Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Avaliação Semântica da Integração da Gestão de Riscos de Segurança em Documentos de Software da Administração Pública

Rodrigo Nunes Peclat

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador
Prof. Dr. Guilherme Novaes Ramos

Brasília
2015

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

Peclat, Rodrigo Nunes
P364a Avaliação semântica da integração da gestão de riscos de segurança em documentos de software da administração pública / Rodrigo Nunes Peclat; orientador Guilherme Novaes Ramos. -- Brasília, 2015.
124 p.

Dissertação (Mestrado - Mestrado Profissional em Computação Aplicada) -- Universidade de Brasília, 2015.

1. riscos de segurança. 2. linguagem natural. 3. mineração de texto. I. Ramos, Guilherme Novaes, orient. II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Avaliação Semântica da Integração da Gestão de Riscos de Segurança em Documentos de Software da Administração Pública

Rodrigo Nunes Peclat

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. Guilherme Novaes Ramos (Orientador)
Departamento de Ciência da Computação/UnB

Prof. Dr. Rommel Novaes Carvalho
Departamento de Ciência da Computação/UnB

Prof. Dr. José Eduardo Malta de Sá Brandão
Instituto de Pesquisa Econômica Aplicada - IPEA

Prof. Dr. Marcelo Ladeira
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 09 de julho de 2015

Dedicatória

Esse trabalho é primeiramente dedicado à Deus, que me permitiu chegar ao final dele, quando muitas vezes, duvidei da possibilidade de fazê-lo.

Em segundo lugar, é dedicado à minha família, especialmente à minha esposa, Giuliana, que foi obrigada a sofrer com minha "ausência" por conta deste mestrado durante quase dois anos.

Aos chefes da Controladoria-Geral da União que acreditaram, desde o início, na importância deste trabalho, ele também é dedicado a vocês.

Por último, mas não menos importante, dedico este trabalho ao meu orientador – Dr. Guilherme Ramos – pela compreensão e apoio fundamentais em todas as dificuldades enfrentadas.

Esse trabalho é para vocês. Esse trabalho é por vocês.

Agradecimentos

Em primeiro lugar, agradeço à Controladoria-Geral da União por permitir minha participação nesse programa de mestrado.

Em seguida, um sincero agradecimento ao corpo docente da Universidade de Brasília que proporcionou, ao longo deste curso, uma extensão da minha maneira de pensar sobre engenharia de software e seus desafios contemporâneos.

Agradecimento especial aos participantes do *survey* realizado ao longo desta pesquisa, bem como ao Departamento de Segurança da Informação e Comunicações do Gabinete de Segurança Institucional da Presidência da República (DSIC/GSI/PR) e a Secretaria de Logística e Tecnologia da Informação do Ministério do Planejamento, Orçamento e Gestão pelo suporte oferecido.

Por último, agredecimentos também são dirigidos a organizações como OWASP, MITRE, SOFTEX e Fatto Consultoria que permitiram, por meio de materiais disponibilizados on-line, a realização deste trabalho.

Resumo

Software seguro é aquele que tem seus riscos de segurança adequadamente geridos. Por recomendação legal, sua produção é um objetivo a ser alcançado pelos seus gestores de tecnologia da informação da Administração Pública Federal (APF). Entretanto, não apenas o governo brasileiro, como também o mercado nacional enfrentam uma escassez de especialistas nesse domínio, que possam fomentar tecnicamente iniciativas obtenção desse tipo de software em suas organizações. Esse fato aliado ao estado atual das técnicas de mineração de texto, particularmente às relacionadas ao processamento de linguagem natural e à classificação multirrótulo, motivam este trabalho no estudo de uma solução que compreenda a semântica de períodos textuais escritos em português e os associe a riscos de segurança previamente definidos, como os do OWASP *Top Ten*.

Busca-se contribuir para a melhoria de editais de licitação de fábricas de software na APF por prover uma opção computacional, às equipes de elaboração e revisão desses documentos, que permita realizar automaticamente avaliações da integração da gestão de riscos de segurança aos métodos descritos nessas especificações. Após estudar essa solução diante de cerca de 120 mil sentenças extraídas de repositórios como OWASP ASVS e termos de referência de aquisições da Administração Pública brasileira, realizou-se um *survey* junto a grupos de engenharia de software e de segurança da informação coletando a avaliação deles sobre uma amostra desses períodos, permitindo a comparação do seu desempenho em relação à opinião especializada por meio de métricas como Precisão, *Recall*, Perda de Hamming e Previsão de Valores Negativos.

Após uma série de modificações sobre essa solução, chega-se uma versão com uma Perda de Hamming significativamente melhor do que a provida pela opinião especializada, bem como com uma capacidade de previsão da ausência de tratamento de risco em sentenças presentes em editais e termos de referência estatisticamente tão boa quanto à dos especialistas envolvidos nesse *survey*, trazendo assim novas perspectivas para trabalhos futuros no desenvolvimento de uma solução computacional a ser utilizada para a obtenção de software seguro em contratações de fábricas de software.

Palavras-chave: riscos de segurança, linguagem natural, mineração de texto

Abstract

Secure software has its security risks well managed regarding vulnerabilities. IT managers in the Brazilian Federal Public Sector (APF) are legally required to strive for software security. However, only a small number of software security professionals are employed in the technical support for development, maintenance and acquisition of such software in their public organizations. This problem, combined with recent advances in text mining, especially in natural language processing and multi-label classification, motivate this work on research for a computational solution that can understand the semantics of sentences in documents written in Portuguese and connect them to previously defined software security risks, such as OWASP Top Ten.

This solution (A2E) can improve the software factories bidding process of the APF by providing the authors and reviewers of technical specifications with a computational tool which can automatically evaluate the integration between security risks management and software processes described in these documents. After applying A2E to more than 120 thousand sentences extracted from OWASP ASVS and past APF specifications, a survey was conducted to compare its performance with the opinion of software engineers and security specialists through objective metrics like Precision, Recall, Hamming Loss and Negative Predictive Values (NPV).

A2E's final version, after a series of improvements in the development process, obtained a significantly better Hamming Loss measure when compared to the specialists' assessments. Additionally, experiments showed that its NPV is statistically as good as the NPV from the surveyed experts. These results bring interesting new perspectives to future of software security in APF biddings.

Keywords: security risks, natural language, text mining

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

A2E Analisador Automático de Editais. 8

APF Administração Pública Federal. 3

ASVS *Application Security Verification Standard*. 24

BSIMM *Building Security in Maturity Model*. 3

CAPEC *Common Attack Pattern Enumeration and Classification*. 35

COBIT 5 *Control Objectives for Information and related Technology*. 1

CVE *Common Vulnerabilities and Exposures*. 63

CWE *Common Weakness Enumeration*. 2

DHS *Department of Homeland Security*. 46

DISA *Defense Information Systems Agency*. 16

eMAG *Modelo de Acessibilidade em Governo Eletrônico*. 89

ESA *Explicit Semantic Analysis*. 19

FTC *Federal Trade Commission*. 16

ISC^2 *International Information Systems Security Certification Consortium*. 3

KDD *Knowledge Discovery in Databases*. 25

NSA *National Security Agency*. 46

NVD *National Vulnerability Database*. 63

OSCIP Organização da Sociedade Civil de Interesse Público. 46

P3TQ *Product, Place, Price, Time, and Quantity.* 25

PCI DSS *Payment Card Industry Data Security Standard.* 16

PCI SSC *Payment Card Industry Security Standards Council.* 46

SANS *SysAdmin, Audit, Networking, and Security.* 16

SEMMA *Sample, Explore, Modify, Model and Assess.* 25

TF-IDF *Term Frequency–Inverse Document Frequency.* 19

Capítulo 1

Introdução

De acordo com o *Control Objectives for Information and related Technology* (COBIT 5) [1], *framework* voltado para governança e gestão corporativa de tecnologia da informação (TI), o alcance efetivo dos objetivos de uma organização requer uma abordagem holística que considere não apenas seus fatores-chaves, mas também a dinâmica de suas interações. Esses fatores, chamados de habilitadores, distribuem-se em categorias como políticas, cultura, competências institucionais. Ainda de acordo com o referido trabalho, são as diferentes necessidades, expectativas e restrições oriundas das partes interessadas de um negócio que provocam customizações desses habilitadores, explicando em parte a diferença do emprego de recursos de TI em organizações distintas. Entre os possíveis objetivos de uma corporação, o COBIT 5 enumera os seguintes:

- gestão dos riscos de negócio;
- conformidade com leis e regulações externas
- disponibilidade e continuidade do negócio
- conformidade com políticas internas

Um elemento comum entre esses objetivos de negócio é a contribuição primária de uma adequada gestão de segurança da informação [1]. Mais especificamente, identificam-se os principais objetivos corporativos que se beneficiariam de uma iniciativa organizacional de desenvolvimento e manutenção de software seguro, o qual, em termos gerais, é aquele que tem seus riscos de segurança adequadamente geridos [2].

Aparentemente um problema puramente tecnológico, a aquisição, o desenvolvimento ou a manutenção de software seguro não é uma tarefa tecnicamente simples. Essa dificuldade pode ser ilustrada pelo "Dilema do Defensor" [3], o qual envolve os seguintes princípios a serem observados por uma equipe de desenvolvimento, que assume o papel de "defensor" de seu software:

- um defensor deve se preocupar com todos os pontos de ataque, enquanto para um atacante basta priorizar o ponto mais fraco;
- um defensor pode se defender apenas contra ataques conhecidos, enquanto um atacante pode aproveitar novas vulnerabilidades, realizando ataques até então desconhecidos;
- um defensor deve estar em constante vigilância, enquanto um atacante pode agir a qualquer momento;
- um defensor deve observar normas e leis, enquanto um atacante pode agir em desconformidade a elas.

Particularmente quanto ao primeiro dos princípios enunciados, o *Common Weakness Enumeration* (CWE)[1], dicionário de fraquezas de software mantido pelo MITRE[2] (organização sem fins lucrativos que opera centros de pesquisa e desenvolvimento financiados pelo governo americano), apresenta a relação de mais de 700 tipos de padrões comportamentais presentes no *design* e na codificação de um programa que podem proporcionar incidentes de segurança (como um ataque bem sucedido). A diversidade de aspectos e observações técnicas existentes nesse catálogo, bem como a heterogeneidade de relacionamentos estabelecidos entre esses padrões (parentesco, pertinência, precedência e dependência entre fraquezas), ressaltam a complexidade relacionada ao escopo do desenvolvimento e da manutenção desses produtos, exigindo não apenas maiores prazos e orçamentos para projetos de desenvolvimento e manutenção, mas também recursos humanos compatíveis com esses trabalhos [4].

Esse desdobramento do "Dilema do Defensor" ajuda explicar a falsa aparência de problema puramente tecnológico da obtenção de software seguro: a definição dos níveis de aceitação dos riscos de uma organização são decorrência de decisões de suas estruturas organizacionais, as quais por sua vez estão influenciadas pela sua cultura [5]; os ataques contra os quais uma equipe de desenvolvimento poderá defender seu software são aqueles que ela conhece, que ela tem familiaridade; a constante vigilância da segurança do software envolve a obtenção de informação de qualidade, num processo contínuo de monitoramento de incidentes intra e interorganizacionais; a observância do ambiente normativo pode dificultar ou facilitar o trabalho de uma equipe de desenvolvimento de sistemas. Em suma, a engenharia de software seguro, a qual poderia ser pensada inicialmente em termos estritamente técnicos, mostra-se relacionada a diferentes habilitadores corporativos de TI como estruturas da organização, cultura, conhecimento, informação, processos e normas.

[1] `http://cwe.mitre.org`
[2] `http://www.mitre.org`

De fato, essa mesma constatação é apresentada em trabalhos que apresentam modelos de maturidade em iniciativas institucionais para obtenção de software seguro [6, 7]. Particularmente quanto à necessidade do conhecimento técnico mínimo para a viabilização de uma iniciativa corporativa de obtenção de software seguro, há o problema da ausência de especialistas nesse domínio, conforme levantamento presente na versão de 2013 do *Building Security in Maturity Model* (BSIMM) [6]. Segundo esse estudo, as organizações de níveis de maturidade mais baixos na gestão dos riscos de segurança de seus software apresentaram em comum baixo número de especialistas em software seguro, sendo que nenhuma das entrevistadas mostrou ausência desse profissional em seu corpo técnico. Ainda de acordo com esse estudo, em organizações que visam adquirir, desenvolver e manter software seguro, seu grupo de especialistas e de "satélites" (patrocinadores da iniciativa ao longo da organização) apresentam um tamanho cuja mediana é, respectivamente, 7 e 4 pessoas. No caso brasileiro, ao mesmo tempo em que se identifica recomendação expressa do Gabinete de Segurança Institucional da Presidência da República (GSI/PR)[3] por meio da Norma Complementar DSIC/GSI nº 17, de 09 de abril de 2013, para a capacitação em certificações como a *Certified Secure Software Lifecycle Professional* (CSSLP)[4], encontra-se o registro pelo *International Information Systems Security Certification Consortium* (ISC^2) de apenas 19 profissionais brasileiros certificados. Em levantamento recente sobre a situação dos recursos humanos de TI na Administração Pública Federal (APF) brasileira [8], o Tribunal de Contas da União envolveu 245 instituições do Poder Executivo, identificando a carência de especialistas, em geral, nas áreas de TI da APF.

À falta desse perfil profissional nas equipes de software brasileiras, adiciona-se a escassez de ferramentas computacionais apoiando atividades como a identificação de riscos de segurança (e o consequente desenvolvimento dos requisitos para tratá-los), conforme revisão sistemática da literatura sobre o desenvolvimento de aplicações *web* seguras [9]. Nesse trabalho é verificado que a maior parte do apoio automatizado existente se concentra no processo de codificação de software, não sendo compatível com o colocado em [4] sobre a integração da gestão desses riscos ao longo do ciclo de desenvolvimento de software, pois processos tradicionalmente antecedentes à implementação do programa, como desenvolvimento de requisitos e análise arquitetural, beneficiam-se apenas indiretamente desse apoio, não evitando o retrabalho sobre seus produtos (requisitos e arquiteturas, os quais são modificados somente após a identificação de falhas de segurança no produto já implementado, aumentando os custos e o cronograma dos projetos).

Nesse cenário, torna-se desejável a utilização de ferramentas que possam apoiar o trabalho de equipes com reduzido número de especialistas (ou mesmo nenhum), buscando-se

[3] `http://www.gsi.gov.br/`

[4] `https://www.isc2.org/csslp/default.aspx`

reutilizar, por meio de repositórios como o CWE, o conhecimento consolidado de profissionais em segurança de software. A fim de empregar essas ferramentas ainda durante as fases de iniciação e elaboração do sistema, cujos objetivos principais envolvem o detalhamento de requisitos e a análise de arquiteturas candidatas, em que cerca de 80% dos artefatos estão escritos em linguagem natural [10], outro requisito desejável é que essas soluções computacionais sejam capazes de processar textos escritos em idiomas como português ou inglês. Para esse contexto, o emprego de uma abordagem algorítimica que seja capaz de processar esses documentos identificando seus principais conceitos apresenta-se como alternativa, criando, assim, oportunidade para emprego de técnicas semânticas de processamento de linguagem natural.

Essas técnicas diferenciam-se dos tradicionais métodos léxicos (ex: Grep[5]) ao focar no significado subjacente das palavras e das expressões presentes em um texto, em vez de simplesmente focar na estrutura morfológica dessas unidades, conforme explica revisão da evolução paradigmática no campo de Processamento de Linguagem Natural (PLN) [11]. Categoriza-se-se como "PLN Taxonômica" o processamento de textos utilizando bases de conhecimento externas, como o CWE ou o OWASP *Top Ten*. Diante disso, algoritmos como o *Explicit Semantic Analysis* (ESA), apresentado originalmente em [12] como meio de mensuração da similaridade semântica entre dois textos arbitrários usando o conhecimento enciclopédico presente na Wikipedia[6], se apresentam como alternativas para experimentos no domíno de software seguro, permitindo o aumento da compreensão da extensão do auxílio que essa técnicas poderiam trazer às organizações interessadas em gerir os riscos de segurança dos software desenvolvidos e mantidos por meio de serviços de desenvolvimento e manutenção de software, também conhecidos como "Fábricas de Software" [13].

1.1 Definição do Problema

Pelo exposto, a segurança das aplicações apresenta-se como uma necessidade de negócio a ser alcançada, mas que enfrenta dificuldades em várias perspectivas - como a de recursos humanos (os quais são escassos) e a de ferramentas computacionais (as quais são concentradas no processo de codificação de software, não favorecendo, assim, o tratamento tempestivo dos riscos de segurança em software). Assim, uma abordagem automatizada de PLN que reutilize o conhecimento especialista existente no assunto se mostra desejável. Particularmente quanto ao governo federal brasileiro, essa solução pode grande utilidade

[5] `http://www.gnu.org/software/grep/manual/grep.html`
[6] `http://pt.wikipedia.org/`

aos processos licitatórios para aquisição de bens e serviços de TI relacionados à obtenção de software seguro.

Isso, devido ao fato da Administração Pública brasileira observar, em licitações e contratos administrativos, a Lei nº 8.666, de 21 de junho de 1993 [14], a qual traz como condição, para a realização de um procedimento licitatório, a existência de um projeto básico (PB), aprovado por autoridade competente, baseado em estudos técnicos preliminares, o qual consolida as informações sobre o serviço a ser contratado, de modo que, nas palavras dessa Lei, "possibilite a avaliação do custo da obra e a definição dos métodos e do prazo de execução" [14]. Na prática, essa característica dos projetos básicos permite aos fornecedores tomar conhecimento sobre qualquer característica técnica previamente conhecida que onere o futuro contrato, como a exigência de processos adicionais para a gestão de riscos de segurança, num cenário de contratação de fábrica de software. De igual modo, nos termos de referência (TRs), planejamentos que substituem os projetos básicos em licitações por pregão eletrônico, conforme Decreto 5.450, de 31 de maio de 2005 [15], há também a necessidade de inclusão, "clara, concisa e objetiva", dos elementos necessários à avaliação de custo. Assim, pelo disposto nesses normativos, espera-se que as organizações explicitem em seus planejamentos os elementos comuns dos produtos e dos processos de sua fábrica contratada.

Configura-se, então, como problemática, a necessidade de um meio mais eficiente para avaliação da integração de riscos de segurança a planejamentos de serviços de software (TRs ou PB's) nesse contexto de escassez de recursos humanos especializados e de soluções computacionais voltadas para análise dessas especificações. A adição a esse problema tanto da possibilidade de reutilização de enciclopédias como o CWE e o OWASP *Top Ten*, como da possibilidade de emprego de trabalhos em PLN Taxonômica que permitam a implementação desse reuso motiva a seguinte pergunta de pesquisa: Como uma abordagem de PLN Taxonômica pode contribuir junto a equipes de planejamento e revisão de contratações de fábricas de software para a gestão dos riscos de segurança de interesse das respectivas organizações?

1.2 Justificativa

A gestão de riscos de segurança em ativos da informação, como software [16], na Administração Pública Federal brasileira é demandada expressamente pelo GSI/PR, por meio da Norma Complementar DSIC/GSI nº 02, de 13 de outubro de 2008 [17]. Nessa Norma, que define a metodologia de gestão de segurança da informação e comunicações de órgãos e entidades brasileiros, o tratamento adequado desses riscos é colocado como requisito para o planejamento eficaz das ações de segurança, trazendo aos gestores públicos a necessi-

dade de desenvolverem os habilitadores apropriados [1] (ex: recursos humanos, processos, normativos, ferramentas, etc.) para a realização dessa gerência de riscos.

Especificamente à produção de software seguro, a necessidade de integrar a gestão de riscos de segurança ao longo dos ciclos de desenvolvimento e manutenção de software é realçada pela publicação da Norma Complementar DSIC/GSI nº 16, de 21 de novembro de 2012 [18], qual não apenas ratifica as diretrizes da já citada Norma Complementar nº 02, mas também apresenta diretrizes incidentes tanto sobre o desenvolvimento e a manutenção por equipes internas de software, quanto por equipes externas (como no caso de aplicações produzidas por serviços do tipo "fábrica"). Dessa forma, diante de todas as exigências normativas apresentadas e suas implicações sobre o custo do software obtido (dado que o escopo desses projetos se torna maior [4]), torna-se essencial a inclusão nos editais dos elementos necessários para garantir, ao máximo possível, que os software adquiridos estejam em níveis de risco de segurança que possam ser aceitos, de acordo com critérios organizacionalmente pré-definidos.

Adiciona-se a essas exigências a questão da materialidade envolvida nessas contratações. A Figura 1.1, gerada a partir de dados do Portal da Transparência[7], apresenta os gastos com software por encomenda de 2009 a 2014, período em que a referenciada Norma Complementar DSIC/GSI nº 02 esteve em vigor. No período, somente com serviços de customização de software, foram gastos mais de 3,5 bilhões de reais. Assim, é válido questionar se foi exigido desses serviços o emprego de métodos específicos para a gestão dos riscos de segurança dos produtos demandados, ou mesmo para quais riscos se exigiu tratamento, aumentando-se, portanto, a necessidade de analisar os termos de referência (ou projetos básicos) elaborados nesse período.

