Predição de Processos com Dados não Estruturados Heterogêneos

Matheus Augusto de Oliveira[1], Flavia Maria Santoro[2], Kate Revoredo[3], Rosa Maria E. Moreira da Costa[2]

[1] Universidade Federal do Estado do Rio de Janeiro – RJ– Brasil
[2] Universidade do Estado do Rio de Janeiro (UERJ) – RJ- Brasil
[3] Universidade Federal do Rio de Janeiro – RJ– Brasil

matheuscruz27@gmail.com, {flavia, rcosta}@ime.uerj.br,
kate.revoredo@dcc.ufrj

Resumo. *Devido ao volume de informações e ao grande número de atividades do dia-a-dia dentro das organizações, o interesse na área de Gestão de Processos de Negócios ou Business Process Management (BPM) apresentou um crescimento notável. Portanto, cresce a necessidade de documentação, controle e otimização de processos e fluxos de trabalho existentes. Neste contexto, o monitoramento de processos ou Business Process Monitoring ganha atenção prevendo algumas características específicas de uma instância do processo em execução com base no histórico da execução do processo (log) e nos dados associados aos seus eventos. Essas previsões podem tornar as decisões de um analista de negócios mais assertivas, já que prever uma certa característica de uma instância em execução em um ponto específico do fluxo de trabalho com certo grau de confiança permite a atenuação de um problema antes que ele aconteça. Desta forma, as ações deixam de ser reativas e se tornam proativas, já que não precisamos esperar que a execução termine para observar o resultado e reagir. Por outro lado, decidir sobre o conjunto de características a considerar para a previsão é uma tarefa difícil. Neste artigo, avaliamos o impacto de características não estruturadas para predição do processo em uma empresa de prestação de serviços em TI.*

Abstract. *Due to the volume of information and the large number of day-to-day activities within organizations, the interest in the Business Process Management (BPM) area presented a remarkable growth and therefore, it increases the need for documentation, control and optimization of the existing workflows. In this context, Business Process Monitoring gain attention predicting some specific characteristics of the running process instance based on the history of the process execution and the data associated to the events. These predictions can make the decisions of a business analyst more assertive, since predicting a certain characteristic of a running instance at some specific point of the workflow with a high degree of confidence, allows the mitigation of a problem before it happens. Thus, actions are no longer reactive and become proactive, since we do not need to wait for the execution to finish to observe the result and react. On the other hand, deciding on the set of characteristics to consider for the prediction is a demanding and sometimes not straight forward task. In this paper, we evaluate the impact of unstructured characteristics for prediction on an IT consulting company.*

1. Introdução

Gerenciamento de Processos de Negócio tem sido bastante discutido e está presente nas instituições, sejam elas empresas privadas ou do setor público. Em Sistemas de Informação, o foco está em técnicas relacionadas como a descoberta, melhoria e o auxílio a tomada de decisão em processos (Francisco e Santos, 2012) (Oliveira e Delbem, 2011). Com o aumento significativo do fluxo de dados e das informações a eles associadas, a gestão desses processos passa a apresentar um novo grau de complexidade. Como apontado por Oliveira e Bertucci (2003), o gerenciamento da informação tornou-se um instrumento estratégico necessário para controlar e auxiliar decisões, através de melhorias no fluxo da informação, do controle, análise e consolidação da informação para os usuários.

Por outro lado, através do uso correto desses novos elementos, a predição de resultados da execução de processos fica mais precisa a cada dia. Com o uso de bases de dados mais robustas e abrangentes, é possível analisar atributos associados à execução do processo, antes não considerados, que por sua vez, também exercem influência no resultado obtido. Porém, junto a esses benefícios, alguns desafios também surgem, como por exemplo, o alto consumo de tempo no processamento dos algoritmos usados para análise (Teinemaa et al., 2016).

Teinemaa *et al.* (2016) propõem um *framework* que faz uso de técnicas de mineração de textos associadas ao monitoramento preditivo, para prever possíveis resultados do processo de contrato de empréstimo de uma agência bancária. Ao coletar informações como a renda e bens em posse do contratante, ou informações pessoais, como idade, tempo e tipo de emprego, analisar e comparar ao histórico existente, provenientes de diferentes casos, é possível apontar futuros atrasos e não pagamentos por parte do cliente. Sendo assim, a agência poderia repensar o tipo de contrato a ser firmado ou negar o valor pedido pelo cliente.

Maggi *et al.* (2103) focam em como a predição pode ajudar na gestão dos processos de atendimento de um hospital, seja no diagnóstico de doenças ou na indicação do tratamento adequado para seus pacientes, com base em casos similares do passado. Esses, dentre outros casos, conseguem exemplificar o quão este tema é abrangente e relevante.

O objetivo deste artigo é reproduzir e analisar os resultados do *framework* de Teinemaa *et al*. (2016) para predição das instâncias do processo de resolução de chamados de uma empresa de consultoria em tecnologia. Mais especificamente, prever se um caso em execução ultrapassará ou não o tempo de resolução estabelecido pelas regras de contrato da empresa com seus clientes. A principal contribuição é a utilização de texto não estruturado heterogêneo, que foi uma limitação deixada em aberto por Teinemaa *et al.* (2016).

O elemento chave (atributo que se deseja prever) é a variável que aponta se a instância do processo teve ou não seu acordo de nível de serviço ultrapassado ao final da sua execução. Para isso, parte do registro dos eventos dos casos executados e seus atributos estruturados e não estruturados foram organizados e utilizados como entrada para o treinamento e teste das funções existentes no *framework*. Assumindo-se as configurações apontadas com melhor performance, por seus desenvolvedores, considerando os tipos de dados do processo e seu comportamento. Isso possibilita também, o reforço ou refutação das hipóteses levantadas anteriormente sobre seu funcionamento e aplicação, sendo a principal delas, a de que o uso de texto, combinado aos outros atributos do processo, melhora a eficiência da previsão, mesmo para casos em que essa informação seja altamente heterogênea.

O artigo está estruturado como nas seguintes seções. Na Seção 2 é apresentada a base teórica, conceitos e técnicas aplicadas. Na Seção 3 são discutidos os trabalhos relacionados. Na Seção 4 é descrita a aplicação do Método de Predição no cenário real escolhido. A Seção 5 conclui o artigo e destaca as contribuições da pesquisa, sugerindo perspectivas futuras.

