Uma Metodologia para Tratamento do Viés da Maioria em Modelos de Stacking via Identificação de Documentos Difíceis

Welton Santos[1], Washington Cunha[1], Celso França[1], Guilherme Fonseca[2] Sergio Canuto[1], Leonardo Rocha[2], Marcos Gonçalves[1]

[1]Universidade Federal de Minas Gerais

[2]Universidade Federal de São João del Rei

{weltonsantos,washingtoncunha,celsofranca,sergiodaniel,mgoncalv}@dcc.ufmg.br

{guilhermefonseca8426,lcrocha}@ufsj.edu.br

Abstract. *Stacking models are effective in automatic document classification by exploring model complementarity. Despite this, there are still situations of failure in the classification of some documents, named here as difficult documents, due to a bias in which most of the learned models point to a class different from the real one. This work presents a first proposal, consisting of two steps, aimed at overcoming failures due to majority bias. First, we train a difficult document detector. Next, we use the detector to direct difficult documents to a meta-classifier specialized in classifying such documents. Empirically, our approach shows promise in isolating the majority bias.*

Resumo. *Modelos de stacking são efetivos na tarefa de classificação automática de documentos explorando a complementariedade entre modelos. Contudo, ainda há situações de falha na classificação de alguns documentos, denominados aqui como documentos difíceis, devido a um viés em que a maioria dos modelos aprendidos apontam para uma classe diferente da real. Este trabalho apresenta uma primeira proposta, composta de dois passos, que visa contornar falhas por viés da maioria. Primeiro, treinamos um detector de documentos difíceis, para depois utilizar o detector para direcionar documentos difíceis para um meta-classificador especialista em tais documentos. Empiricamente, nossa abordagem se mostra promissora no isolamento do viés da maioria.*

1. Introdução

Em tarefas de classificação automática, *stacking* é uma abordagem que combina diferentes modelos de classificação, aprendidos por classificadores distintos (i.e. classificadores base), em um único modelo mais robusto (i.e meta-classificador). A ideia principal é explorar a complementaridade e diversidade do processo de aprendizagem entre os classificadores para superar as deficiências individuais dos modelos base. Modelos de *stacking* têm se mostrado efetivos na tarefa de classificação automática de documentos [Ding and Wu 2020, Gomes et al. 2021], cenário foco deste trabalho. Apesar disso, esses modelos ainda são suscetíveis a falhas quando a maioria dos modelos base falham, comportamento definido aqui como *viés da maioria*.

Para ilustrar esse viés, executamos um modelo de *stacking* para duas base de dados ACM (artigos científicos de Ciência da Computação) e 20NG (notícias de jornais).

Para cada documento erroneamente classificado pelo meta-classificador (*stacker*) contabilizamos quantos modelos base falharam neste documento. Na Figura 1 apresentamos a distribuição de documentos classificados erroneamente pelo meta-classificador pela *Maioria*. Observa-se que as falhas do meta-classificador estão mais concentradas em documentos onde a maioria dos modelos base falham. Mais especificamente, 58% e 50% das falhas do meta-classificador ocorrem quando quatro ou mais (de **7**) modelos base falham para as bases de dados ACM e 20NG, respectivamente. Neste trabalho, denominamos esses documentos em que a maioria dos modelos falha na predição de **documentos difíceis**.

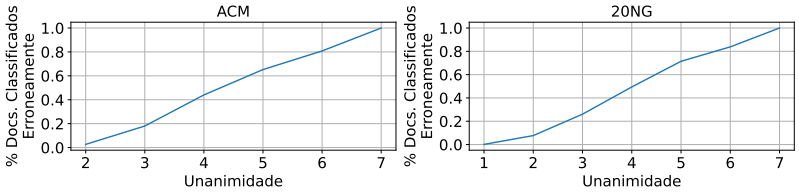

Figura 1. Fração acumulada de documentos classificados erroneamente pelo meta-classificador (eixo y) pela quantidade de modelos base que falharam nos documentos em que o meta-classificador errou (eixo x: *Maioria*).

Vale ressaltar que a detecção de documentos difíceis é benéfica não apenas para modelos de *Stacking*, tem o potencial de aprimorar classificadores base a partir da análise minuciosa das razões de falha desses modelos. Nesse contexto, uma forma de tratar o problema do viés da maioria é desenvolver meta-classificadores especializados em documentos difíceis. Nesse caso, um problema preliminar é identificar, em tempo de classificação (teste), quais são estes documentos. Neste trabalho apresentamos um *pipeline*, composto de duas etapas, para classificar documentos difíceis em modelos *stacking*.

