Doclass: open-source software to support document labeling and classification

M. A. Inuzuka[1], H. A. D. do Nascimento[1], F. S. Almeida[1], B. M. Barros[1], W. A. R. Jradi[2]

[1] Universidade Federal de Goiás, Brazil
{marceloakira,hadn}@ufg.br, {fernandoseverino, brunomattos}@discente.ufg.br
[2] Ultimatum Tecnologia Jurídica, Brazil
walid@ultimatum.com.br

Abstract. This article introduces Doclass, a free and open-source software for the Web that aims to assist in labeling and classifying large sets of documents. The research involved a design science research methodology, guided by the real demands of a legal text processing company. The architecture, several design decisions and the current development stage of the software are presented. Preliminary user experiments for evaluating interactive document labeling are described. As a result, the first version of a system with an architecture composed of a mobile frontend that communicates with a backend through a REST API was published, with satisfactory performance evaluation by the applicant. Other results involve the use of active learning techniques to reduce human effort when performing the classification of documents, as well as the Uncertainty strategy to choose the document to be labeled. The effectiveness of the stop criterion for the active learning technique based on confidence level was tested and proved unsatisfactory, remaining as a future work.

CCS Concepts: • **Computing methodologies** → **Natural language processing**.

Keywords: document classification, active learning, annotation tool, document labeling, legal text

1. INTRODUÇÃO

Este trabalho relata a construção da primeira versão da Doclass[1], uma ferramenta de software para anotação e classificação interativa de documentos textuais jurídicos.

O contexto da pesquisa teve a Ultimatum[2] como ambiente de estudo, uma empresa privada que processa informações extraídas de diários oficiais de entidades jurídicas do país. Um de suas atividades consiste na busca e distribuição de excertos judiciais – tais como intimações, sentenças, acórdãos, etc. – de interesse de seus assinantes. Esse processo envolve a pesquisa diária de cerca de 450 jornais, totalizando a mineração de aproximadamente 22 milhões de publicações mensais.

Para guiar a pesquisa, foi adotada a metodologia de ciência de projeto [Lacerda et al. 2013], a qual tem como foco gerar conhecimento prescritivo por meio da produção de artefatos para a resolução de problemas reais, oriundos de uma organização específica. No entanto, o estudo das especificidades do ambiente organizacional não impede que o conhecimento produzido seja passível de generalização para uma classe de problemas. Outra metodologia bem conhecida é o estudo de caso, cujo objetivo é descritivo e gerador de hipóteses, o que não se encaixa com os anseios deste trabalho.

Após entrevistar os interessados, foi decidido produzir um software que apresentasse evidências

[1]O projeto completo, com código-fonte e documentação, está disponível em https://gitlab.com/ivato/doclass
[2]Ultimatum Tecnologia Jurídica. Mais informações estão disponíveis no endereço https://www.ultimatum.com.br/

para respostas às seguintes questões de pesquisa: Q1 - Qual arquitetura oferece escalabilidade e portabilidade para rotulação e classificação de documentos? Q2 - Quais técnicas ou conjunto de algoritmos contribuem para reduzir o esforço humano de anotação de documentos? Considerando a metodologia adotada, o objetivo do trabalho não é contribuir com avanço do estado da arte, mas sim produzir uma prescrição de um sistema que satisfaça os requisitos mínimos elencados pela requerente. Assim, várias demandas foram avaliadas e, por uma questão de prioridade e escopo, nesta versão não foi implementada classificação multi-rótulo e nem compartilhamento de dados entre usuários.

O restante deste artigo está organizado como segue. A Seção 2 contextualiza o problema apresentando, como os dados são obtidos e os trabalhos relacionados. A Seção 3 descreve o projeto do software, incluindo sua arquitetura, o *pipeline* de atividades, a interface com o usuário e testes. A Seção 4 avalia o desempenho do *software* através de experimentos. Finalmente, na Seção 5, é exposta uma percepção geral sobre os achados da pesquisa e rumos para investigações futuras.

