Agente Inteligente para Classificação de Notícias por Assunto

Alex Souza[1], **José Everardo Bessa Maia**[1]

[1]Universidade Estadual do Ceará - Fortaleza - CE - Brasil

alex.souza@aluno.uece.br, jose.maia@uece.br

Abstract. *The article presents an evaluation of the effectiveness of combinations of document representation models, including a new approach, with multi-label classification algorithms in the implementation of an intelligent agent to classify news by subject. The agent's job is to classify the news into interest classes and distribute them according to the subscriber profile. The document representation models of the TF-IDF, WORD2VEC traditional type and the proposed* `W2VP-IDF` *approach are combined with the KNN, SVM and Decision Tree classifiers and evaluated. The test results show the significantly superior performance of the TF-IDF combination with SVM over all others, which is the recommended combination. The proposed* `W2VP-IDF` *representation obtained the best results among the representation models for the KNN classifier.*

Resumo. *O artigo apresenta uma avaliação da eficácia de combinações de modelos de representação de documentos, incluindo uma nova abordagem, com algoritmos de classificação multirrótulo na implementação de um agente inteligente para classificar notícias por assunto. O trabalho do agente é classificar as notícias em classes de interesse e distribuí-las conforme o perfil dos assinantes. Os modelos de representação de documentos do tipo matriz TF-IDF, WORD2VEC tradicional e a abordagem proposta* `W2VP-IDF` *são combinados com os classificadores KNN, SVM e Árvore de Decisão e avaliados. Os resultados dos testes mostram o desempenho significativamente superior da combinação TF-IDF com SVM sobre todos os outros, que é a combinação recomendada. A representação* `W2VP-IDF` *proposta, obteve os melhores resultados dentre os modelos de representação para o classificador KNN.*

1. Introdução

O grande volume de dados disponível na internet torna difícil para o leitor encontrar informações ou notícias do seu interesse. Os Sistemas de Recomendação (SR) abordam essa problemática, filtrando esse grande volume de dados e recomendando as informações consideradas de maior relevância em conformidade a um perfil de interesse do leitor.

A recomendação de notícias na web difere do cenário de recomendação de outros produtos em vários aspectos: os artigos de notícias têm um ciclo de vida curto, forçando o recomendador a considerar continuamente novos artigos e descartar os desatualizados. Os usuários estão interessados em vários tópicos e domínios variáveis, tornando difícil prever os artigos relevantes em um novo domínio. Um evento inesperado pode ser relevante para um usuário, embora nenhum evento semelhante tenha sido observado no passado. Além desses:

1. Um Sistema de Recomendação de Notícias (SRN) lida com dados não estruturados que podem ser difíceis de interpretar, incompletos e inconsistentes.

2. O usuário geralmente não está disposto a avaliar a relevância das notícias tendo o SRN que interpretar sinais implícitos para aprender os seus interesses e satisfação.
3. Notícias muitas vezes são de interesse para apenas uma área geográfica limitada.

Como resultado, os SRN modernos tendem a adotar soluções híbridas, nas quais a filtragem colaborativa e a recomendação baseada em conteúdo são combinadas [Gulla et al. 2017]. Filtragem colaborativa procura por usuários semelhantes e recomenda artigos de notícias pelos quais esses usuários similares já mostraram algum interesse. A recomendação baseada em conteúdo não envolve outros usuários. A ideia é antes a de criar um perfil de usuário com base no comportamento histórico e recomendar notícias que sejam semelhantes em tópico ao que ele leu no passado.

A experiência indica que os usuários também apreciam estratégias adicionais que impulsionam novas notícias, como as notícias que se tornam populares e as notícias que ocorrem em seu próprio bairro. Alguns sistemas optam por juntar todas essas estratégias em uma única lista de notícias recomendadas, enquanto outros deixam o usuário ativar as estratégias que preferem antes que as recomendações sejam feitas. Além dessas, uma outra abordagem é a simples subscrição nos tópicos de interesse por parte dos próprios usuários, em um catálogo de tópicos apresentado pelo sistema (*subscribe*).