2. Fundamentação Teórica

Nesta seção são brevemente descritos os conceitos de descoberta de conhecimento em base de dados (Seção 2.1), que é utilizado para aprender os modelos preditivos, monitoramento preditivo (Seção 2.2) e o *framework* utilizado como base desse trabalho (Seção 2.3).

2.1 Descoberta de Conhecimento em Base de Dados

Descoberta de Conhecimento em Base de Dados (DCBD) é definida como o processo de extrair informações das grandes bases de dados para descobrir conexões ocultas e prever tendências (Bishop, 2016). A Figura 1 ilustra o processo DCBD.

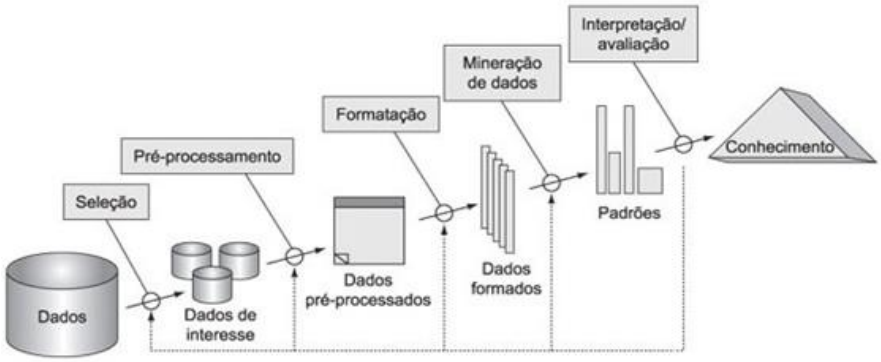

Figura 1 – Processo DCBD (Fayyad *et al.*, 1996).

Geralmente, bases de dados apresentam informações extras que não influenciam no que se deseja analisar, podendo inclusive atrapalhar o processo. Por isso, o primeiro passo as retira da base. O segundo passo é colocar essa base de dados na estrutura interpretada pelo método ou algoritmo que será utilizado na etapa seguinte, que é a mineração de dados. Nessa etapa os padrões são extraídos. Esses podem ser representados através de um modelo de classificação, que tem por objetivo aprender uma função que mapeia, ou classifica, um determinado dado em uma das várias classes predefinidas (Weiss e Kulikowski, 1991).

Neste artigo são utilizados dois modelos de classificação: Florestas Aleatórias (FA) Breiman (2001) e Regressão Logística (RL) (Walker e Duncan, 1967). Esses modelos apresentam bons resultados em várias configurações de problemas e se adaptarem bem ao caso em que os dados são muito escassos (Teinemaa et al., 2016). Floresta Aleatória é baseado na junção de árvores de decisão geradas aleatoriamente. A classificação retornada pela floresta aleatória é a classe mais observada entre as retornadas pelas árvores. Regressão Logística (RL) por sua vez é uma função linear das características do dado, que tem como imagem dois possíveis valores de classe "0" e "1“.

2.2 Monitoramento preditivo

O monitoramento de processos de negócio se refere à análise dos eventos produzidos durante a execução de um processo, com o intuito de avaliar o cumprimento de seus requisitos e sua

performance (Dumas *et al*., 2013). Duas são as abordagens: (i) *offline*, onde usa-se as informações de registros das instâncias finalizadas e (ii) *online*, para analisar a performance do caso em execução.

A gestão de processos de negócio e sua eficácia no auxílio à tomada de decisão está diretamente relacionada ao quanto é possível reduzir o nível de incerteza do resultado de cada uma das instâncias executadas através da predição. Quanto mais rápido e com maior nível de confiabilidade se pode prever os resultados de um processo, melhores decisões são tomadas e mais rapidamente. Utilizando uma restrição do negócio (*business constraint*), definida como um requisito imposto à execução de um processo que separa o comportamento conforme com o não-conforme Pesic e Aalst (2003), o responsável pela gestão do processo irá tomar as medidas necessárias para que se alcance o resultado esperado.

A predição de processo pode ser feita de diversas maneiras. Dentre elas estão as técnicas de mineração de dados, que aprende um modelo preditivo à partir de dados históricos (*log* de eventos de instâncias finalizadas anteriormente). Com o monitoramento preditivo é possível antecipar diversas informações durante a execução do processo, como o tempo necessário para concluir uma tarefa, uma atividade ou operação específica. O grande ganho do monitoramento preditivo é o suporte para que as instituições consigam atingir suas metas previstas para cada um dos casos, podendo interromper a execução, caso percebam que ela sairá do esperado. Outros temas também são abordados, como a estimativa de tempo, seja de violação de prazo (*Service Level Agreement* - SLA) ou previsão do início e fim das tarefas, ou ainda, os casos utilizados para prever os riscos, como exemplificado por Conforti *et al*. (2016).

A Figura 2 ilustra as etapas de predição em processo de negócio. O processo é dividido em dois estágios. No primeiro estágio (*offline*) o modelo preditivo é aprendido. Para isso, são considerados como entrada o modelo de processo e os seus atributos e os dados históricos de execução do processo (*log* de eventos). O aprendizado do modelo preditivo é feito por algoritmos de mineração de dados. No segundo estágio (*online*), o modelo preditivo aprendido é utilizado para avaliar a instância do processo em execução.

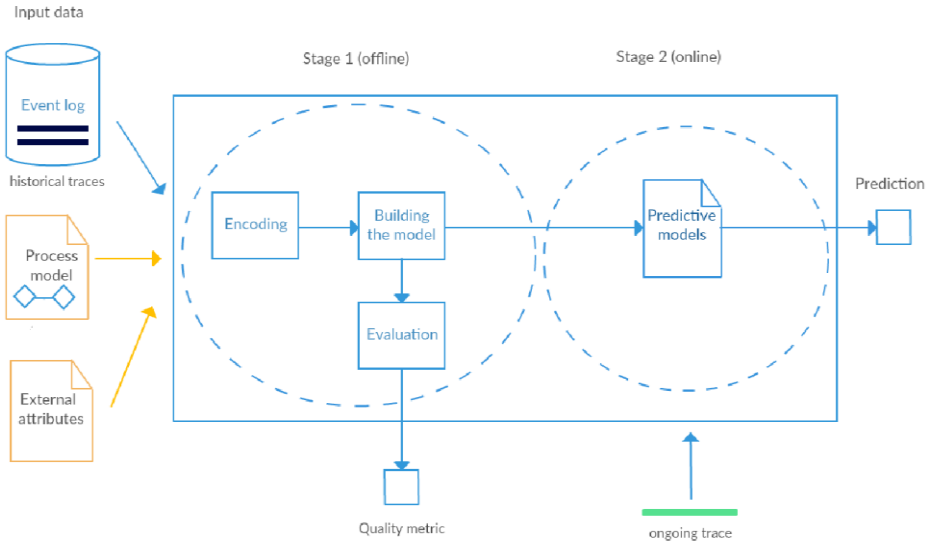

Figura 2 - Predição de processos (Conforti *et al*., 2016).

