UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

PRED.ARG: FERRAMENTA PARA GERAR REPRESENTAÇÕES DE DOCUMENTOS COM BASE EM PAPÉIS SEMÂNTICOS

MATHEUS MARZOLA GOMES
ROBERTA AKEMI SINOARA
SOLANGE OLIVEIRA REZENDE

Nº 425

RELATÓRIOS TÉCNICOS

São Carlos – SP
Abr./2018

Pred.Arg: FERRAMENTA PARA GERAR REPRESENTAÇÕES DE DOCUMENTOS COM BASE EM PAPÉIS SEMÂNTICOS

Matheus Marzola Gomes

Roberta Akemi Sinoara

Solange Oliveira Rezende

Departamento de Ciências de Computação
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - Campus de São Carlos
Caixa Postal 668
13560-970, São Carlos, SP
e-mail: {matheus.marzola.gomes, rsinoara}@usp.br, solange@icmc.usp.br

---

Resumo:

Neste relatório técnico é apresentada a ferramenta Pred.Arg, desenvolvida para gerar representações de coleções de documentos que foram propostas por Persson et al. (2009). As representações de Persson et al. (2009) consideradas neste trabalho são geradas a partir de estruturas de predicador e argumentos identificadas e anotadas em textos escritos em língua natural. Essas representações fazem uso de informações sobre os papéis semânticos, visando a obtenção de atributos mais expressivos e, consequentemente, uma representação mais rica do que a *bag of words*. Esse trabalho foi desenvolvido com o objetivo de possibilitar a comparação das representações de Persson et al. (2009) com outras representações em diferentes tarefas de Mineração de Textos, além de disponibilizar a implementação para pesquisas futuras. A ferramenta possibilita a geração de diferentes representações de coleções de documentos. Ela recebe como entrada um conjunto de documentos pré-processados em um padrão pré-definido e gera como saída um arquivo CSV que representa a coleção de documentos.

Palavras-chaves: Mineração de textos, pré-processamento, representação de documentos, papéis semânticos.

Abr./2018

Sumário

1. Introdução

Com o aumento da quantidade de dados textuais disponível em meio computacional, as técnicas de Mineração de Textos (MT) se tornaram importantes para apoiar a extração de conhecimento de grandes coleções de documentos. Neste sentido, tornou-se necessário o desenvolvimento de técnicas, algoritmos e ferramentas que deem suporte à extração de conhecimento de dados não estruturados.

Existem diversas tarefas de Mineração de Textos, dentre elas pode-se citar classificação e agrupamento. Em alguns desses problemas de MT pode ser necessário reconhecer os padrões além das palavras (vocabulário), levando em conta a semântica dos textos. Portanto, a seleção e a preparação dos textos são as primeiras etapas realizadas para a extração de conhecimento de uma coleção de documentos. Como apresentado na Figura 1, o processo de Mineração de Textos pode ser dividido em cinco etapas: Identificação do Problema, Pré-processamento, Extração de Padrões, Pós-processamento e Utilização do Conhecimento (Rezende, 2003).

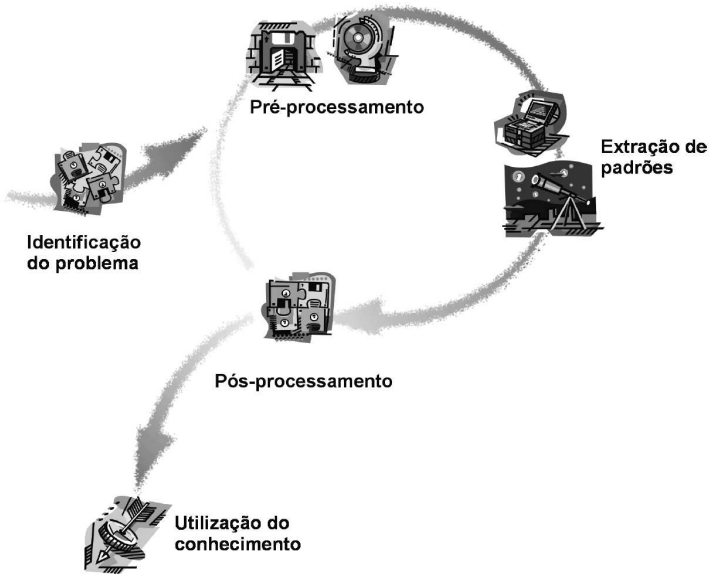

Figura 1: Processo de Mineração de Textos (Rezende, 2003)

