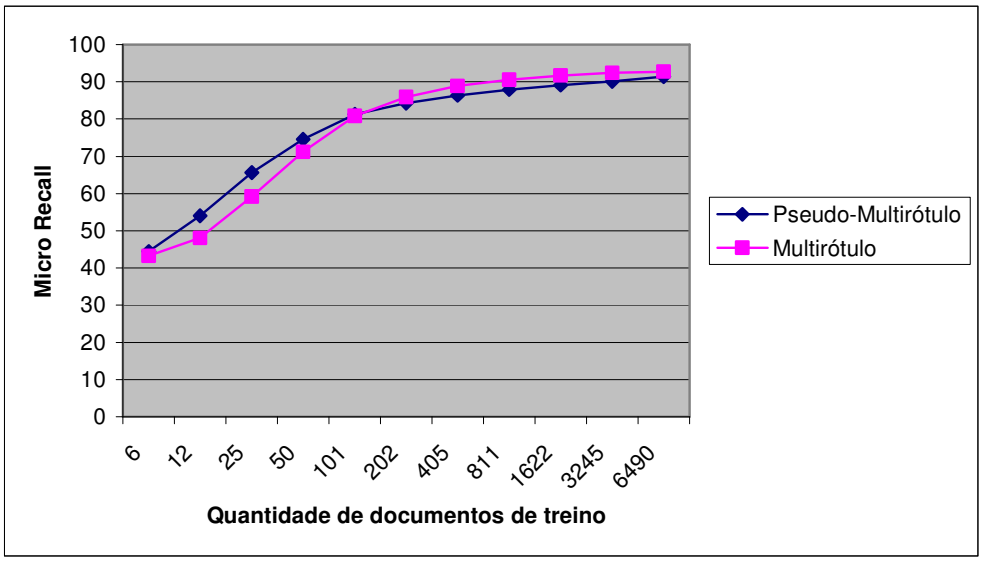

Figura 3 - Resultados Micro Recal para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(10).

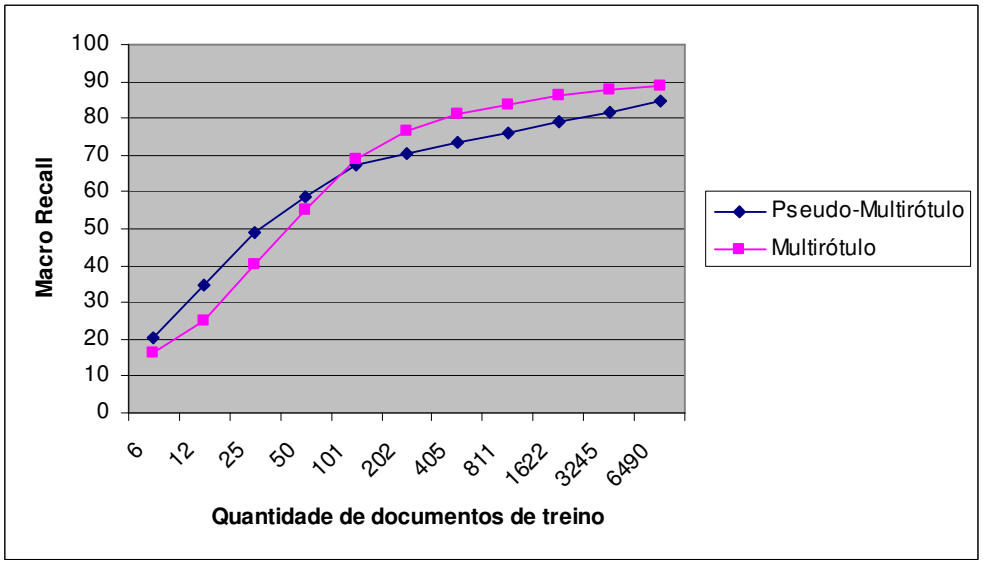

Figura 4 - Resultados Macro Recall para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(10).

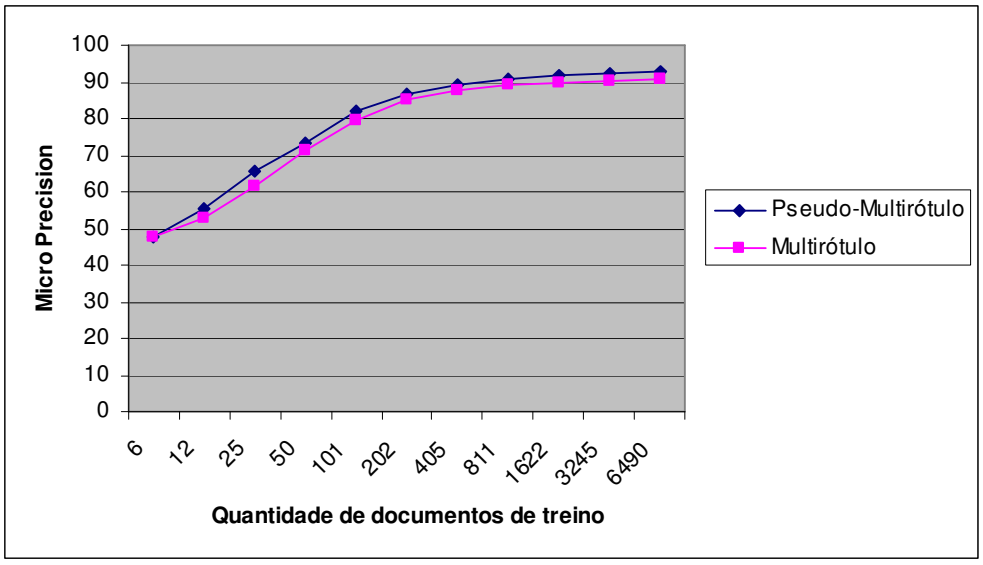

Figura 5 - Resultados Micro Precision para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(10).