2. CONTEXTUALIZAÇÃO

Nesta seção, é apresentado o contexto geral da pesquisa: como os dados foram obtidos e extraídos (2.1) e os trabalhos relacionados encontrados na literatura (2.2).

2.1 Obtenção do conjunto de dados

O processo de extração de dados pela empresa Ultimatum é realizado principalmente pela conversão de arquivos PDF para documentos em formato de texto puro, denominados excertos jurídicos. Tais documentos inicialmente não possuem nenhum tipo de classificação e seus metadados limitam-se à fonte da informação, tais como: data de publicação, órgão publicador, etc. Assim, uma das formas de recuperação de informação é por meio de busca léxica de "strings" no texto dos corpos dos documentos.

Para a construção de um *dataset* para os experimentos utilizou-se arbitrariamente a *string* de busca "audiência de custódia", onde a intenção era obter documentos do tipo "atas de audiência de custódia". Em tais audiências um juiz emite decisões em relação a casos de prisão em flagrante, mantendo o cárcere ou concedendo a liberdade com possíveis medidas cautelares[3]. Assim, obteve-se um lote inicial com 493 documentos mas cuja totalidade não necessariamente correspondia ao tipo desejado, embora o corpo do texto contivesse a *string* de busca. Isto deve-se ao fato de que o termo "audiência de custódia" frequentemente é encontrado em documentos cuja natureza é diversa da pretendida, e. g. *habeas corpus*, *editais de convocação*, etc. Então, para a obtenção de documentos pertencentes apenas ao tipo desejado, foram empregados especialistas humanos que identificaram 152 excertos válidos. Neste contexto, portanto, nos limitamos à classificação binária deste tipo de documento.

2.2 Trabalhos relacionados

Ferramentas de apoio à classificação interativa de textos têm surgido como resultado de investigações científicas nesta área. [Brooks et al. 2015] publicaram o *FeatureInsight*, ferramenta de código fechado para classificação de páginas da internet com suporte visual para seleção de palavras-chaves (*features*) para aprendizado de máquina; porém relataram problemas em aberto, como a dificuldade dos usuários avaliarem a qualidade dos *features* selecionados. Em 2011 [Settles 2011] apresentou o *DUALIST*, uma ferramenta de anotação de textos, que permite ao usuário classificar textos e selecionar *features* que auxiliam na classificação dos documentos e usou técnicas de *active learning* [Settles 2010] para seleção de documentos a serem classificados. A ferramenta *Monkeylearn*[4] é uma das opções comerciais populares para a classificação de textos por meio de rótulos, testar o modelo gerado e que através de um API, suporta integração com outras ferramentas; no entanto, não suporta *active learning*. Outra

[3]Mais informações podem ser obtidas no endereço https://www.cnj.jus.br/sistema-carcerario/audiencia-de-custodia/
[4]Disponível em https://monkeylearn.com/

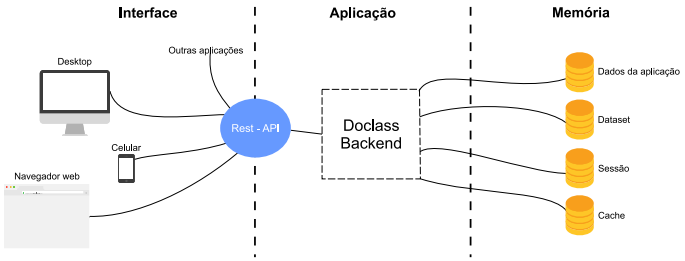


Fig. 1. Arquitetura do projeto experimental

opção comercial de código fechado também popular é o *Prodigy*[5], que suprta *active learning* e vários tipos de classificação como texto e entidades nomeadas. Soluções como o *Brat*[6] e *Doccano*[7] são alternativas de ferramentas *open source* específicas para anotação, ou seja, não suporta classificação interativa e *active learning*. O presente trabalho se diferencia devido à sua modularidade, utilização de tecnologias atuais, ser um projeto *open source* e utilizar-se de técnicas de *active learning* para diminuir o esforço de rotulação por parte dos usuários. Uma tabela comparativa das características de cada uma das ferramentas está disponível na documentação do projeto.