Em qualquer um dos casos, a classificação multirrótulo de textos é um procedimento central na construção de um SRN. O objetivo deste trabalho é avaliar a eficácia da combinação entre modelos de representação de documentos, incluindo o modelo proposto, e algoritmos de classificação na implementação de um agente classificador de notícias. O restante deste artigo está estruturado da seguinte forma. A Seção 2 fornece uma descrição da estrutura do SRN em construção e a Seção 3 contextualiza alguns trabalhos relacionados. A Seção 4 descreve os modelos de representação de documentos, o conceito de classificação multirrótulo e os algoritmos de classificação avaliados. A Seção 5 apresenta a abordagem proposta. Os dados, o plano experimental e uma análise dos resultados são expostos na Seção 6, e o artigo é concluído na Seção 7.

2. O SRN baseado em *publish-subscribe*

A estrutura do SRN está mostrada na Figura 1. A ideia central do sistema é utilizar o paradigma de comunicação *publish/subscribe*, onde os produtores (como por exemplo: CNN, NYTimes, Reuters, UOL, Globo.com) publicam as notícias (*publish*) enquanto os leitores/usuários, aqui chamados assinantes (*subscribe*), inscritos em tópicos do seu interesse passam a receber (*delivery*) notícias apenas desses tópicos.

O software está organizado em torno de três agentes. Um agente de captura, preprocessamento e armazenamento de notícias (*storage*), um agente de inscrição e aprendizagem de perfis de assinantes e um agente classificador multirrótulo de notícias, que é o foco deste trabalho.

O paradigma de comunicação *publish/subscribe* tem propriedades interessantes para esta aplicação, sendo uma delas o desacoplamento entre produtores e assinantes em termos de tempo, espaço e sincronização. Em *publish/subscribe* os geradores de notícias e os assinantes não necessitam estar disponíveis ao mesmo tempo. Isso significa que um assinante pode receber as notificações mesmo que um novo produtor seja criado para determinado contexto. Tal característica tem alavancado o uso de aplicações com comunicação

publish/subscribe, principalmente no contexto de computação móvel, onde a existência de uma conexão muitas vezes não pode ser garantida. Além disso, os eventos publicados são disponibilizados para os assinantes interessados, sem os produtores necessariamente conhecerem os assinantes ou sua localização, e vice-versa [Adhianto et al. 2010].

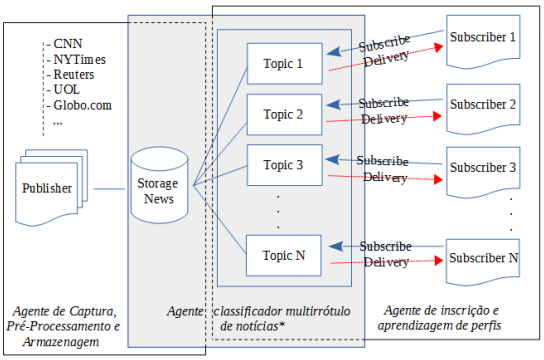

Figura 1. Estrutura do SRN baseada em *Publish-Subscribe*

Note que, mesmo quando os produtores publicam (*publish*) informações já organizadas em "tópicos" enquanto os assinantes especificam seus "tópicos" de interesse, ainda assim, a intenção de informação do assinante pode não está completamente refletida na estrutura dos tópicos, pois ambas podem estar completamente desacopladas. O SRN precisa ser um unificador semântico para os assinantes.

3. Trabalhos Relacionados

Esta seção apresenta uma breve revisão de artigos relacionados com o objetivo de contextualizar a abordagem proposta.

Em [Lewis et al. 2004], os autores utilizaram abordagens de aprendizado supervisionadas que foram amplamente estudadas em experimentos de categorização de texto. Os classificadores utilizados foram: SVM, KNN e Rocchio com TF-IDF como representação de documento, aplicados à coleção de documentos RCV1 (*Reuters Corpus Volume I* - 804414 documentos). Os melhores resultados foram obtidos usando o classificador SVM atingindo 81.6% da medição F1 (micro-média).