[7] `http://www.portaltransparencia.gov.br/`

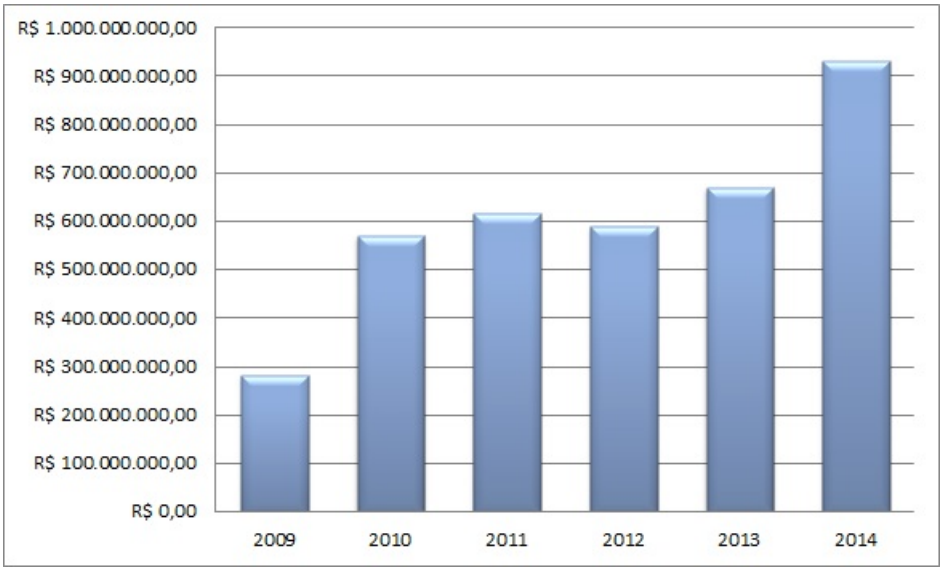

Figura 1.1: Evolução da Materialidade da Aquisição de Software por Encomenda de 2009 a 2014. Fonte: Portal da Transparência[9]

Ao Sistema de Controle Interno do Poder Executivo Federal, de acordo com a Lei nº 10.180, de 06 de fevereiro de 2001, cabe a realização de "auditorias sobre a gestão dos recursos públicos federais sob a responsabilidade de órgãos e entidades públicos e privados". Em decorrência dessa competência, surge a necessidade da avaliação externa da gestão do desenvolvimento, da manutenção e da aquisição de software por essas organizações. O órgão central do referido Sistema - Secretaria Federal de Controle Interno (SFC) - foi incorporado à Controladoria-Geral da União (CGU), pela Lei nº 10.683, de 28 de maio de 2003, trazendo à CGU a necessidade de prover recursos necessários para avaliações mais técnicas desses termos de referência.

Esses fatos justificam a relevância da necessidade de um meio mais eficiente para avaliação da integração da gestão de riscos de segurança a planejamentos de serviços de software (TRs ou PB's), a qual é apresentada como o problema de pesquisa deste trabalho. Salienta-se que a existência de enciclopédias de segurança de software como o CWE e a OWASP[10] e de técnicas computacionais de processamento de linguagem natural como o ESA oportuniza a possibilidade de estudar uma solução computacional que possa prover às equipes de elaboração e revisão de planejamentos de serviços de software de informações sobre a obtenção de software seguro por meio desses planos.

[9] `http://www.portaltransparencia.gov.br/`
[10] `https://www.owasp.org`

1.3 Contribuições Esperadas

A investigação do problema de melhoria da eficiência na avaliação de planejamentos de serviços de software, como TR's e PB's, quanto à integração da gestão de riscos de segurança a esses instrumentos, pode ser considerada a primeira contribuição do presente trabalho. Dessa avaliação, decorre a coleta de informações sobre o domínio estudado, bem como a proposição de diferentes soluções que auxiliam a compreensão dessa problemática.

Em particular, uma proposta de solução denominada Analisador Automático de Editais (A2E) é analisada e submetida a testes que visam simular situações reais de avaliação de termos de referência de contratações de serviços de fábricas de software, realizados pela Administração Pública brasileira. Os resultados dessa análise e dos experimentos a ela associados podem ser considerados como outra contribuição deste texto dissertativo.

Por ultimo, a depender da comparação do desempenho dessa proposta com o de especialistas na área de software seguro, uma última contribuição seria a disponibilização desse protótipo (A2E), bem como de diretrizes para seu uso em verificações dessas especificações de serviços de software, buscando, assim, apoiar suas equipes de elaboração e revisão na produção de documentos técnicos com maior integração entre os métodos de engenharia de software tradicionalmente já exigidos e a gestão dos riscos de segurança incidentes sobre as aplicações produzidas.

1.4 Objetivos

Diante da problemática apresentada e das contribuições esperadas, decorre como objetivo geral a melhoria dessa situação apresentando uma alternativa computacional que auxilie as equipes de elaboração e revisão de especificações de fábricas de software a contornarem parte desse problema, dado que, conforme colocado em [1], sua solução completa envolve o tratamento conjunto de aspectos que envolvem políticas, estruturas, culturas, processos e competências das organizações, os quais estão além do escopo deste trabalho.

A generalidade desse objetivo motiva a proposição de itens específicos a serem atingidos. Três deles se destacam no contexto em análise:

1. propor uma solução computacional para apoio às equipes de elaboração e revisão de especificações de serviços de fábrica de software em suas atividades de avaliação da integração da gestão de riscos de segurança de software a esses documentos;

2. avaliar essa solução de forma a contribuir não apenas com seu desenvolvimento, mas também com o entendimento sobre as dificuldades computacionais no domínio analisado;

3. disponibilizar essa solução, a depender dos resultados dessa avaliação, para grupos interessados em empregá-las em seus trabalhos de elaboração e revisão de especificações.

A fim de alcançar esses três objetivos específicos, intrinsicamente relacionados ao objetivo geral estabelecido, esta dissertação se desenvolverá nos capítulos a seguir.

Capítulo 2

Fundamentação Teórica

No presente capítulo é apresentado o arcabouço teórico que embasa as discussões ao longo deste trabalho. Primeiramente, são referenciados alguns trabalhos correlatos aos conceitos de software seguro e mineração de texto, os quais englobam a maior parte dos fundamentos necessários à proposição de uma solução computacional que torne mais eficiente a avaliação da integração de gestão de riscos de segurança a especificações de serviços de fábrica de software. Entretanto, antes do aprofundamento desses conceitos, conceitua-se o domínio de contratações de soluções de TI pela Administração Pública brasileira, o qual contextualiza o conhecimento expresso nas demais seções deste capítulo.

2.1 Trabalhos Correlatos

Em estudo realizado junto a 67 empresas ao redor do mundo com o objetivo de identificar as diferentes formas que essas entidades tratavam a segurança de seus produtos de software [6], identificou-se 112 atividades distribuídas em 12 práticas, dentre elas uma específica voltada para o desenvolvimento de padrões e requisitos. Um dos resultados apresentados nesse trabalho foi que o comportamento médio das participantes da pesquisa engloba ações de criação e comunicação de seus requisitos e padrões de segurança de software, como a documentação formal deles junto ao público interno e aos contratados relevantes.

Embora esse estudo possa ser utilizado como uma fonte de práticas aceitáveis para o desenvolvimento de software seguro, ele não pode ser tomado como a realidade da indústria. Em *survey* descrito em [19] junto a arquitetos de software espanhóis, foram identificados diversos problemas no que diz respeito à engenharia de requisitos não funcionais e ao papel destes nas arquiteturas desenvolvidas. Particularmente, quanto aos requisitos de segurança, estes foram, de forma unânime entre os entrevistados, não validados durante a produção do software, realçando a deficiência no tratamento de riscos de segurança em aplicações computacionais.

De acordo com [20], recentemente foram apresentados experimentos os quais indicam que a elaboração de requisitos de segurança diretamente vinculados aos objetivos de negócio dos sistemas de informação analisados trazem melhora significativa na acurácia dos riscos identificados. É ressaltada nesse estudo a necessidade de customização de boas práticas de segurança, ou de listas de erros ou riscos, para o contexto em análise. Uma vez que o CWE apresenta uma listagem de cerca de 700 tipos de fraquezas distintas, aumenta-se a necessidade de que a organização, ao contratar o desenvolvimento de seu software, saiba indicar ao contratado quais os riscos relevantes aos processos de negócio envolvidos, a fim de garantir que a equipe de desenvolvimento de sistemas mantenha o foco nas fraquezas mais perigosas para os projetos de software em andamento.

Uma estratégia para incrementar a eficiência na identificação de requisitos em projetos de software, é o apoio de ferramentas computacionais. Estudo sobre o estado da arte em elicitação automatizada de requisitos [10], apresenta a informação de que apenas cerca de 10% dos estudos sobre o assunto tratam da identificam de requisitos não funcionais, como segurança. Além disso, é apresentada uma visão geral, mostrando a predominância de ferramentas semiautomáticas (58,3%) que não utilizam reuso de conhecimento (36,1%) voltadas para a geração de modelos de requisitos (33,3%), os quais devem ser entendidos como aplicações que identificam, classificam e transformam os requisitos para representações de maior abstração (como diagramas). Ferramentas voltadas para a identificação de abstrações em especificações, como conceitos, respondem por cerca de 20% dos trabalhos, enquanto aquelas que reusam conhecimento na forma de bases preexistentes estão relacionadas a 33,3% dos artigos avaliados.

Um primeiro exemplo de solução para classificação de requisitos pode ser encontrado em [21], artigo de 2006, o qual buscou, para cada sentença submetida, a categorização em um dentre 10 diferentes tipos de requisitos (funcional,usabilidade, segurança, etc.). Além da metodologia e do classificador apresentado - baseados em palavras-chaves - disponibilizou ainda uma base de documentos de requisitos provenientes de 15 projetos de desenvolvimento de software, a qual tem sido utilizada de forma recorrente para comparação com o desempenho de novas ferramentas. Nesse sentido, trabalho de 2010 [22], propôs uma nova abordagem que considerasse, além da contribuição de um conjunto de palavras para uma determinada categoria, a ocorrência conjunta de determinados termos. Por último, cita-se proposta de solução computacional presente em [23], a qual consegue melhores resultados do que os algoritmos anteriores ao incorporar em seu processo de classificação o emprego de técnicas customizadas de agregação de sentenças. De fato, conforme registrado, consegue um F-*measure* de 0.382 contra um F-*measure* de 0.239 atingido pelos experimentos de 2006.

Muitas das dificuldades a respeito do reuso desse conhecimento e da identificação de

abstrações em especificações de software estão relacionadas com os desafios relacionados ao processamento de linguagem natural. Em [11], são apresentadas críticas ao foco atual na sintaxe das técnicas desse campo de pesquisa, em vez do foco na semântica ou na narrativa expressa. Particularmente quanto aos métodos de processamento do sentido dos textos, esse trabalho os divide em duas categorias principais: uma delas seria a de técnicas que utilizam de conhecimento externo para o alcance de seus objetivos (PLN Exógeno) e a outra seria a de técnicas que exploram apenas conhecimento intrínseco ao documento analisado (PLN Endógeno). A principal desvantagem da primeira categoria em relação a esta última é a escalabilidade. À medida que o número de conceitos aumenta, torna-se mais difícil viabilizar uma base de conhecimento. Apesar disso, quando a base já existe, permite a realização de trabalhos como o [24], que utilizou a representação de um texto como um conjunto de conceitos sobre a Wikipedia para melhorar os resultados de técnicas padrão de mineração. Esse trabalho exemplifica ainda o emprego de medidas de similaridade semântica entre conceitos presentes em diferentes textos com o objetivo de otimizar os clusters trabalhos pelo autor. A ideia é a partir de um determinado trecho de texto, medir o quão próximo ele está de um conceito em particular.

Uma das medidas utilizadas em [24] em suas comparações foi o *Explicit Semantic Analysis* (ESA). Em [25] é apresentada uma profunda análise desse algoritmo, apresentando as melhorias mais relevantes propostas para ele. O estudo em si propõe adicionalmente a redução da dimensionalidade da base de conhecimento utilizado pela versão clássica do ESA (Wikipedia) para um vetor com cerca de metade de seus conceitos originais. Experimentos indicam que há um ganho relevante no custo computacional em termos de tempo, a uma perda insignificativa de desempenho na medição da similaridade entre documentos.

2.2 Contratações na Administração Pública

De acordo com [26], a Constituição de um país deve ser entendida como sua lei suprema, que orienta todas as suas demais legislações. Na Constituição da República Federativa do Brasil de 1988 [27], é colocado em seu art. 37, inciso XXI, que compras e contratações, salvo casos estabelecidos em lei, devem ser precedidos de procedimento licitatório, que garanta a isonomia entre os concorrentes. Licitação é caracterizada em [28] como um procedimento administrativo, obrigatório para organizações do setor público, que, garantindo a igualdade entre seus participantes, visa selecionar a melhor proposta.

Particularmente quanto à realização desse procedimento, a Lei nº 8.666, de 21 de junho de 1993 [14], que regulamenta o referido art. 37, inciso XXI da Constituição brasileira, estabelece como pré-condição a existência de projeto básico aprovado por autoridade com-

petente, disponível para aqueles interessados em fornecer para a Administração Pública. Esse projeto básico é ainda conceituado em [14] como uma reunião de todos os elementos necessários para caracterizar o serviço a ser contratado, viabilizando a avaliação do seu custo e a definição dos métodos a serem empregados. No caso de bens e serviços de tecnologia da informação, no âmbito do Poder Executivo Federal, em decorrência do art. 9º do Decreto nº 7.174, de 12 de maio de 2010 [29], os critérios utilizados para a escolha dos fornecedores são "menor preço" e "técnica e preço", restringindo este último critério apenas à contratações de objetos predominantemente intelectuais.

Bens e serviços de TI cujos padrões de desempenho e qualidade possam objetivamente ser definidos em edital, estão restringidos, pelo mesmo art. 9º do Decreto nº 7.174, ao critério de menor preço. Particularmente neste caso, em vez de projetos básicos, o instrumento de planejamento denomina-se "termo de referência", os quais, assim como aquele, devem trazer os elementos necessários para que se avaliem o custo do produto (serviço). Em contratações de serviços de desenvolvimento e manutenção de software por meio dessa regra, cresce a importância de projetos básicos ou termos de referências bem elaborados para que o custo do contrato não prejudique a execução das demandas. De acordo com a Instrução Normativa SLTI/MP nº 04, de 12 de novembro de 2010 [30], esses instrumentos de planejamento devem trazer o seguinte conteúdo mínimo:

1. definição do objeto;
2. fundamentação da contratação;
3. descrição da Solução de Tecnologia da Informação;
4. requisitos da solução;
5. modelo de prestação de serviços ou de fornecimento dos bens;
6. elementos para a gestão do contrato;
7. estimativa de preços;
8. adequação orçamentária;
9. definições dos critérios de sanções;
10. critérios de seleção do fornecedor.

No caso desses serviços de desenvolvimento e manutenção de software, cabe a sua equipe de planejamento adequar, a partir de estudos técnicos preliminares [14], cada um dos itens enumerados para o alcance dos objetivos da contratação. Em particular, as contratações de fábricas de software, em decorrência tanto da exigência de procedimentos

relacionados à segurança dos software [31, 18], quanto da necessidade de discriminar os métodos a serem utilizados no fornecimento desses serviços [14], devem trazer consigo elementos suficientes para descrever seu processo de obtenção de software seguro.

2.3 Software Seguro

De acordo com a ISO/IEC 25010:2011 [32] sobre modelo de qualidade de software, segurança é relacionada à proteção contra ações que possam degradar itens de um sistema, sejam elas intencionais ou não. Essa perspectiva da qualidade teria ainda como subcaracterísticas:

1. confidencialidade: relacionada à proteção contra o acesso não autorizado à dados ou às informações geridas;
2. integridade: associada à capacidade de se proteger e de proteger seus dados e informações contra qualquer tipo de alteração indevida;
3. não-repúdio: vincula-se ao grau em que ações ou eventos realizados por meio do software não possam ter sua ocorrência negada;
4. responsabilização: capacidade de um software identificar de forma inequívoca as entidades responsáveis pelas ações realizadas por meio do sistema;
5. autenticidade: pode ser caracterizada como a habilidade de um software em provar que a identidade de uma entidade é realmente verdadeira;
6. conformidade: refere-se à aderência a padrões, boas práticas, normas, etc.

Alem disso, de uma forma resumida, software seguro também pode ser caracterizado como aquele que tem seus riscos de segurança adequadamente geridos [2]. Essa ideia de adequação envolve a integração de atividades especificamente voltadas para a gestão desses riscos ao longo do ciclo de vida desse ativo [5]. De acordo com a ABNT ISO/IEC 27002:2013 [33]a análise de riscos da organização está entre as principais fontes de requisitos de segurança de uma organização. Dessa forma, apresentar-se-á a seguir a conceituação de riscos, requisitos e do relacionamento entre eles.

2.3.1 Riscos de Segurança

De acordo com a ABNT ISO/IEC 27005:2011 [5], risco de segurança pode ser conceituado como uma possibilidade de uma ou mais vulnerabilidades em um conjunto de ativos serem exploradas por uma ameaça. Especificamente no que diz respeito a vulnerabilidades em software, essas são caracterizadas como uma ou mais fraquezas em um produto que

Tabela 2.1: Riscos do OWASP *Top Ten*.

Identificador	**Nome do Risco**
A1	Injeção de Código
A2	Quebra de Autenticação e Gerenciamento de Sessão
A3	*Cross-Site Scripting* (XSS)
A4	Referência Insegura e Direta a Objetos
A5	Configuração Incorreta de Segurança
A6	Exposição de Dados Sensíveis
A7	Falta de Função para Controle de Nível de Acesso
A8	*Cross-Site Request Forgery* (CSFR)
A9	Utilização de Componentes Vulneráveis Conhecidos
A10	Redirecionamentos e Encaminhamentos Inválidos

permitem acesso além do permitido a recursos ou comportamentos (ex: violação de senhas alheias ou autorização de transações críticas por um perfil de usuário incompatível) [34]. As fraqueza por sua vez são definidas pelo MITRE [34], instituição responsável pela manutenção do *Common Weakness Enumeration*, como um padrão comportamental que permite a violação de uma política de segurança (ex: erro de programação - ação humana que produz um código incorreto [35]). Uma prática comum identificada em organizações que alegam manter um programa de produção de software seguro é o uso de listas previamente definidas com riscos ou fraquezas mais relevantes [6]. Entre essas, discute-se a seguir a OWASP *Top Ten* e a CWE/SANS *Top 25*.

OWASP *Top Ten*

O OWASP *Top Ten*[1] é um projeto de conscientização sobre a segurança de aplicações *web* por meio do lançamento periódico de um documento que consolide a opinião de especialistas da OWASP sobre quais os principais problemas nesse tipo de software. As primeiras três versões desse trabalho, que se iniciou em 2003, foram voltadas para vulnerabilidades. As duas últimas, voltaram-se para riscos, os quais, por definição, aumentaram a abrangência da listagem apresentada (devido a cardinalidade 1:n entre riscos e vulnerabilidades).

A versão de 2013, consolidou dados de oito diferentes empresas de segurança de software, envolvendo a análise de mais de 500.000 vulnerabilidades [36]. Nesse trabalho, identificou-se a prevalência dos riscos da Tabela 2.1 na população analisada, além de apresentar uma introdução sobre cada uma dessas categorias, a qual engloba descrições sobre ameaças, fraquezas, vulnerabilidades, impactos técnicos, cenários de ataques, possibilidades de prevenção e referências adicionais.

[1] `https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project`

Apesar da idade desse projeto (mais de 10 anos) e da explícita referência por fontes americanas tradicionalmente envolvidas com segurança computacional como MITRE *Corporation*[2], *Payment Card Industry Data Security Standard* (PCI DSS)[3], *Defense Information Systems Agency* (DISA)[4] e *Federal Trade Commission* (FTC)[5] [36], levantamento publicado em 2013 pela empresa de segurança Veracode[6] [37] (identificada pela consultoria Gartner[7] como uma das líderes do mercado internacional de teste de segurança em aplicações [38]), a partir da análise de cerca de 15.000 aplicações *web*, mostrou que, na primeira submissão ao serviço em nuvem de análise estática de código dessa empresa, os riscos presentes no OWASP *Top Ten* não foram encontrados em apenas 13% delas, o que indica a necessidade de esforços adicionais nas organizações que produzem ou mantêm software para minimizarem a possibilidade de incidentes de segurança bem difundidos como os provocados por essa listagem de riscos.

CWE/SANS *Top 25*

Outra lista de problemas de segurança em software é o CWE/SANS *Top 25*[8], elaborada conjuntamente entre o MITRE e o *SysAdmin, Audit, Networking, and Security* (SANS) *Institute*, a qual apresenta os 25 erros considerados mais perigosos a esse tipo de ativo. Essa conceituação de "perigo" é derivada da análise de três critérios (prevalência, importância e probabilidade de exploração de fraquezas) por profissionais do MITRE e do SANS[9]. Diferentemente do OWASP *Top Ten*, essa lista não é voltada para aplicações *web* (embora alguns dos riscos apresentados pelo trabalho da OWASP possam ser encontrados em projetos de outros tipos de software). Uma descrição dessas fraquezas pode ser encontrada na página do projeto.

O diferencial da listagem apresentada pelo trabalho conjunto do MITRE e do SANS é o suporte do CWE. Enquanto uma descrição de risco no OWASP *Top Ten* traz consigo cerca de 10 diferentes itens, como cenários de ataque e possibilidades de prevenção, uma descrição de fraqueza no CWE, de acordo com a versão 2.8 de seu XML *Schema*[10] (mais recente até então) traz consigo cerca de 30 diferentes atributos, como terminologias, modos de introdução, fatores de exploração e métodos de detecção. Além disso, enquanto os riscos do trabalho da OWASP são apresentados em texto plano, o CWE é estruturado

[2] `http:\www.mitre.org/`
[3] `https:\pt.pcisecuritystandards.org/security_standards/`
[4] `http:\www.disa.mil/`
[5] `https:\www.ftc.gov/`
[6] `http:\www.veracode.com/`
[7] `http://www.gartner.com`
[8] `http:\cwe.mitre.org/top25/`
[9] `https:\www.sans.org/`
[10] `http://cwe.mitre.org/data/xsd/cwe_schema_v5.4.2.xsd`

em uma árvore XML, permitindo a formação de mais de 3000 relacionamentos dos tipos mostrados na Tabela 2.2.