3. PROJETO

Nesta seção, apresentamos o projeto do sistema: sua arquitetura, o *pipeline* de atividades, a interface gráfica com o usuário e os testes de desempenho. Todas as tecnologias empregadas, detalhes e justificativas de utilização estão disponíveis na documentação do projeto.

A Figura 1 ilustra a arquitetura do sistema e como esses componentes se comunicam. Foi projetada em três camadas, de forma a garantir o desacoplamento entre seus componentes. (1) Na **camada de interface** encontram-se os meios de interação do usuário ou integração com outros sistemas. Nesta camada não existe qualquer restrição em relação às tecnologias utilizadas ou ambiente de execução (*desktop*, celular, navegador internet, etc), desde que utilize o protocolo de acesso (API) ao Doclass Backend, construído sob o padrão arquitetural *REST* (Representational State Transfer) [Fielding 2000]. Na versão atual do sistema, apenas a interface para telefones celulares foi implementada. (2) Na **camada de aplicação** está localizado o Doclass Backend, que é responsável por todos os mecanismos de aprendizagem de máquina, bem como os meios de comunicação entre a interface e a camada de memória. (3) Na **camada de memória**[8] foram implementados mecanismos de armazenamento de dados da aplicação, cache, *dataset* e sessão. Esta camada é apresentada em mais detalhes a seguir.

Na Figura 2 é apresentado o fluxo de atividades (*pipeline*) projetado para o sistema. Cada atividade envolveu vários ciclos de experimentação, e as escolhas que se demonstraram tecnicamente vantajosas, estão aqui justificadas: (1) na **abertura** da sessão o usuário identifica-se e o sistema recupera uma sessão aberta anteriormente, caso exista; caso contrário, uma nova sessão é criada e armazenada em um banco de dados em memória. Apesar do padrão arquitetural REST adotar o modelo de comunicação sem estado (*stateless*), na prática a manutenção de estado no servidor se demonstrou vantajosa em desempenho quando ocorrem várias interações com o usuário. (2) logo após a abertura da sessão, o usuário pode escolher um *dataset* para **carga**. O sistema então verifica se o *dataset* já foi carregado anteriormente; caso sim, somente carrega seus metadados, caso contrário, realiza o seu *download*. Experimentalmente, a solução mais efetiva de persistência dos *datasets* foi mantê-los em servidores HTTP em tabelas de formato de valores separados por vírgula (CSV) compactados.

[5]Disponível em https://prodi.gy/
[6]Disponível em https://brat.nlplab.org/index.html
[7]Disponível em https://doccano.github.io/doccano/
[8]Usa-se o termo "camada de memória" pois os dados de *cache* e sessão não são persistentes