Uma representação de documento para classificação de texto, baseada numa extensão da Teoria da Informação denominada LIT (*Least Information theory*) foi proposto por [Ke 2012]. Ele pondera os termos na representação utilizando LIB (*I Binary*) e LIF (*LI Frequency*). Para os testes, utilizaram o classificador KNN (k = 25) combinado com o modelo proposto. Os resultados mais expressivos obtidos para a coleção de documento RCV1 foi usando a ponderação composta LIBxLIF, aonde a medição F1 chegou a 81.8%.

Em [Quispe et al. 2017] os autores propõem um modelo para resolver o problema multi-label, integrando as seguintes técnicas. Primeiro, eles utilizaram indexação semântica latente (LSI) para representação de texto. Em seguida, utilizaram Redes Convolucionais (CNN) para extração de recursos e, finalmente, um *Perceptron Multi Layer* simples como classificador. As métricas obtidas pelo modelo (LSI CNN), para a coleção RCV1, foram: 88.15% de *Precision*, 84.80% de *Recall* e 84.77 de *F1-Score*.

Desde que os *embeddings* de palavras capturam as relações semânticas, vários trabalhos surgiram explorando essa abordagem. Em [He et al. 2018] foram comparados

modelos neurais avançadas (TextCNN, RCNN, HAN) e suas versões evolutivas com base em estrutura de suavidade temporal (TS) e em estrutura de propagação diacrônica (DP), ver [He et al. 2018], utilizando *Word2Vec* como representação de documentos. Os melhores resultados, para a coleção RCV1, foram obtidos pelo modelo (RCNN-DP) com precisão de 95.76%, seguido dos métodos TextCNN-DP (94.38%) e HAN-DP (85.45%).

Esta seção apresentou técnicas e resultados de trabalhos recentes para mostrar os níveis de acurácia obtidos. Note que eles ficam entre 85% e 95%. Os testes apresentados neste trabalho permanece na faixa superior atingindo 91% nos mesmos *datasets*.

4. Métodos

Esta seção descreve brevemente os métodos utilizados neste trabalho. O leitor interessado em apresentações detalhadas deve procurar as referências indicadas.

4.1. TF-IDF

O *TF-IDF (Term Frequency-Inverse Document Frequency)* é uma abordagem clássica da área de Processamento de Linguagem Natural (PLN). Segundo [Baeza-Yates and Ribeiro-Neto 2013] *TF-IDF* é uma medida estatística destinada a medir o grau de importância de uma palavra (termo) para um conjunto de documentos. Ele efetua a equação 1 para cada combinação documento-termo, onde N é o número de documentos na coleção. O valor w_{d_i,t_j} da matriz aumenta proporcionalmente em relação a tf_{d_i,t_j} (número de ocorrências de t_j em d_i). No entanto, esse valor é equilibrado pelo df_{t_j} (número de documentos que contém t_j). Isso auxilia a distinguir o fato da ocorrência de algumas palavras serem mais comuns que outras.

4.2. *Words Embeddings*

A representação por *Word Embedding* é uma das formas de resolver os problemas encontrados em modelos TF-IDF, como a alta dimensionalidade e ambiguidade semântica. Neste trabalho foi avaliado o modelo *Word2Vec* [Mikolov et al. 2014]. *Word2Vec* é um modelo treinado por uma rede neural para encontrar representações de palavras em um espaço de *n* dimensões. Baseia-se no modelo de hipótese distribucional, onde palavras que ocorrem em um contexto similar têm um significado similar.

A partir da representação *embedding* das palavras é necessário obter uma representação de documentos. Para isso, foram avaliadas duas abordagens. A primeira é uma representação na qual um documento (notícia) é representado pelo vetor média dos vetores das palavras que o compõe (*Word2Vec averaging*), utilizada em [Kim et al. 2017]. A segunda, uma abordagem de média ponderada proposta neste trabalho, será descrita na Seção 5.

4.3. Classificação multirrótulo

Classificação multirrótulo multi-classe difere de classificação de rótulo único principalmente pela possibilidade de considerar a correlação entre rótulos como fonte de informação na classificação.