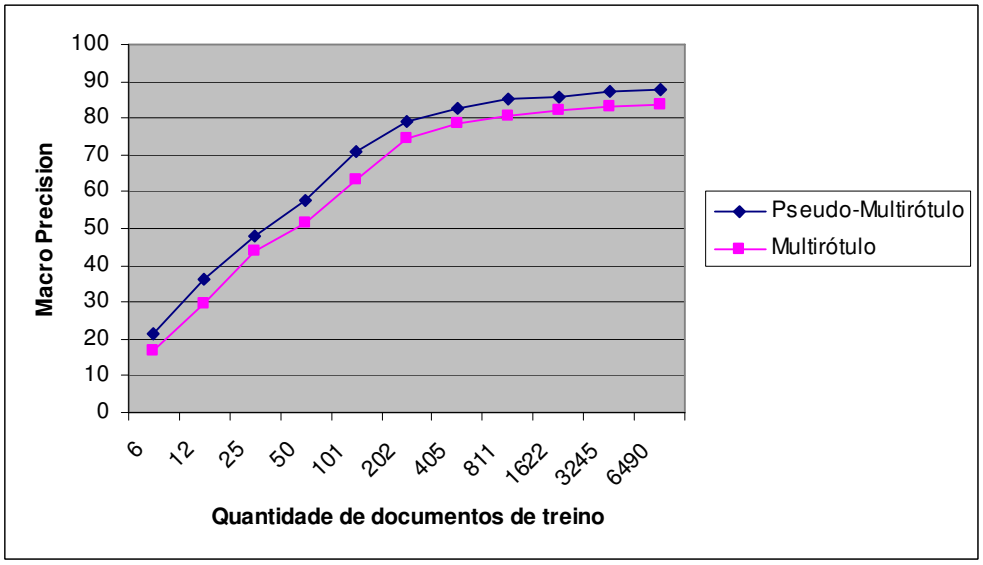

Figura 6 - Resultados Macro Precision para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(10).

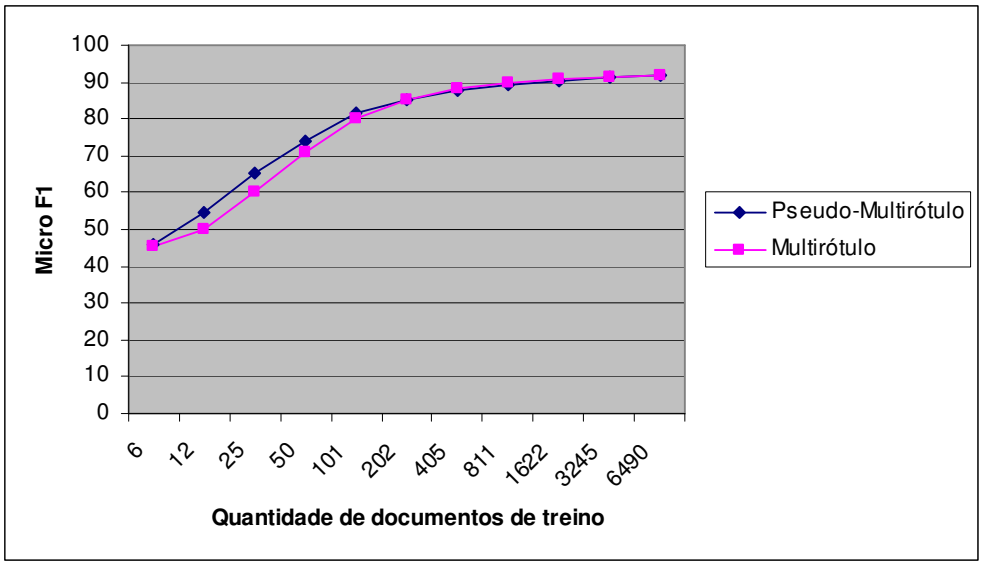

Figura 7 - Resultados Micro F1 para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(10).

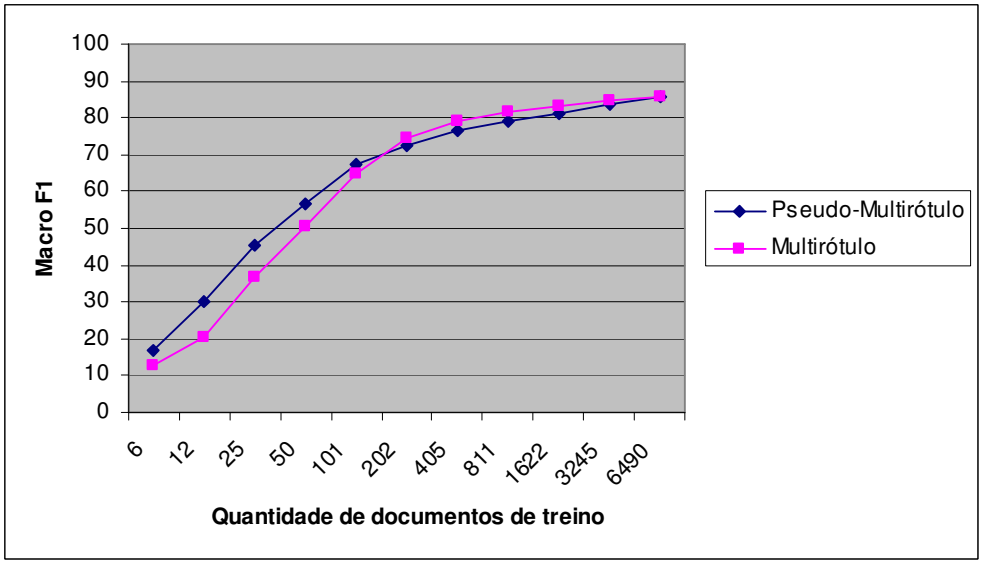

Figura 8 - Resultados Macro F1 para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(10).

4.3.2
Experimentos com a base Reuters R(90)

Conforme concluído nos experimentos de Sebastiani & Debole [2004], a construção indutiva de classificadores na base R(10) é menos árdua que na base

R(90). Por isso, foram realizados experimentos para os dois algoritmos propostos na base R(90).

Assim como no conjunto R(10), primeiramente foi realizado um experimento utilizando a partição ModApté, com 7.770 documentos de treinamento, 3.019 documentos de teste, vocabulário de 24.244 palavras e 365 combinações das 90 categorias da base.