O início do processo é a etapa de Identificação do Problema. Nessa etapa, um especialista em Mineração de Textos, juntamente com um especialista do domínio, tomam decisões acerca das coleções de textos utilizadas, definindo os objetivos e a utilidade dos resultados. A representação das coleções textuais se dá na etapa de Pré-processamento. Essa etapa busca preparar os dados de entrada para representá-los adequadamente a fim de colocá-los em um formato aceito pelos algoritmos utilizados na etapa de Extração de Padrões. Muitas técnicas têm sido propostas e desenvolvidas na área, porém, algumas apresentam limitações quando a semântica do texto tem grande importância para a tarefa de mineração, sendo necessário buscar outras alternativas (Sinoara et al., 2017).

Após resolver as duas etapas anteriores, é possível realizar a etapa de Extração de Padrões. Nessa etapa, um algoritmo de aprendizado de máquina será utilizado para realizar a tarefa desejada e extrair os padrões nos dados que já estão pré-processados. Assim que os padrões são obtidos, a etapa de Pós-processamento é realizada com o objetivo de avaliar o conhecimento descoberto segundo os objetivos traçados nas primeiras etapas de Mineração de Textos. Por fim, a etapa de Utilização de Conhecimento possibilita que os usuários utilizem o conhecimento extraído quando este atende os objetivos estabelecidos anteriormente.

O resultado do processo de Mineração de Textos depende da qualidade das decisões tomadas nas primeiras etapas do processo. Na etapa de pré-processamento, o conjunto de textos pode sofrer diversos tipos de tratamento. Um desses tratamentos é a remoção de *stopwords*, que remove palavras que prejudicam o desempenho de algoritmos de extração de padrões. Outra atividade importante na etapa de pré-processamento é a representação e estruturação dos documentos. Um dos métodos mais tradicionais para representação dos documentos é a *bag of words* (BOW). Nesse método os documentos são representados por uma matriz documento-termo, na qual os documentos são dispostos em linhas e os termos (palavras) em colunas. Cada célula da matriz possui o valor correspondente ao peso do termo para o documento especificado. Normalmente esse peso é dado pela frequência do termo no documento.

A BOW tem bom desempenho na classificação ou agrupamento dos documentos em tópicos específicos, porém, não permite a incorporação de características semânticas presentes nos textos (Sinoara et al., 2017). É possível observar essa limitação com duas sentenças simples. Por exemplo, considere os seguintes documentos:

- **D1**: "João matou o bandido"
- **D2**: "O bandido matou o João"

A representação desses documentos utilizando uma BOW é apresentada na Figura 2. Utilizando uma *bag of words* as representações dos documentos são idênticas, porém, as sentenças têm sentidos opostos. Essa limitação da representação BOW pode ter grande impacto em problemas que requerem a diferenciação dos documentos além do vocabulário.

	João	Matou	Bandido
Documento 1	1	1	1
Documento 2	1	1	1

Figura 2: Representação dos documentos D1 e D2 utilizando a *bag of words*

Portanto, um importante desafio no processo de Mineração de Textos é encontrar uma representação de documentos que leve aos resultados esperados de acordo com a necessidade

da tarefa. Assim, outros métodos de representação dos documentos surgiram visando solucionar problemas envolvendo a semântica. Um exemplo para representação textual é o *Latent Dirichlet Allocation* (Blei et al., 2003) que é uma alternativa baseada na extração de tópicos. Tal método representa a semântica dos documentos de maneira latente e, ou mesmo tempo, também atua como um método de redução de dimensionalidade. Considerando a semântica de maneira mais explícita, tem-se como exemplo os modelos propostos por Persson et al. (2009). Para representar os documentos, Persson et al. (2009) utiliza termos desambiguados e informações de papéis semânticos.

Nesse contexto, desenvolveu-se a ferramenta PRED.ARG, apresentada nesse relatório. O objetivo dessa ferramenta é possibilitar a comparação das representações de Persson et al. (2009) com outras representações, em diferentes tarefas de Mineração de Textos, além de disponibilizar a implementação para pesquisas futuras. Tal ferramenta utiliza um conjunto de documentos desambiguados e com as estruturas de predicador e argumentos anotadas. Como saída, é gerado uma matriz documento-termo no formato CSV.