Para uma descrição formal do problema de classificação multirrótulo, seja $X = (x_1, ..., x_n)$ um conjunto finito de instâncias (documentos, notícias) e $Y = (y_1, ..., y_m)$ um conjunto finito de rótulos (tópicos) tal que cada instância $x_i \in X$ está associada a

múltiplos rótulos de classe Y_i, onde $Y_i \subseteq Y$. Dado um conjunto de exemplos de treinamento $S = \{(x_1, Y_1), ..., (x_n, Y_m)\}$ a tarefa de aprendizagem de máquina é construir, a partir de S um classificador $f : X \rightarrow 2^Y$ capaz de estimar uma função alvo desconhecida $\varphi : X \rightarrow 2^Y$. Neste contexto, o conjunto potência 2^Y representa todos os possíveis perfis de assinante ou todos possíveis rótulos para uma notícia. O problema de classificação multiclasse com um único rótulo é um caso especial do problema multirrótulo no qual a cada instância é atribuído apenas um rótulo.

Ao problema multirrótulo, para cada um dos classificadores estudados, foi aplicada a estratégia *One-Against-All* (Um-contra-todos) – que constrói $|C|$ modelos de predição binários, onde $|C|$ é o número de classes. Cada modelo é treinado para separar uma das classes das demais. Portanto, o i-ésimo modelo é treinado considerando que todos os exemplos da i-ésima classe pertencem à classe positiva, enquanto os exemplos das outras classes pertencem à classe negativa. Quando um exemplo de classe desconhecida é apresentado, ele é classificado por cada um dos $|C|$ modelos de predição e recebe a classe relativa ao modelo que obtiver o melhor resultado [Hsu and Lin 2002].

4.4. Classificadores

Os três seguintes classificadores foram avaliados neste estudo:

KNN (*K-Nearest Neighbor*): O *KNN* Classifica novas amostras de acordo com as K amostras do conjunto de treinamento mais próximas a essas novas amostras. O *KNN* usa uma medida de distância para definir a semelhança (proximidade) de uma amostra com outra [Duda et al. 1995]. Dado um conjunto de n pares $\{(x_1, \theta_1), ..., (x_n, \theta_n)\}$, em que x_i toma valores de um espaço X, e θ_i toma valores de um conjunto $1, 2, ..., M$. Considera-se cada θ_i como o índice do tópico a que pertence o *i-ésimo* indivíduo, e cada x_i o resultado de um conjunto de medições feitas sobre aquele indivíduo. Se é dado um novo par (x, θ), no qual apenas o valor de x é conhecido e deseja-se estimar θ a partir do conjunto de pontos classificados corretamente, $x_n^{'} \in \{x_1, x_2, ..., x_n\}$ é o vizinho mais próximo de x se $min\ d(x_i, x) = d(x_n^{'}, x)$, com $i = 1, 2, ..., n$. A regra vizinho mais próximo decide que x pertence ao tópico $\theta_n^{'}$ de seu vizinho mais próximo $x_n^{'}$. A distância d é determinada por uma métrica de similaridade, geralmente a distância Euclidiana (utilizada no artigo, com K = 5). Apesar de sua simplicidade, o *KNN* apresenta um bom desempenho, mas possui algumas desvantagens como alto custo computacional para calcular a distância entre a nova amostra e todas outras do conjunto de treinamento; baixa precisão em espaços de características muito elevados e dificuldade em se definir o melhor valor do parâmetro K.

SVM (*Support Vector Machines*): No *SVM*, a classificação se baseia na margem de separação das classes. Assim, o objetivo do treinamento do *SVM* consiste em encontrar um hiperplano separador ótimo, ou seja, aquele em que a distância de separação entre as classes é máxima, chamado hiperplano de margem máxima. As amostras que estão situadas sobre as margens são as mais informativas para a criação do limite de decisão da classificação e são chamadas de vetores suporte. A classificação pode ser realizada tanto no espaço original dos atributos quanto em um espaço de características projetado através de uma função de kernel. Assim, problemas que não são linearmente separáveis no espaço original podem tornar-se linearmente separáveis no espaço de características. À medida que a dimensão do espaço de características aumenta, também aumenta a probabilidade desse problema se tornar linearmente separável. A habilidade de separar dados

com distribuição não linearmente separável depende da escolha da função kernel, e deve ser analisada de acordo com o domínio do problema. Os kernels mais usados são: Linear, Polinomial e Função de base radial (RBF), no artigo foi utilizado: Kernel Linear.