As tabelas 7, 8 e 9 apresentam os resultados do algoritmo pseudo-multirótulo. Já as tabelas 10, 11 e 12 apresentam os resultados do algoritmo mulirótulo.

Categoria	Recall	Precision	F1
acq	97,64	92,73	95,12
corn	35,71	74,07	48,19
crude	93,12	67,69	78,40
earn	98,25	95,36	96,78
grain	69,80	84,55	76,47
interest	64,12	85,71	73,36
money-fx	91,62	68,91	78,66
ship	67,42	88,24	76,43
trade	88,03	47,03	61,31
wheat	64,12	85,71	73,36

Tabela 9 – Resultados do algoritmo pseudo-multirótulo na base R(90) para as 10 categorias com maior quantidade de documentos de treinamento.

Micro Recall	72,58
Micro Precision	83,14
Micro F1	77,50
Macro Recall	19,13
Macro Precision	38,93
Macro F1	21,92

Tabela 10 – Resultados globais do algoritmo pseudo-multirótulo na base R(90).

Classe	Segundos
Treinamento	5,750
Classificação	56,344

Tabela 11 – Tempo de execução da fase de treinamento e da fase de classificação do algoritmo pseudo-multirótulo na base R(90).

Categoria	Recall	Precision	F1
acq	96,80	97,62	97,21
corn	87,50	62,03	72,59
crude	96,30	88,78	92,39
earn	97,98	98,61	98,29
grain	89,26	97,08	93,01
interest	74,05	86,61	79,84
money-fx	94,41	75,78	84,08
ship	79,78	85,54	82,56
trade	90,60	84,80	87,60
wheat	83,10	62,11	71,08

Tabela 12 – Resultados do algoritmo multirótulo na base R(90) para as 10 categorias com maior quantidade de documentos de treinamento.

Micro Recall	79,68
Micro Precision	77,70
Micro F1	78,68
Macro Recall	29,07
Macro Precision	41,99
Macro F1	30,73

Tabela 13 – Resultados globais do algoritmo multirótulo na base R(90).

Classe	Segundos
Treinamento	18,594
Classificação	20,344

Tabela 14 – Tempo de execução da fase de treinamento e da fase de classificação do algoritmo multirótulo na base R(90).

Após o experimento utilizando a partição ModeApté, foi realizado um experimento para verificar como se comportam os dois algoritmos em função da quantidade de documentos de treinamento.

Para isso, para cada um dos algoritmos, foram geradas 10 partições aleatórias, onde 7.770 documentos eram de treinamento e 3.019 documentos eram de teste.

Para cada partição rodou-se o algoritmo 11 vezes, mantendo-se os documentos de teste fixos (3.019 documentos) e variando a quantidade de documentos de treinamento pertencentes ao conjunto de 7.770 documentos, iniciando com 7.770 e dividindo por 2 até a quantidade de documentos de treinamento chegar a 7.

Uma vez gerada a curva para cada um das 10 partições calculou-se a média das 10 curvas.

As figuras abaixo apresentam os resultados do experimento para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo.

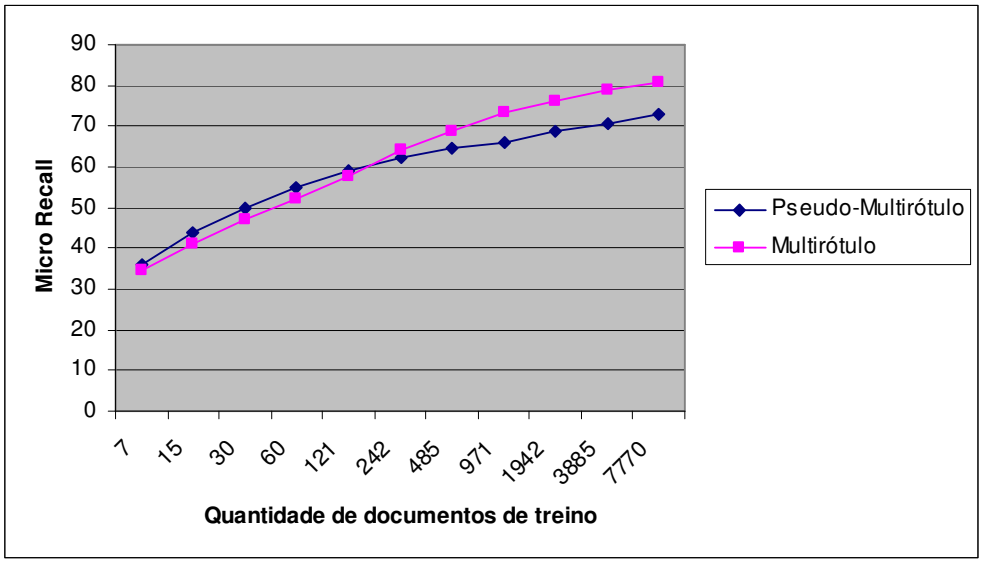

Figura 9 – Resultados Micro Recall para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(90).

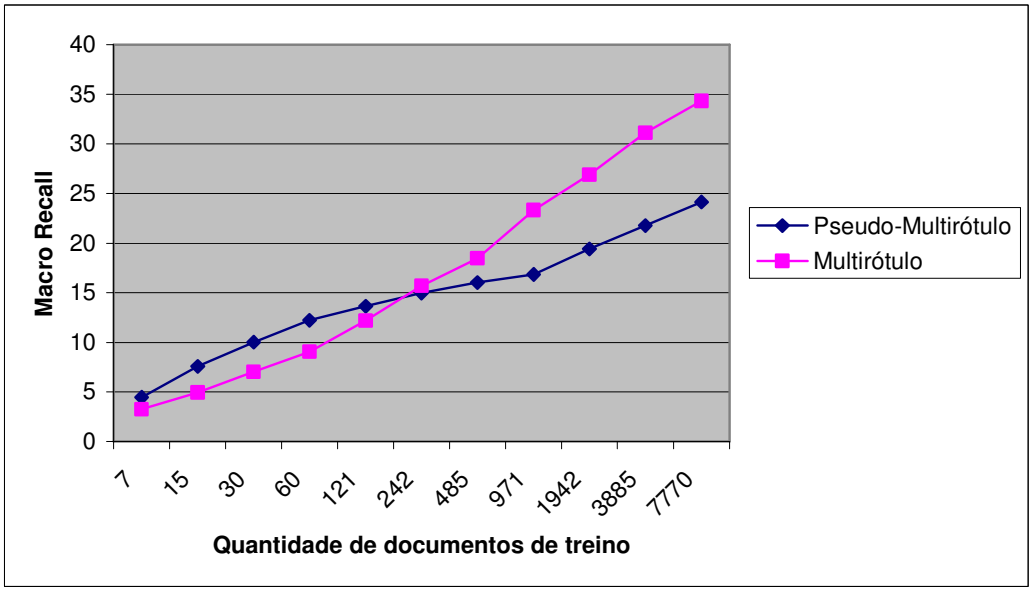

Figura 10 – Resultados Macro Recall para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(90).