Tabela 2.2: Tipos de Fraquezas do CWE/SANS *Top 25*

Papel no Relacionamento	**Papel da Contraparte**	**Descrição do Relacionamento**
HasMember	*MemberOf*	agregação entre fraquezas
ChildOf	*ParentOf*	especialização/generalização entre fraquezas
Requires	*RequiredBy*	dependência entre fraquezas
StartsWith	*StartsChain*	papel de uma fraqueza dentro de um grupo
CanPrecede	*CanFollow*	sequenciamento de fraquezas

O CWE/SANS Top 25 é proposto em [39] como parte de uma estrutura para mensuração da qualidade de software, no que diz respeito aos seus aspectos de segurança. Assim como o OWASP *Top Ten*, essa listagem de fraquezas é empregada pela Veracode como critério para a liberação do uso de uma aplicação em um ambiente de produção. Segundo essa empresa, 31% das aplicações não *web* submetidas ao seu serviço em nuvem de análise estática deixam de apresentar os erros indicados por essa listagem. Entre os possíveis motivos poderia estar a abrangência, uma vez que, segundo análise da produtora de soluções em análise estática de código Coverity[11] [40], os riscos do OWASP *Top Ten*, presentes como categorias de fraquezas no CWE, são mapeados para cerca de 190 delas, enquanto os erros de software apresentados no CWE/SANS Top 25 se relacionam com um número em cerca de 50% inferior (90 fraquezas). Cabe ressaltar que a Coverity tem como clientes empresas como Symantec[12] e RSA[13], referências em segurança da informação.

Em complemento às noções aqui apresentadas sobre software seguro, em razão das especificações de serviços de software presentes nos TRs e PB's da APF estarem escritas em linguagem natural, uma proposta de automatização da análise desses documentos envolve necessariamente a aplicação de técnicas oriundas da área de "Mineração de Textos", a qual é apresentada a seguir.

2.4 Mineração de Textos

Mineração de textos pode ser conceituada como a conjugação de técnicas oriundas de áreas como mineração de texto, aprendizado de máquina, processamento de linguagem natural e

[11] `http://www.coverity.com/`
[12] `http://www.symantec.com`
[13] `http://www.emc.com/domains/rsa/index.htm`

gerenciamento de conhecimento para resolução de problemas relacionados à identificação de padrões em grande volume de documentos [41]. Comparado à mineração de dados, em que a informação buscada está implícita num conjunto de dados, a mineração de texto volta-se para a identificação da informação expressamente registrada em sentenças e parágrafos, mas que necessita ser adequadamente processada por máquinas, o que pela natureza não estruturada desses dados, traz novas dificuldades a essa espécie de mineração, como a extração de termos ou expressões de interesse [42].

Devido aos textos objeto deste estudo terem sido escritos em linguagem natural, apresentar-se-á a seguir uma visão geral dessa área de conhecimento, bem como de seu paradigma semântico. Na sequência, abordar-se-á a classificação multirrótulos, referenciada ao longo deste texto devido à necessidade de avaliar a pertinência de um trecho de texto com várias categorias de risco. O emprego de uma metodologia de mineração será tratado ao se abordar os últimos três itens desta seção: CRISP-DM, Qualidade de Dados e Avaliação de Desempenho.

2.4.1 Processamento de Linguagem Natural

De acordo com trabalho realizado sobre a evolução das pesquisas na área de Processamento de Linguagem Natural (PLN), caracterizou-se este campo do conhecimento como aquele voltado para a análise e representação automática da linguagem humana [11]. Ressalta-se a necessidade do desenvolvimento das pesquisas nessa área para viabilizar a extração por máquinas do conhecimento presente em fontes textuais [43]. Identificam-se três paradigmas básicos em PLN [11]:

1. sintático: voltado voltado para a extração de conhecimento sobre um texto a partir da identificação de palavras e expressões considerados mais importantes;

2. semântico: ao contrário do paradigma sintático que foca apenas na forma das palavras e das expressões utilizadas, o paradigma semântico busca considerar o significado desses componentes. Abordagens são baseadas em "conceitos", os quais podem ser definidos como unidades de conhecimento em [24] (ex: um artigo da Wikipedia [44]), e podem ser subdivididas basicamente em dois grupos: PLN exógeno, o qual se utiliza de conhecimento externo, como uma Wiki, para extração da semântica de partes de um texto, ou PLN endógeno, o qual se utiliza do conhecimento já presente num corpo de textos para, por meio de técnicas de aprendizagem de máquina, obter a semântica de trechos específicos.

3. pragmático: paradigma mais incipiente dentre os citados que visa proporcionar às máquinas a capacidade de compreensão de narrativas textuais - séries de acontecimentos comunicados por escritos [11].

Afirma-se ainda no citado trabalho sobre evolução das pequisas de PLN que hoje há o predomínio de ferramentas pertencentes ao primeiro paradigma, com uma tendência de crescimento do paradigma semântica. Ao longo deste texto dissertativo, este modelo exerce importância fundamental pela noção de similaridade entre conceitos que ele proporciona. Uma medida dessa similitude é dado pelo *Explicit Semantic Analysis* (ESA) e pela sua versão baseada em grafo (ESA-G), os quais já apresentaram um bom desempenho na correlação de seus julgamentos com aqueles provenientes de avaliadores humanos [24].

Explicit Semantic Analysis

Em [44], propôs-se um algoritmo para medição de similaridade semântica entre textos utilizando-se a Wikipedia como sistema de conceitos - estrutura responsável pela organização destes. Esse algoritmo, denominado de *Explicit Semantic Analysis* (ESA), teve comprovado por meio de experimentos uma a correlação com o julgamento humano superior a de outras medidas concorrentes como a obtida por meio da representação de um texto em vetores de palavras (*bag-of-words*) ou como o *Latent Semantic Analisys* [45]. A ideia principal dessa medida é a representação vetorial de partes de um texto num espaço de conceitos, permitindo, assim, a correlação entre cada uma dessas partes e descrições de ideias previamente escolhidas (como artigos da Wikipedia).

Dessa forma, ainda segundo este último exemplo, considerando que cada artigo do Wikipedia possa ser representado como um vetor num espaço de palavras $\mathbb{P}$, teríamos para cada artigo A_i dessa enciclopédia, $i \in [1, W]$, em que W é a quantidade de artigos da Wikipedia, a seguinte representação vetorial:

$$\overrightarrow{A}_i = (p_{1i}, ..., p_{ni}) \tag{2.1}$$

em que p_{ji} é o peso da j-ésima palavra pertecente a $\mathbb{P}$, com $j \in [1, |P|]$, na representação de $\overrightarrow{A}_i$. Na proposta original do ESA, cada p_{ji} representa o *Term Frequency–Inverse Document Frequency* (TF-IDF) da j-ésima palavra w_j no artigo A_i ($TF - IDF(w_j, A_i)$), o qual é definido nos seguintes termos [41]:

$$TF - IDF(w_j, A_i) = TermFreq(w_j, A_i) * log(N/DocFreq(w_j)) \tag{2.2}$$

em que a função $TermFreq(w_j, A_i)$ é o número de vezes que a palavra w_j se repete no artigo A_i (frequência) e a função $DocFreq(w_j)$ é o número de artigos da Wikipedia que contém essa palavra.

De forma análoga à representação dos artigos da Wikipedia, um trecho de texto genérico T, quando submetido ao ESA, é representado da seguinte forma:

$$\overrightarrow{T} = (p_{1T}, ..., p_{nT}) \tag{2.3}$$

contudo, as representações 2.1 e 2.3 ainda são baseadas apenas nos léxicos presentes no corpo de texto analisado (no caso, a Wikipedia). A contribuição semântica entre cada artigo A_i aparece quando se calcula o cosseno da representação vetorial de T com cada artigo A.

$$\overrightarrow{T}'' = \left(\frac{\langle T \cdot A_1 \rangle}{\|T\| * \|A_1\|}, ..., \frac{\langle T \cdot A_{\text{nº de artigos da Wikipedia}} \rangle}{\|T\| * \|A_{\text{nº de artigos da Wikipedia}}\|} \right) \tag{2.4}$$

essa representação $\overrightarrow{T}''$ é considerada a representação semântica do trecho textual T no espaço de conceitos da Wikipedia. Após representar dois ou mais trechos nesse espaço no espaço, torna-se possível calcular o cosseno entre eles, sendo este valor considerado a similaridade semântica entre desses fragmentos textuais.

Uma visão geral do ESA é colocada na Figura 2.1 [44]. Observar que o mapeamento dos textos T para os artigos da Wikipedia é feito pelo seu interpretador semântico. Em seguida, outro componente realiza a comparação vetorial no novo espaço de representação.

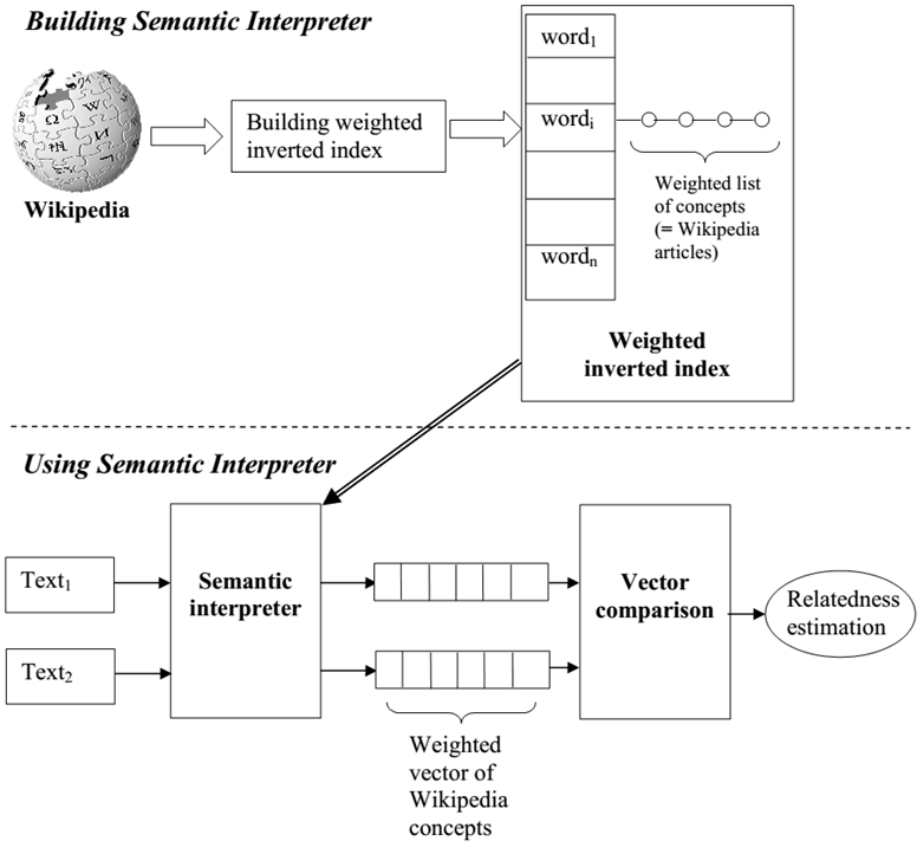

Figura 2.1: Visão geral do ESA

ESA-G

Conforme exposto em [44], o ESA, ao contrário de algumas outras medidas de similaridade semântica baseadas no Wikipedia, não explora a estrutura de *links* presentes em cada artigo, desconsiderando, assim, as ligações entre eles. A fim de aprimorar essa medida, é proposto em [46] o emprego, juntamente com o ESA, de uma técnica denominada "caminhada aleatória" (*random walk*) em grafos, originando a medida denominada de ESA-G (ESA *Graph*). A ideia seria, a partir da representação fornecida pelo ESA e de um grafo do sistema de conceitos utilizado (como a Wikipedia) mostrando a existência de arestas entre os nós (artigos) dessa enciclopédia, mapear um trecho de texto T para um vetor $\overrightarrow{T}_{\text{ESA-G}}$ cujas coordenadas representam a distribuição da probabilidade de que, ao falarmos de T, estejamos falando de um certo artigo dessa enciclopédia.

De forma objetiva, considera-se que, ao submeter o trecho T ao ESA-G, obtém-se, primeiramente, a representação $\overrightarrow{T}''$, dada pela Equação 2.4. A partir disso, considerando-se uma matriz quadrada M_{nxn}, em que n é o número de artigos da Wikipedia e um fator de amortecimento c, obtém-se $\overrightarrow{T}_{\text{ESA-G}}$ da seguinte forma [47]:

$$\overrightarrow{T}_{\text{ESA-G}} = cM * \overrightarrow{T}_{\text{ESA-G}} + (1 - c)\overrightarrow{T}'' \tag{2.5}$$

Em trabalho de clusterização de conceitos que abrangeu a comparação entre algoritmos de similaridade semântica [24], é mostrado que a correlação com o julgamento humano obtido pelo ESA-G pode ultrapassar o desempenho obtido pelo ESA. Detalhes mais aprofundados sobre métodos alternativos de se calcular a Equação 2.5 podem ainda ser encontrado em [48].

Uma das possíveis aplicações para o ESA (e o ESA-G) é a classificação de textos [49]. No domínio em estudo, um texto pode se relacionar, simultaneamente, a vários diferentes rótulos, caracterizando a então denominada "Classificação Multirrótulo" apresentada a seguir.

2.4.2 Classificação Multirrótulo

Em análise sistemática de medidas de desempenho em tarefas de classificação [50], é apresentada a seguinte diferenciação entre essas atividades:

1. classificação binária dado duas classes C_1 e C_2, uma determinada instância submetida ao classificador pode ser associada a apenas uma delas;

2. classificação multiclasse uma extensão da classificação binária em que se tem n classes (C_1 à C_n), em que n $\in$ {conjunto dos números naturais maiores que 2};

3. classificação multirrótulo ao contrário do classificador multiclasse que pode associar à apenas uma dentre n classes, o classificador multirrótulo pode associar, ao mesmo tempo, uma instância a quaisquer das categorias disponíveis;

4. classificação hierárquica uma instância, inicialmente, deve ser classificada em apenas uma categoria C_j (j $\in$ {conjunto dos números naturais}), dentre n disponíveis. Num segundo momento, ela é associada a categorias hierarquicamente subordinadas a C_j.

Dentre as atividades citadas, a de classificação multirrótulo exerce importância em campos como categorizaçao de textos, categorização de músicas em emoções, elaboração de ações dirigidas de *marketing* e anotações de significados em imagens e sons [51]. De acordo com revisão de algoritmos de aprendizagem multirrótulo [52], são duas as categorias em que esse tipo de classificação se subdivide:

1. transformação de problemas em que um problema de classificação multirrótulo é alterado para vários problemas de regressão ou classificação binária;

2. adaptação de algoritmos em que um algoritmo previamente existente de classificação binária ou multiclasse é modificado para suportar dretamente a classificação multirrótulo.

Ainda de acordo com [52], uma prática comum nesse tipo de classificação é a calibração de limiares com o objetivo de dividir os rótulos associados a uma instância em dois conjuntos: relevantes ou irrelevantes.

Emprego de Limiares

De acordo com [53], são três os tipos básicos de limiares a serem utilizados numa tarefa de classificação:

1. limiar a nível de categoria SCUT (*score cut*): para cada possível categoria, é estabelecido uma pontuação mínima que servirá de limiar para identificar as instâncias relevantes (instância x é relevante para a $Categoria_i \iff f_i(x) \geq limiar_i$, em que f_i: {conjunto de instâncias} $\rightarrow$ {conjunto dos números reais} é uma função que atribui a cada instância uma pontuação com respeito à $Categoria_i$).

2. limiar a nível de instância RCUT (*row cut*): para cada instância submetida ao classificador, é calculado um limiar específico, sendo relevantes as instâncias que excederem esse limite de corte (x é relevante para a $Categoria_i \iff f_i(x) \geq limiar_{instancia_x}$)

3. limiar a nível de proporção PCUT (*proportion cut*): para cada possível categoria, é definido previamente um número p de instâncias esperadas. Dessa forma, uma instância será relevante para aquela categoria se estiver entre as p de maior pontuação.

De acordo com trabalho sobre a otimização de limiares para uso por classificadores multirrótulos [54], é apresentado um algoritmo de otimização de limites SCUT a partir da maximização de sua medida micro *F-measure* [51], correspondente à média harmônica da Precisão e do *Recall* desse procedimento. Doravante, no corpo deste texto, os limiares obtidos por essa técnica serão chamados de "micro-SCUT".

No tocante aos limiares t obtidos a nível de instância, uma abordagem frequente descrita em [52] é a suposição de serem descritos de forma linear pela seguinte equação:

$$t(x) = \langle \overrightarrow{W} \cdot \overrightarrow{f(x)} \rangle + \mathrm{b} \tag{2.6}$$

em que f é definida como f_i : {conjunto de instâncias} $\rightarrow$ {conjunto dos números reais}^{n}, em que n é o número de categorias, tal que f_i, $1 \leq i \leq n$, é a i-ésima coordenada de $f(x)$, correspondente à pontuação obtida para a $Categoria_i$, restando, assim, $\overrightarrow{W}$ (um vetor de n pesos - um para cada categoria) e b, respectivamente variáveis dependentes e independente, para serem determinados. Ainda de acordo com [52], isso pode ser feito pela aplicação dos métodos dos mínimos quadrados para resolução do seguinte problema:

$$\underset{\{\overrightarrow{W}, viés\}}{min} \sum_{i=1}^{\text{número de instâncias}} (\langle \overrightarrow{W} \cdot \overrightarrow{f(x_i)} \rangle + \text{viés} - s^2(x_i)) \tag{2.7}$$

em que:

$$s(x_i) = \arg\min_{a \in \Re}(\|y_j | y_j \in Y_i, f(x_i, y_i) \leq a\| + \|y_k | y_k \in \bar{Y_i}, f(x_i, y_i) \geq a\|) \tag{2.8}$$

o que, em outras palavras, é uma função que identifica o limiar a ótimo para uma dada instância. Reparar que a é o valor que minimiza o número de erros na divisão do conjunto $\{f_1(x), ..f_n(x)\}$ em dois subconjuntos: C_1 - soma do número de rótulos incorretamente atribuídos, mas que estão acima do limiar; C_2 - soma do número de rótulos incorretamente não atribuídos, mas cujas pontuações estão abaixo do limiar.

Diante do exposto, o cálculo dos limiares RCUT pode ser sumarizado pelo Algoritmo 1.

Algoritmo 1 Cálculo de limiares RCUT

Entrada: k instâncias $x_i, i \in 1, ..., k, k \in N$, e função pontuação f

1: Obter a representação $f_i = (y_1, ..., y_n)$ de cada instância, $n \in N$
2: Para cada instância, obter seu respectivo $s(x_i)$, Equação 2.8, obtendo $S = (s(x_1), ..., s(x_k))$
3: Formar matriz M em que sua i-ésima linha corresponda à $f(x_i)$
4: Concatenar como (n+1)-ésima coluna de M a coluna $c = (1, ..., 1)$
5: Resolver a equação: $MT = S$, em que $T = (w_1, ..., w_n, b)$
6: Para cada nova instância p, seu limiar RCUT será: $t(p) = (f_1(p), ..., f_n(p)) \cdot (w_1, ...w_n) + b$

Para exemplificar melhor a diferença entre a utilização de limiares micro-SCUT e limiares RCUT, considere o exemplo a seguir: deseja-se representar a sentença "verificar se a flag HTTPOnly é usada em todos os cookies que não requerem, especificamente, acesso do JavaScript", presente no OWASP *Application Security Verification Standard* (ASVS) [55], em um espaço com nove diferentes dimensões. Cada uma dessas dimensões têm um diferente limiar SCUT. Por exemplo, a aplicação da técnica [54] referida nesta Subseção traz para a dimensão 1 o valor de 0.1727542, enquanto para a dimensão a dimensão 4 o valor de 0.2899110. A execução do Algoritmo 1 permite a identificação do limiar RCUT de valor 0.1215475. A Tabela 2.3 resume esses valores.

Tabela 2.3: Representação vetorial de uma sentença e limiares SCUT e RCUT

Dimensão	**Representação Vetorial**	**Limiares micro-SCUT**	**Limiar RCUT**
D1	0.12005231	0.1727542	
D2	0.12931900	0.1828478	
D3	0.21528875	0.2152887	
D4	0.11783063	0.2899110	
D5	0.05251930	0.1158384	0.1215475
D6	0.08866075	0.1133172	
D7	0.15627696	0.1720086	
D8	0.12005231	0.1858632	
D9	0.00000000	1.0000000	

Como se pode visualizar na Tabela 2.3, tomando-se os limiares micro-SCUT como base, somente a categoria associada à dimensão D3 seria associada à sentença representada, uma vez que apenas a terceira coordenada desse vetor é maior que o limiar micro-SCUT correspondente. Contudo, caso tomássemos o limiar RCUT discriminado nessa tabela como referência, a sentença em questão seria associada às categorias relacionadas às dimensões D2, D3 e D7 por essas coordenadas serem maiores que o valor RCUT

apresentado. De fato, o período textual em análise está associado aos riscos D2 - "Quebra de Autenticação", D3 - *"Cross-Site Scripting"* e D7 - "Falta de Função para Controle de Acesso".

As técnicas até aqui discutidas necessitam ser integradas ao longo de um arcabouço de processos de trabalho que tragam confiabilidade metodológica a um projeto de mineração de texto. Nesse contexto, na sequência, apresenta-se o CRISP-DM.

2.4.3 CRISP-DM

Para a realização confiável de um projeto de mineração, incorporando atividades e produtos reconhecidamente necessários e suficientes, torna-se importante adotar um *framework* de processos adequado. A fim de avaliar essa adequação, é apresentado em [56] uma comparação entre opções como *Knowledge Discovery in Databases* (KDD), *Sample, Explore, Modify, Model and Assess* (SEMMA), *CRISP-DM* e o *Catalyst* (*Product, Place, Price, Time, and Quantity* (P3TQ)), a qual caracteriza CRISP-DM como uma das opções de maior detalhamento. Ainda de acordo com [57], ele não apenas é o *framework* de maior utilização em todo o mundo, como também num mapeamento para 14 diferentes metodologias de mineração, é identificado como base para aquelas elaboradas a partir do ano 2000.

O CRISP-DM, conforme [58], em sua versão 1.0, constitui-se de duas partes: um Modelo de Referência que apresenta uma visão geral de suas fases, tarefas, saídas e um Guia de Usuário, o qual, por sua vez, apresenta orientações sobre a execução de um projeto de mineração, trazendo sugestões para as fases e tarefas previstos naquele Modelo. No âmbito desse *framework*, a mineração de dados pode ser decomposta nas seguintes fases:

1. Entendimento do Negócio: foco no entendimento dos objetivos e requisitos do domínio do problema;
2. Entendimento dos Dados: a ênfase dessa fase é a obtenção de impressões iniciais acerca dos dados disponíveis para a mineração;
3. Preparação dos Dados: visa a construção da base de dados a ser minerada;
4. Modelagem: orientada para a seleção de técnicas de mineração com a associada otimização de seus parâmetros para a solução do problema em questão;
5. Avaliação: envolve a revisão do alcance dos objetivos de negócio inicialmente estabelecidos, como também a revisão dos passos do CRISP-DM até então executados; e
6. Implantação: Foco na entrega do produto que pode ser desde um simples relatório final ao estabelecimento de um processo de mineração num cliente.