Decision Tree: É um algoritmo de classificação (ou regressão) constituido essencialmente uma série de decisões *if-else*. Os dados vão sendo particionados em subconjuntos e alguma medida de pureza dos subconjuntos vai sendo avaliada para decidir quando parar. A medida mais frequentemente utilizada é a entropia do subconjunto e a decisão sobre os pontos de particionamento é baseada em ganho de informação. Um tratamento detalhado para classificação pode ser encontrado em [Caraciolo 2009]

5. Abordagem `W2VP-IDF`

Uma abordagem comum utilizada para ir da representação *embedding* das palavras para uma representação de documentos é a representação na qual um documento (notícia) é representado pelo vetor média dos vetores dos termos que o compõe (*Word2Vec averaging*) [Kim et al. 2017]. A proposta deste trabalho, ao invés, é representar um documento pela média dos vetores dos termos, ponderada pela importância dos termos na discriminação dos documentos. A intuição por trás desta proposta é que um mesmo termo, presente em diferentes documentos, apresenta poder discriminatório diferente. Ela será identificada pelo acrônimo `W2VP-IDF` (*Word2Vec* ponderado por *IDF*). Algoritmo 1 apresenta o pseudocódigo para obter a representação de documentos `W2VP-IDF`.

Algorithm 1: Pseudocódigo da Representação de Documento `W2VP-IDF`

```
Data: DOCUMENTOS (Notícias), WORD2VEC (Vetor Word2Vec de cada
      termo da coleção)
Result: Vetor W2VP-IDF de cada Documento
while Existir Termos na Coleção do
    Calcula_IDF_Termo()   //Monta o vetor: T-IDF (Termo e IDF do Termo);
while Existir Documentos na Coleção do
    while Existir Termos no Documento do
        W2VP-IDF(documento) = Média (WORD2VEC(termo) *
        T-IDF(Termo));
```

A ponderação por termos já foi vista antes em [Rei and Cummins 2016] e [Liu and Yang 2012]. Entretanto, de forma diferente. Em [Rei and Cummins 2016] *Embeddings (Word2Vec)* é ponderado por *IDF*, para representar sentenças, onde cada palavra recebe um peso separado dado por $IDF(w) = log(N/1 + n_w)$, onde w é a palavra, N é o numero total de sentenças da coleção e n_w é o número de sentenças em que a palavra-alvo w ocorre.

Já em [Liu and Yang 2012], os autores propõem um novo método de ponderação chamado *TF-IDF-CF* baseado no *TF-IDF*, com a adição de frequência de classe, a fim de melhorar a precisão que é um dos principais problemas da classificação de texto. A fórmula utilizada é a seguinte: $a_{ij} = log(tf_{ij}) * log(N + 1.0/n_j) * n_{cij}/N_{ci}$, onde n_{cij} representa o número de documentos onde o termo j aparece dentro da mesma classe que o documento i pertence, N_{ci} representa o número de documentos dentro da mesma classe que o documento i pertence.

Por outro lado, nesse trabalho, a ponderação *IDF* é dada pela fórmula:

$$IDF(w) = log\left(\frac{N}{df_t}\right) + 1 \quad (1)$$

onde w é o termo (palavra), N é o numero total de documentos da coleção e df_t é o número de documentos em que o termo w ocorre. A composição $log() + 1$ em vez de log garante que os termos com *IDF* igual a zero não sejam suprimidos inteiramente.