Na primeira etapa, introduzimos uma estratégia para detectar documentos difíceis. Para isso, propomos um conjunto de *meta-features* que caracterizam os modelos de classificação em relação às suas predições, aplicadas no conjunto de treinamento, tanto para os documentos difíceis quanto os demais. Essas informações são então utilizadas em um processo de treinamento de um modelo capaz de distinguir os documentos difíceis dos "*fáceis*". Na segunda etapa, os documentos difíceis são direcionados a um meta-classificador especialista, enquanto os demais são direcionados ao meta-classificador tradicional. Avaliamos o detector de documentos difíceis considerando duas coleções de dados distintas, ACM e 20NG. Nossos resultados demonstraram a efetividade do detector proposto, abrindo espaço para futuros trabalhos nesta frente.

2. Trabalho Relacionados

Estratégias de *Stacking* combinam a informação de múltiplos classificadores heterogêneos (e.g., SVMs, redes neurais, árvores de decisão) para formar modelos mais efetivos [Džeroski and Ženko 2004]. Esta estratégia se beneficia da complementaridade entre modelos aprendidos por diferentes classificadores para superar as possíveis deficiência de classificadores individuais. Em classificação automática de documentos, as propostas de *stacking* mais consolidadas são compostas pela integração entre modelos tradicionais de aprendizado de máquina (kNN, regressão logística) com outros modelos, tais como *Random Forest* e SVM[Wahba et al. 2022, Singhal and Kashef 2023]. Recentemente, com o crescente interesse em modelos baseados em redes neurais profundas, os mesmos vêm sendo utilizados em trabalhos atuais em propostas de *stackings* híbridos [Chowanda and Muliono 2022, Subba and Kumari 2022, Gomes et al. 2021].

Dos trabalhos citados, destacamos a proposta [Gomes et al. 2021], na qual os autores apresentam modelos de *stacking* robustos e diversificados combinando classificadores neurais e não neurais com múltiplas estratégias de representação de documentos. Apesar dos bons resultados, os autores argumentam que há espaço para melhorias. No estudo, os autores destacam que há um número relevante de documentos para os quais a maioria dos modelos apontam para uma classe que não é a real desses documentos. Denominamos o problema relatado de *viés de maioria*, e nosso objetivo é apresentar e avaliar uma primeira solução que visa contorná-lo. Por fim, vale ressaltar o trabalho [Penha et al. 2019] que tentar predizer a performance de um classificador individual, uma tarefa relacionada, mas diferente da proposta aqui.

3. Proposta

Na Figura 2 apresentamos a visão geral da nossa abordagem, basicamente composta de duas etapas. Na primeira etapa, introduzimos uma estratégia para detectar documentos difíceis. Para isso, propomos um conjunto de *meta-features* que caracterizam os modelos de classificação em relação às suas predições, aplicadas no conjunto de treinamento, tanto para os documentos difíceis quanto os demais. No treinamento, os documentos são tidos como difíceis se a classe correta do mesmo não é predita por mais de 50% dos modelos que compõem o *stacking*. Essas informações são então utilizadas em um processo de treinamento de um modelo capaz de distinguir os documentos difíceis dos demais.

Na segunda etapa, os documentos de teste passam pelo detector o qual define quais deles são considerados difíceis, que por sua vez são direcionados a um meta-classificador especialista enquanto os demais são direcionados ao meta-classificador tradicional. Na Seção 3.1 detalhamos as meta-features propostas e como as mesmas são utilizadas na construção do detector de documentos difíceis. Na Seção 3.2 descrevemos nossa proposta para construção do meta-classificador especialista para documentos difíceis (que não é o foco deste artigo).

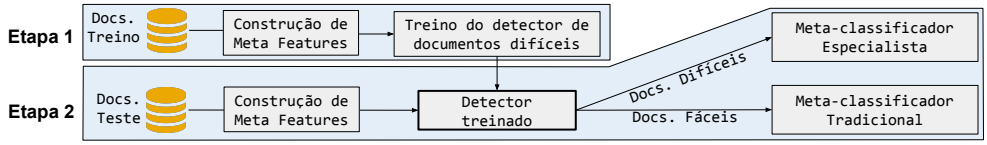

Figura 2. Visão geral da proposta.