Um dos desafios do monitoramento preditivo é a escolha das características consideradas para a construção do modelo preditivo. Teinemaa *et al*. (2016) discutiram o benefício da consideração de dados não estruturados combinados com dados estruturados. Na próxima seção detalhamos como o *framework* desenvolvido pelos autores combinou esses diferentes tipos de dados para a predição.

2.3 *Framework* para predição de processos

Teinemaa *et al*. (2016) analisaram como o uso de texto livre, tido como atributo não estruturado do processo e a sua combinação com os dados estruturados, pode contribuir para o monitoramento preditivo de processos de negócio. A proposta apresenta um *framework* capaz de lidar com *logs* de eventos com essas características e seus resultados são comparados às técnicas existentes. Para isso, técnicas de mineração de texto foram utilizadas para extrair informações relevantes de cada instância de processo enriquecendo o *log* de eventos.

Para confirmar se o resultado apresenta melhorias ou não, um estudo de caso foi feito em *logs* de eventos de dois processos reais de uma empresa financeira: (i) processo de recuperação de dívidas, cujo resultado é o reembolso parcial do valor negociado previamente ou o encaminhamento do caso para uma agência externa de cobrança e (ii) um processo de oferta de contratos em que o resultado é a assinatura ou não de um empréstimo do cliente em potencial. O objetivo então, é conhecer se o valor devido será pago no prazo definido, para o primeiro processo e se o cliente fechará um novo contrato, para o segundo. Em ambos os casos, essa previsão dever ser feita em tempo de execução do processo, tendo como entrada o histórico das ações realizadas e das mensagens trocadas até o dado momento.

A primeira etapa do *framework* é a construção de modelos e extração das características de texto. Ambos feitos com base nas mensagens associadas a cada um dos eventos no registro. Diferentes técnicas e parâmetros foram utilizadas para criar os modelos, visando verificar qual apresenta melhor resultado de acordo com as características específicas de cada *log*.

No segundo momento, as características extraídas dos documentos de texto são combinadas com os atributos estruturados, ambos adicionados a um vetor de tamanho fixo e com seus valores representados por números. Isso é feito para que os algoritmos de mineração de dados consigam interpretar suas informações e assim treinar os classificadores para previsão, que é a última etapa do *framework*.

Para estruturar o texto, os métodos utilizados foram o BoNG; Taxas de contagem de *log* com Naïve-bayes (NB) que é baseado no modelo BoNG, mas ponderado com as taxas de registro do NB (Allahyari *et al*., 2017); Alocação latente de Dirichlet, onde o modelo de texto é representado por tópicos abordados pelos documentos; e Paragraph Vector (PV), onde não somente os termos, mas também suas sequências são extraídas para criar o modelo. Os dois últimos são detalhados por Blei (2003). BoNG (*bag-of-n-grams*) parte dos mesmos princípios do modelo *bag-of-words* ou saco-de-palavras, e tem o texto representado por uma coleção de palavras que independem da gramática. Esse grupo pode ser formado separando cada termo independentemente e mostrando a frequência com que eles aparecem. Como exemplo, a frase "Não posso falar agora, te ligo depois", poderia ser representada por {"não":1, "falar":1, "agora":1, "te":1, "ligo":1, "depois":1}. Neste caso, porém, não é possível representar a ordem em que essas palavras estão dispostas no documento. O modelo BoNG soluciona esse problema, uma vez que a seleção das palavras é feita em grupos de *n*-palavras. A

representação do mesmo documento em grupos de 2 palavras ficaria, então, como a seguir {“não posso”:1, “posso falar”:1, “falar agora”:1, ...}.

Como os documentos de texto não apresentam um padrão a ser seguido, são executados o tratamento e a limpeza de dados em cada um dos *logs*, visando manter somente informações relevantes para o contexto. Após essa etapa inicial, o *framework* é divido em dois componentes. O primeiro, apresentado na Figura 3 é *off-line*. Ele realiza a estruturação dos dados, combinando a sequência de ambos os tipos de atributos dos casos no histórico para treinar os classificadores que, no segundo componente (*online*) fará uso desses dados para realizar as previsões dos casos em execução.

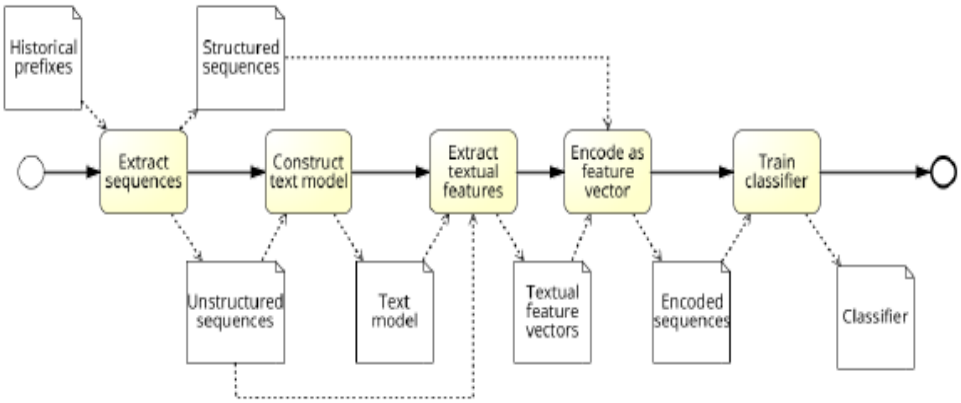

Figura 3 - Componente offline do *framework* (Teinemaa *et al*., 2016).