Devido à maior desestruturação presente nos dados de uma mineração de textos em relação à mineração tradicional, projetos deste tipo apresentam maior ênfase no entendimento e na preparação desses documentos [42].

2.4.4 Qualidade dos Dados

Conforme colocado em [59], a maior parte dos trabalhos voltados para a qualidade de dados está voltada para dados estruturados como tabelas e estatísticas. Em publicação voltada para a qualidade da informação [60], são apresentados exemplos de dados não estruturados como planos estratégicos e relatórios de desempenho avaliados segundo critérios distribuídos em três grandes categorias:

1. qualidade intrínseca: extensão na qual as informações correspondem à realidade;
2. qualidade contextual: extensão na qual a informação é adequada para o contexto em que seu destinatário se encontra; e
3. segurança/acessibilidade: extensão na qual a informação está disponível.

Em trabalho específico sobre a qualidade de dados utilizados em projetos de mineração de dados, são apresentadas na Tabela 2.4 [61] categorias e subdimensões que, além de incorporarem as categorias acima, serão utilizadas ao longo deste texto dissertativo para avaliar documentos de interesse para as atividades de mineração executadas.

Ressalta-se que, conforme colocado em [60], nem sempre é conveniente a aplicação de todos esses critérios a um objeto para sua avaliação. Cada cenário de uso ensejará a adoção de um subconjunto de dimensões da Tabela 2.4.

2.4.5 Avaliação de Desempenho

Um dos processos do CRISP-DM é o de modelagem. Segundo [58], em seu âmbito os testes de qualidade são projetados e realizados sobre os modelos propostos. Duas questões fundamentais que permearam essas discussões foram a definição das instâncias a serem testadas e as métricas utilizadas para mensuração do desempenho dos modelos propostos.

Estratificação

De acordo com [42], para a previsão do desempenho de um classificador é necessário que seu desempenho seja aferido sobre um conjunto que não tenha sido utilizado em sua formação, o qual ele denomina de conjunto de teste. Além disso, apresenta em complemento as noções de conjuntos de treinamento e validação, os quais destinam-se respectivamente

Tabela 2.4: Categorias e subdimensões da qualidade da informação

Categoria	**Subdimensão**	**Definição**
Qualidade Intrínseca	Acurácia	Ausência de erros
	Objetividade	Correspondência para a realidade
	Credibilidade	Extensão em que a informação é plausível
	Reputação	Julgamento das pessoas, em geral, sobre a qualidade da informação
Qualidade Contextual	Relevância	Conexão com o assunto em questão
	Valor Adicionado	Capacidade de aprimorar um produto
	Tempestividade	Entrega no momento correto
	Completude	Existência de todos os elementos necessários
	Quantidade de Informação	Existência de informação útil na quantidade adequada
Qualidade Representacional	Interpretabilidade	Capacidade de explicar o significado
	Facilidade de Entendimento	Capacidade de compreender o significado
	Concisão	Ausência de detalhes desnecessários
	Consistência de Representação	Ausência de contradições
Acessibilidade	Acesso	Facilidade para obter a informação
	Conveniência	Facilidade de uso da informação
	Segurança	Salvaguarda da informação contra sabotagem, crimes ou ataques

para aprendizado e para otimização de parâmetros. Juntamente a esses conceitos, é desejável que esses conjuntos sejam representativos da população de instâncias em questão. Nesse contexto, a intersecção entre os elementos utilizados para teste e os presentes nos demais grupos é conhecida como ressubstituição [42], a qual enviesa de forma otimista a aferição de desempenho.

A fim de dividir uma população de instâncias previamente definida, de forma a permitir o treinamento (ou validação) e teste no desenvolvimento de um classificador multirrótulo, torna-se necessário separar esses conjuntos da melhor forma possível, buscando-se a representatividade e a independência desses grupos. Conforme exposto em [62], a possibilidade de um exemplo estar associado de forma concomitante a mais de um categoria, traz uma dificuldade adicional para a referida divisão.

A divisão inadequada do conjunto pode causar um desbalanceamento das classes entre eles, o que pode gerar enviesamento do classificador e consequentemente queda de desempenho quando exposto a novas situações [63]. A própria definição de balanceamento em conjuntos multirrótulos é discutida: refere-se ao balanceamento das categorias individualmente ou ao balanceamento das possíveis combinações entre elas? Supondo n classes iniciais , tem-se 2^n arranjos, o que traz a dificuldade adicional do tratamento de um número exponencial de estratos. Assim, em domínios como o de identificação automatizada de requisitos, de acordo com as considerações sobre a escassez de instâncias previamente rotuladas [22], a aplicação desta última abordagem pode se tornar indesejável.

Métricas

Na revisão de classificadores multirrótulo apresentada em [52], as métricas de desempenho são sumarizadas em dois grandes grupos: baseadas em exemplo e baseadas em rótulos. Como o próprio nome diz, o primeiro conjunto consiste de medidas que têm como objeto erros ou acertos na atribuição de categorias à uma instância (ex: precisão de 60% na classificação de um exemplo). Já o segundo se volta para medições a nível das classes disponíveis (ex: precisão de 60% na classificação para uma certa categoria).

A fim de apresentar as medidas utilizadas ao longo deste trabalho, considere que x_i é a i-ésima instância do conjunto de exemplos disponibilizado, $L = \{y_1, ...y_q\}$, $q \in \mathbb{N}$ é o conjunto dos q possíveis rótulos a serem atribuídos à x_i, $h(x_i) = \{h_1(x_i), ..., h_m(x_i)\}$ é o conjunto dos m rótulos, $m \leq q$, atribuídos à x_i pelo classificador e $Y_i \subset L$ é o conjunto dos rótulos que realmente estão associados à x_i. Sejam ainda $\bar{h(x_i)}$ e $\bar{Y_i}$, respectivamente, os complementares de $h(x_i)$ e Y_i em relação a L. Assim, listam-se as seguintes métricas por instâncias.

1. *Precisão*$_i$, *Recall*$_i$ e F-*score*: essas medidas são definidas em [51] da seguinte forma (p é o número de instâncias analisadas):

$$Precisão_i(h) = \frac{1}{p}\sum_{i=1}^{p}\frac{|Y_i \bigcap h(x_i)|}{|h(x_i)|} \tag{2.9}$$

$$Recall_i(h) = \frac{1}{p}\sum_{i=1}^{p}\frac{|Y_i \bigcap h(x_i)|}{|Y_i|} \tag{2.10}$$

$$\text{F-}score(\beta, h) = \frac{(1+\beta^2)\cdot Precisão_i(h)\cdot Recall_i}{\beta^2 \cdot Precisão_i(h) + Recall_i(h)} \tag{2.11}$$

A primeira delas, a Precisão, refere-se à quantidade de acertos do classificador, enquanto o *Recall* refere-se à quantidade de *labels* relevantes retornados por ele. Essa diferença pode ser exemplificada ao analisarmos instância x_i associada aos rótulos $y_k, y_w, y_s \in L$, cujo $h(x_i) = \{y_k\}$. Aqui, observa-se que, embora a precisão do classificar seja 1, o *Recall* é de apenas 1/3, pois a maior parte dos resultados pertinentes não foram identificados por h.

O F-*score* é medida que combina a Precisão e o *Recall* em uma só. A métrica em questão tem um parâmetro associado - $\beta \in \Re$ - o qual se relaciona ao balanceamento entre Precisão e *Recall*: se $0 \leq \beta < 1$, então ocorre um privilégio da primeira; se $\beta > 1$ priviligia-se a segunda. No caso em que $\beta = 1$, obtém-se equilíbrio entre as medidas citadas e o F-*score* corresponde à média harmônica delas (também denominada F-*measure*). Ressalta-se ainda que, em estudo sobre a otimização de limiares em classificadores multirrótulos [54], é descrito um algoritmo para determinação desses valores (SCUT) de forma a maximizar o valor dessa medida.

2. Perda de Hamming: essa medida é definida em [52] conforme a seguinte equação:

$$hloss(h) = \frac{1}{p}\sum_{i=1}^{p}\left|\frac{h(x_i) \triangle Y_i}{q}\right| \tag{2.12}$$

em que $\triangle$ (distância simétrica) representa a soma dos elementos dos rótulos erroneamente atribuídos $S_1 = |h(x_i) - Y_i|$ com a soma dos rótulos erroneamente esquecidos $S_2 = |Y_i - h(x_i)|$ e q representa o número total de rótulos no espaço analisado. Supondo que uma certa instância x seja representada num espaço multirrótulo $\{y_1, ..., y_5\}$ tal que $h(x) = \{y_1, y_2, y_3\}$ é o conjunto de rótulos retornado pelo classificador e $Y = y_3, y_5$ é o conjunto de rótulos realmente associado à instância. De acordo com a definição, temos $\left|\frac{h(x_i)\triangle Y_i}{q}\right| = 3/5$. Observar ainda que, no pior caso ($h(x)$ e Y disjuntos), essa diferença será igual a 1. Salienta-se ainda que, podendo se escrever $|A - B|$ (A, B conjuntos) como $|A \cap \bar{B}|$, torna-se possivel afirmar que, enquanto a Precisão, o *Recall* e o F-*score* focam na intersecção entre $h(x_i)$ e Y_i, a

Perda de Hamming considera ainda os complementos desses conjuntos ($\overline{h(x_i)}$ e $\overline{Y_i}$) na medição do desempenho do classificador.

3. *Negative Predictive Value*$_i$: a partir da definição presente em [64] e da consequente analogia com a Precisão, o (NPV_i) pode ser formulado da seguinte forma:

$$NPV_i(h) = \frac{1}{p}\sum_{i=1}^{p}\frac{|\overline{Y_i}\bigcap\overline{h(x_i)}|}{|\overline{h(x_i)}|} \tag{2.13}$$

Um exemplo pode ser utilizado para mostrar o funcionamento dessa medida. Seja mais uma vez uma instância x, classificada para três rótulos $h(x) = \{y_1, y_2, y_3\}$, mas associada realmente à apenas dois $Y = y_3, y_5$. A classificação proposta, ao mesmo tempo que prevê três relacionamentos, também descarta outros dois possíveis $\overline{h(x)} = \{y_4, y_5\}$. A ideia principal do NPV é medir a "precisão" dessa não-classificação. Considerando que os rótulos realmente não associados à instância em questão são $\overline{Y} = \{y_1, y_2, y_4\}$, temos que o NPV para esse exemplo é 1/2. O resultado diz que, nesse caso específico, apenas metade dos rótulos não previstos pelo classificador realmente não estão relacionados a x.

As medidas anteriormente apresentadas focam o desempenho do classificador h por instâncias. Outra forma de mensurar esse categorizador seria pelo resultado obtido em cada uma das q classes. De acordo com [51], essas métricas se dividem em macro e micro médias. Considerando que em cada categoria existe um ou mais dos seguintes membros:

1. verdadeiros positivos (vp): quantidade de instâncias x_i que são atribuídas de forma correta a uma categoria;
2. verdadeiros negativos (vn): quantidade de instâncias x_i que corretamente não são atribuídas a uma categoria;
3. falsos positivos (fp): quantidade de instâncias x_i que são atribuídas incorretamente a uma categoria;
4. falsos negativos (fn): quantidade de instâncias x_i que incorretamente não são atribuídas a uma categoria.

e que a Precisão e o NPV em uma classe específica $y_j \in L$ são funções dessas quatro quantidades, dadas respectivamente por:

$$Precisao_{y_j}(vp_{y_j}, vn_{y_j}, fp_{y_j}, fn_{y_j}) = \frac{vp_{y_j}}{fp_{y_j} + vp_{y_j}} \tag{2.14}$$

$$NPV_{y_j}(vp_{y_j}, vn_{y_j}, fp_{y_j}, fn_{y_j}) = \frac{vn_{y_j}}{fn_{y_j} + vn_{y_j}} \tag{2.15}$$

temos que medidas como como Precisão e NPV (designadas genericamente como B) são dadas, em suas versões macro e micro, respectivamente, por:

$$B_{macro}(h) = \frac{1}{q}\sum_{t=1}^{q} B\left(\frac{vp_{y_t}}{fp_{y_t} + vp_{y_t}}\right) \tag{2.16}$$

$$B_{micro}(h) = B\left(\sum_{t=1}^{q} vp_{y_t}, \sum_{t=1}^{q} vn_{y_t}, \sum_{t=1}^{q} fp_{y_t}, \sum_{t=1}^{q} fn_{y_t}\right) \tag{2.17}$$

O seguinte exemplo pode ajudar a compreender a diferença entre essas médias. Considere que se deseja calcular a Precisão a nível de categoria sobre os resultados apresentados na Tabela 2.5. De acordo com a Equação 2.14 e a Equação 2.16, tem-se:

1. categoria 1: Precisão = 1/2;
2. categoria 2: Precisão = 2/3;
3. categoria 3: Precisão = 0/2;

o que resulta numa macro precisão igual a média aritmética dessas três, o valor de 2/9.

Em oposição, a tabela em questão também poderia ter a Precisão derivada de uma micro média conforme a Equação 2.17. Neste caso, precisaríamos somar todos os verdadeiros positivos e falsos negativos apresentados para então calcular a métrica desejada. Neste caso, tem-se um total de 3 vp's e 4 fp's, resultando em uma micro-Precisão igual a 3/7.

Tabela 2.5: Representação matricial dos resultados obtidos por um classificador sobre três instâncias

Instância	**Categoria 1**	**Categoria 2**	**Categoria 3**
Instância 1	vp	fp	-
Instância 2	fp	vp	fp
Instância 3	-	vp	fp

Observe que enquanto as medidas macro atribuem igual peso a todas as classes, a medida micro privilegia na contagem aquelas que têm maior número de representantes, conforme expresso em [50]. Assim, no projeto de um classificador, a opção por apenas uma delas revela a importância atribuída nos experimentos às diferentes classes neles presentes.

Por último, ressalta-se que não há relacionamento direto entre as métricas por categoria apresentadas e as métricas por instância anteriormente mostradas. Por exemplo, é teoricamente possível que um classificador tenha uma alto desempenho m relação a uma categoria e um baixo desempenho nas demais. A micro e a macro média é que proporcionarão uma ideia geral da métrica em questão sobre todas as categorias.

Survey

De acordo com [65], um *survey* não pode ser confundido com um questionário, *email*, ou um simples intrumento, consiste em um processo de descrição, comparação ou explicação de conhecimentos, atitudes e comportamentos. Segundo comparação entre métodos empíricos em engenharia de software presente em [66], *surveys* são empregados para a descrição de uma população.

Em trabalho específico sobre a utilização de *surveys* em engenharia de software [65], esse conceito é descrito como um processo que compreende as seguintes atividades:

1. fixação de objetivos mensuráveis;
2. planejamento;
3. obtenção de recursos;
4. projeto do *survey*;
5. preparo do instrumento de coleta;
6. validação do instrumento;
7. seleção de participantes;
8. administração do instrumento de coleta;
9. análise dos dados;
10. apresentação dos resultados.

Especificamente quanto ao *design* desse processo, três objetivos são colocados em [67]: resistência ao viés, adequação e otimização do custo. O primeiro deles busca obter resultados o mais próximo possível da realidade. Já o segundo orienta quanto à complexidade do processo para não torná-la excessiva. O terceiro, diretamente relacionado com o segundo, exige que a administração do processo não esgote os recursos disponíveis.

Quanto à avaliação dos questionários empregados na realização de *surveys*, trabalho de avaliação de questionários apresentado em [68], apresenta as seguintes estratégias:

1. validade de face - realizada por diretamente por julgadores externos, os quais, muitas vezes, não têm suficiente treinamento nesse tipo de atividade. Sua subjetividade termina por fragilizá-la como estratégia de validação;

2. validade de conteúdo - envolve uma revisão detalhada, por especialistas no domínio em análise e por respondentes, do escopo do questionário. Quando se trata de um instrumento novo, numa área de pesquisa ainda sem artefato similar, esta estratégia se torna a única forma preliminar aceitável de validação;

3. validade de critério - busca comparar o instrumento objeto de análise com outros similares emitidos na mesma área de estudo;

4. validade do constructo - forma de validação que avalia a convergência ou a divergência de diferentes conjuntos de dados coletados segundo o mesmo instrumento. Geralmente exige um tempo maior de experiência para ser realizada.

Ainda em [67] é discutida a questão de determinação do tamanho da população a ser avaliada e das taxas de resposta em um *survey*. Particularmente quanto à administração por meio eletrônico, como a Internet, é exposto o problema do baixo índice de respostas e apresentada uma discussão sobre a adequabilidade de uma taxa de resposta de cerca de 20%.

Capítulo 3

Solução Proposta

A fim de estudar o desempenho de uma abordagem semântica de processamento de linguagem natural na identificação de períodos de texto relacionados a certos riscos de segurança, torna-se necessário, inicialmente, definir qual método será utilizado, bem como o porquê de sua escolha. Além disso, para viabilizar experimentos sobre essa abordagem, de modo a estudar seu comportamento em cenários como os de processamento de sentenças reconhecidamente de segurança e de processamento de termos de referência de contratações de fábricas de software, exige-se uma implementação que seja funcionalmente aplicável a essas situações de uso. Assim, este capítulo inicia-se pela exposição dos motivos que levaram à opção pelo ESA, para, em seguida, propor a adaptação desse algoritmo, instanciada pelo protótipo chamado "A2E", que seja voltada para a classificação de textos em categorias de risco previamente estabelecidas.

Registra-se ainda que os códigos-fontes envolvidos no desenvolvimento da solução proposta e dos experimentos a serem apresentados no próximo capítulo se encontram em `https://github.com/rnpeclat/A2E`. Além disso, uma proposta arquitetural dessa ferramenta se encontra no Apêndice A.

3.1 Abordagem Semântica e Escolha do ESA

Em trabalho sobre a evolução dos estudos na área de processamento de linguagem natural [11], é apresentado que as ferramentas de PLN predominantemente em uso dissociam a representação léxica do conceito representado (por exemplo, uma abordagem puramente sintática seria capaz de identificar o relacionamento entre as palavras "autenticar" e "autêntico" pelo compartilhamento de radical, mas não seria capaz de associá-las à expressão "controle de acesso"). Essa limitação é superada, em grande parte, por ferramentas desenvolvidas sob o paradigma semântico de PLN, que busca associar significados a essas representações textuais. A existência de bases de conhecimento especialista sobre o do-

mínio de segurança de software como o CWE e a OWASP permite que novas soluções em PLN sejam projetadas incorporando essa expertise, categorizadas em [11] como "PLN Taxonômica".

O conhecimento chave no domínio do problema é o de gestão de riscos de segurança de software, os quais podem ser associados a várias fraquezas distintas. Um risco como "Injeção de Código" pode advir tanto de aspectos da geração de uma página *web* (CWE-79), quanto de proteções inexistentes a nível de sistema operacional (CWE-78). Analogamente, um risco como "Exposição de Dados Sensíveis" relaciona-se não apenas com deficiências no tráfego de informação sensível em claro pela Internet (CWE-311), como também a regras de negócio sigilosas inadequadamente inseridas no código fonte de uma aplicação (CWE-312). Um conjunto aparentemente pequeno de riscos, como os do OWASP *Top Ten*, pode estar relacionado a quase 200 fraquezas distintas, conforme visto no Capítulo 2, exigindo um saber na equipe que participará da gestão desses riscos que pode ser encontrado em enciclopédias não estruturadas, como as publicadas pela OWASP, e, até mesmo, em enciclopédias semiestruturadas (disponibilizadas em XML) como o CWE e o *Common Attack Pattern Enumeration and Classification* (CAPEC)[1]. Uma proposta geral da incorporação dessas bases de conhecimento em soluções de PLN pode ser encontrada em [41].

O ESA é um exemplo desse tipo de proposta, ao apresentar a vantagem de permitir a utilização de um conjunto de conceitos previamente determinado, como a Wikipedia, para a internalização de conhecimento de especialistas num assunto em particular. Embora o trabalho original sobre o ESA [44] tenha usado esse repositório de conhecimento, a literatura registra a utilização de outros repositórios de dados, como mostrado em [69], pelo uso do Reuters (relacionado aos textos de notícias), ou [24], pelo uso do HE50 (também relacionado a um conjunto de notícias).

Além disso, há a questão do desempenho do ESA, diante de outras medidas de similaridade semântica. Em [24], é apresentado o desempenho superior dessa medida, e do ESA-G, em termos de correlação (ρ-*pearson* [70]) quando comparado ao obtido por outras medidas como o *Latent Semantic Indexing* (LSI) [45], o cosseno entre duas representações baseadas em *bag of words* e o julgamento realizado por especialistas. De fato, conforme é afirmado em [24] com respeito à base HE50, o ESA e sua variação se mostram mais consistentes com a média das classificações atribuídas por julgadores humanos do que os próprios julgadores em si. Um exemplo de experimento que ilustra essa afirmação pode ser encontrado em [71]. A Tabela 3.1 [24] apresenta essa comparação entre as diferentes técnicas de cálculo de similaridade.

[1] `http://capec.mitre.org/`

Tabela 3.1: Comparação da Consistência com o Julgamento Humano Obtida por Diferentes Métodos de Cálculo de Similaridade.