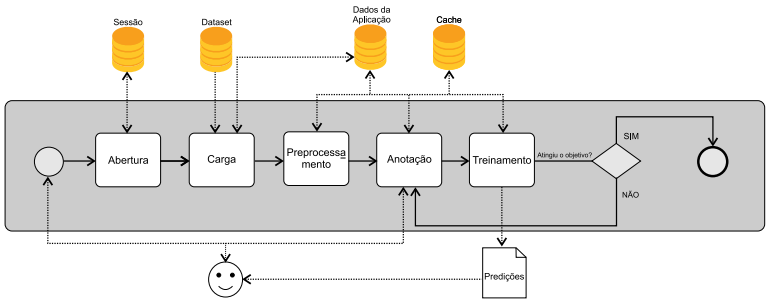


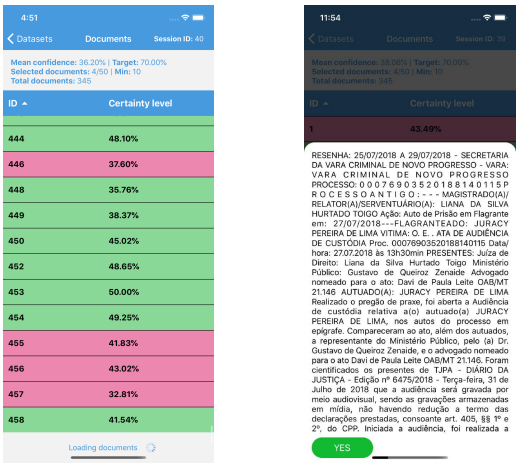
Fig. 2. Pipeline do projeto experimental

Desta forma, somente os metadados dos textos de cada documento do *dataset* são armazenado na base de dados de aplicação e o conteúdo de cada texto é carregado temporariamente em memória para preprocessamento. (3) No **preprocessamento**, foram filtrados *stopwords* do texto e cada token foi vetorizado utilizando-se a técnica *Term Frequency–Inverse Document Frequency* (TF-IDF) [Luhn 1957]. (4) Na **anotação**, o usuário classifica instâncias fornecidas pelo sistema Doclass através da associação a rótulos que são armazenados na base de dados da aplicação. Para reduzir o esforço de rotulação de documentos foi empregada a técnica de *active learning*. (5) No **treinamento**, o sistema gera o modelo a partir das instâncias anotadas e sob demanda, prediz a classe de cada documento através de um algoritmo de aprendizado de máquina. O sistema deve então avaliar se a predição atingiu um objetivo de qualidade; caso não, solicita mais instâncias para anotação, caso sim, não solicita mais instâncias e para o processo de treinamento.

A Figura 3(a) é uma captura da tela da interface gráfica do celular. Esta interface consiste em uma tabela de documentos, na qual cada linha corresponde a um documento identificado pelo id (identificação), com sua respectiva propriedade *certainty level* (nível de confiança) e a classificação do texto como sendo ata de audiência de custódia ou não; na cor verde e vermelho, respectivamente. As linhas da tabela podem ser ordenadas ao clicar-se em uma dessas propriedades informadas no cabeçalho. Acima do cabeçalho é informado a confiança média (mean confidence), o objetivo (target), a quantidade de documentos selecionados (selected documents) do máximo possível (50), o mínimo de documentos obrigatórios para seleção (10) e o total de documentos disponíveis para seleção (345). A Figura 3(b) apresenta o conteúdo de um documento já classificado como ata de audiência de custódia, conforme o único botão disponível na cor verde e com valor 'YES'.

A tabela I apresenta o tempo médio nos ambientes de desenvolvimento[9] e produção[10] nas atividades de carga com largura de banda 5 Mbps e 30 Mbps; preprocessamento sem e com utilização de cache; treinamento para cada documento selecionado para 25 e 50 passos; e predição para 100 e 200 documentos. Os testes foram automatizados através de um *script* em *Python* e utilizando-se a média de 5 rodadas[11]. No ambiente de produção o tempo de carga não foi medido, pois o *dataset* estava armazenado no mesmo hospedeiro. Nota-se também que a utilização o desempenho do *cache* no sistema de arquivos do ambiente de produção foi ruim, comparando-se com o *cache* do ambiente de desenvolvimento que utilizou disco rígido SSD. O tempo médio de treinamento para cada documento selecionado aumentou 21% e 6% nos ambientes de desenvolvimento e produção, quando se aumentou a quantidade de 25 para 50 documentos treinados, respectivamente. O tempo de predição praticamente dobrou quando se aumentou 100 para 200 documentos em ambos tipos de ambiente. Esse resultado nos levou a implementar predições sob demanda, ou seja, ajustadas dinamicamente conforme a rolagem da janela; assim garante-se escalabilidade da aplicação independentemente do tamanho do *dataset*.