A outra parte é feita pelo componente *online* definido pelo *framework*, conforme a Figura 4. Nesse momento que é realizada a predição e classificação da instância em execução. Para sua execução, o grau de confiança mínimo precisa ser adicionado como variável de entrada. Quando executado para previsão do resultado de um caso em andamento, com *k* eventos realizados, se a probabilidade apontada pelo classificador for maior do que esse limite, o resultado é apontado como positivo. Enquanto esse requisito não for alcançado, o *framework* continua monitorando as próximas atividades. Se o evento observado for terminal, o resultado da previsão é classificado como negativo.

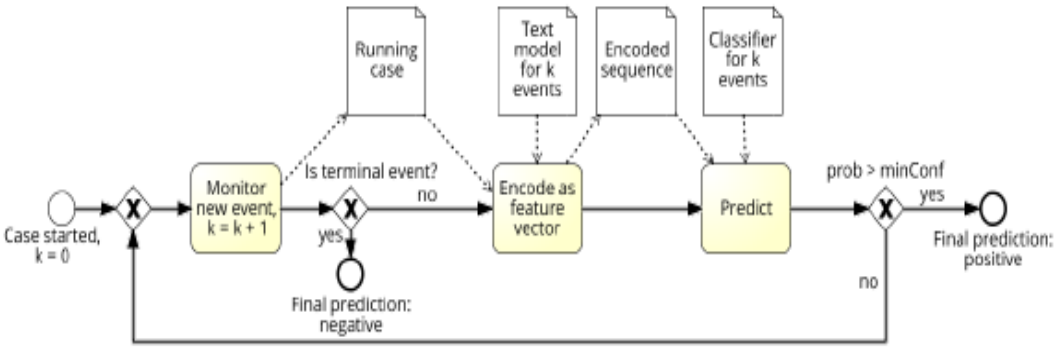

Figura 4 - Componente online do *framework* (Teinemaa *et al*., 2016).

O *framework* foi avaliado com relação a precisão, *recall* e *f-score* comparando um modelo que foi aprendido a partir de dados estruturados com um modelo aprendido a partir de dados

estruturados e não estruturados. Os *logs* de ambos os processos foram divididos aleatoriamente em duas partes. Uma para treinar os classificadores no componente *offline* e a segunda, menor, para testar no componente *online*.

Dois modelos foram considerados floresta aleatória e regressão logística. O primeiro apresentou um resultado mais satisfatório do que o segundo. Além disso, a avaliação do *framework* mostrou que o uso de informações extraídas de texto combinadas aos outros atributos do processo, melhora a previsão da resolução do caso em execução.

Uma das limitações apontadas por esse trabalho foi com relação aos dados não estruturados considerados. Eles eram textos associados ao processo, com uma formatação conhecida, sendo assim uma análise com relação a textos mais heterogêneos ficou em aberto.

3. Trabalhos relacionados

Esta seção apresenta brevemente outras aplicações do monitoramento preditivo em processos para a resolução de problemas. Áreas de nichos completamente diferentes podem fazer uso de técnicas e conceitos muito parecidos.

Maggi *et al*. (2013), por exemplo, combinam a descoberta de processos baseada no controle de fluxo de eventos com a baseada no fluxo de dados, Eles mostram como essa combinação pode ser usada para auxiliar nas decisões dos médicos durante uma análise clínica. Isso é feito com base nas informações de casos similares. Já De Leoni e Aalst (2013) analisaram o *log* de eventos de um processo para lidar com empréstimos solicitados pelos clientes de um instituto de crédito.

Yeshchenko *et al*. (2018) propuseram uma abordagem de integração de elemento de contexto externo aos processos de negócio em métodos de predição. Os autores desenvolveram uma técnica que associa os sentimentos sobre de notícias *online* ao *log* do processo para a tarefa de predição de tempo restante de execução de suas instâncias.

Também inserido no nicho das organizações, sejam privadas ou não, trabalhos são voltados para o monitoramento de riscos, como mostram Conforti *et al*. (2016) e Conforti *et al*. (2013). Outros possuem uma visão mais acadêmica, como o apresentado em Senderovich *et al*. (2016), que usa as redes de petri estocásticas generalizadas (GNSPs) com o objetivo de melhorar a acurácia das previsões.

4. Análise do Método de Predição em Cenário Real

Utilizando como base os resultados do estudo apresentado em Teinemaa *et al*., (2016), esse artigo visa contribuir para a análise de como a combinação de dados estruturados e não estruturados pode auxiliar na eficácia do monitoramento preditivo para processos de negócio. Além disso, textos com estrutura livre foram considerados, estendendo a análise feita em Teinemaa *et al*., (2016).

O *log* de eventos estudado contém registros dos *tickets* de um processo de resolução incidentes na área de Tecnologia da Informação (TI) e redes, de uma empresa de consultoria localizada no Rio de Janeiro. Essas informações foram registradas durante o ano de 2015 através do sistema de Gerenciamento de Serviços de Tecnologia de Informação (ITSM) utilizado pela empresa. Dentro desse *log* encontram-se informações de cada *ticket* e os atributos relacionados a cada um dos eventos executados durante o processo de resolução dos incidentes. Esses atributos podem representar informações do *ticket*, como tipo de

atendimento e a sua prioridade, ou dados relacionados às atividades executadas. No primeiro caso, as informações são coletadas no início da execução do processo e são mantidas durante a execução. Já no segundo, a cada atividade, a informação do atributo pode alterar.

O atributo utilizado para análise da influência do texto no processo é o corpo dos *e-mails* trocados pelos solicitantes do atendimento e os membros das equipes solucionadoras. Caso em determinado *ticket ou e-mails* não tenham sido trocados, então este atributo tem valor zero.

Um atributo extra é considerado para indicar se o processo foi executado dentro da SLA definida ou não. O atributo é denominado MissedSLA, e tem como possíveis valores "0" e "1", respectivamente. Esses valores são obtidos através da comparação do tempo de execução com o prazo pré-estabelecido para as diferentes prioridades.

A seguir, estão detalhados o processo de negócio, o *log* estudado, as etapas de execução do método de predição e os resultados encontrados.

4.1 Processo utilizado

Como mencionado anteriormente, o *log* utilizado apresenta registros dos casos de um processo de tratamento de incidentes (também chamados de *tickets*) relacionados ao setor de tecnologia de informação encontrados pelos usuários da mesma empresa. O objetivo desse processo é executar a operação de resolução dos problemas de maneira eficaz e dentro do prazo estabelecido em contrato com seus clientes.