Método	ρ **de Pearson**
Julgamento entre avaliadores	0.6
Bag of Words	0.42
LSI	0.6
ESA	0.72
ESA-G	0.77

Essa indicação de superioridade do ESA e da sua variante ESA-G aliada ao fato de que, dentre os métodos presentes na Tabela 3.1, esses dois são os mais voltados para a internalização do conhecimento de enciclopédias como OWASP *Top Ten* e CWE, motivam a escolha inicial desses algoritmos para a produção de representações de períodos em espaços vetoriais de conceitos pelo A2E. Salienta-se que o melhor desempenho em um domínio (como o de notícias em língua inglesa exemplificado pelo HE50) não é garantia de superioridade em um outro domínio, como o de editais e termos de referência de fábrica de software escritos em português. Contudo, a escassez de registros do emprego de soluções semânticas de processamento de linguagem natural neste domínio permite adotar, os resultados da Tabela 3.1 não como uma comprovação de superioridade, mas como um indicador de que o ESA e o ESA-G podem ser as melhores escolhas para o domínio em questão.

3.2 Adaptações do ESA para uma Versão Inicial do A2E

Com a finalidade de incorporar a utilização do ESA ao A2E, buscou-se realizar adaptações nos seus dois componentes principais: o interpretador semântico do ESA e o seu classificador. A seguir, são apresentadas considerações a respeito desses ajustes.

3.2.1 Adaptação no Interpretador Semântico do ESA

Como já colocado, a proposta presente em [44] utilizou a Wikipedia para a formação de seu sistema de conceitos. No A2E, a Wikipedia foi substituída por um repositório voltado para o domínio de segurança de software. Para os experimentos a serem realizados neste trabalho, utilizaram-se a versão brasileira do OWASP *Top Ten* e uma proposta de tradução do SANS/CWE *Top 25*, as quais trouxeram redução da custo de memória exigido pela execução do A2E: enquanto, no trabalho de [44], se utilizou um índice mapeando cerca de 300.000 termos para cerca de 170.000 artigos, o A2E mapeia cerca de 2000 termos

para cerca de 10 conceitos. Para cada um desses mapeamentos, seja na versão original do ESA, seja nesta adaptação, a associação entre termos e conceitos é quantificada pelo TF-IDF dessa relação.

Dessa forma, o índice do interpretador semântico do ESA pode ser representado como uma matriz $M_{TermosxConceitos}$ em que sua entrada M_{ij} significa o TF-IDF do i-ésimo termo dessa matriz em relação ao j-ésimo conceito presente no sistema de conceitos utilizado, por exemplo, o OWASP *Top Ten*.

3.2.2 Adaptação no Classificador do ESA

A proposta inicial do ESA não tem um classificador, dado que seu propósito original não é a categorização de documentos e sim a medição de similaridade semântica entre eles. Em [49], apresenta-se uma das primeiras propostas de classificação com o ESA sem um conjunto de treinamento explícito, baseando-se apenas no conhecimento incorporado em seu sistema de conceitos para realizar categorizações. Como o problema apresentado na Seção 1.1 envolve a análise de fragmentos de texto e não a simples comparação entre dois ou mais trechos de documentos, uma das primeiras mudanças introduzidas no ESA foi a de retornar, no lugar do cosseno entre dois textos, o conceito que mais se aproxima semanticamente do texto submetido ao A2E. Dessa forma, cada sentença é associada a um conceito específico.

Entretanto, como já colocado na Seção 3.1, uma sentença pode se relacionar a mais de um risco de segurança, o que expõe uma limitação da abordagem acima. Assim, duas opções surgem como candidatas para a constituição de uma versão inicial do A2E, entre elas:

1. retorno dos conceitos correspondentes aos k maiores cossenos: Para um $k \in \mathbb{N}$ previamente definido, o classificador do ESA retorna a lista ordenada dos conceitos correspondentes aos k maiores cossenos relacionados ao texto analisado;

2. retorno com base em limiares: Para a representação de um texto num espaço de conceitos, o classificador do ESA somente associa esse trecho a uma categoria específica caso o limiar utilizado permita. Conforme estudo sobre o assunto apresentado em [53], são estratégias básicas de limiares o RCUT, o SCUT e o PCUT. Neste trabalho, o emprego do PCUT tornou-se prejudicado pela escassez de estimativas confiáveis quanto à proporção de sentenças associadas a cada um dos riscos do OWASP *Top Ten* em editais e termos de referência de contratações de serviços de software da Administração Pública brasileira. Entretanto, as estratégias RCUT e

SCUT apresentam-se como alternativas viáveis, uma vez que podem ser derivadas a partir da análise do OWASP ASVS[2].

3.3 Funcionamento do A2E

De acordo com as considerações sobre alterações iniciais a serem realizadas no ESA, a estrutura geral do A2E, a partir da proposta original de [44], pode ser vista na Figura 3.1.

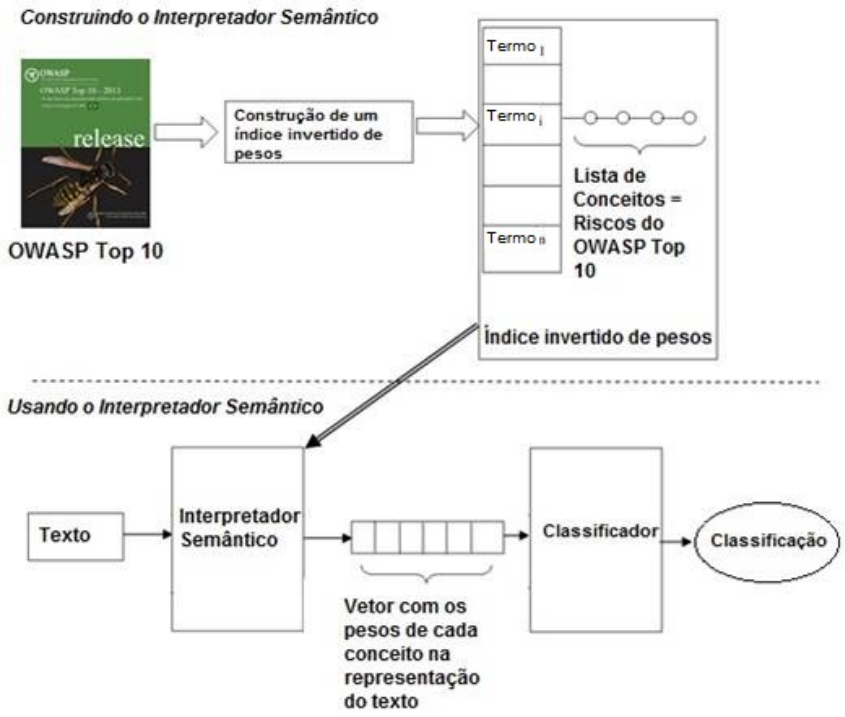

Figura 3.1: Visão geral da proposta inicial de ajuste do ESA usada pelo A2E

Uma vez que o diagrama acima é derivado do ESA, ele pode ser explicado utilizando-se os conceitos apresentados na Subseção 2.4.1, especificamente pode ser melhor compreendido comparando-se a Figura 3.1 com a Figura 2.1. Inicialmente, observa-se que, em vez de termos a Wikipedia como sistema de conceitos, altera-se para uma enciclopédia voltada exclusivamente para riscos de segurança de software, no caso a OWASP *Top Ten*. Ela será utilizada para se formar um índice termos para conceitos $M_{termosXconceitos}$, em que cada célula M_{ij} representa o TF-IDF do i-ésimo termo presente no corpo de textos do OWASP *Top Ten* em relação ao j-ésimo conceito. Essa matriz é necessária para computar a representação de um trecho de texto submetido ao A2E, o qual primeiramente é tratado como um vetor de termos, no espaço semântico almejado. Para finalizar, este novo vetor

[2] `https://www.owasp.org`

é encaminhado ao classificador, o qual, com base nas suas coordenadas, irá associá-lo a uma ou mais das possíveis categorias presentes no sistema de conceitos utilizado.

Esse funcionamento também é ilustrado no Algoritmo 2.

Algoritmo 2 Classificar Texto em Categorias do OWASP *Top Ten*

Entrada: Texto T e matriz de termos por conceitos $M_{termosXconceitos}$

1: pré-processamento de (T)
2: representação de (T) como um vetor $\overrightarrow{T}$ num espaço de termos
3: cálculo da representação ($\overrightarrow{T''}$) num espaço de conceitos
4: classificar, a partir das k maiores coordenadas de $\overrightarrow{T''}$, o texto T em k categorias

Sobre o Algoritmo 2 ressaltam-se os seguintes pontos:

1. o pré-processamento explicitado no primeiro passo abrangerá funcionalidades como a homogeneização da formatação de T e a sua preparação para os passos seguintes;
2. a representação de T realizada no Passo nº 2 é em seu espaço de termos (palavras) que o compõem. Dessa forma, tem-se ao final dele uma representação $\overrightarrow{T} = (a_1, ..., a_{termos}), termos \in N$, em que termos é o número total de diferentes palavras presentes nesse texto;
3. ao fim do Passo nº 3, obtém-se uma outra representação vetorial de T, denominada $\overrightarrow{T''}$, no espaço de conceitos previamente escolhido, obtendo-se:

$$\overrightarrow{T''} = \left(\frac{\overrightarrow{T} \cdot M_{i1}}{\|\overrightarrow{T}\| * \|M_{i1}\|}, ..., \frac{\overrightarrow{T} \cdot M_{iconceitos}}{\|\overrightarrow{T}\| * \|M_{iconceitos}\|} \right) \tag{3.1}$$

em que $\frac{\overrightarrow{T} \cdot M_{ij}}{\|\overrightarrow{T}\| * \|M_{ij}\|}$ é o cosseno [72] entre a representação vetorial de T e a representação vetorial do j-ésimo conceito de M.

3.4 Pontos de Variação do A2E

De acordo com [73], pontos de variação são recursos arquiteturais que objetivam dotar uma certa arquitetura de flexibilidade suficiente para buscar uma melhor configuração a partir de um arcabouço básico. Nesse sentido, diante do exposto sobre a estrutura do A2E, incluindo o registrado no Apêndice A, são as seguintes opções arquiteturais a serem testadas ao longo dos experimentos a serem realizados no Capítulo 4:

1. ponto de variação no interpretador semântico:

(a) uso do ESA ou do ESA-G para medição da similaridade semântica entre termos e categorias de risco;

(b) uso do OWASP *Top Ten* ou do CWE/SANS Top 25 como sistemas de conceitos utilizados para representação vetorial das sentenças, seja num espaço de palavras-chaves, seja num espaço semântico de conceitos;

2. ponto de variação no classificador:

   (a) uso de limiares a nível de categoria (SCUT) ou limiares a nível de instância (RCUT).

Assim, no próximo capítulo, buscar-se-á, dentre as opções apresentadas, aquela de melhor desempenho.

Capítulo 4

Experimentos

O presente Capítulo traz uma visão resumida dos experimentos realizados com o objetivo de identificar uma versão do A2E que se mostre útil à análise da integração da gestão de riscos de segurança em planejamentos de contratações de fábricas de software. Inicialmente, tópicos relevantes da metodologia utilizada são abordados, para, em seguida, se aprofundar a descrição das atividades realizadas e dos principais resultados obtidos ao longo de duas fases de trabalho: uma voltada para a avaliação de sentenças textuais oriundas da área de segurança de software, outra com foco em editais e termos de referências de licitações já realizadas pela Administração Pública brasileira. Por fim, um *survey* com especialistas em segurança da informação e em desenvolvimento de software é detalhado coletando subsídios para a comparação do desempenho do A2E com o desempenho desses profissionais na tarefa de classificação multirrótulo de períodos de texto em riscos de segurança previamente definidos.

4.1 Metodologia Utilizada

O projeto de desenvolvimento do A2E foi basicamente estruturado em duas grandes fases: uma voltada para a correta identificação de requisitos de segurança de software, outra de adaptação do algoritmo obtido ao fim da Fase 1 para a identificação de sentenças relacionadas a riscos de software em editais e termos de referência de serviços do tipo fábrica de software. Para o ciclo de vida desse projeto, optou-se por uma estratégia iterativa e incremental [74] de execução dos macroprocessos do CRISP-DM [58], visando atingir os objetivos propostos. A Tabela 4.1 apresenta uma visão geral sobre as atividades realizadas ao longo das iterações do projeto.

Cabe ressaltar que os resultados do macroprocesso “Entendimento do Negócio”estão registrados principalmente no capítulo “Introdução” desta dissertação, no qual os objetivos e a avaliação do contexto foram discutidos. Além disso, registra-se que quanto ao

Tabela 4.1: Estrutura analítica das fases do projeto de desenvolvimento do A2E

EAP - Fases 1 e 2			
Entendimento dos Dados	**Preparação dos Dados**	**Modelagem**	**Avaliação**
Coleta de Dados	Limpeza de Dados	Projeto de Testes	Avaliação dos Resultados
Descrição dos Dados	Formatação de Dados	Construção de Modelos	Revisão de Processos
Exploração dos Dados	-	Avaliação de Modelo	Identificação de Trabalhos Futuros
Avaliação da Qualidade dos Dados	-	-	-

macroprocesso "Implantação", o presente trabalho não visa, num primeiro momento, a disponibilização de uma ferramenta em sua versão final, limitando suas entregas ao relatório final que se consubstancia neste texto dissertativo. Por último, com a finalidade de privilegiar a execução das principais atividades deste projeto, optou-se por registrar apenas os itens considerados necessários à repetibilidade e à análise dos experimentos.

4.2 Análise de Requisitos de Segurança com A2E

4.2.1 Contexto

Um dos objetivos relacionados ao desenvolvimento do A2E é estudar a adaptação da abordagem de PLN Exógena escolhida, no caso o ESA, ao domínio de segurança de software por meio de experimentos relacioados a requisitos específicos dessa área. Para seu alcance, as Fases deste projeto utilizaram dois conjuntos de dados: descrições de riscos de segurança em aplicações *web* contidas no OWASP *Top Ten* e os requisitos de segurança propostos no OWASP ASVS. O primeiro conjunto foi utilizado para formar o sistema de conceitos, cujo papel é internalizar no A2E o conhecimento enciclopédico especialista necessário à gestão de riscos específicos em contratações de serviços de software. Já o segundo conjunto foi utilizado para os testes iniciais sobre esse protótipo, os quais direcionaram algumas decisões arquiteturais e de *design* da ferramenta, como o número de classificações (riscos) retornadas para cada instância (requisito), ou os valores de limiares a nível de linha e coluna que devem ser utilizados para a melhora dos resultados obtidos nessa classificação dos exemplos a serem submetidos ao A2E na fase seguinte - relacionada a editais e termos de referência de contratações de serviços de software.

A escolha inicial do OWASP *Top Ten* decorreu de ser uma enciclopédia sobre riscos de software com uma versão em português validada por uma comunidade de especialistas, o que não ocorre necessariamente em outras fontes como a Wikipedia. Ressalta-se que no domínio ao qual se destina primariamente o A2E (Administração Pública Federal brasileira), apesar de não existir uma exigência normativa de que os riscos do OWASP *Top Ten* sejam tratados nos trabalhos de desenvolvimento, manutenção e aquisição de software realizados, esse trabalho é explicitamente referenciado [75] pelo GSI/PR (órgão responsável pelas questões de segurança da informação do Poder Executivo Federal) como padrão internacional sobre o qual se derivam seus denominados "requisitos mínimos necessários à Segurança das Infraestruturas Críticas da Informação" [75].

Considerando que a comparação entre CWE/SANS *Top 25* e OWASP *Top Ten* foi reservada para a Fase 2, discute-se então a escolha do OWASP ASVS. A utilização desse padrão de verificação foi motivada pelas seguintes razões:

1. requisitos escritos em português: uma vez que o A2E destina-se à análise de editais e termos de referências escritos em linguagem natural, torna-se necessário utilizar como conjuntos de validação e teste das versões desse protótipo, ao longo deste trabalho, sentenças escritas em língua portuguesa;

2. validado por uma comunidade de especialistas: de fato, visitando a página do projeto OWASP ASVS, identifica-se a participação de mais de 30 profissionais da área de software seguro, além de três desenvolvedores brasileiros envolvidos na elaboração de uma versão em português desse documento, conforme o projeto OWASP *Portuguese Language*[1].

3. existência de uma categorização que favorece o mapeamento dos requisitos do OWASP ASVS para os riscos do OWASP *Top Ten* Essa facilidade de mapeamento é decorrente do uso das descrições do CWE para a identificação do relacionamento entre as 14 categorias do OWASP ASVS e as 10 do OWASP *Top Ten*. Salienta-se que a utilização desse catálogo de fraquezas contribuiu para um mapeamento menos enviesado do que um que fosse realizado baseado somente no conhecimento do autor.

Cabe registrar ainda que foi incorporado aos experimentos desta Fase, como parte de um conjunto de validação, uma amostra de 121 sentenças retirada da Lei nº 8.666, de 21 de junho de 1993. Essa escolha traz as seguintes justificativas:

1. necessidade redução de falsos positivos: como exposto, esperar-se-ia que um versão do A2E validada apenas com o OWASP ASVS apresentasse um classificador que

[1] `https://www.owasp.org`

fosse de cerca forma enviesado tal que, para qualquer trecho submetido ao A2E, se obtivesse que a sentença seria pertencente a uma das descrições de risco existentes no sistema de conceitos utilizado, ainda que ela, semanticamente, não se relacionasse com essas categorias;

2. necessidade de não aumentar o número de falsos negativos: para evitar enviesamento da ferramenta no sentido de classificar uma categoria como sendo de "segurança" (ou de "não segurança"), tomou-se uma amostra da Lei nº 8.666 de mesmo tamanho do OWASP ASVS, isto é, 121 sentenças, reduzindo o risco de que um possível desbalanceamento de categorias levasse ao A2E preferir uma a outra;

3. necessidade de considerar durante o ajuste dos limiares do A2E sentenças do domínio "Licitações e Contratos": editais, projetos básicos e termos de referência têm um conteúdo mínimo legalmente previsto, respectivamente na Lei nº 8.666 e na Instrução Normativa SLTI/MP nº 04, de 12 de novembro de 2010, [30], apresentando, assim, intersecção com outros domínios de conhecimento, além de segurança de software, como o de "Licitações e Contratos". Dessa forma, foram extraídos trechos da referida Lei, por ser ela a principal referência desse domínio no Direito Administrativo brasileiro, como parte do conjunto de validação do A2E.

Apresentadas essas considerações iniciais, são experimentos a serem realizados na presente Fase:

1. comparação do desempenho entre as estratégias de classificar uma instância em um número fixo ou variável de categorias:

   (a) objetivo: identificar, dentre as possíveis estratégias de classificação pertencentes ao escopo deste projeto, a de melhor desempenho para o A2E;

   (b) considerações: importante ressaltar que trechos de especificações podem se relacionar a várias categorias de risco ao mesmo tempo. Como exemplo, considere o seguinte trecho: *O acesso ao sistema deverá utilizar SSL/TLS*. Esse requisito se relaciona tanto ao risco de exposição de dados sensíveis, uma vez que, por exemplo, as credenciais de acesso trafegarão encriptadas, como também se relaciona ao risco de quebra de autenticação, pois uma proteção a essas credenciais é também uma proteção adicional ao processo de controle de acesso. Ao mesmo tempo, requisitos como *utilizar apenas certificados digitais emitidos pela ICP-Brasil*[2], analisados de forma isolada, dificilmente poderiam ser relacionados a outro risco no OWASP *Top Ten* além do relacionado à salvaguarda de informações sigilosas. Apesar da teoria apresentada indicar uma

[2] `http://www.iti.gov.br/icp-brasil`

opção arquitetural para o A2E, classificando períodos em um número variável de categorias, esta estratégia não seria interessante a essa ferramenta caso apresentasse desempenho inferior (principalmente em termos de Precisão ou NPV) ao da estratégia de classificação em número fixo de riscos, pois para equipes de software carentes de especialistas em segurança, aumenta-se a necessidade de que o resultado retornado pela ferramenta seja confiável a ponto subsidiar análises iniciais de editais;

2. ajuste de limites de corte a serem utilizados sobre editais e termos de referência:

   (a) objetivo: transferir para o classificador a ser utilizado na Fase 2 do projeto informações úteis sobre sentenças de segurança;

   (b) considerações: caso se opte pela utilização de limiares para a tomada de decisão no processo de classificação de trechos de editais e termos de referência, ajustar-se-ia os parâmetros de duas estratégias de *thresholding*: corte a nível de linha (RCUT) e corte por pontuação (SCUT). Parâmetros são calibrados, respectivamente, visando minimizar a função erro de classificação e otimizar o F-*score* tomando um $\beta = 0.5$. Importante notar que, ao longo dos experimentos deste capítulo, esse valor de β será tomado com o intuito de atribuir uma importância maior à precisão do classificador em relação ao seu *recall*, uma vez que, dada a carência de profissionais de software seguro nas instituições da Administração Pública brasileira, é necessário que as saídas produzidas pelo A2E se apresentem da forma mais correta possível para que as equipes que as utilizarão possam tomar decisões nelas baseadas, sem impactos significativos nas avaliações realizadas da integração da gestão de riscos em planejamentos de fábricas de software da APF.

4.2.2 Entendimento e Descrição de Bases de Dados

Além dos documentos citados na seção anterior - OWASP *Top Ten*, OWASP ASVS e Lei nº 8.666 - foram utilizadas publicações sobre licitações e contratos do TCU [76] e da CGU [77], bem como guias de implementação do Programa MPS-BR[3] (sobre melhoria de processos de software) para a criação de conceitos, adicionais aos riscos da OWASP, sobre esses dois assuntos. Essa necessidade foi derivada da natureza dos editais e termos de referência de fábrica de software, os quais não são documentos voltados para a produção de um sistema específico, mas sim voltados para a especificação de um serviço que permita o desenvolvimento e a manutenção de um portfólio de aplicações, com uma

[3] `http://www.softex.br/mpsbr/MPS`

significativa quantidade de suas sentenças, por exigência da referida Lei, relacionadas a aspectos de "Licitações e Contratos", bem como de "Processos de Software". É por meio desses conceitos que se torna possível ao A2E, nesta Fase, classificar uma sentença como semanticamente relacionada a essas duas categorias.