O restante deste relatório técnico está organizado da seguinte maneira. Na Seção 2 os conceitos envolvendo pré-processamento e papéis semânticos são brevemente explicados. Nessa seção, também são apresentadas as representações propostas por Persson et al. (2009). Na Seção 3 são apresentadas as funcionalidades da ferramenta e os padrões de entrada e saída. Um exemplo de uso e uma breve avaliação das representações geradas são apresentados na Seção 4. Por fim, na Seção 5 são apresentadas as considerações finais deste trabalho.

2. Representações com base em papéis semânticos

Tomando como base o modelo *bag of words* por ser o modelo mais tradicional, observa-se uma extração da frequência de palavras isoladas, utilizadas como atributos da representação no modelo espaço-vetorial. Em contraste, Persson et al. (2009) propõem o uso de atributos mais ricos. Para isso, os autores fazem uso de diferentes tarefas de Processamento de Língua Natural, para fazer anotações nos textos. A primeira é a desambiguação lexical de sentidos, na qual as palavras são desambiguadas e anotadas com o sentido específico. A segunda é a anotação sintática, obtendo-se anotações de sujeito, verbo e objeto, chamadas de *VSO triples*. A terceira é a anotação de papéis semânticos, na qual são identificadas e rotuladas as estruturas de predicador e argumentos. Na ferramenta apresentada nesse relatório técnico, foram consideradas somente as propostas que fazem uso de desambiguação e anotação de papéis semânticos.

Para entender a proposta de Persson et al. (2009), alguns conceitos devem ser apresentados. Na anotação de papéis semânticos, cada sentença é processada visando a identificação de dois elementos. O primeiro é o "predicador" que, na maioria das vezes, são verbos. O segundo elemento são os argumentos, que são os termos que acompanham o verbo, podendo ser sujeito, objeto ou outro elemento de sintaxe. Os argumentos são rotulados de acordo com a sua relação com o verbo, ou seja, de acordo com o papel que eles exercem na sentença. Na notação do PropBank[1] (Palmer et al., 2005), os argumentos podem receber rótulos numerados de Arg0 a Arg5.

Para ilustrar esse processo, Persson et al. (2009) apresenta o exemplo da Figura 3. Nesse exemplo, é utilizada a sentença *"Chrysler plans new investment in Latin America"* para demonstrar o processo de desambiguação e anotação dos papéis semânticos. A primeira etapa importante é identificar o predicador que, nesse caso, são dois: *plans* e *investment*. No inglês, essas duas palavras podem ser usadas em contextos diferentes, portanto ocorre um trabalho de desambiguação, que é importante para a estrutura de papéis semânticos proposta pelo autor. Após a desambiguação, os argumentos são extraídos de acordo com seus respectivos predicadores, recebendo um rótulo (Arg0, Arg1 ou Arg2), finalizando o processo de anotação dos papéis semânticos.

Com base nessas anotações, além da anotação sintática, seguintes conjuntos de atributos são propostos por Persson et al. (2009).

- *Predicates* (código 010000): nesse conjunto cada atributo é formado pelo predicador desambiguado. O predicador é numerado de acordo com o seu sentido (*sense*), ou seja, um verbo pode assumir mais de um sentido, sendo necessário a identificação, concatenando com um *label* (verbos sem *sense* identificado são ignorados).
- *VSO triples*(código 001000): nesse conjunto os atributos são baseados na estrutura