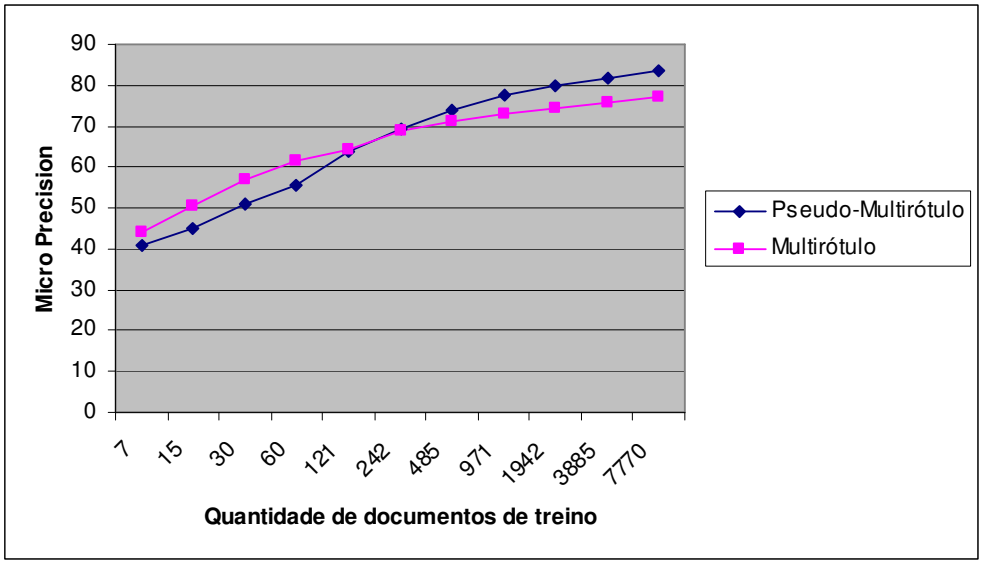

Figura 11 – Resultados Micro Precision para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(90).

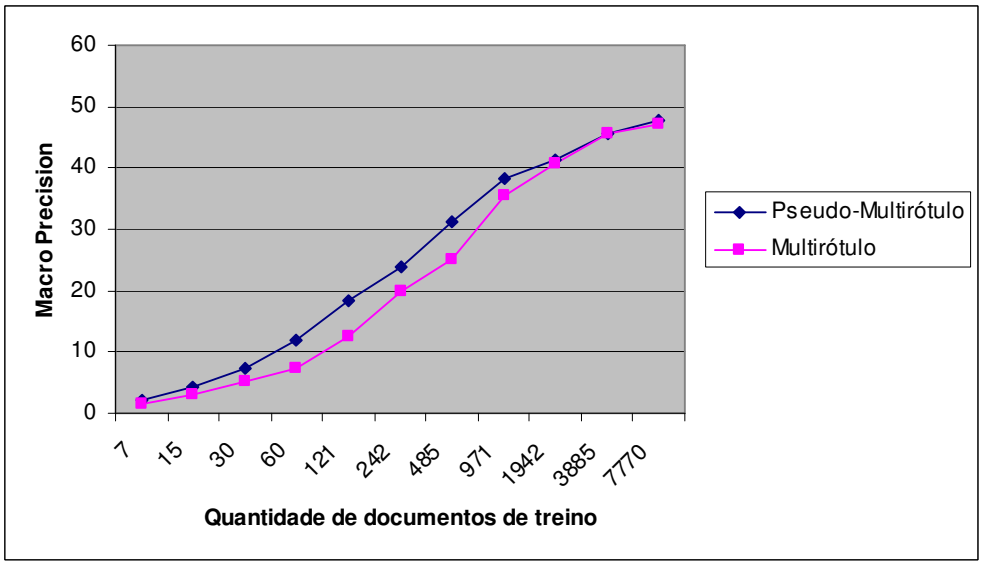

Figura 12 – Resultados Macro Precision para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(90).

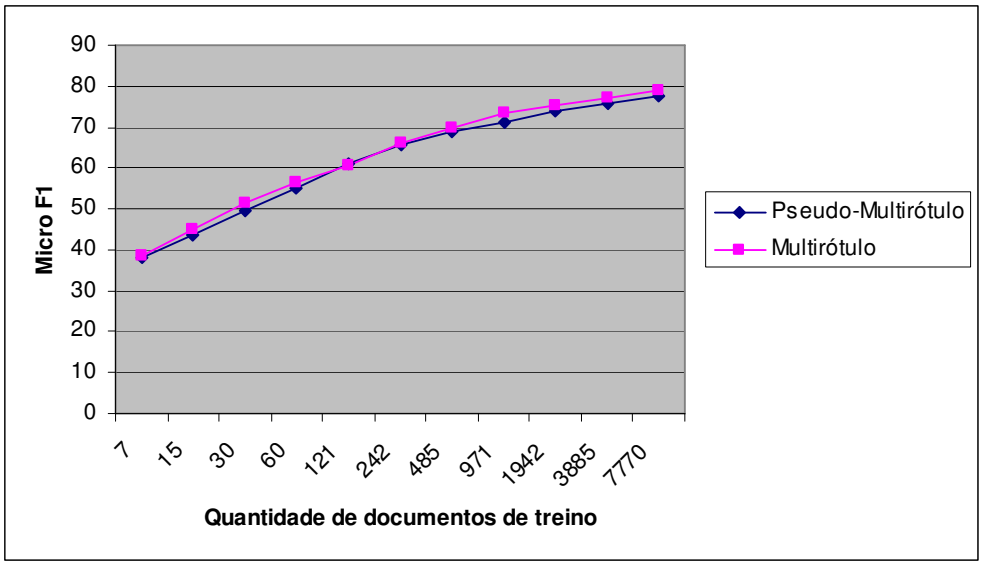

Figura 13 – Resultados Micro F1 para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(90).