3.1. Detector de Documentos Difíceis

O primeiro passo de nossa metodologia é realizar uma experimentação cruzada de k partes (*folds*) no conjunto de treinamento, em que $k-1$ *folds* são utilizados para criar os modelos base do *stacking*, bem como o meta-classificador do *stacking*, e a parte restante (conjunto de validação) usada para ser classificada pelo *stacking* gerado. Esse processo é repetido k vezes até que todos os documentos do treinamento tenham sido, em algum momento, classificados por um *stacker*. A partir disso, todos os documentos do treino são separados em duas classes: 1) difíceis: documentos para os quais mais de 50% dos classificadores do *stacking* erraram sua classe correta; 2) fáceis: os demais documentos.

Uma vez identificados os documentos difíceis no treinamento, para treinar o modelo de detecção de documentos desses documentos propomos um conjunto de sete *meta-features* baseadas nas predições de cada um dos modelos base do *stacking* descritos ante-

riormente[1]. São elas: (i) **Tamanho da maior concordância (TC)** – maior quantidade de classificadores que assinalaram a mesma classe; (ii) **Divergência (Div)** – atributo binário que indica se o classificador diverge ou não da maior concordância; (iii) **Número de classes (NC)** – conjunto de classes únicas preditas pelos modelos base para um documento; (iv) **Entropia (EP)** – entropia da distribuição de classes preditas para um documento; (v) **Confiança (Conf)** – a probabilidade do classificador sobre a predição de um documento; (vi) **Taxa de acerto da classe (AC)** – razão entre a quantidade de predições corretas de um modelo para a quantidade de predições da classe; e (vii) **Peso da classe (PC)** – razão entre a quantidade de documentos da classe pelo total de documentos da base.

Para assegurar que as probabilidades dos modelos base não destoam da distribuição de probabilidade real para os documentos, propomos também utilizar a calibragem dos algoritmos tradicionais com o método *Isotonic* [Niculescu-Mizil and Caruana 2005] e os modelos neurais com o método *Temperature Scaling* [Desai and Durrett 2020]. Por fim, definidas as classes de cada documento e construídas as *meta-features*, utilizamos um classificador *Random Forest* para a geração do modelo de detecção de documentos difíceis.

3.2. Meta-Classificador Especialista

Para tratar problemas relativos ao alto desbalanceamento e ruído dos documentos difíceis, treinamos um meta-classificador especialista em uma amostra balanceada de documentos difíceis e fáceis. Nosso objetivo é especializar o meta-classificador nestes documentos e isolar o viés dos documentos fáceis, muitos mais numerosos, sobre o classificador. Note que esse balanceamento é feito apenas no treino. Na fase de teste do modelo, os *folds* permanecem com a distribuição original da base de dados, que é bastante enviesada para os documentos fáceis. Note que nosso foco neste artigo é detectar documentos difíceis. Deixamos a avaliação do meta-classificador especialista para trabalhos futuros.

4. Experimentos e Avaliações

4.1. Setup

Como base para os modelos de *stacking*, buscamos inspiração em [Gomes et al. 2021]. Dentre os 18 classificadores utilizados nos experimentos naquele trabalho, mantivemos os mais efetivos que incluem: XLNet, BERT, SVM, Regressão Logística (RL) e KNN com TF-IDF (TF) e KNN e SVM com Meta-Features de Centroides [Canuto et al. 2019, Cunha et al. 2020], totalizando sete modelos base[2]. Mantivemos também os mesmos processos para produção de probabilidades do treino e teste e configuração e otimização de parâmetros tanto para os modelos base como para o meta-classificador.

Para detecção de documentos difíceis empregamos o algoritmo *Random Forest*[3]– com o seguinte conjunto de parâmetros: 300 árvores (*N Estimators*), profundidade máxima das árvores (*Max Depth*) 8 e balanceamento de classe (*Class Weight*) com *subsample*. Para avaliar o detector aplicamos validação cruzada com 10 *folds* e estratificação.[4] Além disso, neste trabalho utilizamos duas bases de dados: **ACM** consiste

[1]Como usaremos 7 classificadores nos nossos experimentos, teremos ao total de 49 (7x 7) features.

[2]Com estes sete classificadores, mantivemos os mesmos níveis de efetividade de [Cunha et al. 2021] a um custo muito menor.