Qualidade Intrínseca

Primeiramente, não foram identificados prejuízos léxicos ao se converter os documentos originais do OWASP *Top Ten* e do OWASP ASVS, bem como as coletâneas do TCU, CGU e MPS.BR. Além disso, essas bases, incluindo entre elas o conteúdo da Lei nº 8.666, foram por equipes heterogêneas desvinculadas do presente trabalho, aumentando a objetividade em sua utilização. A questão da credibilidade é realçada ao se notar a importância de suas organizações autoras: OWASP, Softex, CGU, TCU e Congresso Nacional. A OWASP é uma organização internacional sem fins lucrativos cujo trabalho em riscos de segurança foi explicitamente referenciado pelo Gabinete de Segurança da Informação da Presidência da República brasileira como padrão internacional a ser observado. A Associação para Promoção da Excelência do Software Brasileiro (Softex) é uma Organização da Sociedade Civil de Interesse Público (OSCIP)[4], cujo trabalho em melhoria de processo de software apresenta resultados positivos como o presente em [78]. A Controladoria-Geral da União, órgão central de controle interno do Poder Executivo, o qual tem responsabilidade constitucional de avaliar o cumprimento do orçamento deste Poder. Também na Constituição brasileira, estão o TCU e o Congresso Nacional, ambos envolvidos na fiscalização de gastos públicos. Quanto à reputação desses textos, além da explícita recomendação feita pelo Governo Brasileiro em [75], frisa-se o emprego do OWASP *Top Ten* pela Microsoft, pela *National Security Agency* (NSA) e pelo *Payment Card Industry Security Standards Council* (PCI SSC), conforme apresenta a página desse Projeto da OWASP[5]. Já o OWASP ASVS é citado como boa prática pelo *Department of Homeland Security* (DHS) dos EUA e pelo SANS *Institute*. Por sua vez, a Lei nº 8.666, parágrafo único, indica a amplitude da utilização desse normativo: *além dos órgãos da administração direta, os fundos especiais, as autarquias, as fundações públicas, as empresas públicas, as sociedades de economia mista e demais entidades controladas direta ou indiretamente pela União, Estados, Distrito Federal e Municípios* [14]. As coletâneas de TCU e CGU são utilizadas pelos seus jurisdicionados (incluindo, no presente caso, todas as organizações do Poder Executivo) para a resolução de dúvidas quanto às suas contratações, enquanto o MPS.BR traz a reputação de realmente ser útil para a melhoria dos resultados de pequenas e médias organizações produtoras de software, conforme exposto em [78].

[4] `http://www.planalto.gov.br/CCIVIL_03/leis/L9790.htm`

[5] `https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project`

Tabela 4.2: Distribuição dos requisitos do OWASP ASVS.

Categoria	**Quantidade**
Arquitetura	6
Autenticação	15
Gerenciamento da Sessão	13
Controle de Acesso	15
Validação de Entradas	9
Codificação de Saídas	10
Criptografia	10
Tratamento de Erros	12
Proteção de Dados	6
Segurança da Comunicação	9
Segurança do HTTP	7
Investigação de Códigos Maliciosos	2
Segurança Interna	3

Qualidade Contextual

Sob a perspectiva da qualidade contextual, alguns problemas foram identificados sobre a completude dos dados. De fato, o OWASP ASVS apresenta, originalmente, seus requisitos distribuídos em 14 categorias, como mostrado na Tabela 4.2, as quais se relacionam de forma diferenciada com as categorias de risco presentes no OWASP *Top Ten*, implicando em desbalanceamento de exemplos entre elas. Como exemplo dessa diferenciação, observa-se que no ASVS, as classes "Autenticação" e "Gerenciamento de Sessão" correspondem a 23% do total dessa base, enquanto que a categoria "Validação de Entradas", diretamente relacionada a riscos como injeção de código e XSS, corresponde a apenas 7,4%.

Essa exemplificação traz dois problemas: primeiro, o aumento do risco de enviesamento da solução (A2E) no sentido de tratar melhor algumas classes de risco que outras (pelo maior número de instâncias nos conjuntos de treinamento e validação [42]); segundo, um enviesamento que possa não corresponder à realidade dos incidentes de segurança em aplicações: no exemplo tratado, observa-se que um maior número de exemplos está relacionado ao risco de "Quebra de Autenticação e Gerenciamento de Sessão" do OWASP *Top Ten*, apesar de termos na primeira colocação o risco de "Injeção de Código" (sem falar no risco de XSS, colocado em terceiro lugar no mesmo *ranking*). Para a realização dos experimentos no âmbito deste trabalho, este segundo problema não foi tratado, enquanto, o primeiro problema acabou levando à versão inicial do A2E a considerar apenas um subconjunto dos riscos do OWASP *Top Ten*.

Ainda sob o ponto de vista contextual, a Tabela 4.3 caracteriza, em meados de 2015, a defasagem dos documentos utilizados neste trabalho. Contudo, supõem-se que, no presente caso, o efeito de trabalharmos com essas versões, em algum grau desatualizada, seja

reduzido, uma vez que o que importa realmente nesses textos é a existência de termos que possam caracterizar a área de licitações e contratos. Apesar dessa desatualização, a amplitude dessas coletâneas que abordam, de forma geral, toda a Lei nº 8.666, a quantidade de informação nelas contida (mais de mil páginas sobre o assunto) e a importância de serem homologadas justamente pelos órgãos de controle do domínio estudado (Administração Pública Federal) contribuem significativamente para a qualidade contextual desses textos.

Tabela 4.3: Defasagem dos documentos utilizados neste projeto

Documento	**Última Atualização**
OWASP *Top Ten*	2013
OWASP ASVS versão 1.0	2009
Lei nº 8.666	2014
Coletâneas do TCU e da CGU sobre Licitações e Contratos	2011
Guias do MPS-BR	2013

Qualidade de Representação dos Documentos

Pelo caráter técnico dos documentos citados na Tabela 4.3, essas enciclopédias de conhecimento são melhor compreendidas por alguns grupos especialistas: as bases da OWASP aos profissionais de desenvolvimento seguro de software, enquanto a Lei nº 8.666 e as coletâneas do TCU e da CGU são melhor compreendida por profissionais com maior preparo em "Direito Administrativo", como servidores públicos, advogados etc. Consideração análoga pode ser feita para o MPS.Br, a qual é melhor entendida por profissionais que trabalhem com desenvolvimento, manutenção e aquisição de software. Quanto à acessibilidade dessas informações, como já mencionado, estão publicamente disponibilizadas em seus respectivos sítios eletrônicos.

Neste ponto, apresentam-se comentários sobre alguns dados derivados. Para os experimentos elencados nesta fase, tornou-se necessário relacionar os requisitos presentes no OWASP ASVS com uma ou mais das categorias de risco presentes no OWASP *Top Ten*. Um problema que surgiu como decorrência da escassez de trabalhos correlacionando esses documentos foi propor uma classificação a esses riscos que fosse compatível com a opinião de especialistas em segurança de software. Para a realização desses experimentos, uma categorização inicial foi utilizada pelo autor, sem contudo evitar os seguintes problemas:

1. objetividade: uma vez que essa classificação reflete a análise crítica de uma única pessoa, a objetividade desses dados ficou prejudicada;

2. credibilidade e reputação: diferentemente do OWASP *Top Ten* e do OWASP ASVS homologadas pela OWASP, essa classificação não foi submetida para homologação, o que compromete essas características da qualidade.

Contudo, torna-se necessário dizer que, sob o ponto de vista contextual, a classificação proposta foi relevante e adicionou valor ao trabalho, pois sem ela, os experimentos não poderiam ser realizados, dado que a mensuração de seu desempenho envolveu a comparação do retornado pelo A2E (classificação proposta pela ferramenta) contra o que, teoricamente, deveria seria o resultado verdadeiro (classificação proposta pelo autor). Para minimizar os dois problemas enumerados, utilizou-se como estratégia mapear primeiramente os requisitos do OWASP ASVS para fraquezas presentes no CWE e, somente após esse mapeamento, propor a classificação em uma das categorias do OWASP *Top Ten*. Isso se deve principalmente aos metadados presentes no CWE, exemplificados no Capítulo 2, como os relacionamentos entre as diferentes fraquezas, os quais permitem percorrer o grafo de relacionamentos existentes nessa base. Já Figura 4.1 apresenta uma visão geral do processo de mapeamento dos requisitos do ASVS para o OWASP *Top Ten*.

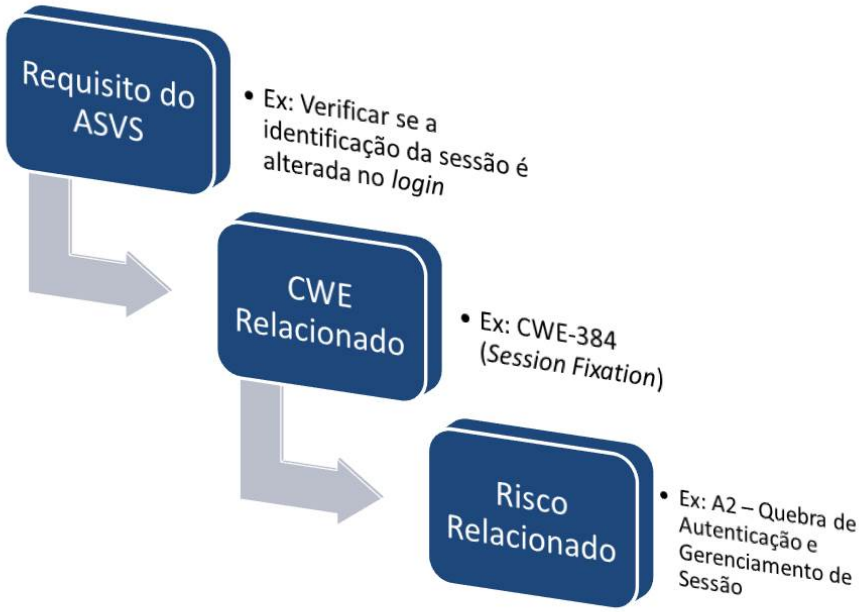

Figura 4.1: Estrutura de uma fraqueza de software presente no CWE..

4.2.3 Limpeza e Formatação dos Dados

Uma vez coletados os dados, desenvolveu-se um algoritmo de limpeza e formatação em R aplicado tanto sobre os sistemas de conceitos (ex: riscos do OWASP *Top Ten* e as categorias adicionais formadas pelas publicações do TCU e da CGU - "Licitações e Contratos" - e pelas da Softex - "Processos de Software"), quanto sobre as instâncias do ASVS submetidas ao A2E para validação e teste. A seguir, apresenta-se o pseudocódigo desse procedimento:

Algoritmo 3 Preparar Texto para a Mineração

Entrada: T - um texto plano (ou conjunto de textos)

Objetivo: T está pronto para a Mineração

1: $corpoSentencas \leftarrow formaCorpo(T)$
2: $corpoSentencas \leftarrow retiraEspaçosEmBranco(corpoSentencas)$
3: $corpoSentencas \leftarrow converteSentencasMinusculo(corpoSentencas)$
4: $corpoSentencas \leftarrow removePontuacao(corpoSentencas)$
5: $corpoSentencas \leftarrow removerStopWords(corpoSentencas, \text{“}portuguê\text{s”})$
6: $corpoSentencas \leftarrow removerStopWords(corpoSentencas, \text{“}inglê\text{s”})$
7: $corpoSentencas \leftarrow stemizacao(corpoSentencas, \text{“}portuguê\text{s”})$
8: $corpoSentencas \leftarrow stemizacao(corpoSentencas, \text{“}inglê\text{s”})$

A escolha em trabalhar com Corpos (definidos como conjuntos de documentos textuais [79]), está relacionada à escolha tecnológica de utilização do pacote TM[6] do R. Neste pacote, ainda segundo [79], Corpos são sua principal estrutura, os quais, um vez formados, possibilitam uma série de transformações como as presentes no Algoritmo 3. A seguir uma breve descrição das funções presentes nesse procedimento:

1. formaCorpo(texto T): recebe um texto plano e transforma em um corpo;
2. retiraEspaçosEmBranco(corpo C): recebe um corpo C e retorna um novo Corpo C' em que os espaços em branco em sequência de C foram transformadas em apenas um só. Como exemplo, uma sentença como "isso é um teste" tornar-se-ia "isso é um teste";
3. converteSentencasMinusculo(corpo C): recebe um corpo C e retorna um novo Corpo C' em que reduz cada uma das letras de C para minúsculo. Dessa forma, um termo como "Segurança" seria o mesmo que "segurança", facilitando a contagem da frequência de um termo dentro de um documento;

[6] `http://cran.r-project.org/web/packages/tm/index.html`

4. removePontuacao(corpo C): recebe um corpo C e retorna um novo Corpo C' com todas as pontuações existentes em C extraídas. Importante para evitar termos diferentes como "segurança", "segurança," e "segurança.";

5. removeStopWords(corpo C, string idioma): recebe um corpo C e retorna um novo corpo C' com todas *stopwords* daquele "idioma" em C extraídas. Essa função torna-se importante ao desejarmos focar a análise apenas nas palavras mais relevantes de um texto, dado que *stopwords*, pela definição contida em [80], adicionam pouco valor à recuperação de informação. No Algoritmo 3, essa função foi chamada tanto em português, quanto em inglês, devido ao fato da área de informática brasileira ser influenciada de forma intensa por estrangeirismos oriundos da língua inglesa [81];

6. stemizacao(corpo C, string "idioma"): recebe um corpo C e retorna um novo Corpo C' com todas as palavras constantes em C reduzidas às suas raízes morfológicas. Analogamente ao que ocorreu no caso da aplicação da função "removeStopWords(corpo C, string idioma)", é aplicada tanto relativamente ao idioma português quanto relativamente ao inglês, devido aos estrangeirismos da área de informática brasileira [81]. Salienta-se que, devido ao fato dos sufixos ingleses e dos sufixos portugueses extraídos serem distintos, não se espera uma dupla "stemização" sobre um mesmo radical de um termo presente num documento em análise pelo A2E.

Por último cabe ressaltar que, devido ao fato do ESA não considerar de forma explícita os relacionamentos entre os conceitos presentes em seu espaço semântico de representação de sentenças, muitos dos experimentos realizados ao longo do estudo do A2E exploraram dois cenários: um cuja representação é unicamente derivada do interpretador do ESA (consequentemente sem relacionamentos entre os conceitos); outro cenário cuja representação é um híbrido da aplicação do ESA e do *Page Rank*, denominada ESA-G, derivada do uso do seguinte grafo, não direcional, de relacionamento entre categorias de risco, como definido na Figura 4.2:

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	ADM	PSW
A1	0	0	1	1	0	0	0	0	1	0	0	1
A2	0	0	0	1	0	1	1	1	1	0	0	1
A3	1	0	0	0	0	0	0	1	1	0	0	1
A4	1	1	0	0	0	1	0	0	1	0	0	1
A5	0	0	0	0	0	1	0	0	1	0	0	1
A6	0	1	0	1	1	0	0	0	1	0	0	1
A7	0	1	0	0	0	0	0	1	1	0	0	1
A8	0	1	1	0	0	0	1	0	1	0	0	1
A9	1	1	1	1	1	1	1	1	0	1	0	1
A10	0	0	0	0	0	0	0	0	1	0	0	1
ADM	0	0	0	0	0	0	0	0	0	0	0	0
PSW	1	1	1	1	1	1	1	1	1	1	0	0

Figura 4.2: Grafo de relacionamento entre conceitos do A2E.

em que são apresentados os relacionamentos entre as 10 categorias de risco do OWASP *Top Ten*, um conceito relacionado a licitações e contratos (doravante denominado "ADM") e um conceito relacionado à melhoria de processo de software (doravante denominado "PSW").

Cada célula desse grafo corresponde a um valor binário (0 ou 1), em que 0 significa a ausência de relacionamento entre os conceitos e 1 significa a existência de relacionamentos. A fim de conseguir uma qualidade intrínseca aceitável em termos de reputação, apresentando uma classificação derivada de uma renomada entidade em segurança de software (MITRE), adotou-se como critério, para o estabelecimento desses valores, a análise das entradas correspondentes no CWE aos riscos do OWASP *Top Ten* 2013. Os relacionamentos entre essas categorias encontram-se na Tabela 4.4, os quais são ainda ilustrados por alguns CWE-IDs nelas presentes que se relacionam.

Os relacionamentos presentes na Tabela 4.4 são ainda complementados pelas seguintes considerações:

Tabela 4.4: Relacionamentos entre fraquezas catalogadas pelo CWE.

Categoria	CWE	Tipo de Relacionamento	CWE	Categoria
A1	CWE-74	*ParentOf*	CWE-79	A3
A1	CWE-74	*ParentOf*	CWE-99	A4
A2	CWE-862	*ParentOf*	CWE-639	A4
A2	CWE-311	*ChildOf*	CWE-934	A6
A2	CWE-287	*ChildOf*	CWE-935	A7
A2	CWE-613	*RequiredBy*	CWE-352	A8
A3	CWE-79	*PeerOf*	CWE-352	A8
A4	CWE-668	*ParentOf*	CWE-200	A6
A5	CWE-209	*ChildOf*	CWE-200	A6
A7	CWE-287	*CanFollow*	CWE-613	A8

1. a descriçao do risco A9 no CWE considera que ele pode estar relacionado a qualquer outro risco;
2. considerou-se que a categoria "ADM" (relacionado aos conceitos do Direito Administrativo vinculados a licitações e contratos) não se relaciona diretamente com nenhuma das categorias relacionadas a engenharia segura de software;
3. considerou-se que todas as categorias de risco de software se relacionam com a categoria "PSW" (conceito relacionado a processos de software).

A produção desses dados derivados a partir do CWE, faz com que eles herdem a qualidade dos dados desse dicionário, o qual, conforme já dito, pode ser comparado à qualidade do OWASP *Top Ten*.

4.2.4 Modelagem e Resultados dos Experimentos

Experimento 1: Comparação do desempenho entre as estratégias de classificar uma instância em um número fixo ou variável de categorias

Buscou-se, primeiramente, um resultado que ajudasse a distinguir qual a melhor classificador para o A2E entre um que retornasse um número pré-definido de rótulos e outro que retornasse um número variável. Dessa forma, o experimento consistiu em comparar seus desempenhos sobre os requisitos presentes no OWASP ASVS. Para isso, estruturou-se o sistema de conceitos do A2E com os 12 conceitos apresentados na Figura 4.2. Os conceitos "ADM" e "PSW" foram adicionados para complementar o conjunto de categorias do OWASP *Top Ten*, uma vez que os textos de editais e termos de referências de contratações de fábricas de software, seja por exigências normativas, como as descritas na Seção 2.2, seja pelo conteúdo diversificado de especificações de software, como apresentado em [22], não se limitam ao domínio de segurança, abrangendo também tópicos de

engenharia de software e de direito administrativo sem relacionamento direto com aquela área de conhecimento. Dessa forma, propôs-se o conceito "ADM" para ser associado àquelas sentenças relacionadas principalmente aaspectos técnicos de licitações e contratos e o conceito "PSW" para tratar de aspectos específicos de processos de software das especificações analisadas.

Inicialmente, executou-se o A2E em doze diferentes versões: cada uma delas relacionada ao total de classificações retornada pelo algoritmo. Dois cenários foram identificados para a realização desse experimento:

1. representação baseada no ESA;
2. representação baseada no ESA-G.

Os dados dos experimentos sobre representações baseadas unicamente no ESA e sobre representações baseadas no ESA-G podem ser encontradas, respectivamente, na Tabela 4.5 e na Tabela 4.6 .

Tabela 4.5: Versões do A2E sobre o ESA classificando em um número de categorias previamente determinadas

nº Categorias Retornadas	**Precisão**	***Recall***
1	0.2685950	0.1405647
2	0.2896006	0.4271350
3	0.2238292	0.6939394
4	0.1887052	0.8008953
5	0.1767218	0.8811295
6	0.1718320	0.9266529
7	0.1725207	0.9710744
8	0.1714876	0.9855372
9	0.1683884	0.9938017
10	0.1670799	0.9958678
11	0.1669484	1.0000000
12	0.1666667	1.0000000

De acordo com as tabelas apresentadas, observa-se que, para uma estratégia de classificação em um número pré-definido de rótulos, à medida que o número de categorias retornadas pelo classificador diminui, aumenta-se a sua Precisão. Esse comportamento não é desejável para classificadores multirrótulos, os quais, conforme Seção 2.4.2, podem associar a uma única instância várias ou até mesmo todas as categorias disponíveis. Os resultados mostrados na Tabela 4.5 (a qual usou a representação proporcionada pelo ESA) e na Tabela 4.6 (a qual usou a representação proporcionada pelo ESA-G) mostram que, quanto mais uma instância se relaciona simultaneamente com diferentes *labels*, mais inadequado se torna a utilização de uma abordagem como a testada.

Tabela 4.6: Versões do A2E sobre o ESA-G classificando em um número de categorias previamente determinadas

nº Categorias Retornadas	**Precisão**	***Recall***
1	0.3719008	0.1681818
2	0.2772039	0.2575069
3	0.2331267	0.3213499
4	0.2152204	0.3855372
5	0.1896006	0.4092287
6	0.1725207	0.4347796
7	0.1609111	0.4650826
8	0.1553030	0.5012397
9	0.1486455	0.5314050
10	0.1507576	0.5994490
11	0.1528613	0.6938705
12	0.1666667	1.0000000

A primeira abordagem, neste fase, de classificação em um número variável de categorias foi realizada empregando-se limiares RCUT, tanto sobre a representação gerada pelo ESA, quanto pelo ESA-G. A Tabela 4.7 e a Tabela 4.8 mostram que, em comparação com a abordagem de retorno de um número pré-definido (arbitrário) de rótulos, a estratégia de associação em um número variante de *labels* apresenta desempenho em termos de Precisão similar ao apresentado pelo método de associação a um número arbitrário de grupos, mas em termos de *Recall* seu desempenho é bem superior (chegando a ser cerca de 6 vezes maior).