O processo se inicia quando um usuário de outro setor da empresa percebe algum problema que bloqueie ou atrapalhe seu trabalho e esteja diretamente relacionado com tecnologia. Os diferentes níveis de impacto definem a prioridade dos chamados, e consequentemente suas SLAs.

O processo de tratamento se inicia quando um colaborador da empresa abre um chamado através da ferramenta de gerenciamento de serviços de TI, adicionando informações iniciais como, por exemplo, qual o tipo de problema enfrentado. Esse *ticket* é, então, direcionado para a equipe de nível 1, que faz a triagem das informações e verifica qual equipe de suporte deve ser acionada para resolução do incidente.

Caso o problema possa ser resolvido pela própria equipe, o *ticket* continua na fila da equipe e as ações necessárias são tomadas diretamente por um de seus membros. Caso o problema seja causado por uma área mais específica, o atendente transfere o *ticket* para a equipe que possui capacidade de resolução. Esse *ticket* pode passar pelas demais equipes até sua resolução, caso o problema esteja relacionado a mais de uma área técnica.

Após as atividades consideradas suficientes pelo responsável, o solicitante é notificado com a proposta de resolução e pode confirmar se o problema inicial for resolvido, encerrando as atividades para o caso, ou se ainda existem problemas a serem solucionados . Nesse caso, o *ticket* retorna para quem propôs a solução e permanece nesse ciclo até que a resolução atenda aos requisitos. Quanto menos vezes o chamado precisar voltar para a equipe solucionadora, melhor é o tempo de resolução. Sendo assim, o melhor caso é aquele em que a primeira solução é aceita pelo usuário.

4.2 *Log* de eventos

O *log* de eventos utilizado possui registros da execução de *tickets* do primeiro trimestre do ano de 2015. O arquivo contém informações referentes a 6.337 *tickets* resolvidos durante o

período e as atividades necessárias para resolução de cada um, divididos entre as 84 diferentes classificações de chamado.

Dentre os 6.337 *tickets*, 3.370 foram resolvidos dentro da SLA (SLAMissed=0) e 2.967 fora (SLAMissed=1). Em média, a quantidade necessária de passos para a resolução do problema é de 39. Pode-se observar também os extremos em que, no menor caso, esse número foi de somente 10 e no maior, que precisou de 357 até ser resolvido. A grande maioria dos casos não possui esse número acima de 50. Em uma análise direta, observa-se que quanto maior o número de atividades executadas, maior a probabilidade de ultrapassar a SLA.

Os atributos do *log* considerados para análise estão descritos na Tabela 1, assim como o tipo de dado armazenado.

Tabela 1 - Tipos de dados do *log*.

Nome do atributo	Descrição
CaseID	Número de identificação do *ticket* no registro.
EventID	Número que identifica e representa a ordem em que a atividade foi executada durante o processo.
EventName	Classe que identifica e classifica os diferentes tipos de atividades possíveis no processo.
Priority_ID	Número usado para identificar a prioridade, de acordo com o impacto de cada *ticket*.
ServiceType	Classe que indica o tipo de problema para o atendimento.
a_body	Corpo dos *e-mails* trocados durante as atividades de cada *ticket*.
SLAMissed	Classe que indica para cada caso, se o problema foi resolvido dentro do prazo estabelecido, de acordo com as prioridades.

A data de resolução e abertura dos incidentes não foram utilizadas, uma vez que a coluna "SLAMissed" indica se o caso foi resolvido ou não dentro do tempo esperado, não havendo necessidade de calcular o tempo total da carga de trabalho. Seguindo o objetivo deste artigo de verificar a combinação de dados estruturados e texto livre para a melhoria dos resultados de predição, somente o corpo do *e-mail* foi mantido para a análise. As informações de remetente e destinatário são consideradas dados semiestruturados.

Por possuir diversas informações não relevantes ao contexto do que é tratado durante a resolução dos *tickets*, foi realizada uma limpeza no texto. Apresentações e assinaturas, por exemplo, foram retiradas, assim como dados de telefone dos atendentes e diferentes *links* para o *site* da empresa. Além disso, um pré-processamento foi executado para remover as chamadas "palavras vazias". Essas palavras são importantes para o entendimento do texto entre seus locutores, por serem utilizadas para definir gênero ou ligar as diferentes frases do texto, por exemplo. Mas, para o algoritmo de aprendizado não acrescentam muito valor ao que se é aprendido, uma vez que são palavras muito comuns . Portanto, aparecem com muita frequência nas mensagens. Alguns exemplos são 'e', 'o', 'a', 'em' e 'no'. Além desse, outros tratamentos foram aplicados e estão detalhados na sequência.

4.3 Execução do método de predição

Uma vez que o estudo realiza uma comparação entre diferentes métodos para construção do modelo de texto e extração das características de maior impacto para o processo, o modelo

utilizado foi o *bag-of-words*. Ele apresenta uma melhor desenvoltura nos resultados, em comparação aos demais.

Como o objetivo de aprimorar o processo de predição, após o processamento do texto, um novo atributo foi adicionado, contendo a quantidade de palavras utilizadas durante as mensagens para cada *ticket*. Dessa forma, é possível analisar também como a quantidade de palavras presentes em cada *ticket* afeta a resolução do problema. A Figura 5 ilustra esse comparativo e é possível verificar uma redução no número de *tickets* dentro da SLA quando a quantidade de palavras passa de 1.900.

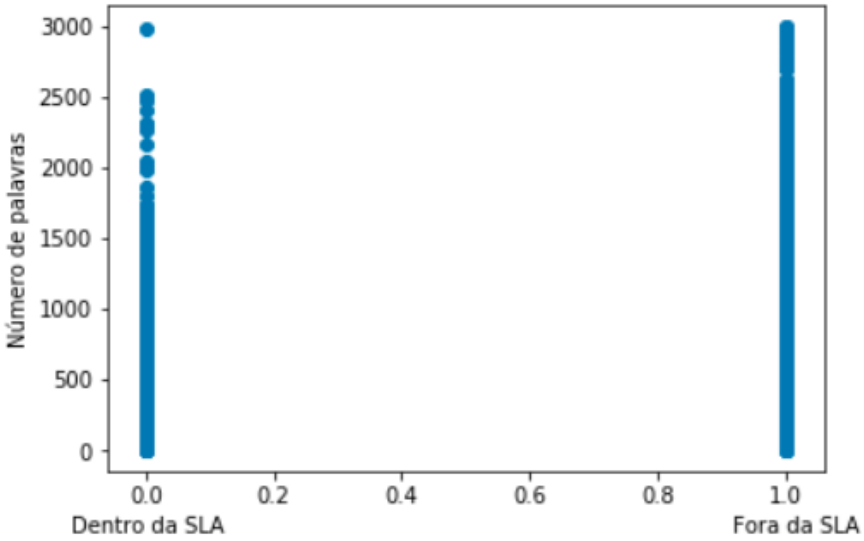

Figura 5 - Classificação por quantidade de palavras.