6. Dados e Resultados

6.1. Dados

Os testes foram realizados na coleção de notícias *Reuters (RCV1-Reuters Corpus Volume 1)*, que contém mais de 800.000 notícias em inglês manualmente classificadas em 103 tópicos (Economia, Ciência e Tecnologia, Esportes, Corporativo, entre outras) e disponibilizadas, para fins de pesquisa, pela *Reuters, Ltd.*

A coleção encontra-se preprocessada, ou seja, tokenizada, sem *stopwords* (palavras com baixo poder discriminativo (artigos, preposições)) e com *stemming* (radicalização das palavras), além de apresentar suporte multirrótulo. Devido à limitação computacional, utilizou-se 33.149 notícias divididas em 103 tópicos distintos. Mais sobre o preprocessamento de texto veja em [Baeza-Yates and Ribeiro-Neto 2013].

6.2. Plano experimental

O passo a passo do procedimento experimental para avaliar as configurações do Agente Inteligente classificador de notícias foi:

- Importar a coleção de notícias *Reuters (RCV1)*;
- Criar 20 *datasets* de treinamento e 20 de testes selecionados aleatoriamente e independentes.
- Criar as representações *TF-IDF, Word2Vec* e `W2VP-IDF` da coleção (Os dois últimos usando CBOW (*Continuous Bag of Words*) e Vetor de Recursos de 100 e 600 dimensões da biblioteca GENSIM do *Python*);
- Classificadores (para cada *dataset*)
  - **Treinamento** - Treinar os classificadores *SVM, DTree e KNN* (usando validação cruzada) utilizando as representações de documentos *TF-IDF, Word2Vec* e `W2VP-IDF`;
  - **Teste** - Proceder a classificação dos *datasets* de teste para todas as combinações classificador/representação de documento;
- Mostrar a tabela comparativa de *precision*, *recall* e *f1-score* das combinações e selecionar aquela mais promissora;
- Mostrar os gráficos detalhados sobre *precision* e *recall* (*Average Precision* e *Curve ROC*) da combinação escolhida.

6.3. Resultados

Precision e *Recall* são métricas para avaliar a qualidade de saída do classificador. Na recuperação de informações, a *precision* é uma medida da relevância do resultado, enquanto o *recall* é uma medida de quantos resultados realmente relevantes são retornados. A *F1-Score* é interpretada como uma média ponderada do *precision e recall*.

Uma das abordagens utilizadas para obter *precision, recall, f1-score* é por micro-média, que calcula as métricas globalmente, contando o total de verdadeiros positivos, falsos negativos e falsos positivos. A Tabela 1 mostra o desempenho alcançado por cada Classificador aplicado aos modelos de Representação de Documentos (Repres.Doc) nos conjuntos de teste. Os resultados foram calculados pelas micro-médias do *precision*, *recall* e *f1-score (F1)*. Os treinamentos utilizaram 3 iterações da validação cruzada *3-fold* estratificada, para os conjuntos de treinamento. Os valores em negrito indicam os cinco melhores resultados obtidos dentre as combinações.

Coleção de Documentos: ***RCV1***						
Classificador	Repres.Doc.	Dimensões	***Precision***	***Recall***	***F1***	*Time(min)*
SVM	***TF-IDF***	-	**0.9136**	**0.7477**	**0.8224**	**256**
SVM	*Word2Vec*	100	0.8768	0.6409	0.7397	101
SVM	***Word2Vec***	**600**	**0.8901**	**0.6823**	**0.7724**	**626**
SVM	W2VP-IDF	100	0.8633	0.6463	0.7391	592
SVM	**W2VP-IDF**	**600**	**0.8694**	**0.7218**	**0.7887**	**636**
DTree	*TF-IDF*	-	0.7127	0.6977	0.7045	261
DTree	*Word2Vec*	100	0.5682	0.6014	0.5838	127
DTree	*Word2Vec*	600	0.5734	0.6188	0.5955	598
DTree	W2VP-IDF	100	0.5670	0.6012	0.5835	592
DTree	W2VP-IDF	600	0.5731	0.6179	0.5946	626
KNN	*TF-IDF*	-	0.8322	0.6819	0.7496	243
KNN	*Word2Vec*	100	0.8500	0.6968	0.7651	119
KNN	*Word2Vec*	600	0.8506	0.6970	0.7661	617
KNN	**W2VP-IDF**	**100**	**0.8508**	**0.6991**	**0.7674**	**592**
KNN	**W2VP-IDF**	**600**	**0.8499**	**0.6989**	**0.7670**	**620**