[9]S.O. Ubuntu 18.04.1 em processador Intel i7-7500U CPU @ 2.70GHz com 16 GB de memória RAM e HD SSD
[10]MV Debian GNU/Linux 10 em processador Intel Xeon vCPU X5675 3.07 GHz c/ RAM de 1 GB e disco virtual
[11]Os detalhes de utilização deste script estão descritos na documentação do sistema



(a) (b)

Fig. 3. Interface: (a) documentos e seus valores de confiança; (b) conteúdo de um documento e opções de rotulamento.

Table I. Tempos de execução de atividades

Ambiente	Carga		Preprocessamento		Treinamento		Predição	
	5Mbps	30Mbps	s/ Cache	c/ Cache	25 passos	50 passos	100 docs	200 docs
Desenvolvimento	7,32s	2,26s	0,71s	0,03s	3,05s	3,71s	0.44s	1.08s
Produção	NA	NA	0,67s	1,98s	2,65s	2,81s	0,37s	0,90s

4. EXPERIMENTAÇÃO

Dois experimentos foram realizados com o intuito de avaliar o sistema e identificar possibilidades de melhoria. Neles, empregou-se o conjunto de documentos obtido conforme a Seção 2.1. Esse foi dividido entre treinamento e teste, sendo a partição de teste correspondendo a um terço do total. O conjunto de treino foi então subdivido em dois subconjuntos, L e U, representando, respectivamente, o grupo dos documentos já rotulados pelo humano e o grupo dos documentos ainda não rotulados pelos humanos. O rótulo correto de cada documento de U existia, mas foi assumido provisoriamente que ele estava ausente, sendo recuperado apenas quando necessário e empregando o conceito de "consultar um oráculo", comum na abordagem de *active learning*. Os testes iniciaram sempre com o mesmo conjunto L contendo três documentos previamente rotulados e com um método de classificação (no caso do presente estudo, um SVM[12]) treinado sobre esse conjunto. Além disso, uma classe foi predita para cada documento de U usando o classificador, e uma medida de confiança acerca dessa predição foi obtida usando a função *predict_proba()* da biblioteca Python Scikit-learn. Toda vez que um documento de U era rotulado, ele era movido para o conjunto L, o método de classificação era retreinado sobre o novo L e as classes e as medidas de confiança eram recalculadas para os elementos remanescentes de U. O desempenho do classificador foi medido em termos de acurácia, sendo esta calculada usando apenas o conjunto de teste. Os detalhes dos experimentos são apresentados a seguir.

4.1 Primeiro Experimento

O primeiro experimento consistiu em um estudo piloto com o objetivo de verificar como os usuários se sairiam no uso do Doclass. Em particular, no emprego dos níveis de confiança para guiar a escolha dos documentos a serem rotulados. Oito usuários, envolvendo alunos de graduação, ex-alunos e docentes, participaram do experimento. O sistema foi ajustado para apresentar uma tela como aquela

[12]O método *Support Vector Machine*, ou SVM, foi selecionado por ter fácil implementação, ter rápido tempo de treinamento e inferência e apresentar bons resultados na tarefa de classificação de documentos[Mayor and Pant 2012].

Table II. Dados dos testes realizados pelos usuários e dos testes com as estratégias *uncertainty* e *Aleatória*.

Experimento	Passos	Acurácia	Confiança Média	σ
Usuário 1	39	**0,986486**	0,904444	0,096574
Usuário 2	35	0,777027	0,852583	0,149603
Usuário 3	8	0,358108	**0,939887**	0,059245
Usuário 4	35	0,797297	0,858085	0,148640
Usuário 5	9	0,358108	0,749888	0,042089
Usuário 6	28	0,871622	0,862033	0,130286
Usuário 7	17	0,885135	0,884689	0,163129
Usuário 8	10	0,358108	0,924086	0,045162
Média para os Usuários	22,6	0,673986	0,871961	0,103894
Uncertainty	38	0,966216	0,829571	0,112414
Aleatória: Melhor	43	0,952703	0,862216	0,158743
Aleatória: Mediano	37	0,878378	0,854150	0,133096
Aleatória: Pior	13	0,641892	0,794516	0,00644