[1]PropBank: `http://verbs.colorado.edu/~mpalmer/projects/ace.html`

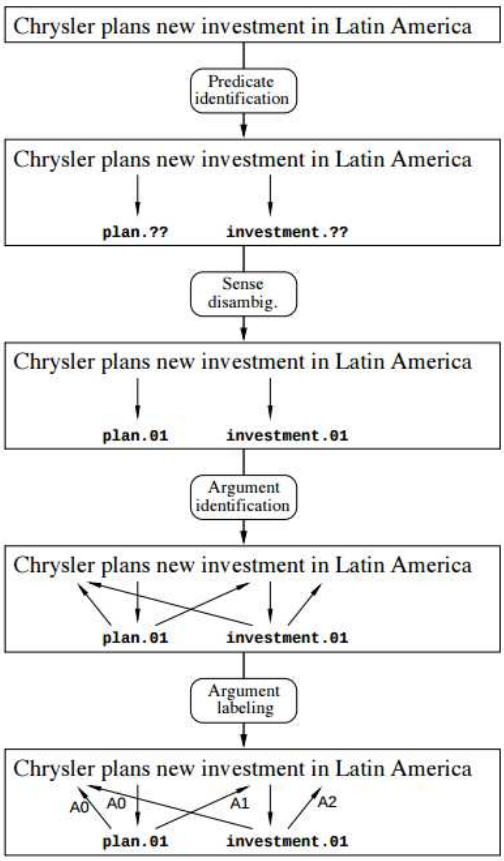

Figura 3: Processo de identificação e anotação de estrutura de predicador e argumentos (Persson et al., 2009)

"sujeito-verbo-objeto", no qual para cada verbo são extraídos o principal objeto e sujeito da sentença caso eles existam, seguindo o padrão verbo#sujeito#objeto.

- *Argument 0*(código 000100): nesse conjunto cada atributo é a concatenação do predicador com o seu respectivo *sense* e o argumento 0 identificado na sentença.
- *Argument 1*(código 000010): nesse conjunto cada atributo segue o mesmo padrão do anterior, porém, com o argumento 1.
- *Arguments 0 and 1*(código 000001): nesse conjunto cada atributo consiste no predicador concatenado com o seu *sense*, argumento 0 e argumento 1. Segue o padrão "verbo.*sense*#argumento0#argumento1".

Além desses conjuntos de atributos, Persson et al. (2009) também consideraram a tradicional *bag of words* (código 100000). Cada conjunto recebeu um código específico possibilitando a identificação de combinações (concatenações) dos atributos, totalizando 64 possíveis combinações.

Nas avaliações experimentais de Persson et al. (2009), os testes foram feitos com os 64 conjuntos de atributos. De acordo com Persson et al. (2009) o uso de termos baseados

em estruturas de predicador e argumentos demonstrou uma melhora significativa quando combinados com a *bag-of-words*, gerando bons resultados em trabalhos de classificação e confirmando ser uma alternativa viável para representação de documentos na etapa de pré-processamento.

A ferramenta PRED.ARG, descrita neste relatório, possibilita a geração de 4 conjuntos de atributos formados por estruturas de predicador e argumentos que foram propostos por Persson et al. (2009). Além disso, a ferramenta também possibilita a geração de representações combinando um desses conjuntos de atributos com a *bag of words*, totalizando, assim, 8 opções diferentes de modelos de representação. Na próxima seção a ferramenta PRED.ARG é descrita, apresentando uma visão geral do funcionamento e detalhes das entradas e saídas.

3. Ferramenta PRED.ARG

Nessa seção é apresentada uma visão geral da ferramenta PRED.ARG, que foi desenvolvida para gerar os modelos propostos por Persson et al. (2009). São apresentadas as bibliotecas utilizadas e alguns padrões técnicos necessários para execução do programa. Um exemplo de uso por linha de comando é introduzido, assim como os argumentos de entrada para a execução. Um padrão de arquivos de entrada deve ser utilizado e são demonstrados na Seção 3.2, em conjunto com os arquivos de saída que são gerados na execução da ferramenta.

3.1 Visão Geral da ferramenta

Uma visão geral da ferramenta PRED.ARG é apresentada na Figura 4. O principal objetivo da ferramenta é possibilitar a construção de modelos propostos por Persson et al. (2009). Cabe ao usuário escolher a representação que deseja, tendo 4 opções disponíveis:

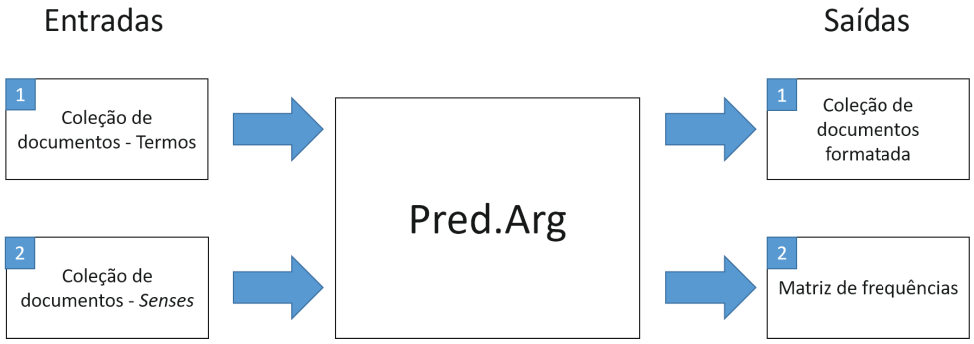

Figura 4: Entradas e saídas da ferramenta PRED.ARG

- *predicateA0*;
- *predicateA1*;
- *predicate*;
- *predicate+predicateA1*.

Também é possível gerar representações combinando um desses conjuntos com uma *bag of words*. Na Tabela 1 são apresentadas as representações geradas pela PRED.ARG, juntamente com os códigos dessas representações de acordo com Persson et al. (2009).

Conforme apresentado na Figura 4, a ferramenta recebe como entrada a coleção de documentos em dois formatos: *senses* e estruturas de predicador e argumentos. Assim, é necessário pré-processar a coleção de documentos, para realizar a desambiguação lexical de sentidos e a anotação de papéis semânticos. Como resultado da desambiguação, obtém-se os *senses* presentes nos documentos. Já com a anotação de papéis semânticos

Tabela 1: Conjuntos gerados pela ferramenta e seus respectivos códigos, segundo Persson et al. (2009)

Conjunto	Código
BOW+predicateA0	100100
BOW+predicateA1	100010
BOW+predicate	110000
BOW+predicate+predicateA1	110010
predicateA0	000100
predicateA1	000010
predicate	010000
predicate+predicateA1	010010

são identificados os predicadores e seus argumentos (chamados de termos neste relatório). Os termos (Entrada 1 da Figura 4) e os *senses* dos textos (Entrada 2 da Figura 4) devem ser fornecidos no formato apresentado a seguir.

- Estrutura de predicador e argumentos: `[V]_teve_[A0]_Hakkinen_[A1]_pneu`
- *Senses*: `teve;bn:00089242v`

Para exemplificar esses formatos, considere o seguinte texto, extraído do documento "ESP_FORMULA_1.20040509.147.txt" da coleção *BEST sports*[2] (Sinoara & Rezende, 2018).

D1:*"Schumacher vence a quinta seguida e iguala o recorde de Nigel Mansell. Como previsto, o alemão Michael Schumacher venceu neste domingo o GP da Espanha de Fórmula 1, disputado em Barcelona. Schumacher largou na pole position com sua Ferrari mas foi surpreendido pela Renault do italiano Jarno Trulli, que pulou da quarta colocação no grid para a liderança na largada."*

Os arquivos pré-processados dos termos e *senses* para o documento **D1** são apresentados, respectivamente, nas Figuras 5 e 6. Detalhes sobre a estrutura de pastas na qual esses documentos devem ser organizados são apresentados nas próximas subseções.

```
[V]_vence_[A0]_Schumacher_[A1]_quinta
[V]_iguala_[A0]_Schumacher_[A1]_recorde_Nigel_Mansell
[V]_venceu_[A0]_alemao_Michael_Schumacher_[A1]_GP_Espanha_Formula_1_Barcelona
[V]_largou_[A0]_Schumacher_[A1]_pole
[V]_surpreendido_[A0]_Renault_italiano_Jarno_Trulli_que_colocacao_grid_lideranca_largada
[V]_pulou_[A0]_que_[A1]_colocacao
```

Figura 5: Conteúdo de um arquivo de estruturas de predicador e argumentos

A ferramenta PRED.ARG foi desenvolvida em Python e faz uso das seguintes bibliotecas para auxiliar em algumas tarefas:

[2]Coleção *BEST sports*: `http://sites.labic.icmc.usp.br/rsinoara/bestsports/`

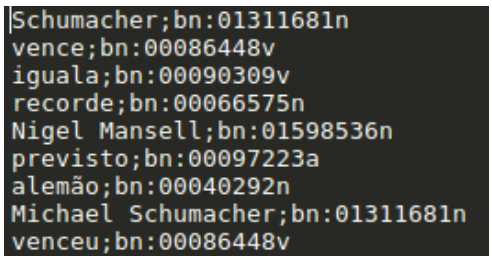

Figura 6: Conteúdo de um arquivo de *senses*

- re - Bilbioteca para uso de expressões regulares.
- argparse - Biblioteca para organização e estruturação de argumentos de entrada.
- sklearn[3] - Biblioteca de Aprendizado de Máquina de código aberto para Python. Scikit-learn é a principal biblioteca utilizada no desenvolvimento da PRED.ARG devido aos diversos algoritmos de Aprendizado de Máquina disponíveis em código aberto. Para contagem de frequência foram utilizados os padrões TF e TF-IDF.

3.2 Utilização da ferramenta PRED.ARG

Nessa seção são apresentados os argumentos de entrada para funcionamento da ferramenta PRED.ARG, assim como a estrutura em que os arquivos de entrada devem estar organizados. Por fim, um exemplo de saída é demonstrado de acordo com o documento **D1** da Seção 3.1.

3.2.1 Argumentos de entrada

O programa é executado por meio de linha de comando no terminal, contando com os seguintes argumentos de entrada.

- –**terms:** Nome da pasta contendo os arquivos com os termos (predicadores e argumentos no padrão apresentado na Figura 5).
- –**senses:** Nome da pasta contendo os arquivos com os *senses* no padrão apresentado na Figura 6.
- –**result:** Nome da pasta para salvar os arquivos de saída.
- –**freq:** Medida utilizada para calcular os pesos dos termos ("TF" ou "TF-IDF").
- **-id:** ID da representação ("0" para *predicate+A0*, "1" para *predicate+A1*, "2" para *predicate* e "3" para concatenação de "1" e "2").

[3]Scikit-learn: `http://scikit-learn.org/stable/`

- **–bow:** Argumento opcional, nome do arquivo contendo uma *bag of words* no formato CSV, que será concatenada com o conjunto de atributos gerados a partir das estruturas de predicadores e argumentos.

A seguir é apresentado um exemplo de utilização.

```
python3 PredArg.py --terms \Terms --senses \Senses --result \Result
--freq tf -id 0 --bow arquivo_bow.csv
```

3.2.2 Estrutura de diretórios dos arquivos de entrada

Alguns padrões são necessários para execução do programa. As pastas de entrada (termos e *senses*) devem conter subpastas com a identificação das classes (rótulos) dos documentos. Dentro dessas subpastas, os arquivos devem apresentar os termos e *senses* conforme o formato exemplificado na Seção 3.1 e separados por uma quebra de linha, de modo a seguir o padrão das Figuras 5 e 6. A seguir, a Figura 7 exemplifica a organização das subpastas.

Figura 7: Subpastas correspondente às classes dos documentos

3.2.3 Arquivos de saída

São criados dois tipos de saída. Um deles é a pasta principal "Resultados" (Saída 1 da Figura 4), contendo as mesmas subpastas e arquivos dos termos e *senses*. Dentro desses arquivos, são salvos os atributos dos documentos no formato padrão de Persson et al. (2009), conforme exemplificado na Figura 8.

```
vence.bn:00086448v#Schumacher
iguala.bn:00090309v#Schumacher
venceu.bn:00086448v#alemao_Michael_Schumacher
largou.bn:00087364v#Schumacher
surpreendido.bn:00083054v#Renault_italiano_Jarno_Trulli_que_colocacao_grid_lideranca_largada
```

Figura 8: Conteúdo dos arquivos de saída no padrão de (Persson et al., 2009)

A segunda saída é a função principal da ferramenta: criar a matriz que representa a coleção de documentos (Saída 2 da Figura 4). Na mesma pasta onde o programa foi executado,

será criado um arquivo CSV contendo os atributos e as suas frequências. O nome desse arquivo é dado de acordo com a medida de frequência utilizada e a concatenação ou não com a *bag of words*. Por exemplo, caso seja utilizado a medida TF e a concatenação com a *bag of words*, o nome do arquivo gerado será "frequency_tf+bow".

4. Exemplo de uso

Para exemplificar o uso da ferramenta, foram utilizadas duas coleções de textos com 3 configurações diferentes (*datasets*) cada.