Tabela 1. Resultados das combinações - Repres. de documentos / Classificador

Na métrica *f1-score* podemos observar que o classificador *SVM* em conjunto com as representações de documentos: *TF-IDF*, W2VP-IDF (600) e *Word2Vec* (600) obtiveram os melhores resultados dentre as combinações, seguido do *KNN* com a representação de documentos proposta W2VP-IDF (100 e 600). Vemos que utilizando representação de documentos *Embeddings*, apenas o classificador *KNN* apresentou melhor desempenho do que usando a tradicional representação *TF-IDF*. As combinações *DTree* com W2VP-IDF, *Word2Vec* ou *TF-IDF* obtiveram um desempenho muito abaixo do esperado para essa coleção de documentos em relação as demais combinações.

Observamos que quanto maior a dimensionalidade do vetor de recursos (*Embeddings*), melhores são os resultados, mas, em contrapartida, o tempo computacional aumenta significativamente. Apenas a combinação *SVM* com W2VP-IDF e *Word2Vec* obtiveram um ganho significativo, na métrica: *f1-score*, ao mudar a dimensionalidade do vetor de recursos de 100 para 600, nas demais, a melhoria foi relativamente baixa.

O modelo de representação de documentos proposto: W2VP-IDF alcançou os melhores resultados dentre as combinações, ao utilizá-lo com o classificador *KNN*, como podemos ver, obteve-se uma melhoria no *F1-Score* de aproximadamente 2.33% em relação ao tradicional *TF-IDF*. Já combinando a representação W2VP-IDF com os demais classificadores não obtivemos ganhos, em relação ao *TF-IDF*, para a coleção RCV1.

Analisando individualmente a combinação classificador/representação de docu-

mentos mais bem avaliada, *SVM com TF-IDF*, agora não mais utilizando *cross-validation*, mas sim, a divisão da coleção em 70% para treinamento e 30% para teste, obtivemos uma *Average Precision* de 90% para todas os tópicos, que corresponde a Área sob a Curva *Precision-Recall* (PR-AUC), apresentada na Figura 2.

As Figuras 3 mostram curva ROC média (A) e detalhada (B). Na Figura 3 (A), a Curva ROC média para todos os tópicos confirma o que já estava revelado na curva *precision-recall* (Figura 2). Já na Figura 3 (B), que mostra um detalhamento por tópico, pode-se ver que o agente tem desempenho satisfatório em praticamente todos os tópicos.

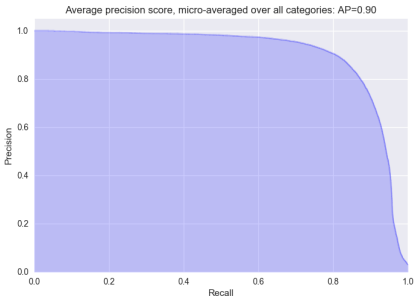

Figura 2. Área sob a Curva *Precision-Recall* (PR-AUC) - *Average Precision* (90%).

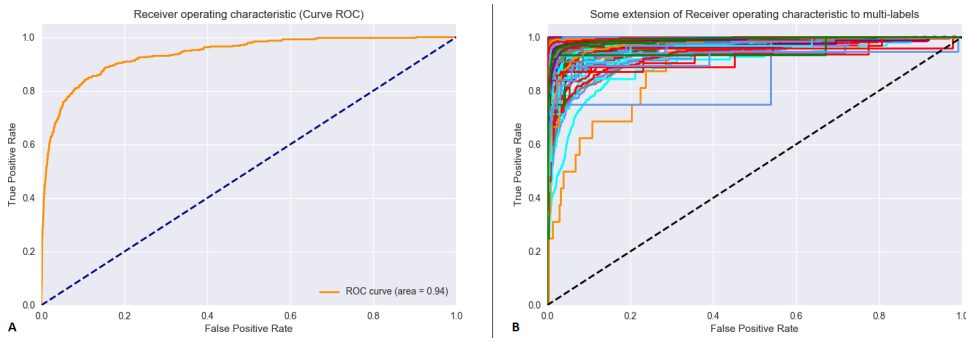

Figura 3. Curva ROC (A) e Curva ROC detalhada por tópico (B) (*Reuters (RCV1)*).