[3]*Classificador Random Forests*: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[4]O código utilizado nos experimentos está disponível em: https://github.com/warSantos/MS.git

em uma coletânea de artigos científicos de ciência da computação com mais de 24 mil documentos distribuídos em 11 classes; **20NG** contém mais de 18 mil notícias de diferentes jornais distribuídos em 20 classes. A Tabela 1 contém a distribuição dos documentos.

Tabela 1. Quantidade de documentos em cada base. Número (e percentagem) total e por classe.

Base de dados	N Docs	N Fáceis	N Difíceis
ACM	24897	19877 (79.83)%	5020 (20.16)%
20NG	18846	16942 (89.89)%	1904 (10.10)%

4.2. Resultados e Discussão

Apresentamos a seguir uma avaliação de efetividade do detector de documentos difíceis para as bases de dados da ACM e 20NG. Na Tabela 2 temos a Macro-F1, precisão e revocação obtida pelo detector para documentos fáceis e difíceis isoladamente, além da Macro-F1 média. No geral, o detector apresenta efetividade promissora, sustentando 75.25 e 81.00 pontos de Macro-F1 para as bases ACM e 20NG. Como esperado, o detector possui maior facilidade em identificar documentos fáceis atingindo elevada precisão, revocação e Macro-F1 92.3, 84.7 e 88.2 para ACM e 97.1, 96.0 e 95.6 para 20NG, respectivamente. Em relação aos documentos difíceis, foco deste trabalho, apesar da precisão mediana atingida na ACM de 55.2 o detector obteve uma alta revocação de 74.2. Na base 20NG, foi obtida uma precisão de 67.7 e uma revocação de 75.0.

Tabela 2. Desempenho do detector de documentos difíceis. Computamos as métricas Macro-F1 (F1), Precisão (P) e Revocação (R) para os documentos fáceis (Fác.) e difíceis (Dif).

Base de Dados	F1-Média	F1-Dif.	F1-Fác.	P-Fác.	P-Dif.	R-Fác.	R-Dif.
ACM	75.2	68.8	88.2	92.3	55.2	84.7	74.2
20NG	81.0	67.7	95.6	97.1	67.9	96.0	75.0

Note que uma revocação relativamente alta, próxima a 75% nas duas bases, significa que estamos passando para a segunda fase da nossa metodologia a maior parte dos documentos difíceis, o que serve bem ao nosso objetivo. A dificuldade relativa na detecção de documentos difíceis é justificada por diferentes fatores, incluindo o desbalanceamento de classes, além da dificuldade inerente dos modelos base em corretamente classificar esses documentos. De qualquer forma, mostramos que a detecção de documentos difíceis é viável, tornando a detecção desses documentos uma frente promissora para melhorias dos modelos de *stacking*, como indicado na Tabela 3. É importante destacar que para base ACM (base mais desafiadora - desbalanceada) o limite superior (UpperBound – assumindo um detector perfeito) atinge considerável melhora na efetividade (73.74 para 75.68), sendo um sinal positivo do potencial da abordagem a ser explorado.

Tabela 3. Macro-F1 do stacking, pipeline, e limite superior do pipeline (com detector perfeito).

ACM	Stacking	Pipeline	UpperBound	20NG	Stacking	Pipeline	UpperBound
	73.74	69.72	75.68		92.18	89.79	92.34

Figura 3 apresenta uma análise da importância das features usando o método de redução de entropia das árvores do modelo *Random Forest*. Na Figura 3 temos o ranking das 20 *meta-features* mais importantes (eixo x) pela importância média de cada *meta-feature* sobre os 10 *folds* da validação cruzada para cada base de dados (eixo y). Podemos observar que as *meta-features* com foco na entropia da predição e tamanho da concordância dos modelos base dominam o topo do ranking. Além disso, notamos que os modelos que utilizam a representação TF-IDF (e.g., KNN, SVM, RL) produziram *meta-features* mais importantes do que as redes neurais para essa tarefa, o que é interessante e merece um aprofundamento em trabalhos futuros.