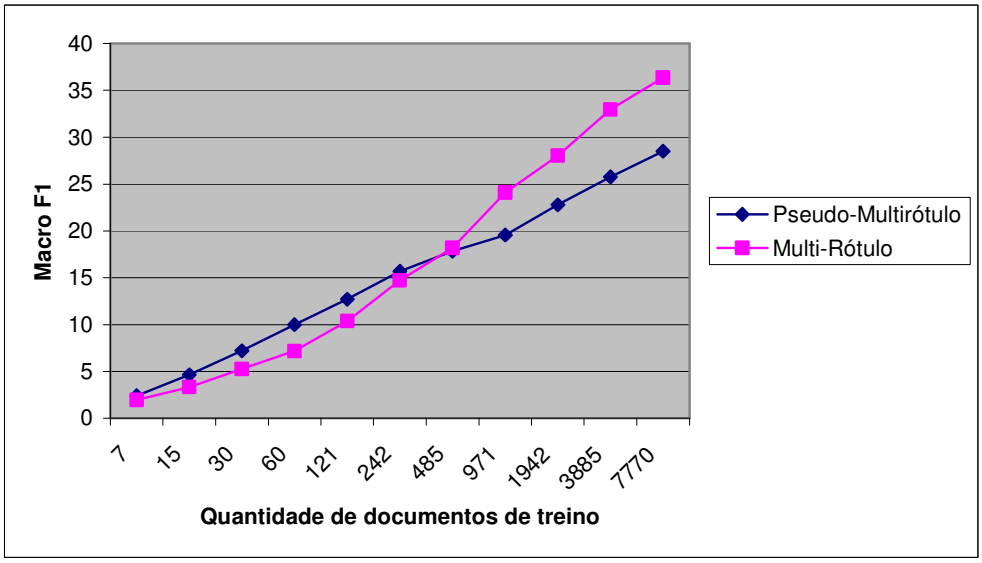

Figura 14 – Resultados Macro F1 para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base R(90).

4.3.3
Conclusões dos experimentos com as bases da Reuters

Com base nos experimentos realizados na base R(10) e base R(90) usando a partição fixa ModApté, pode-se observar que o algoritmo multirótulo mostrou-se mais eficiente, uma vez que possui valores maiores para Micro F1 e Macro F1.

Vale ressaltar também que o algoritmo multirótulo apresenta valores maiores para Micro Recall e Macro Recall, porém apresenta valores menores para Micro Precision e Macro Precision, o que sugere que o algoritmo multirótulo possui maior recall em detrimento de uma menor precision.

Com relação aos experimentos com partição aleatória, observa-se que conforme se aumenta a quantidade de documentos de treinamento, as medidas Micro Recall e Macro Recall dos algoritmos crescem até chegar a um ponto em que as medidas Micro Recall e Macro Recall do algoritmo multirótulo ultrapassam as medidas Micro Recall e Macro Recall do algoritmo pseudo-multirótulo.

4.3.4
Experimentos com a base Ohsumed

Nos experimentos foi utilizado um subconjunto da base Ohsumed definido pelos primeiros 20.000 documentos de 1991 que possuem abstracts, classificados em 23 categorias sobre doenças.

Primeiramente, foi realizado um experimento com uma partição fixa definida por 6.286 documentos de treinamento e 7.643 documentos de teste, um vocabulário de 27.871 palavras e 756 combinações das 23 categorias da base.

As tabelas 15, 16 e 17 apresentam os resultados do algoritmo pseudo-multirótulo. Já as tabelas 18, 19 e 20 apresentam os resultados do algoritmo mulirótulo.

Categoria	Recall	Precision	F1
Pathology	28,15	51,53	36,41
Cardiovasc	82,63	58,71	68,65
Immunolog	50,65	60,27	55,04
Neoplasms	84,46	62,23	71,66
Dig.Syst.	39,56	79,87	52,91

Tabela 15 – Resultados do algoritmo pseudo-multirótulo na base Ohsumed para 5 categorias.

Micro Recall	38,73
Micro Precision	62,53
Micro F1	47,83
Macro Recall	23,70
Macro Precision	71,13
Macro F1	28,48

Tabela 16 – Resultados globais do algoritmo pseudo-multirótulo na base Ohsumed.

Classe	Segundos
Treinamento	10,672
Classificação	727,516

Tabela 17 – Tempo de execução da fase de treinamento e da fase de classificação do algoritmo pseudo-multirótulo na base Ohsumed.

Categoria	Recall	Precision	F1
Pathology	59,96	43,81	50,63
Cardiovasc	79,63	68,07	73,40
Immunolog	55,83	56,07	55,95
Neoplasms	77,51	73,21	75,30
Dig.Syst.	65,03	55,09	59,65

Tabela 18 – Resultados do algoritmo multirótulo na base Ohsumed para 5 categorias.

Micro Recall	52,75
Micro Precision	60,04
Micro F1	56,16
Macro Recall	37,63
Macro Precision	65,40
Macro F1	43,27

Tabela 19 – Resultados globais do algoritmo multirótulo na base Ohsumed.

Classe	Segundos
Treinamento	40,281
Classificação	97,857

Tabela 20 – Tempo de execução da fase de treinamento e da fase de classificação do algoritmo multirótulo na base Ohsumed

Após o experimento utilizando a partição fixa, foi realizado um experimento para verificar como se comportam os dois algoritmos em função da quantidade de documentos de treinamento.

Para isso, para cada um dos algoritmos, foram gerados 10 partições aleatórias, onde 6.286 documentos eram de treinamento e 7.643 documentos eram de teste.