Essas são algumas das análises diretas que podem ser feitas observando os dados relativos às mensagens de texto trocadas durante um atendimento. Elas exemplificam bem como a troca de informações na forma de linguagem natural pode afetar o resultado de um processo e reforçam a necessidade desse e outros trabalhos sobre o tema. A seguir estão as etapas de execução até o treinamento e teste dos princípios abordados.

4.3.1 Preparação do log

Após o pré-processamento do texto e adição de um novo atributo, os valores nominais de cada atributo foram discretizados permitindo que os algoritmos de mineração escolhidos pudessem ser aplicados.

Após a preparação inicial do *log*, os próximos passos se deram através do processamento do texto e criação do vetor de características. O conteúdo do texto foi utilizado para extração desses vetores. Isso indica que o tamanho do vetor é dado de acordo com o tamanho do dicionário de palavras selecionado. Dessa forma, por se tratar de documentos que podem conter um grande número de palavras, as características extraídas irão apresentar um grande volume, criando vetores muito esparsos, dado que diversas palavras devem aparecer poucas vezes em textos diferentes. Isso reduz o desempenho, pois o algoritmo tem que percorrer cada um e aprender pouco com os vários valores nulos que estão representados.

Visando melhorar o desempenho de processamento e trazer informações relevantes ao conteúdo, uma limpeza e seleção de palavras foi realizada. Ela ocorreu da seguinte forma:

1. Extração do texto não relevante para a resolução dos incidentes: Normalmente, *e-mails* apresentam uma estrutura similar para cada tipo de usuário e empresa, como assinaturas,

sites web, telefones de contato e apresentações. Nessa etapa, o objetivo foi retirar esse tipo de informação, que normalmente não fazem parte do contexto descritivo do problema e nem das atividades realizadas durante o processo. Além de excluir informações que não agregam para a previsão, as retiradas desses termos também auxilia no desempenho do algoritmo, pois o número de palavras lidas é menor. No exemplo a seguir, pode-se ver um corpo do *e-mail* como seria registrado no *log*:

Bom dia!

Não consigo enviar mensagens do meu e-mail corporativo.
Não está funcionando para receber novas mensagens também.

Podem verificar, por favor?

Att,
Nome Sobrenome
Analista de Marketing
Tel.:(xx) 0000-0000

Na sequência, como o mesmo texto ficaria somente após o primeiro filtro das informações:

Não consigo enviar mensagens do meu e-mail corporativo.

Não está funcionando para receber novas mensagens também.

2. Retirada de pontuação: Para que seja possível passar a entonação e consequentemente, dar o sentido desejado ao que está escrito, a pontuação se faz presente nos *e-mails* trocados. Porém, para o modelo de texto a ser criado, esses caracteres não agregam valor, além de aumentarem o número de informação a ser computada. Nesse segundo passo, a tarefa foi retirar esses marcadores antes de separarmos as palavras dos textos.
3. Retirada das palavras vazias: Mais uma vez, para dar sentido a escrita diversas palavras são utilizadas, como conectores ou artigos para definição do gênero, e consequentemente não refletem o conteúdo da conversa. Durante essa etapa esses vocábulos foram retirados. A mensagem passaria a ser representada como a seguir:

Não consigo enviar mensagens e-mail corporativo
Não funcionando receber novas mensagens

4. Criação de *tokens*: Esta etapa é essencial para a criar o modelo de texto, pois é nesse momento que as palavras são separadas e adicionadas às posições no vetor. Como houve a retirada de pontuação, as palavras estão separadas através dos espaços em branco no texto. Para que o algoritmo não computasse separadamente as palavras somente por conter letras maiúsculas, todas foram normalizadas para minúsculo. No exemplo, um vetor x, contendo os *tokens* retirados do texto teria seus valores indicados por x = {‘não’, ’consigo’, ..., ’novas’, ’mensagens’}.
5. *Stemming*: O termo em inglês é utilizado em processamento de linguagem natural, e define a prática de se extrair o radical dos termos presentes no texto. Em outras palavras, as diferentes formas das palavras derivadas da mesma raiz, são todas reduzidas a ela. Para exemplificar, os vocábulos “apresentar”, “apresentação”, “apresentando”, seriam reduzidas à “apresent”, uma vez que somente os sufixos são diferentes. Como cada língua apresenta características específicas, foi utilizada a função *stemm* do Python para

português. Seguindo a nova etapa, as informações no vetor ficariam x = {“não”, “conseg”, ..., “funciona”, “receb”, “nova”, “mensage”}

Esse tratamento das informações contidas nas mensagens de texto foi importante para evitar conteúdo que interferisse no resultado negativamente. Além disso, ajudou a reduzir a quantidade de informação presente nos documentos de texto, melhorando a velocidade de processamento. O uso desses dados para extração do vetor de características e treinamento dos classificadores estão detalhados a seguir.

4.3.2 Extração dos vetores de características

Como cada uma das mensagens está representada por uma lista de palavras, ou *tokens*, o próximo passo foi convertê-las em um vetor de características. Dessa forma, ele pode ser utilizado como entrada pelos métodos de aprendizado de máquina, que não trabalham com texto bruto diretamente. Para isso, foi preciso criar um dicionário com as palavras a serem utilizadas. Como a redução do número de termos apresentou uma perda considerável de predição e baixo ganho no desempenho de processamento, as palavras presentes nos diferentes documentos foram mantidas.

Seguindo as etapas do modelo *bag-of-words*, foram contadas quantas vezes cada uma das palavras aparecem em cada documento. Isso quer dizer que cada uma delas corresponde à uma coluna da tabela e os valores em cada linha representam a quantidade de vezes que elas aparecem no documento associado a cada atividade. Sendo assim, atividades que não tiveram troca da mensagem tiveram seus valores iguais a zero.