Tabela 4.7: Comparação entre a classificação usando RCUT e a classificação usando um número fixo de categorias (ambos usando a representação do ESA)

Método	**nº de Categorias**	**Precisão**	***Recall***
Arbitrário	2	0.2896006	0.4271350
RCUT	Variável	0.3457300	0.9745179

Tabela 4.8: Comparação entre a classificação usando RCUT e a classificação usando um número fixo de categorias (ambos usando a representação do ESA-G)

Método	**nº de Categorias**	**Precisão**	***Recall***
Arbitrário	1	0.3719008	0.1681818
RCUT	Variável	0.3267906	0.9469697

Em termos práticos, isso significa que dado dois classificadores - $h_{arbitrario}$ e h_{RCUT} - e uma instância x pertencente a seis diferentes categorias (de um total de dezoito), $h_{arbitrario}$ retorna apenas três rótulos, acertando apenas 1, o h_{RCUT} é capaz de retornar os dezoito, acertando seis deles. Dessa forma, enquanto, em termos de precisão, os classificadores

se equivalem, em termos de *Recall* se diferenciam significativamente (consequentemente, maior F-*score*). Esse fato motiva a alteração da solução inicial proposta no Capítulo 3 para um classificador baseado em limiares previamente calculados.

Ainda nessa série de experimentos, conforme Tabela 4.9, classificadores micro-SCUT, tanto sobre sobre a representação fornecida pelo ESA, quanto pelo ESA-G, superaram o F-*score*, $\beta = 0.5$, obtido pelo classificador RCUT, fazendo com que chegássemos ao final da Fase concluindo pela alteração do A2E para a utilização, pelo seu classificador, de limiares ao nível das categorias do sistema de conceitos (micro-SCUT). A Figura 4.3 ilustra essa nova proposta.

Tabela 4.9: Comparação do F-*score* de versões do A2E usando limiares RCUT e limiares micro-SCUT

Método	**F-Score(β=0.5)**
RCUT ESA	0.396955496
RCUT ESA-G	0.376045760
Micro-SCUT ESA	0.513392900
Micro-SCUT ESA-G	0.514150900

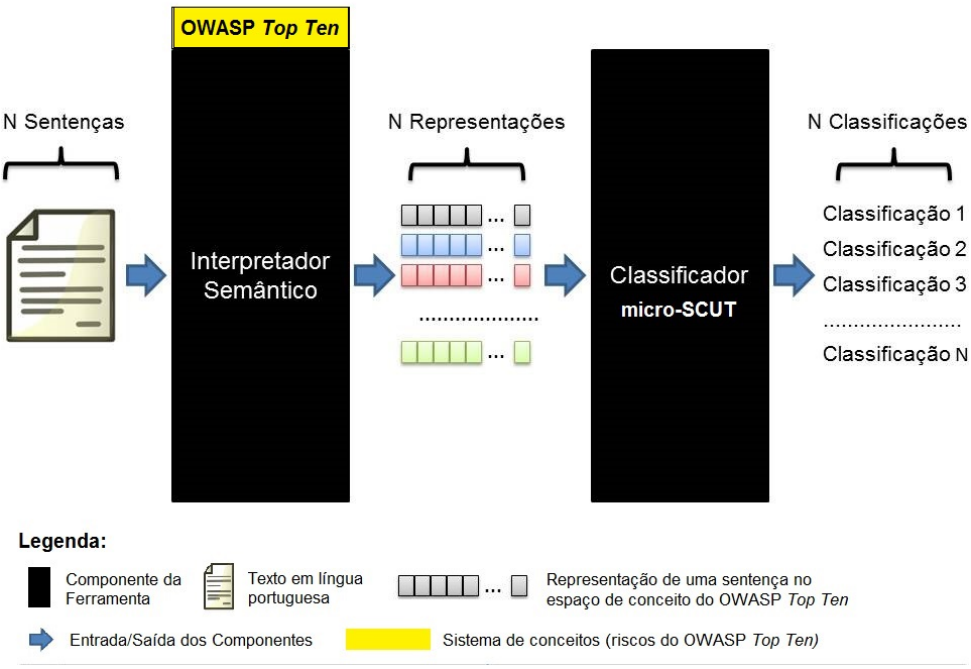

Figura 4.3: Atualização da proposta arquitetural do A2E após a Fase 1.

Como pode ser visto, ela se difere da proposta final presente no Capítulo 3 pelo fato de incorporar um classificador baseado em limiares do tipo micro-SCUT (isto é, valores

pré-fixados por categorias derivados a partir de micro-médias). Entretanto, no intuito de resolver o problema estabelecido no Capítulo 1 deste trabalho, é necessário submeter o A2E ao domínio de editais e termos de referência de software, uma vez que eles contém as sentenças carentes de análises pelas equipes da APF. Assim, na próxima seção, o foco se transfere de períodos textuais próprios da área de segurança (como requisitos do OWASP ASVS) para períodos textuais de planejamentos de fábricas de software.

Experimento 2: ajuste de limites de corte a serem utilizados sobre editais e termos de referência

Apesar do desempenho superior do micro-SCUT sobre o categorização RCUT, mostrado na Tabela 4.9, visando a generalização equivocada dessa superioridade para um domínio de sentenças distinto do desta etapa, identiicaram-se os parâmetros de corte utilizados por esses classificadores para reuso na fase seguinte deste projeto. O objetivo é auxiliar os classificadores a serem utilizados para avaliação das sentenças de editais e termos de referências a reconhecerem sentenças de segurança a partir da reutilização dos parâmetros identificados no presente experimento.

Para limiares RCUT, aplicando-se o algoritmo discriminado na Subseção 2.4.2, obtém-se os seguintes valores:

Tabela 4.10: Parâmetros RCUT derivados da análise de sentenças do OWASP ASVS

Coordenadas	**Pesos com ESA**	**Pesos com ESA-G**
Injeção de Código	0.3992294162	0.15903205
Quebra de Autenticação e Gerenciamento de Sessão	0.2011728559	-0.21212909
Cross-Site Scripting (XSS)	0.5925382806	0.19073372
Referência Insegura e Direta a Objetos	0.5057066805	0.68459573
Configuração Incorreta de Segurança	-0.0887798161	-0.28170525
Exposição de Dados Sensíveis	0.2241932971	-0.17036750
Falta de Função para Controle de Nível de Acesso	-0.0093435886	0.04855821
Cross-Site Request Forgery (CSFR)	0.6185197525	-0.01516673
Utilização de Componentes Vulneráveis Conhecidos	-0.2369603592	-0.64267873
Redirecionamentos e Encaminhamentos Inválidos	-0.0924531841	0.07503965
Licitações e Contratos (ADM)	0.8875048457	-0.13884444
Processos de Software (PSW)	0.0466442004	0.41211170
Fator Independente (b)	-0.0003761309	0.10917932

Como se pode observar da Tabela 4.10, os valores (pesos) utilizados por um classificador RCUT são dependentes da representação vetorial das sentenças a serem categorizadas no espaço semântico escolhido. Conceitos como o de "CSRF" e "Exposição de Dados Sensíveis" sensíveis podem contribuir para o aumento do limiar de corte em uma representação e para a diminuição quando outra representação é adotada. Esses valores ainda permitem visualizar a influência que os relacionamentos interconceitos podem exercer nesses valores: com o ESA, o qual não considera explicitamente essa influência, o conceito "ADM" tem o maior valor, ressaltando a ortogonalidade em relação ao demais conceitos mostrada na Figura 4.2; já com o ESA-G, essa influência do conceito ADM é reduzida, uma vez que não se relaciona com nenhuma outra categoria. O contrário ocorre com o conceito "PSW", que ao se relacionar com todos os outros grupos (com exceção do "ADM"), passa a exercer maior influência na determinação do valor do limiar de corte, quando as representações são derivadas do ESA-G.

O método de corte a nível de categoria micro-SCUT, o qual obteve, segundo Tabela 4.9, melhor desempenho nesta Fase, também teve seus parâmetros identificados para reaproveitamento na próxima etapa do projeto (análise de sentenças oriundas de editais e termos de referências). Dessa forma, buscando a otimização do F-*score* do classificador do A2E sobre o conjunto de sentenças do OWASP ASVS, utilizando-se o algoritmo proposto em [54], obteve-se o conjunto dos limiares responsáveis pelos resultados da referida tabela. Contudo, uma vez que o *threshold* obtido para a categoria ADM é igual a zero, o que poderia aumentar significativamente a intersecção entre sentenças do domínio de segurança e sentençs sobre licitações e contratos, apresenta-se na Tabela 4.11, os parâmetros obtidos para um F-*score* com $\beta = 0.05$, os quais são maiores ou iguais aos obtidos para um $\beta = 0.5$, e resultam em valores de 0.7686704 e 0.5824196, respectivamente, para o F-*score* obtido pelo micro-SCUT com ESA e com ESA-G.

A Tabela 4.11, de forma análoga à Tabela 4.10, reforça a dependência entre os valores dos parâmetros e a representação escolhida para as sentenças no espaço semântico desejado. De fato, os valores apresentados são diferentes dependendo do algoritmo de similaridade semântica utilizado. Além disso, ressaltam-se alguns fatos relevantes sobre esses resultados. O primeiro deles é que enquanto com o ESA-G o conceito de maior limiar é o "ADM", com o ESA, essa e a categoria de menor valor de corte. Isso implica numa maior facilidade do micro-SCUT ESA, sobre o micro-SCUT ESA-G, para detectar sentenças que não sejam de segurança, o que pode ser uma propriedade importante ao se analisar domínios como os da Fase 2, em que se busca diferenciar semanticamente sentenças a respeito de tópicos de segurança de software das demais sentenças presentes em editais e termos de referências. De forma análoga, a utilização da representação baseada no ESA também facilita a categorização de instâncias para o conceito "PSW".

Tabela 4.11: Parâmetros SCUT derivados de sentenças do OWASP ASVS

Conceito	**Limiar de corte com ESA**	**Limiar de corte com ESA-G**
Injeção de Código	0.053971552	0.1886919
Quebra de Autenticação e Gerenciamento de Sessão	0.025793099	0.1856520
Cross-Site Scripting (XSS)	0.087200780	0.2008286
Referência Insegura e Direta a Objetos	0.103589818	0.1491837
Configuração Incorreta de Segurança	0.039569175	0.1651007
Exposição de Dados Sensíveis	0.050553499	0.2202904
Falta de Função para Controle de Nível de Acesso	0.041678865	0.2037498
Cross-Site Request Forgery (CSFR)	0.057940263	0.2130503
Utilização de Componentes Vulneráveis Conhecidos	0.056766682	0.2724496
Redirecionamentos e Encaminhamentos Inválidos	0.042235578	0.1792086
Licitações e Contratos (ADM)	0.004756405	0.3077144
Processos de Software (PSW)	0.033816107	0.2024116

4.3 Avaliação de Sentenças de Editais de Contratação de Software da Administração Pública Brasileira

Com o intuito de atender ao objetivo específico de estudar a adaptação do A2E a editais de fábricas de software, apresentam-se a seguir suas principais análises e resultados, as quais podem ser agrupadas em dois experimentos-chaves:

1. comparação entre as estratégias de classificação micro-SCUT e RCUT;
2. comparação entre o desempenho do A2E utilizando o OWASP *Top Ten* e o CWE/SANS *Top 25*;

Diferentemente do que ocorreu com os experimentos realizados na Fase 1 deste projeto, a listagem apresentada compartilha o mesmo objetivo comum: identificar a estratégia de melhor desempenho no descarte de sentenças que não sejam de segurança de software. Para isso, tomar-se-á como referencial, ao longo da presente Fase, o fato de que, especificações de software não costumam apresentar mais do que 10% dos seus requisitos como sendo de segurança, conforme foi identificado no estudo sobre análise arquitetural registrado em [22]. De fato, considerando que o modelo de qualidade proposto no padrão internacional ISO/IEC 25010:2011 [32], relativo a modelo de qualidade de software, tem

oito diferentes dimensões, das quais segurança é apenas uma delas, esperar-se-ia, caso a distribuição dos requisitos fosse uniforme, resultado similar, que correspondesse a cada um desses grupo 12,5% do documento.

Considerando que o objeto de análise desta Fase são termos de referência de serviços de desenvolvimento e manutenção de software, não especificações de um produto em si, os quais ainda dedicam grande parte do corpo de seu texto a questões de direito administrativo, gerenciamento de serviços e engenharia de software [13], a tendência é a identificação de uma proporção ainda menor de trechos relacionados a software seguro em relação ao total de períodos textuais contidos nesses documentos. Dessa forma, quanto mais a eliminação pelas opções avaliadas em cada experimento se aproximar de 90% das sentenças, mais ela será considerada quantitativamente apropriada.

Ressalta-se ainda a ciência de que um maior descarte de sentenças pode vir acompanhado de um maior número de falsos negativos (Subseção 2.4.5). Contudo, essa opção está relacionada ao custo, no domínio em análise (especificações de fábricas de software) de um falso positivo e de um falso negativo decorrentes do uso do A2E. A Figura 4.4 permite a realização da comparação entre os custos desses achados (supondo que, caso haja verificação, o erro do A2E será identificado). De acordo com essa figura, os custos esperados em cada uma dessas situações seriam dados pela Equação 4.1 e pela Equação 4.2:

$$\text{custo de falso positivo}(CFP) = P * CV + (1 - P) * CCA \tag{4.1}$$

$$\text{custo de falso negativo}(CFN) = P * CV + (1 - P) * CCE \tag{4.2}$$

Considerando-se que o custo de correção de uma aplicação, por abranger todo o ciclo de desenvolvimento, é maior do que o custo de correção de sua especificação, tem-se que o custo esperado de um falso positivo é maior do que o de um falso negativo na situação em análise. Dessa forma, conforme já colocado, observa-se como critério de análise de diferentes estratégias de configuração do A2E, o número de sentenças que a ferramenta consegue descartar, supondo que não estão relacionadas a riscos de segurança.

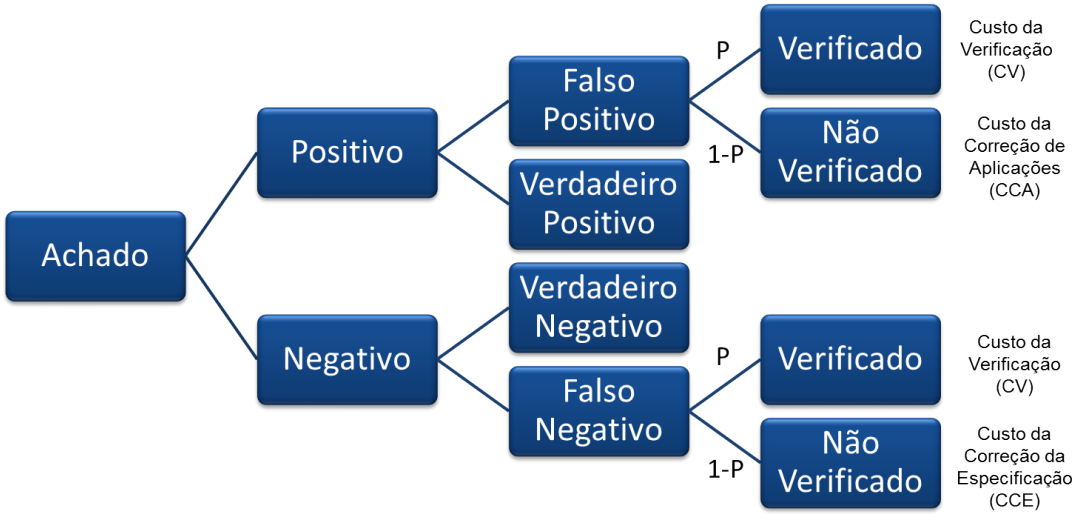

Figura 4.4: Árvore de decisão comparando os custos de um falso positivo e de um falso negativo oriundos do A2E.

4.3.1 Preparação e Entendimento dos Editais e Termos de Referência

Para os experimentos realizados nesta fase, foi escolhido um conjunto de editais e termos de referência disponibilizados pela consultoria Fatto[7]. Esse conjunto é constituído de 105 documentos, elaborados durante o período de 2006 a 2012, distribuídos conforme Figura 4.5.

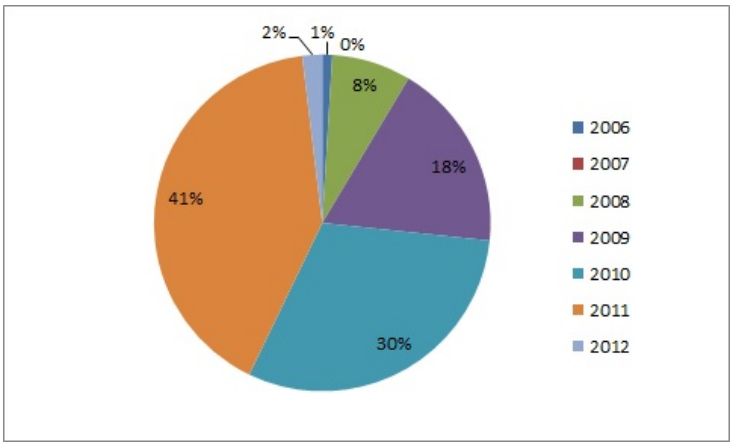

Figura 4.5: Distribuição por ano dos editais e termos de referência analisados.

[7] http://fattocs.com/pt

A escolha dessa conjunto, o qual representa uma amostra não aleatória de contratações nesse intervalo de tempo, teve como principal justificativa a acessibilidade desse material, não observada na obtenção da população de documentos correspondente, não apenas pelo armazenamento disperso dessas informações (no sítio eletrônico da organização licitante), como pela própria disponibilização em si, dado que muitas desses especificações não são mais encontradas nos portais de suas instituições elaboradoras.

A possibilidade de trabalhar com essa base da Fatto trouxe como contrapartida a utilização de especificações mais antigas, dado que os documentos mais recentes apresentam, no ano de 2015, cerca de 3 anos de vida. Entretanto, assume-se, pela exigência da Lei nº 8.666, desde 1993, da explicitação de todos os elementos necessários para viabilizar a avaliação dos custos de um contrato, que o desempenho do A2E diante desses insumos é um importante indicador do desempenho que o A2E teria com editais e termos de referência mais recentes, pois a necessidade das organizações em tratar riscos de segurança presentes em seus produtos não se alterou nesse período. Além disso, assume-se, pelo mesmo motivo da necessidade de explicitação de todos os elementos necessários para caracterizar custos [14], que a completude e a quantidade de informação presente nessa documentação é compatível com a existente nas contratações mais contemporâneas. Esses fatos servem para caracterizar a qualidade contextual do material em questão.

Já do ponto de vista da qualidade intrínseca dos textos, assumindo-se que a documentação coletada seja fidedigna com a originalmente apresentada pelos órgãos e entidades públicos, obtém-se uma acurácia compatível com a obtida nas bases utilizadas na Fase 1, em que problemas relacionados a essa característica não foram identificados. Quanto à objetividade, por corresponderem a contratações reais, pode ser considerada como alta. Em relação a aspectos como credibilidade e reputação, as especificações analisadas na Fase 2 deste projeto refletem a heterogeneidade técnica entre diferentes órgãos da Administração Pública brasileira, diversidade esta que pode ser identificada em levantamentos do Tribunal de Contas da União sobre a diversidade dos recursos humanos de TI no âmbito da Administração Pública Federal [8]. Assim, pode-se afirmar que editais produzidos por organizações distintas terão, pela natureza de suas organizações, níveis diferenciados de qualidade intrínseca.

Considerando o aspecto contextual da amostra de instâncias analisadas nesta Fase 2 pelo A2E, temos que os editais e termos de referência coletados adicionam grande valor à compreensão desse protótipo devido ao fato de se relacionarem diretamente com o domínio final de sua aplicação, aumentando, assim, a importância de realmente se considerar os experimentos desta Fase no *design* do A2E, buscando a redução de seu número falsos positivos de forma que permita a utilização da ferramenta por equipes sem especialistas em segurança de software.

Sob o ponto de vista representacional, as informações ali presentes, basicamente relacionadas a aspectos de "Licitações e Contratos" e "Engenharia de Software", privilegiam o entendimento e a interpretação por aqueles com conhecimentos básicos nesses domínios.

Outra base utilizada nesta Fase foi o CWE. Esse dicionário de fraquezas de software consolida o conhecimento de várias fontes relacionadas à segurança desse tipo de ativo, conforme pode ser visualizado na Figura 4.6. Isso aliado ao patrocínio direto do *Department of Homeland Security*[8] dos EUA, órgão responsável pela segurança interna desse país, contribui para a credibilidade e a reputação dos dados presentes nesse repositório. Sua objetividade é realçada pelo seu relacionamento com bases de vulnerabilidades como *National Vulnerability Database* (NVD) e *Common Vulnerabilities and Exposures* (CVE), as quais abrangem em diversos tipos de software. Único ponto desfavorável no emprego do CWE, no contexto em questão, é a ausência de uma versão em português consolidada por especialistas em segurança de software. Diante disso, a fim de viabilizar uma comparação entre o uso do OWASP *Top Ten* e o uso do CWE como sistema de conceitos, optou-se por trabalhar com o CWE/SANS *Top 25*[9], o qual consiste num subconjunto de 25 CWE-ID's, avaliados por especialistas em segurança como os de maior pontuação no CWSS[10], sistema de pontuação de fraquezas de software desenvolvido pelo MITRE, considerando os critérios de disseminação, importância e probabilidade de exploração. A partir dessa escolha, tomou-se na íntegra a representação em inglês de cada uma dessas fraquezas associando-as aos correspondentes do OWASP *Top Ten*. Dessa forma, chegou-se à distribuição presente na Tabela 4.12.