Para cada partição rodou-se o algoritmo 11 vezes, mantendo-se os documentos de teste fixos (7.643 documentos) e variando a quantidade de documentos de treinamento pertencentes ao conjunto de 6.286 documentos, iniciando com 6.286 e dividindo por 2 até a quantidade de documentos de treinamento chegar a 6.

Uma vez gerada a curva para cada uma dos 10 partições calculou-se a média das 10 curvas.

As figuras abaixo apresentam os resultados do experimento para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo.

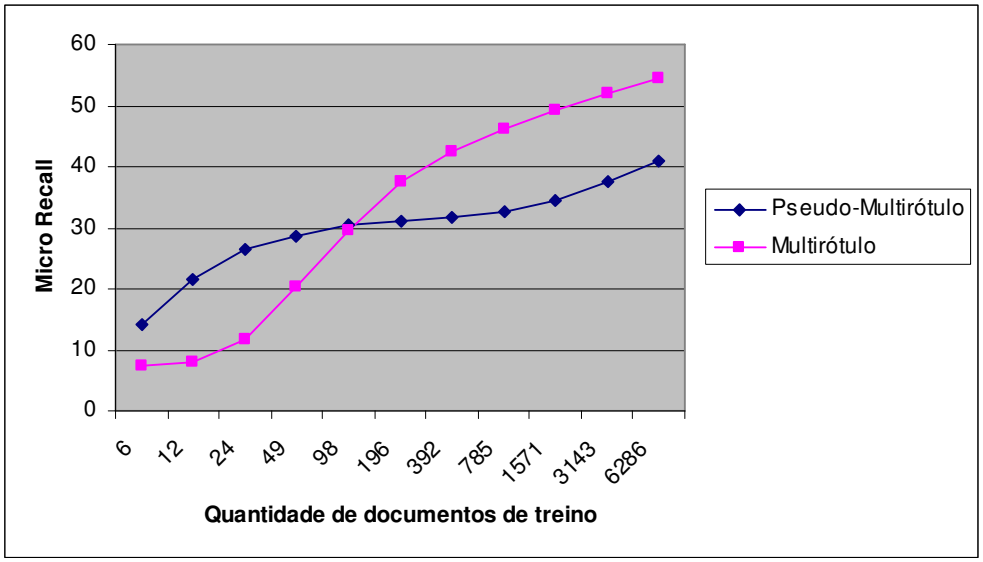

Figura 15 – Resultados Micro Recall para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base Ohsumed.

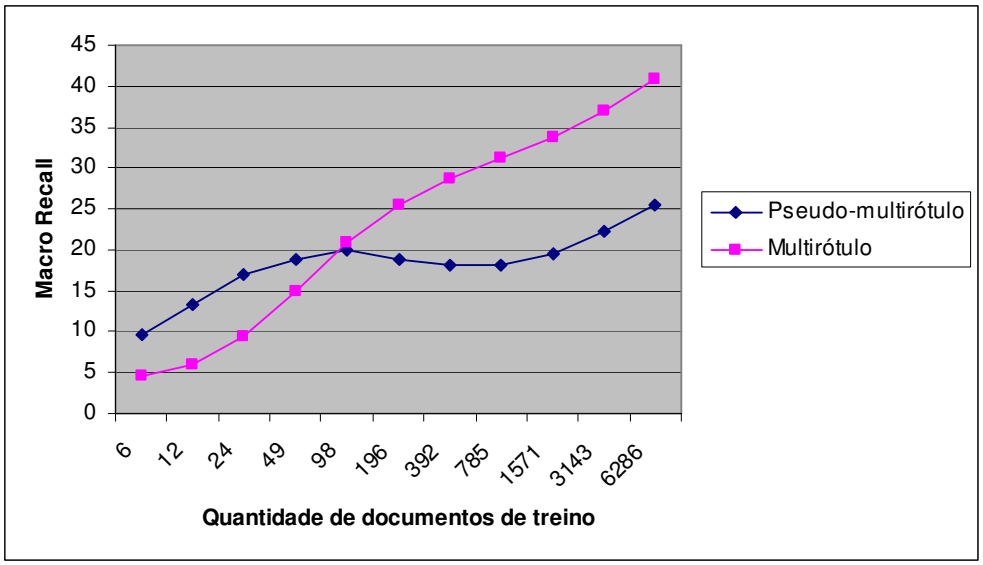

Figura 16 – Resultados Macro Recall para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base Ohsumed.

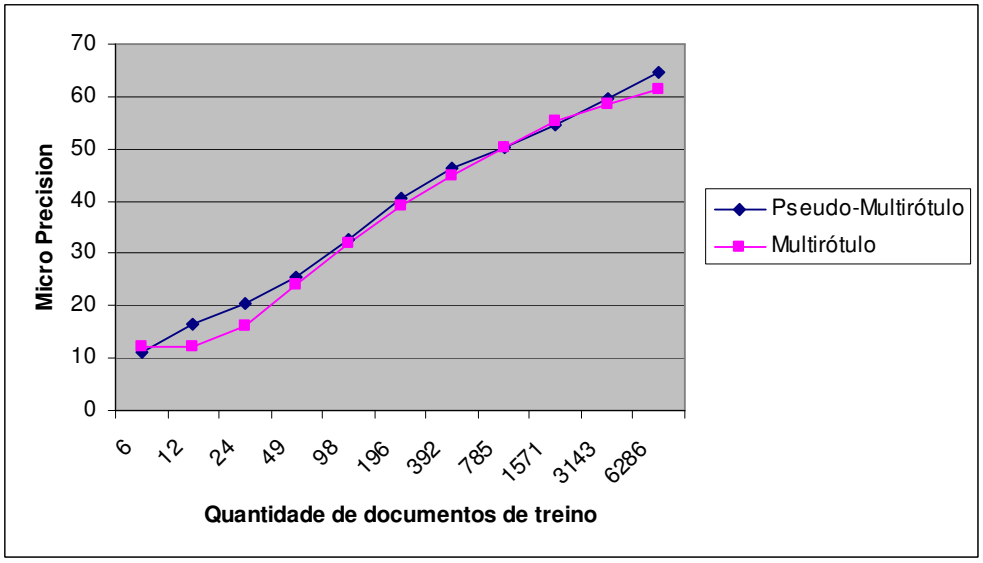

Figura 17 – Resultados Micro Precision para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base Ohsumed.