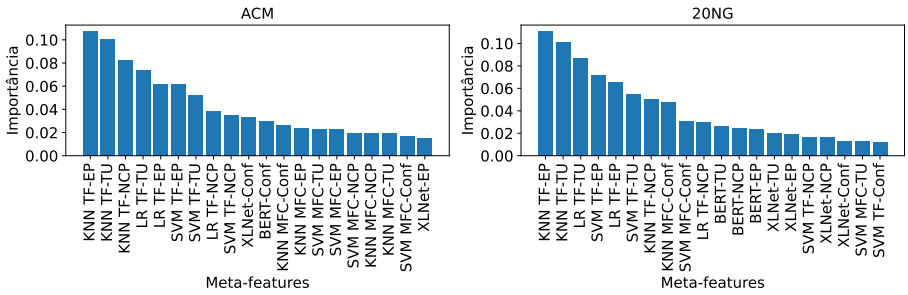

Figura 3. ***Ranking* das 20 *meta-features* mais importantes (Redução de entropia).**

5. Conclusão

Neste trabalho introduzimos um novo *pipeline* para *stacking* em duas etapas: detecção de documentos difíceis com um novo conjunto de *meta-features* e uma nova estratégia para gerar meta-classificadores especialistas em documentos difíceis. Além disso, realizamos um estudo da efetividade do detector de documentos difíceis proposto na primeira etapa em duas bases de dados (ACM e 20NG) amplamente utilizadas na literatura. Nossos experimentos se mostraram promissores resultados na destilação de documentos entre difíceis e fáceis, atingindo 75.25 e 81.00 pontos de Macro-F1 para as bases ACM e 20NG, respectivamente. Nosso trabalho abre espaço para esforços futuros direcionados a melhoria de *stacking*, onde se destacam duas frentes promissoras: geração de novas *meta-features* para detecção de documentos difíceis e evolução do meta-classificador especialista com técnicas de engenharia de *features* (e.g. seleção de instâncias [Cunha et al. 2023]).

Agradecimentos: Este trabalho foi parcialmente financiado pelo CNPq, CAPES, FAPEMIG, FAPESP e Amazon-AWS.

Referências

Canuto, S., Salles, T., Rosa, T. C., and Gonçalves, M. A. (2019). Similarity-based synthetic document representations for meta-feature generation in text classification. In *Proceedings of the 42nd International ACM SIGIR Conference*.

Chowanda, A. and Muliono, Y. (2022). Indonesian sentiment analysis model from social media by stacking bert and bi-lstm. In *2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)*.

Cunha, W., Canuto, S., Viegas, F., Salles, T., Gomes, C., Rosa, T., Gonçalves, M. A., and Rocha, L. (2020). Extended pre-processing pipeline for text classification: On the role of meta-feature representations, sparsification and selective sampling. *IP&M*, 57.

Cunha, W., Mangaravite, V., Gomes, C., Canuto, S., Viegas, F., França, C., Almeida, J. M., Rosa, T., Rocha, L., and Gonçalves, M. A. (2021). On the cost-effectiveness of neural and non-neural approaches and representations for text classification. *IP&M*, 58.

Cunha, W., Viegas, F., França, C., Rosa, T., Rocha, L., and Gonçalves, M. A. (2023). A comparative survey of instance selection methods applied to nonneural and transformer-based text classification. *ACM Computing Surveys*.

Desai, S. and Durrett, G. (2020). Calibration of pre-trained transformers. In *arXiv preprint arXiv:2003.07892*.

Ding, W. and Wu, S. (2020). A cross-entropy based stacking method in ensemble learning. *J. of Intelligent & Fuzzy Systems*, 39:1–12.

Džeroski, S. and Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54.

Gomes, C., Goncalves, M., Rocha, L., and Canuto, S. (2021). On the cost-effectiveness of stacking of neural and non-neural methods for text classification: Scenarios and performance prediction. In *Findings of the ACL-IJCNLP 2021*.

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *ICML'05*.

Penha, G., Campos, R., Canuto, S., Gonçalves, M., and Santos, R. (2019). Document performance prediction for automatic text classification. In *European Conference on IR Research (ECIR)*.

Singhal, R. and Kashef, R. (2023). A weighted stacking ensemble model with sampling for fake reviews detection. *IEEE TCSS*.

Subba, B. and Kumari, S. (2022). A heterogeneous stacking ensemble based sentiment analysis framework using multiple word embeddings. *Computational Intelligence*.

Wahba, Y., Madhavji, N., and Steinbacher, J. (2022). Reducing misclassification due to overlapping classes in text classification via stacking classifiers on different feature subsets. In *Proceedings of the 2022 FICC, Vol.*