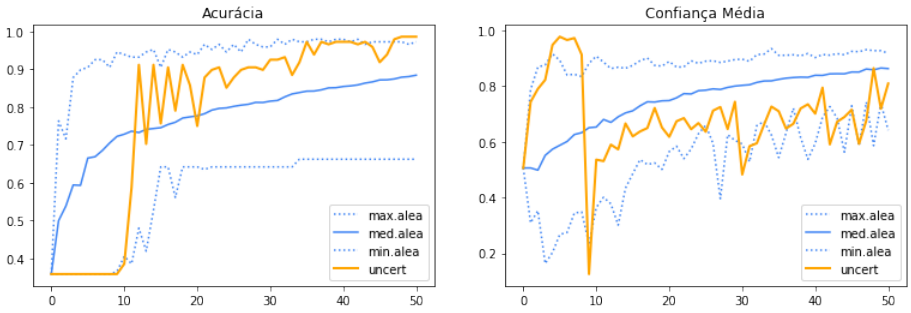
descrita na Figura 3(a). Aqui o usuário poderia escolher um documento, o que o levava à tela da Figura 3(b). Nesse momento, o usuário tinha duas opções: (1) aplicar um rótulo ao documento, o que imediatamente o transferia de U para L, forçava o re-treino do classificador e atualizava a classe e a medida de confiança dos demais itens de U; ou (2) cancelar a visualização do documento. Em ambos os casos, o sistema retornava à tela anterior para a escolha de outro documento. Os usuários tiveram liberdade para escolher qual documento rotular. O sistema foi modificado, no entanto, para sempre apresentar e utilizar o rótulo correto (já conhecido), a fim de evitar variação na qualidade das respostas dos usuários e para reduzir o tempo total do experimento. Além disso, cada usuário partiu das mesmas condições iniciais e podia rotular documentos até que uma das seguintes condições de parada fosse satisfeita: (i) pelo menos 8 documentos novos tivessem sido rotulados e a média da confiança dos itens em U menos o seu desvio padrão fosse maior ou igual a 0,7; ou (i) 50 novos documentos tivessem sido rotulados. A intenção era que as condições de parada finalizassem o processamento assim que o classificador alcançasse um nível de efetividade suficiente para rotular sozinho todos os demais documentos, no conjunto U. A ação dos usuários a cada passo, incluindo a acurácia do classificador na iteração corrente, foi registrada para avaliação posterior.

Para fins de comparação com as escolhas feitas pelos usuários, 101 testes totalmente automatizados foram realizados utilizando estratégias de *activite learning* para a escolha dos documentos a serem rotulados. Um deles adotou a estratégia *Uncertainty*, na qual o elemento de U com menor confiança na classificação foi escolhido para ser rotulado[13]. Os demais 100 testes adotaram a estratégia *Aleatória*, onde um documento qualquer de U era escolhido. Em todos os testes automáticos, as configurações iniciais e os procedimentos de atualização do classificador e das predições (bem como as condições de parada) foram os mesmos daqueles empregados nos testes com humanos.

A Tabela II mostra os resultados obtidos com os testes. As primeiras linhas referem-se aos testes dos usuários, incluindo a média de seus valores. Em seguida, estão os resultados alcançados no teste automático usando a estratégia *Uncertainty* e nos casos "melhor", "mediano" e "pior" entre os 100 testes com a estratégia *Aleatória*. As colunas da tabela indicam quem produziu o resultado, a quantidade de passos até atingir uma condição de parada, a acurácia final do classificador (sobre o conjunto de teste), a confiança média dos documentos em U na última iteração e o desvio padrão da dessa medida de confiança (σ). Os maiores valores de acurácia e de confiança estão destacados em negrito.