7. Conclusão

Neste trabalho foram avaliadas diversas combinações de classificador e representação de documentos para implementar um agente classificador de notícias por assunto. Para avaliar o desempenho das combinações propostas, foram conduzidos experimentos com a coleção de notícias *Reuters RCV1*. De maneira geral, os melhores resultados foram alcançados pela combinação *SVM* com *TF-IDF*, apresentando vantagem até mesmo sobre combinações utilizando métodos de *Word Embeddings* (como o *Word2Vec* e o `W2VP-IDF` proposto). O modelo de representação de documento, `W2VP-IDF`, foi bem avaliado quando combinado com o classificador: *KNN* para a coleção de documentos *RCV1*. Podemos dizer que essa combinação, tirou o melhor proveito da densidade semântica da técnica *embedding* proposta.

Os resultados aqui obtidos são comparáveis, ou melhores em alguns casos, aqueles obtidos em publicações recentes que utilizaram a mesma coleção de documentos. Por

exemplo, os resultados obtidos para Precision e F1-measure em [Quispe et al. 2017] foram 88,15% e 84,77%, e em [Ke 2012] foram 84,60% e 75,40% (TF-IDF), enquanto neste trabalho obteve-se 91,36% e 82,24%, respectivamente. Isso mostra que essa implementação é competitiva com o estado da arte.

A evolução em andamento deste trabalho está sendo avaliar estratégias mais elaboradas de classificação multirrótulo que considere a correlação entre rótulos e melhorias na representação `W2VP-IDF` proposta, para com isso tirar melhor proveito da densidade semântica de *embeddings* e obter resultados mais expressivos.

Referências

Adhianto, L., Banerjee, S., Fagan, M., Krentel, M., Marin, G., Mellor-Crummey, J., and Tallent, N. R. (2010). Hpctoolkit: Tools for performance analysis of optimized parallel programs. *Concurrency and Computation: Practice and Experience*, 22(6):685–701.

Baeza-Yates, R. and Ribeiro-Neto, B. (2013). *Recuperação de Informação-: Conceitos e Tecnologia das Máquinas de Busca*. Bookman Editora.

Caraciolo, M. P. (2009). Introdução a Árvores de decisão para classificação e mineração de dados. Acessado em 21-01-2018 a url: http://aimotion.blogspot.com.br/2009/04/artigo-introducao-arvores-de-decisao.html.

Duda, R. O., Hart, P. E., and Stork, D. G. (1995). Pattern classification and scene analysis 2nd ed. *ed: Wiley Interscience*.

Gulla, J. A., Zhang, L., Liu, P., Özgöbek, Ö., and Su, X. (2017). The adressa dataset for news recommendation. pages 1042–1048. ACM.

He, Y., Li, J., Song, Y., He, M., and Peng, H. (2018). Time-evolving text classification with deep neural networks. In *IJCAI*, pages 2241–2247.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425.

Ke, W. (2012). Least information document representation for automated text classification. *Proceedings of the American Society for Info. Science - Technology*, 49(1):1–10.

Kim, H. K., Kim, H., and Cho, S. (2017). Bag-of-concepts: Comprehending doc. rep. through clustering words in distr. repres. *Neurocomputing*, 266:336–352.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, pages 361–397.

Liu, M. and Yang, J. (2012). An improvement of tfidf weighting in text categorization. *International proceedings of computer science and information technology*.

Mikolov, T., Chen, K., Corrado, G., Dean, J., Sutskever, L., and Zweig, G. (2014). word2vec.

Quispe, O., Ocsa, A., and Coronado, R. (2017). Latent semantic indexing and convolutional neural network for multi-label and multi-class text classification. In *Computational Intelligence (LA-CCI), 2017 IEEE Latin American Conference on*, pages 1–6. IEEE.

Rei, M. and Cummins, R. (2016). Sentence similarity measures for fine-grained estimation of topical relevance in learner essays. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 283–288.