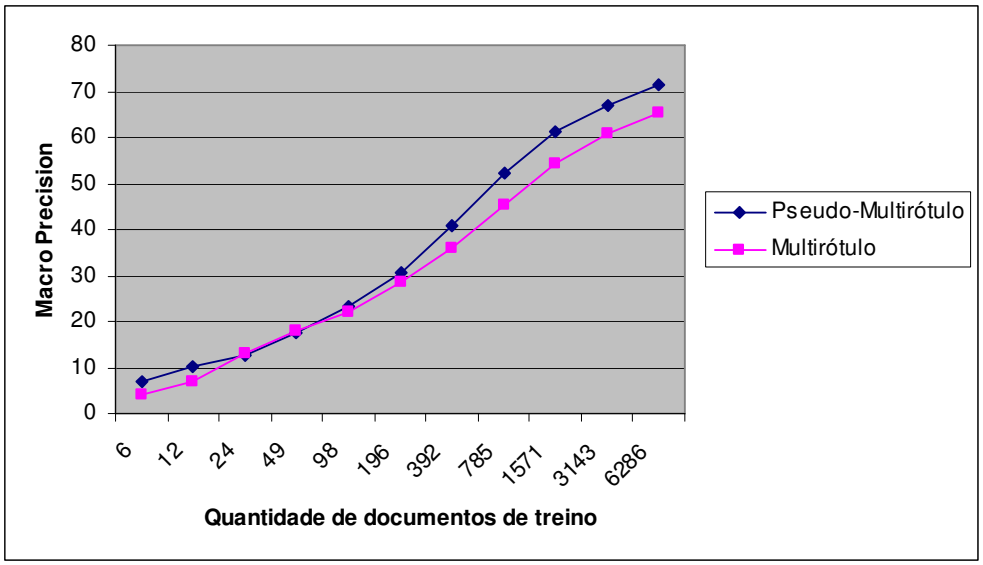

Figura 18 – Resultados Macro Precision para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base Ohsumed.

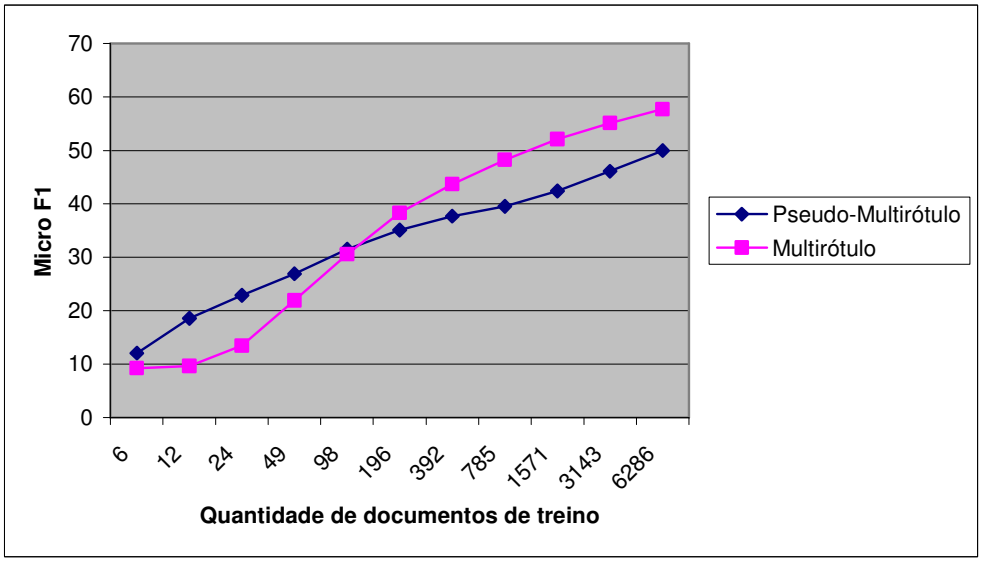

Figura 19 – Resultados Micro F1 para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base Ohsumed

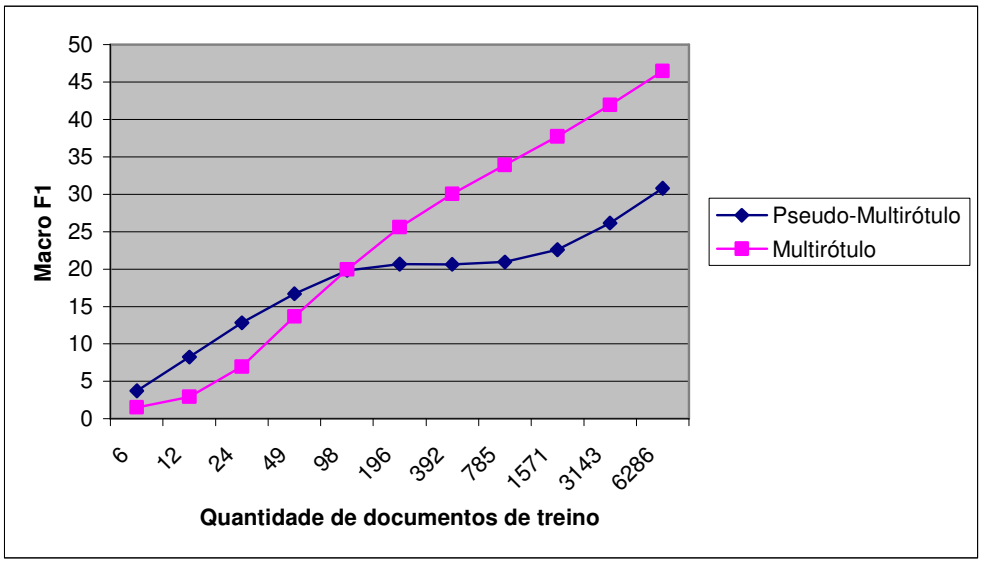

Figura 20 – Resultados Macro F1 para o algoritmo pseudo-multirótulo e para o algoritmo multirótulo na base Ohsumed

4.3.5
Conclusões dos experimentos com a base Ohsumed

Com base nos resultados da partição fixa, o algoritmo multirótulo obteve resultados melhores que o algoritmo pseudo-multirótulo, chegando a uma diferença de 51% entre os valores Macro F1, explicado por um aumento de 58% da medida Macro Recall.