Pela tabela, percebe-se que o teste automatizado utilizando a estratégia *Uncertainty* alcançou uma acurácia de aproximadamente 0,96, o que pode ser considerado elevado. Os testes com a estratégia *Aleatória* apresentaram uma grande variação de acurácia, de 0,64 a 0,95 aproximadamente, tendo o caso mediando obtido 0,878378. Por outro lado, apenas um usuário humano conseguiu superar

[13]Essa estratégia foi escolhida por ter sido a mais eficaz em experimentos automatizados previamente realizados. Uma descrição mais detalhada do método por ser encontrada em [Lewis and Gale 1994]



(a) (b)

Fig. 4. (a) Evolução da acurácia e (b) Evolução da confiança média do classificador, nas estratégias *Uncertainty* e *Aleatória* de seleção de itens para rotulamento.

em acurácia os métodos automatizados para a escolha do documento a ser rotulado (Usuário 1, com acurácia de 0,986486). Todos os demais usuários chegaram a resultados inferiores ao do *Uncertainty* e, em alguns casos, inferiores até mesmo ao do pior caso do teste com a estratégia *Aleatória*. Analisando-se com mais cuidado a situação, verifica-se que os testes humanos com baixa acurácia apresentaram uma parada prematura pela ativação da condição de parada (i), com poucos passos de iteração mas alta medida média de confiança. Interessantemente, dois desses usuários afirmaram terem iniciado o teste seguindo uma estratégia parecida ao método com *Uncertainty*. Isso sugere que o uso da confiança média como um dos elementos da condição de parada pode não ter sido uma decisão adequada. De fato, a correlação de Pearson entre as colunas Confiança Média e Acurácia, excluindo-se a linha da média, é de apenas 0,004749, o que demonstra que a confiança média não é uma boa estimativa de acurácia. Em contra-partida, a correlação de Pearson entre a quantidade de passos e a acurácia é de 0,865028, e entre o desvio padrão da confiança e a acurácia é de 0,729255.

4.2 Segundo Experimento

O segundo experimento procurou investigar mais profundamente a relação entre a medida de confiança média e a acurácia do classificador. Para tanto, somente os testes automatizados foram empregados. Tanto o teste com a estratégia *Uncertainty* quanto os 100 testes automatizados com a estratégia *Aleatória* tiveram seus passos estendidos até que a condição de parada (ii) fosse atingida, e não mais a condição (i). Isso garantiu que todos os 101 testes resultassem em 50 documentos rotulados, além dos 3 documentos iniciais. Os dados de acurácia (tomados sobre o conjunto de teste) e de confiança média (calculados sobre o conjunto U) dos classificadores foram registrados a cada passo.

A Figura 4 sumariza esses dados na forma de gráficos de evolução da acurácia e da confiança média do classificador. A linha mais grossa nas imagens (também destacada em cor laranja) apresenta os valores do teste usando a estratégia *Uncertainty*. As linhas mais finas (e em cor azul), no caso da figura à esquerda, são referentes aos valores "máximo", "médio" e "mínimo" de acurácia dos 100 testes usando a estratégia *Aleatória* tomados a cada passo. Na figura à direita, essas linhas mais finas representam os valores "máximo", "médio" e "mínimo" da confiança média nos mesmos 100 testes.

Pela figura, observa-se que a estratégia *Uncertainty* não apresenta bons resultados de acurácia nos primeiros 10 passos mas que essa medida rapidamente evolui e consegue atingir valores superiores à acurácia alcançada pela estratégia *aleatória*. Verifica-se também que a confiança média para o teste com a estratégia *Uncertainty* apresenta um comportamento oposto ao da acurácia nos primeiros 10 passos, atingindo logo um patamar muito alto e depois caindo. Isso pode explicar a razão pela qual os usuários que seguiram uma abordagem parecida à da estratégia *Uncertainty* tenham sofrido uma parada prematura de seus experimentos pela condição (i). Também sugere que, caso os usuários tivessem tido mais tempo, seus resultados de acurácia poderiam melhorar significativamente. Análises

de correlação entre as séries temporais da confiança média e da acurácia tanto para o teste com *Uncertainty* quanto para os testes com a estratégia *Aleatória* não apresentaram valores significativos.