A primeira coleção de textos é chamada de *Best Sports*[4] (BS) (Sinoara et al., 2016). Essa coleção é composta por 283 documentos de esportes em português separados em pastas divididas em suas categorias, cada um sendo um esporte. Essa base tem 3 *datasets* para problemas de classificação: *semantic*, *topic* e *topic-semantic*. A *semantic* classifica desempenho (vitória ou derrota) de atletas brasileiros, a *topic* classifica o esporte e, por fim, a *topic-semantic* utiliza as duas formas de classificação.

A segunda coleção de textos é chamada de *SemEval-2015 Aspect Based Sentiment Analysis* (SE) (Pontiki et al., 2015). Essa coleção é composta por documentos de opiniões sobre hotéis, restaurantes e *laptops*. Em Sinoara et al. (2016) foram construídos 3 datasets com essa coleção: *polarity* (polaridade), *product* (produto) e *product-polarity* (produto-polaridade). A *polarity* classifica a polaridade da avaliação do produto (positivo, negativo ou neutro), a *product* classifica o tipo do produto e a *product-polarity* junta as últimas duas maneiras de classificação.

A ferramenta foi executada para essas duas coleções, nas 4 opções (conjuntos de atributos) disponíveis no programa. Para cada uma, foi gerado o arquivo CSV final contendo a tabela de frequências de acordo com o padrão selecionado.

Após gerar as representações de cada *dataset*, ou seja, os arquivos de frequência, foram executados alguns experimentos de classificação para exemplificar a utilidade da ferramenta. Os experimentos foram executados por meio da ferramenta Text Categorization (Rossi, 2016), utilizando os seguintes algoritmos de classificação.

- IMBHN^{C} e IMBHN^{R}. Foram utilizadas as taxas de correção de 0,01, 0,05, 0,1, 0,5, com o número máximo de iterações em 1000 e o erro dos mínimos quadrados com critério de parada de 0,01.
- J48. O parâmetro *confidence factor* foi ajustado para 0,25, 0,2 e 0,15.
- KNN (*K-nearest neighbor*). O parâmetro k utilizou os seguintes valores: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 25, 35, 45, 55. O algoritmo rodou com as distâncias euclidiana e cosseno, também com a opção de voto com peso ou sem peso.
- NB (*Naive Bayes*).
- MNB (*Multinomial Naive Bayes*).

[4] `http://bestsports.com.br/db/notarqhome.php`

- SMO (*Sequential Minimal Optimization*). Nesse algoritmo foram consideramos três tipos de kernel: linear, polinomial (expoente=2) e RBF (*Radial Basis Function*). Os valores considerados para cada tipo de kernel foram 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0, 1, 10, 10^2, 10^3, 10^4, 10^5. .

Os melhores resultados para cada *dataset*, considerando-se todos os algoritmos e variações de parâmetros, são apresentados na Tabela 2. Nessa tabela, o melhor resultado para cada *dataset* é apresentado em negrito. É possível extrair algumas conclusões da utilidade das representações utilizando papéis semânticos propostas por Persson et al. (2009). A execução sem a concatenação com a *bag of words* demonstrou resultados muito abaixo dos resultados obtidos com a *bag of words*, indicando que o uso dessas representações obtém melhores resultados de modo mais preciso ao se concatenar com a *bag of words*. Esse fato está de acordo com os resultados obtidos por Persson et al. (2009), visto que os melhores resultados apresentados pelos autores referem-se a representações que apresentam a combinação dos atributos da *bag of words* com outros atributos enriquecidos.

Tabela 2: Melhores valores de acurácia obtidos na avaliação experimental

Dataset	BS-semantic	BS-topic	BS-topic-semantic	SE-polarity	SE-product	SE-product-polarity
bow	68.9532	**100.0000**	66.8596	84.5438	**99.5077**	**83.6811**
predicate	54.0640	73.8300	40.6158	72.3848	78.7699	57.1725
predicate+bow	**69.3103**	**100.0000**	**67.5862**	84.9067	99.1373	82.2011
predicate+predicateA1	53.0788	76.6133	41.3424	71.8970	76.4363	57.7853
predicate+predicateA1+bow	68.9532	**100.0000**	64.6921	83.9235	99.0139	82.1966
predicateA0	53.0296	77.4015	48.0788	70.6775	63.9175	49.5754
predicateA0+bow	68.2389	**100.0000**	65.3818	**85.0331**	99.3842	81.9527
predicateA1	45.1847	68.4729	34.9261	67.9750	59.9985	41.9587
predicateA1+bow	68.9532	**100.0000**	65.7389	84.6582	99.1388	82.1951