4.3.3 Aplicação do TF-IFD

Após criar o vetor de características, o próximo passo foi normalizar os valores encontrados para cada termo. Existem palavras que podem aparecer diversas vezes nos documentos, mas que não adicionam informações tão relevantes para o contexto. Caso não sejam tratadas, essas palavras podem sobrepujar outras que não aparecem tantas vezes, mas que podem ser específicas sobre o domínio estudado, por exemplo.

Para que esses dados não dominassem o modelo com uma alta contagem e influenciassem a previsão de forma negativa, foi utilizada a técnica de pontuação TF-IDF. Essa abordagem compara a frequência que as palavras aparecem nos documentos separadamente com o quão raras são entre todos os outros. Nesse caso, as expressões frequentes no corpo do *e-mail* analisado, e em todos os outros documentos, têm sua pontuação penalizada. O efeito disso é que palavras distintas recebem um peso maior se comparadas àquelas que são muito utilizadas. A partir de então, os termos deixam de ser representados pela sua frequência no documento e passam a ser representados pelo seu valor ponderado.

Após essas etapas, o *log* está pronto para o aprendizado, já que possui seus valores representados por vetores de números de tamanho fixo. A combinação dos valores para as variáveis estruturadas e do texto processado de cada linha foi utilizada pelo modelo como treino e teste para prever o valor da classe associada como resultado. Em outras palavras, o valor de cada linha da coluna “SLAmissed” apresenta uma combinação de eventos, de atributos associados e de palavras presentes no corpo dos *e-mails*, que é utilizada para mapear o comportamento padrão através dos algoritmos de aprendizado de máquina.

4.3.4 Treinamento dos classificadores

Para treinar os classificadores, após toda a etapa de preparação, os registros do *log* foram separados em grupos diferentes, para que o conjunto de dados de treinamento fosse diferente do utilizado para teste e dessa forma, o resultado fosse o mais próximo possível do caso real.

Como linha de base, foram utilizados os resultados obtidos do treinamento e teste utilizando somente as informações disponíveis como dados estruturados. Assim, é possível comparar se a combinação dos diferentes atributos realmente apresenta uma melhora de resultado. Para comparação, outro teste foi realizado considerando somente as características extraídas do corpo dos *e-mails*. As Tabelas 2 e 3 mostram os resultados de precisão, *recall* e *f-score* quando considerando Floresta Aleatória e Regressão Logística.

Esses resultados mostram uma melhora em relação ao aos modelos aprendidos apenas considerando os dados estruturados. Esse comportamento é mais acentuado para os casos positivos.

Tabela 2 - Relatório de classificação - dados estruturados e não estruturados (FA).

	Dados estruturados			Dados estruturados & não estruturados		
SLAMissed	Precisão	*Recall*	*F-score*	Precisão	*Recall*	*F-score*
0	0.62	0.63	0.62	0.69	0.79	0.74
1	0.57	0.56	0.56	0.71	0.61	0.66
Media/Total	0.60	0.60	0.60	0.70	0.70	0.70

Tabela 3 - Relatório de classificação – dados estruturados (RL)

SLA Missed	Dados Estruturados		Dados estruturados & Não estruturados			
	Precisão	*Recall*	*F-score*	Precisão	*Recall*	*F-score*
0	0.53	0.87	0.66	0.70	0.79	0.74
1	0.46	0.13	0.20	0.72	0.62	0.67
Media/Total	0.50	0.52	0.44	0.71	0.71	0.71

Um outro comportamento observado é que o algoritmo utilizado possui um melhor desempenho para analisar os casos que não apresentam desvio. Ou seja, esse método é mais preciso quando o objetivo for monitorar os *tickets* que são finalizados dentro da SLA. Como no contexto em que esta pesquisa está inserida, o objetivo era monitorar e tentar prever os casos que ultrapassarão o prazo inicial (para auxílio a tomada de decisão), esse resultado indica que esse pode não ser o melhor algoritmo, mesmo que a diferença entre os casos não seja tão grande.

A linha de base para esse algoritmo apresenta um resultado muito diferente do que foi obtido com o primeiro. Observa-se uma grande melhora em relação a previsão dos casos reais positivos, *Recall* de 0,87 (comparado aos 0,67 do primeiro algoritmo), mas que ao mesmo tempo apresenta um piora acentuada em relação aos casos falso positivos com *Recall* bem abaixo do primeiro caso, 0,13. Para essa configuração, a regressão logística tende a classificar os casos como não desviantes, o que, na média entre os valores para casos positivos e negativos, faz com que seu resultado acerte em menos do que 50% dos casos (*F-score* = 0,44), sendo pior do que um palpite aleatório.

O resultado da combinação dos atributos também apresenta progresso. Como a linha de base desse algoritmo teve desempenho muito baixo, a melhora é ainda mais significativa.

Assim como o primeiro algoritmo, a regressão logística consegue prever com mais precisão os casos não desviantes. Porém, com uma diferença menor em relação a esse para os *tickets* cuja resolução que fogem da SLA pré-estabelecida. Isso mostra que, apesar de ainda não ser ideal, funciona melhor para previsão em um *log* com as características do contexto desse projeto. Isso pode ser explicado, uma vez que a regressão logística funciona melhor onde os dados são esparsos, que é o caso do modelo *bag-of-words*.

4.4 Discussão dos resultados

Como mencionado anteriormente, algumas características influenciam diretamente no tempo de resolução dos chamados. Um exemplo disso é a quantidade de atividades necessárias até que se chegue à resolução do problema. Além dessa característica presente no processo, as informações atreladas a essas atividades também podem exercer grande influência, como tipo do *ticket* ou a fila de atendimento em que ele está associado, visto que pessoas diferentes operam de modo diferente.

Neste artigo, o objetivo foi verificar como as mensagens trocadas durante a execução de cada *ticket* influenciam no tempo de resolução e entender como alguns dos algoritmos de aprendizado de máquina se comportam para um *log* com características semelhantes. No contexto descrito, a base de dados apresenta uma quantidade balanceada de valores positivos e negativos para o que se deseja prever.