5. DISCUSSÕES E CONCLUSÃO

Neste artigo foi apresentado o estado atual de desenvolvimento do Doclass, sistema que está sendo liberado como software livre para apoio à classificação de textos jurídicos. Conceitos de escalabilidade e de portabilidade foram considerados quando projetando a arquitetura do sistema, baseada em 3 camadas: interface com usuário, aplicação (*backend*) e memória. A arquitetura é inteiramente suportada por tecnologias livres que foram integradas, testadas, validadas pelo demandante e documentadas.

Para a meta de reduzir o esforço humano de classificação de documentos, o emprego de um classificador semi-supervisionado que melhora o seu desempenho usando *active learning* é a solução. A questão está agora em decidir em quais níveis de interação o usuário deve atuar (se ele deve ou não escolher o documento a ser rotulado) e a condição de parada de sua atividade de rotulação.

Ainda é cedo para qualquer conclusão sobre quem – humanos ou estratégias automatizadas – possui melhor capacidade de decidir quais documentos devem ser rotulados. Uma forma menos arriscada seria a adoção de procedimentos automatizados usando *Uncertainty*, onde o usuário é responsável apenas pela rotulagem do documento escolhido. Novos estudos devem se aprofundar nessa questão.

Em condições reais, não há como garantir que um classificador treinado sobre alguns poucos itens apresentará um bom desempenho na classificação de elementos de um conjunto não rotulado. Por outro lado, alguma estimativa sobre esse desempenho é necessária a fim de possibilitar a definição de uma condição de parada que minimize o esforço humano de rotulamento. Com base nos dois supracitados experimentos é possível concluir que a confiança média não foi uma escolha adequada para produzir tal estimativa, já que apresenta baixa correlação com a acurácia, medida de desempenho adotada no estudo. Critérios de parada baseados em outras medidas de confiança, como algumas propostas de Zhu et al. [2010], podem ser boas alternativas, devendo ser avaliadas em um estudo futuro.

Por fim, o desenvolvimento da Doclass continua através de pesquisas que estendem o estado inicial do projeto como, por exemplo: o emprego da ferramenta para classificação multi-rótulo, utilização de outros algoritmos de aprendizado de máquina, suporte a *crowdsourcing* para permitir a classificação colaborativa de um grande conjunto de documentos.

REFERENCES

Brooks, M., Amershi, S., Lee, B., Drucker, S. M., Kapoor, A., and Simard, P. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, Chicago, IL, USA, pp. 105–112, 2015.

Fielding, R. T. *Architectural styles and the design of network-based software architectures*. Ph.D. thesis, University of California, Irvine, 2000.

Lacerda, D. P., Dresch, A., Proença, A., and Antunes Júnior, J. A. V. Design Science Research: método de pesquisa para a engenharia de produção. *Gestão & Produção* 20 (4): 741–761, Nov., 2013.

Lewis, D. D. and Gale, W. A. A sequential algorithm for training text classifiers. In *SIGIR '94*, B. W. Croft and C. J. van Rijsbergen (Eds.). Springer London, London, pp. 3–12, 1994.

Luhn, H. P. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development* 1 (4): 309–317, 1957.

Mayor, S. and Pant, B. Document classification using support vector machine. *International Journal of Engineering Science and Technology* vol. 4, pp. 1741–1745, 04, 2012.

Settles, B. Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2010.

Settles, B. Closing the Loop: Fast, Interactive Semi-Supervised Annotation With Queries on Features and Instances. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* vol. 1, pp. 12, 2011.

Zhu, J., Wang, H., Hovy, E., and Ma, M. Confidence-based stopping criteria for active learning for data annotation. *ACM Transactions on Speech and Language Processing* 6 (3): 3:1–3:24, Apr., 2010.