No geral, as opções *"predicate+bow"* e *"bow"* atingiram o maior número de melhores resultados. Variando a representação adotada, o algoritmo MNB destaca-se dentre todos com melhores resultados nos *datasets BS-topic*, *SE-polarity*, *SE-product* e *SE-product-polarity*. No caso do *BS-topic-semantic* e *BS-semantic*, o KNN e IMBHN$^{\mathrm{R}}$ se sobressaem nos resultados, respectivamente.

Portanto, é possível observar que em alguns trabalhos de classificação em que a semântica tem importância, o padrão predicador-argumentos pode atingir bons resultados quando utilizado em conjunto com uma *bag of words*.

5. Considerações Finais

Na Mineração de Textos, diversas técnicas de representação textual têm sido desenvolvidas com o objetivo de tornar possível a extração de conhecimento de coleções de documentos. Nesse processo, a semântica dos textos pode ser de grande relevância para tarefas de classificação ou agrupamento, sendo importante considerar além das palavras de maneira independente, como na tradicional representação *bag of words*.

Em Persson et al. (2009), são utilizadas informações obtidas por meio da anotação de papéis semânticos para melhorar a representação de documentos. O objetivo da anotação de papéis semânticos é atribuir uma identificação (ou rótulo) para os elementos das sentenças dos documentos, permitindo a identificação de informações sobre os eventos relatados, como, por exemplo, quem fez e quem recebeu a ação. Persson et al. (2009) apresentam a proposta de modelos para representação de documentos considerando os papéis que os termos apresentam nas sentenças de cada documento.

Nesse Relatório Técnico foi apresentada a ferramenta Pred.Arg, que foi desenvolvida com objetivo de possibilitar a comparação das representações propostas por Persson et al. (2009) com outras representações em diferentes tarefas de Mineração de Textos. A fim de disponibilizar a implementação para pesquisas futuras, a Pred.Arg foi disponibilizada no site do Laboratório de Inteligência Computacional (LABIC) em `http://labic.icmc.usp.br/material/16`.

[§]Esse trabalho foi desenvolvido com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da FAPESP (processo nº 2013/14757-6, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP)). As opiniões, hipóteses e conclusões ou recomendações expressas neste material são de responsabilidade dos autores e não necessariamente refletem a visão do CNPq ou da FAPESP.

Referências

Blei, D. M., A. Y. Ng, & M. I. Jordan (2003). Latent dirichlet allocation. *Journal of Machine Learning Research 3*(Jan), 993–1022.

Palmer, M., D. Gildea, & P. Kingsbury (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics 31*(1), 71–106.

Persson, J., R. Johansson, & P. Nugues (2009). Text categorization using predicate-argument structures. In *NODALIDA 2009: Proceedings of the 17th Nordic Conference of Computational Linguistics*, pp. 142–149.

Pontiki, M., D. Galanis, H. Papageorgiou, S. Manandhar, & I. Androutsopoulos (2015). Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval 2015: Proceedings of the 9th International Workshop on Semantic Evaluatio*, pp. 486–495.

Rezende, S. O. (Ed.) (2003). *Sistemas Inteligentes: Fundamentos e Aplicações*. Editora Manole.

Rossi, R. G. (2016). *Classificação automática de textos por meio de aprendizado de máquina baseado em redes*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Sinoara, R. A. & S. O. Rezende (2018). BEST sports: a portuguese collection of documents for semantics-concerned text mining research. Relatório Técnico 424, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

Sinoara, R. A., R. G. Rossi, & S. O. Rezende (2016). Semantic role-based representations in text classification. In *ICPR 2016: Proceedings of the 23rd International Conference on Pattern Recognition*, pp. 2314–2319.

Sinoara, R. A., R. B. Scheicher, & S. O. Rezende (2017). Evaluation of latent dirichlet allocation for document organization in different levels of semantic complexity. In *CIDM'17: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, pp. 2057–2064.