Outro ponto é que diferentemente do apresentado por Teineema *et al*. (2016), o corpo dos *e-mails* associados às atividades é escrito pelos próprios usuários e não seguem nenhuma estrutura padrão. Assim, foi possível estudar as hipóteses levantadas durante o artigo citado para o comportamento em textos mais heterogêneos.

Uma das questões foi que o seu artigo informa que os resultados obtidos com floresta randômica foram melhores quando comparados aos da regressão logística, o que não foi observado em nossa pesquisa. Por mais que a linha de base tenha apresentado um resulto ruim para algoritmo, sua previsão apresenta resultado tão bom quanto o de floresta randômica, ao combiná-los com as informações extraídas dos textos.

Com base nas análises feitas durante a apresentação das matrizes de confusão e relatórios de classificação, observa-se que mesmo com comportamento intermediário diferente para cada um dos casos, previsão dos falsos positivos e negativos, por exemplo, o resultado reforça que a utilização de dados extraídos de texto ajudam na eficiência para previsão de um determinado valor para uma classe, ao final do processo em andamento, já que em todos os casos o valor para *F-score* sofreu grande acréscimo.

5. Conclusões

O principal objetivo da pesquisa foi o de reforçar ou refutar a hipótese de que a combinação de características extraídas de texto livre e atributos estruturados inerentes ao processo, melhora a eficiência na predição de resultado para os casos em andamento. Para alcançar esse objetivo, outros estudos foram utilizados como base da pesquisa, em especial e de Teinemaa *et al*., (2016), que realizou testes com diversos modelos de texto para processamento de linguagem natural e algoritmos de aprendizagem de máquina. Os resultados e análise desse artigo serviram como ponto de partida para as definições do projeto. Um exemplo foi a escolha do modelo de texto a ser utilizado.

Por mais que o comportamento dos algoritmos tenha sido um pouco diferente do mencionado no artigo, os resultados obtidos reforçam a hipótese de que essa combinação melhora consideravelmente a predição em processos de negócio. Isso mostra mais uma vez o quão importante é a área de processamento de linguagem natural dentro do contexto atual, onde a quantidade e velocidade de mensagens trocadas na internet aumenta gradativamente.

Um dos principais pontos de dificuldade ao se trabalhar com texto livre é a quantidade de informação extra existente nos documentos estudados. Para lidar com essas questões, existem diversas técnicas de limpeza, com o objetivo de obter o mínimo possível do conteúdo que apresente dados que sejam relevantes para o processo.

Neste artigo, diversas dessas técnicas foram aplicadas para eliminar as palavras irrelevantes ao contexto. Porém, por ser tratar da troca de *e-mails* entre o suporte e o cliente, muitas das mensagens analisadas possuíam parte da sequência anterior, dentro do mesmo corpo. Caso fosse aplicado para todos os casos, uma solução seria a seleção somente do último *e-mail* trocado. Como isso não se aplica ao *log*, todas as mensagens foram mantidas para que não houvesse a perda de conteúdo, porém essa solução gera tópicos repetidos.

Como trabalhos futuros, serão realizados estudos com outros algoritmos de aprendizado máquina e mineração de texto e em um *log* com características semelhantes. Além disso, será considerada a análise do impacto de descrições erradas dos problemas, durante a resolução.

6. Referências Bibliográficas

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B. and Kochut. K. (2017). A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. In Proceedings of KDD Bigdas, Halifax, Canada, August 2017, 13 pages.

Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer, ISBN 0-387-31073-8.

Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003). Latent dirichlet allocation. The Journal of machine Learning research 3 (2003), 993–1022.

Breiman, L. (2001), Random forests. Machine learning 45(1), 5-32.

Conforti R., Fink S., Manderscheid J., Röglinger M. (2016). PRISM – A Predictive Risk Monitoring Approach for Business Processes. In: La Rosa M., Loos P., Pastor O. (eds) Business Process Management. BPM 2016. Lecture Notes in Computer Science, vol 9850. Springer, Cham.

Conforti, R., de Leoni, M., Rosa, M. L., Aalst, W. M. P. (2013). Supporting risk-informed decisions during business process execution. In: Proc. of CAiSE. pp. 116–132.

Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. (2013). Fundamentals of Business Process Management. Springer.

Fayyad, U; Piatetsky-Shapiro, G; Smyth, P. From Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence, 1996.

Francisco, R. Portela Santos, E. A. (2011). Aplicação da Mineração de Processos como uma prática para a Gestão do Conhecimento. Simpósio Brasileiro de Sistemas de Informação (SBSI). Salvador, BA. 447-484.

Maggi, F.M., Francescomarino, C. D., Dumas, M., Ghidini, C. (2013). Predictive Monitoring of Business Processes. arXiv:1312.4874v2 [cs.SE], 19 Dec 2013.

Oliveira, D., Delbem, A. (2018) Compreendendo e Prevendo o Processo Legislativo na Câmara dos Deputados do Brasil. Simpósio Brasileiro de Sistemas de Informação, SBSI 2018. Caxias do Sul.

Oliveira, M. e Bertucci, M. G. E. S. (2003). A pequena e média empresa e a gestão da informação. Informação & Sociedade: Estudos, João Pessoa, v. 13, n. 2.

Pesic, M., Aalst, W.M.P. (2006). A Declarative Approach for Flexible Business Processes Management. In: BPM Conference 2006 Workshops. pp. 169–180.

Senderovich, A., Shleyfman, A., Weidlich, M., Gal, A. and Mandelbaum, A. (2016). P3-Folder: Optimal Model Simplification for Improving Accuracy in Process Performance Prediction. BPM 2016, LNCS 9850, pp. 418-436.

Teinemaa, I.; Dumas, M.; Maggi, F. M.; Francescomarino, C. D. (2016). Predictive Business Process Monitoring with Structured and Unstructured Data. In Business Process Management, pages 401–417.

Walker, SH; Duncan, DB (1967). "Estimation of the probability of an event as a function of several independent variables". Biometrika. 54: 167–178. doi:10.2307/2333860.

Weiss, S. I., and Kulikowski, C. (1991). Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning, and Expert Systems. San Francisco, California: Morgan Kaufmann.

Yeshchenko A., Durier F., Revoredo K., Mendling J., Santoro F. (2018) Context-Aware Predictive Process Monitoring: The Impact of News Sentiment. In: Panetto H., Debruyne C., Proper H., Ardagna C., Roman D., Meersman R. (eds). OTM 2018 Conferences.