Como na base da Reuters, o algoritmo multirótulo apresenta um aumento nos valores da medida recall e uma diminuição no valor da medida precision, porém o valor menor da medida precision não é suficiente para afetar o valor da medida F1, que em todos os experimentos se mostrou maior para o algoritmo multirótulo.

Nos experimentos com partições aleatórias verificou-se também que o algoritmo multirótulo obteve resultados piores que o algoritmo pseudo-multirótulo para poucos documentos de treinamento, porém, quando o número de documentos de treinamento chega a 98, os valores de Micro F1 e Macro F1 do algoritmo multirótulo ultrapassam os valores do algoritmo pseudo-multirótulo, chegando a uma diferença de 66%.

4.3.6
Comparação com outros trabalhos

Nesta seção, serão comparados os resultados obtidos pelo algoritmo multirótulo, que apresentou melhores resultados, com resultados de outros trabalhos na literatura de classificação de textos.

A grande dificuldade encontrada para realizar a comparação é a divergência de medidas utilizadas para avaliar o desempenho dos algoritmos propostos. Muitos trabalhos, por exemplo, apresentaram seus resultados através da medida "breakeven point", que só faz sentido em classificadores que usam limiares.

4.3.6.1.
Reuters

Bennett et al. [2002] propuseram um método de combinação de classificadores e testou esse método na base de dados Reuters R(10), utilizando quatro classificadores conhecidos na literatura: árvore de decisão, support vector machine, naive Bayes binário e naive Bayes multinomial. A seguir serão apresentados os resultados individuais dos classificadores, o resultado gerado pela combinação dos quatro classificadores (Strive-S) e por último os resultados dos algoritmos propostos neste trabalho:

Método	Macro F1
Árvore de decisão	78,46
Support vector machine	84,80
Naive Bayes binário	65,74
Naive Bayes multinomial	76,45
Strive-S (norm)	87,49
Naive Bayes pseudo-multirótulo	**84,74**
Naive Bayes multirótulo	**85,86**

Tabela 21 – Resultados de Bennett et al. [2002] na base R(10)

Já Sebastiani & Debole [2004] comparam a dificuldade de três subconjuntos da base de dados da Reuters, R(10), R(90) e R(115) comumente utilizados na literatura. Para realizar a comparação foram realizados testes através dos classificadores Rochhio, *K*-NN (K-nearest neighbor) e Support vector machine. A média dos resultados obtidos pelos três classificadores para a base R(10) e R(90) são apresentados na tabela 22 e tabela 23, assim como os resultados obtidos neste trabalho:

Método	Micro F1	Macro F1
Rochhio, *K*-NN e Support vector machine	85,223540	72,393364
Naive Bayes pseudo-multirótulo	**93,05**	**84,74**
Naive Bayes multirótulo	**93,17**	**85,86**

Tabela 22 – Resultados de Sebastiani & Debole [2004] na base de dados R(10)

Método	Micro F1	Macro F1
Rochhio, *K*-NN e Support vector machine	78,707075	52,659655
Naive Bayes pseudo-multirótulo	**77,50**	**21,92**
Naive Bayes multirótulo	**78,68**	**30,73**

Tabela 23 – Resultados de Sebastiani & Debole [2004] na base de dados R(90)

No trabalho de Yang & Liu [1999] é comparado o desempenho de cinco classificadores de texto na base de dados R(90): SVM (Support vector machine), *K*-NN (K-nearest neighbor), LLSF (Linear Least Squares Fit), NB (Naive bayes baseado em um modelo misto multinomial, proposto por McCallum [1999]) e NNet (Rede neural). Os resultados desse experimento serão apresentados na tabela 24, assim como os resultados dos algoritmos naive Bayes pseudo-multirótulo (NBPM) e naive Bayes multirótulo (NBM) propostos neste trabalho.

Método	Micro Recall	Micro Precision	Micro F1	Macro F1
SVM	81,20	91,37	85,99	52,51
KNN	83,39	88,07	85,67	52,42
LLSF	85,07	84,89	84,98	50,08
NNet	78,42	87,85	82,87	37,65
NB	76,88	82,45	79,56	38,86
NBPM	**72,58**	**83,14**	**77,50**	**21,92**
NBM	**79,68**	**77,70**	**78,68**	**30,73**

Tabela 24 – Resultados de Yang & Liu [1999] na base de dados R(90)

4.3.6.2. Ohsumed

Alessandro Moschitti [2003b] realizou um estudo relativo à aplicação de técnicas de processamento de linguagem natural na classificação automática de textos. Desta forma, informações sintáticas, como nomes próprios e substantivos compostos, foram contempladas na representação dos documentos.

Em um dos seus experimentos, Alessandro testou os algoritmos Parametrized Rochhio (proposto pelo autor) e Support Vector Machine, na base de documentos médicos Ohsumed, utilizando a representação básica "bag-of-words" sem nenhuma informação sintática. Os resultados desse experimento serão apresentados na tabela 25, assim como os resultados dos algoritmos propostos neste trabalho.

Outros experimentos foram realizados incluindo informação sintática na representação dos documentos, porém não são comparáveis com este trabalho, cujos experimentos se basearam apenas na representação simples "bag of words".

	PRC	SVM	**NBPM**	**NBM**
Categoria	F1	F1	**F1**	**F1**
Pathology	50.58	48.5	**36,41**	**50,63**
Cardiovasc	77.82	80.7	**68,65**	**73,40**
Immunolog	73.92	72.8	**55,04**	**55,95**
Neoplasms	79.71	80.1	**71,66**	**75,30**
Dig.Syst.	71.49	71.1	**52,91**	**59,65**

Tabela 25 – Resultados de Moschitti [2003b] em 5 categorias da base de dados Ohsumed

Método	Micro F1
PRC	65,80
SVM	68,37
NBPM	**47,83**
NBM	**56,16**

Tabela 26 – Resultados globais de Moschitti [2003b] na base de dados Ohsumed