Tabela 4.12: Proposta de mapeamento do CWE/SANS *Top 25* para o OWASP *Top Ten*

Riscos de Segurança	**Fraquezas**			
A1	CWE-78		CWE-89	
A2	CWE-307	CWE-862	CWE-863	
A3	CWE-79		CWE-807	
A4	CWE22	CWE-732	CWE-250	
A6	CWE-311	CWE-327	CWE-759	CWE-798
A7	CWE-306	CWE-676	CWE-829	
A8	CWE-352			
A10	CWE-601			
Buffer Overflow	CWE-120	CWE-131	CWE-134	CWE-190

Esse mapeamento utilizou-se do campo *Relationships* existente na descrição de cada uma das fraquezas descritas no CWE/SANS *Top 25*. Dessa forma, basicamente se explorou relacionamentos do tipo *ChildOf* para identificar em qual categoria do OWASP *Top*

[8] `http://www.dhs.gov`
[9] `http://cwe.mitre.org/top25/index.html`
[10] `https://cwe.mitre.org/cwss/cwss_v1.0.1.html`

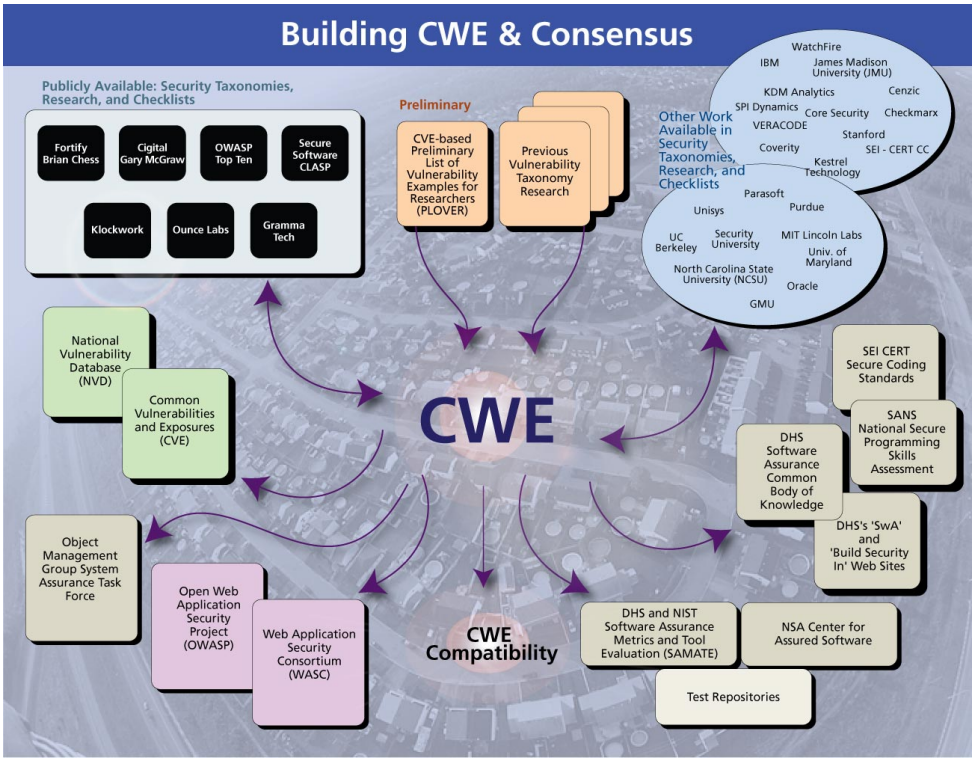

Figura 4.6: Visão geral da consolidação de fontes proporcionada pelo CWE. Fonte: MITRE [82]

Ten esses CWE-ID 's pertencem. Como se pode ver na Tabela 4.12, dois componentes dessa listagem não foram mapeados por não ter sido identificado vínculo com nenhum dos 10 riscos do OWASP *Top Ten*: CWE-434 *Unrestricted Upload of File with Dangerous Type* e CWE-494 *Download of Code Without Integrity Check*. Além disso, um risco explicitamente citado no OWASP *Top Ten*, mas não listado entre os 10 maiores, foi incluído nesse mapeamento: *buffer overflow*. Ressalta-se ainda que:

1. não houve mapeamento para o risco A5 (Configuração Incorreta de Segurança), uma vez que nenhuma das fraquezas listadas pelo CWE/SANS *Top 25* está, de acordo com o CWE, associada a essa categoria de risco;

2. não houve mapeamento para o risco A9 (Utilização de Componentes Vulneráveis Conhecidos), uma vez que o CWE apresenta que essa categoria, ao contrário das outras, não pode ser descrita em termos de erros nos processos de *design*, implementação e implantação que ocorrem ao longo do ciclo de desenvolvimento de um software.

Apresentado esse mapeamento, executou-se procedimento similar ao Algoritmo 3, com exceção da remoção de *stopwords* e da extração de sufixos referentes à língua portuguesa (pois o CWE se encontra todo em inglês). Após esse limpeza dos dados, formou-se uma

matriz $M_{\text{termos x conceitos}}$ em que cada entrada M_{ij} apresenta o TF-IDF do i-ésimo termo em relação ao j-ésimo conceito. Para viabilizar o uso dessa matriz, uma proposta de tradução de seus termos, feita pelo autor deste trabalho, foi utilizada. Contudo, a ausência desta proposta por uma comunidade de especialistas em segurança de software potencialmente fragiliza a acurácia desses dados.

Uma vez que o uso de apenas 10% das palavras mais frequentes em cada conceito não implica perda significativa de desempenho de um classifcador [41], realizou-se a restrição de seu conjunto de termos a uma pequena fração de si mesmo, mantendo nesse subconjunto apenas aquelas palavras que, ao menos para uma categoria de risco, esteja entre os 10% de maior TF-IDF.

Algoritmo 4 Tradução de descrições do CWE/SANS *Top 25* para uso pelo A2E

Entrada: Textos em inglês do CWE/SANS *Top 25* e mapeamento para o OWASP *Top Ten*;

1: Agrupar os textos nos riscos correspondentes do OWASP *Top Ten*;
2: Gerar a matriz $M_{termosxconceitos}$ referenciada na Seção 3.3;
3: Para cada coluna dessa matriz, identificar os termos que têm os maiores TF-IDF (10%);
4: Traduzir cada um desses termos para sua versão correspondente em português;
5: Substituir os termos em inglês da matriz $M_{termosxconceitos}$ pelos termos em português;
6: Tomar como novo índice do A2E a submatriz $M'_{\text{termos traduzidos x conceitos}}$ de M;

A tradução desses termos também pode ser encontrada em `https://github.com/rnpeclat/A2E`.

4.3.2 Modelagem e Resultados dos Experimentos

Experimentos 1 e 2: comparações entre as estratégias de classificação micro-SCUT e RCUT e entre os sistemas de conceitos OWASP *Top Ten* e CWE/SANS Top 25

Inicialmente, a partir dos limiares previamente calculados na Fase 1, ainda se utilizando do OWASP *Top Ten* como sistema de conceitos, submeteu-se um conjunto de aproximadamente 120 mil sentenças, correspondentes ao total de períodos textuais extraídos dos editais e termos de referências disponibilizados pela consultoria Fatto, ao processamento do A2E, obtendo-se os dados presentes na Tabela 4.13, que mostra o desempenho superior da filtragem micro-SCUT quando combinada com a representação gerada pelo ESA na redução desse conjunto inicial de períodos textuais, ao utilizar como regra o descarte das

sentenças classificadas como “ADM” (isto é, pertencentes ao domínio de licitações e contratos administrativos), apresentando maior redução do número de sentenças que seriam de segurança de software (cerca de 95%), conforme mostra a Figura 4.7.

Tabela 4.13: Distribuição das classificações propostas pelo A2E/OWASP na análise de editais e termos de referência

Categorias / Métodos	**Micro-SCUT ESA-G**	**Micro-SCUT ESA**	**RCUT ESA-G**	**RCUT ESA**
A1	2631	507	23177	18378
A2	645	505	29074	4659
A3	555	61	20295	8707
A4	1171	53	17384	4324
A5	364	2042	16336	11106
A6	170	2977	28348	111063
A7	129	1347	16659	9497
A8	289	1316	21454	4334
A9	249	20	89779	10071
A10	67	196	12863	8177
ADM	92328	112160	23327	29961
PSW	103812	97436	114378	23014

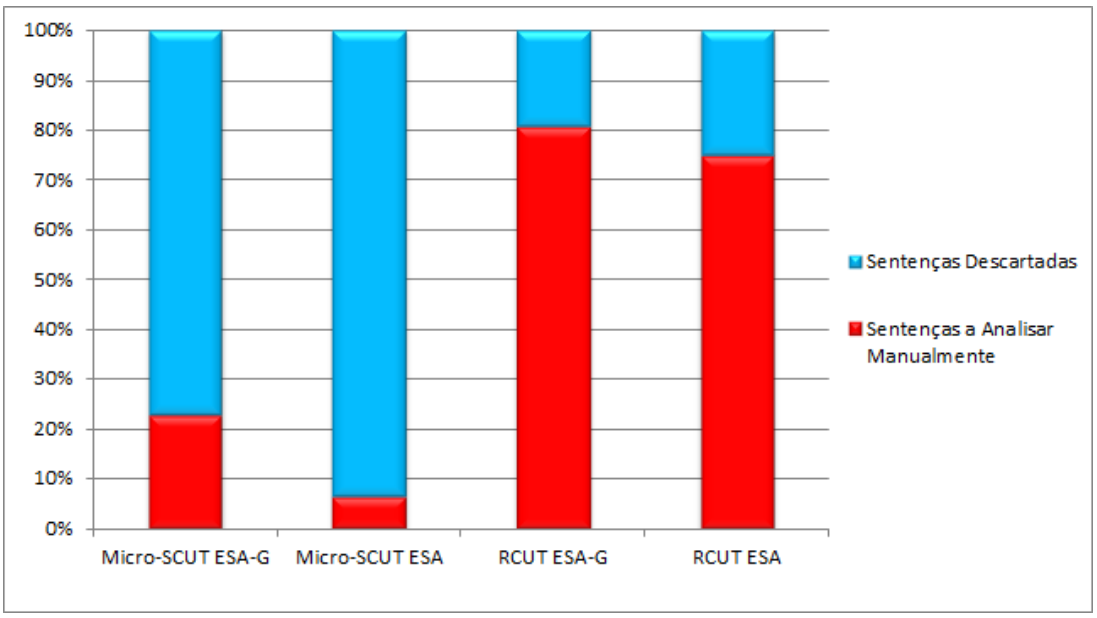

Figura 4.7: Comparação da redução de sentenças providas pelos diferentes métodos da Tabela 4.13.

Observar que a redução apresentada na Tabela 4.13 é contabilizada da seguinte forma (Regra 1): uma sentença x é considerada estar associada a um risco R se, e somente se, x é classificada como R e, ao mesmo tempo, não é classificada como "ADM". Caso, além dessa regra, também exijamos que x esteja relacionada ao conceito "PSW" (Regra 2), obtém-se a distribuição presente na Figura 4.8. Nesta figura, mostra-se a quantidade de sentenças remanescentes pela aplicação da Regra 1 e da Regra 2, respectivamente, em azul e vermelho.

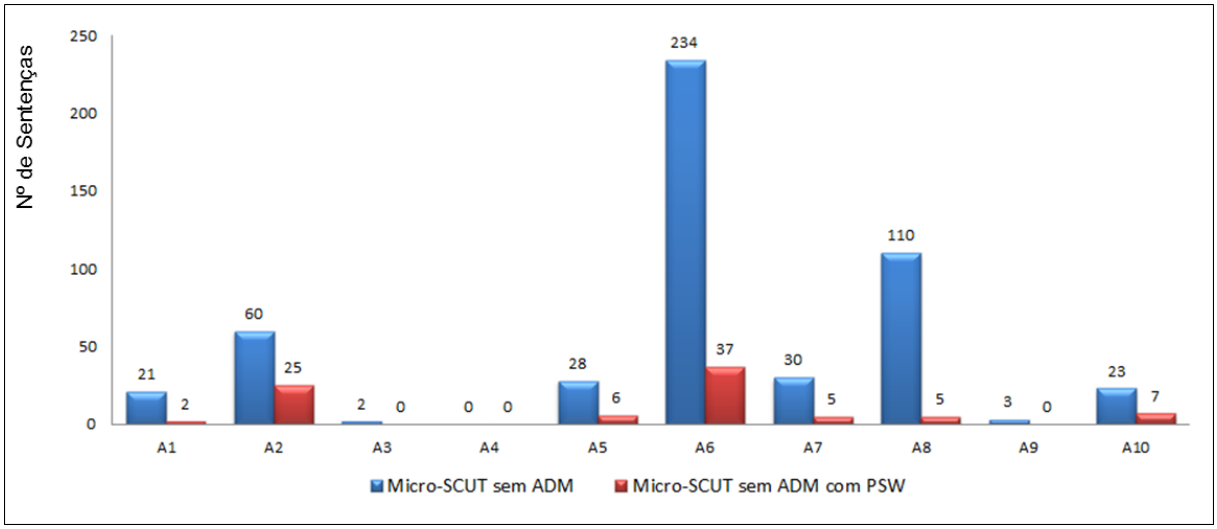

Figura 4.8: Comparação entre estratégias de uso do método micro-SCUT com ESA.

Conjunto análogo de experimentos foi também realizado utilizando as descrições do CWE/SANS *Top 25* no lugar das do OWASP *Top Ten*, modificando assim o sistema de conceitos até então utilizado. Os dados desse novo experimento estão na Tabela 4.14. A potencial redução do trabalho manual provida por cada método testado com o A2E, tendo o esse novo sistema de conceitos (subconjunto do OWASP *Top Ten* descrito pelo CWE/SANS *Top 25*), pode ser visualizado na Figura 4.9. Uma comparação entre a redução provida pelo micro-SCUT ESA usando o OWASP *Top Ten* e o CWE/SANS *Top 25*) pode ainda ser encontrada na Figura 4.10.

Tabela 4.14: Distribuição das Classificações Propostas pelo A2E/CWE na Análise de Editais e Termos de Referência

Categorias / Métodos	**Micro-SCUT ESA-G**	**Micro-SCUT ESA**	**RCUT ESA-G**	**RCUT ESA**
A1	5957	7204	9115	11068
A2	14273	538	29462	9131
A3	1041	336	10892	8688
A4	0	1584	40513	15696
A6	7604	2825	3075	4945
A7	8441	659	5297	10087
A8	455	209	4958	1083
A10	0	0	837	6638
Buffer Overflow	4798	1	8022	5600

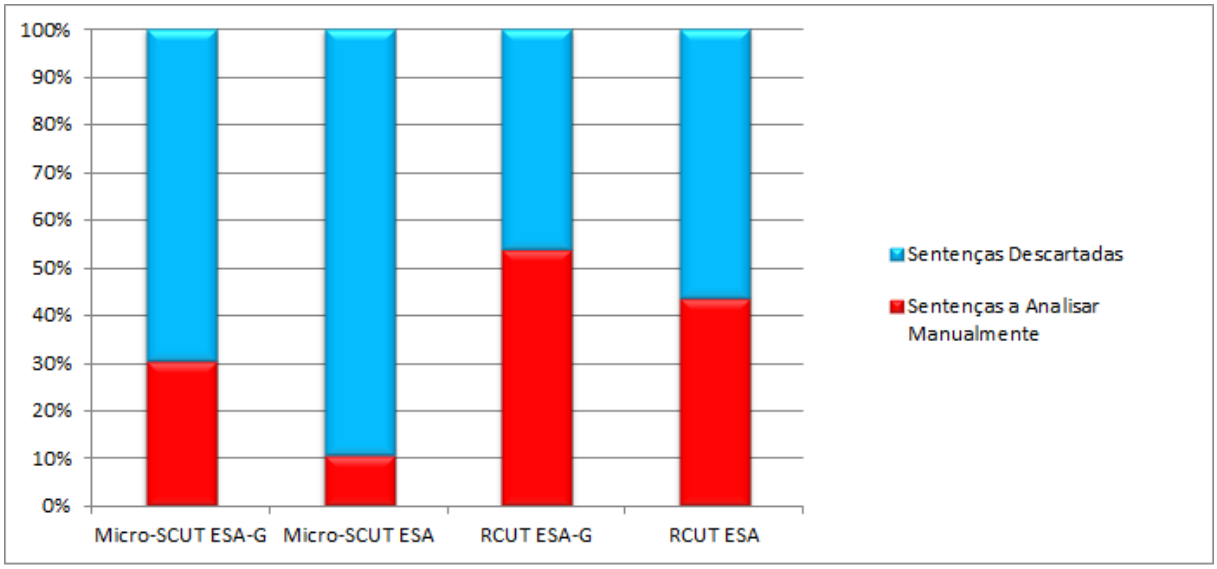

Figura 4.9: Comparação da redução de sentenças providas pelos diferentes métodos da Tabela 4.14.

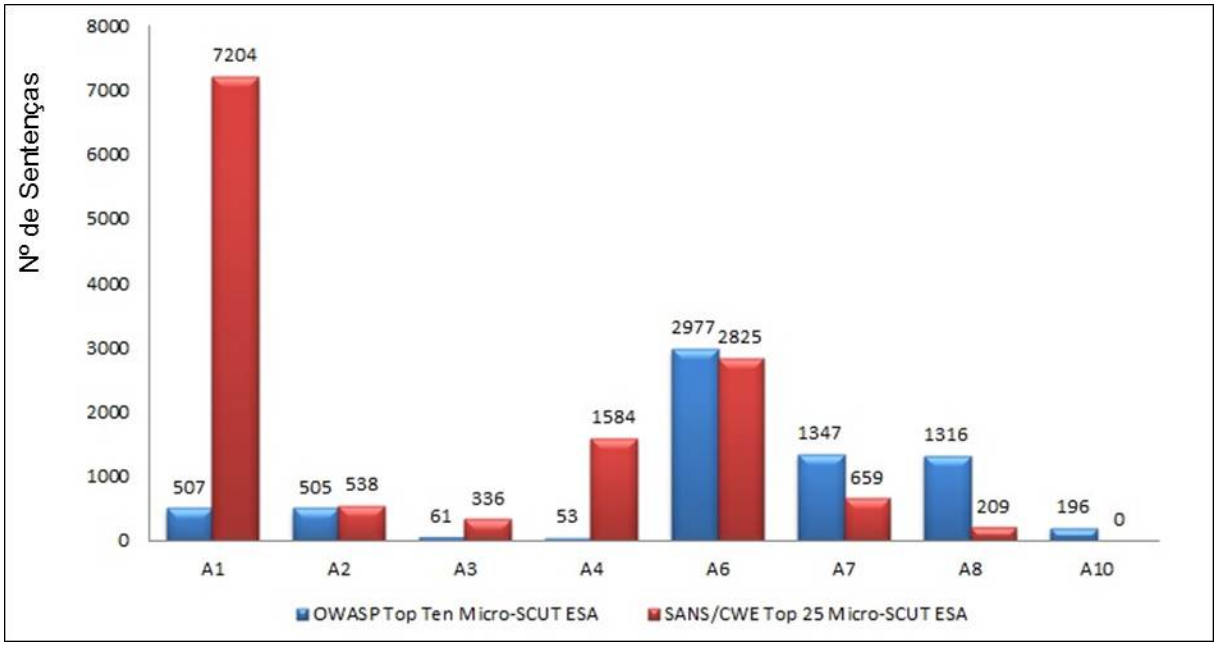

Figura 4.10: Comparação entre os achados utilizando o OWASP *Top Ten* e o CWE/SANS *Top 25*.

A comparação presente na Figura 4.9 também mostra a superioridade do classificador que utiliza micro-SCUT no descarte de sentenças, obtendo, quantitativamente, o resultado mais próximo do esperado - redução do total de sentenças submetidas a cerca de 10% do conjunto original. Ainda se adotando esse critério, a Figura 4.10 indica a superioridade do emprego do A2E com descrições de riscos do OWASP *Top Ten* sobre o A2E utilizando as descrições do CWE/SANS *Top 25*. Entretanto, apesar dessa maior eliminação de períodos textuais empregando o OWASP *Top Ten* como sistema de conceitos, esse fator, por si só, não pode ser considerado como determinante para sua escolha.

Até então, os experimentos apresentados focaram em um único aspecto quantitativo da classificação. Entretanto, para que o A2E possa ser útil aos seus usuários, é necessário que a qualidade de sua classificação seja aceitável, isto é, que apresente aos seus usuários períodos textuais que possam auxiliar a tomada de decisão dessas equipes quanto à integração ou não de gestão de riscos nas especificações de serviços de fábricas de software. Para isso, nova análise foi realizada envolvendo a amostragem do conjunto de 1807 sentenças $P1$, classificadas pelo A2E em um ou mais riscos do OWASP *Top Ten* utilizando-se a combinação das técnicas de Micro-SCUT e ESA, além da referida enciclopédia como sistema de conceitos. Em experimentos da Fase 1, a variância obtida na medição da precisão (a nível de linha - RCUT) foi de 0.057392, o que, pela fórmula de determinação de tamanho de amostra aleatória sem reposição [83], tomando-se um grau de confiança (α) de 95% e um erro máximo de estimação igual a 0.05, resultou na necessidade de se-

lecionar 85 sentenças distintas de $P1$. O objetivo era submeter esse subconjunto a um *survey* [65] com especialistas em segurança de software para utilizar essas respostas como *benchmark* na medição do desempenho do A2E, proporcionando a possibilidade de medir o desempenho do A2E tomando os julgamentos desses profissionais como referência. Contudo, a avaliação inicial da amostragem de $P1$ indicou que a maioria das sentenças ali presentes sequer eram diretamente relacionadas ao conceito de processo de software "PSW" (ex: sentenças relacionadas estritamente a aspectos de Direito Administrativo), trazendo o risco de desperdício dos julgamentos desses profissionais em software seguro, levando-se à consideração de outras alternativas como a utilização do CWE/SANS Top 25 como sistema de conceitos.

Assim, amostrou-se o conjunto $P2$ (resultado da classificação realizada pelo A2E, em iguais condições às do parágrafo anterior, apenas mudando a descrição do sistema de conceitos para as traduções dos principais termos oriundos da representação em inglês do CWE/SANS *Top 25*), composto de 12768 sentenças, de forma análoga ao realizado sobre $P1$, obtendo-se um subconjunto que, numa análise preliminar, se aproximava muito mais de engenharia de software (seja se relacionando a riscos de segurança, seja se relacionando a processos de software) do que a amostra obtida de $P1$ (a qual praticamente não apresentava sentenças desse domínio). Assim, optou-se por trabalhar sobre $P2$.

A fim de melhor compreender como as sentenças de segurança se diferenciam das demais, recorreu-se novamente à submissão de exemplos do OWASP ASVS ao A2E para analisar como as distribuições de probabilidade desses períodos se comportam. A Figura 4.11 apresenta para cada uma das categorias do $P2$ duas funções densidade: uma (em azul) correspondente ao grupo que trata do risco de software e outra (em vermelho) que com este não se relaciona. Observa-se em ambas maior concentração de elementos em pontuações mais baixas, com a diferenciação entre elas tornando-se maior à medida que o valor atribuído pelo A2E aumenta.

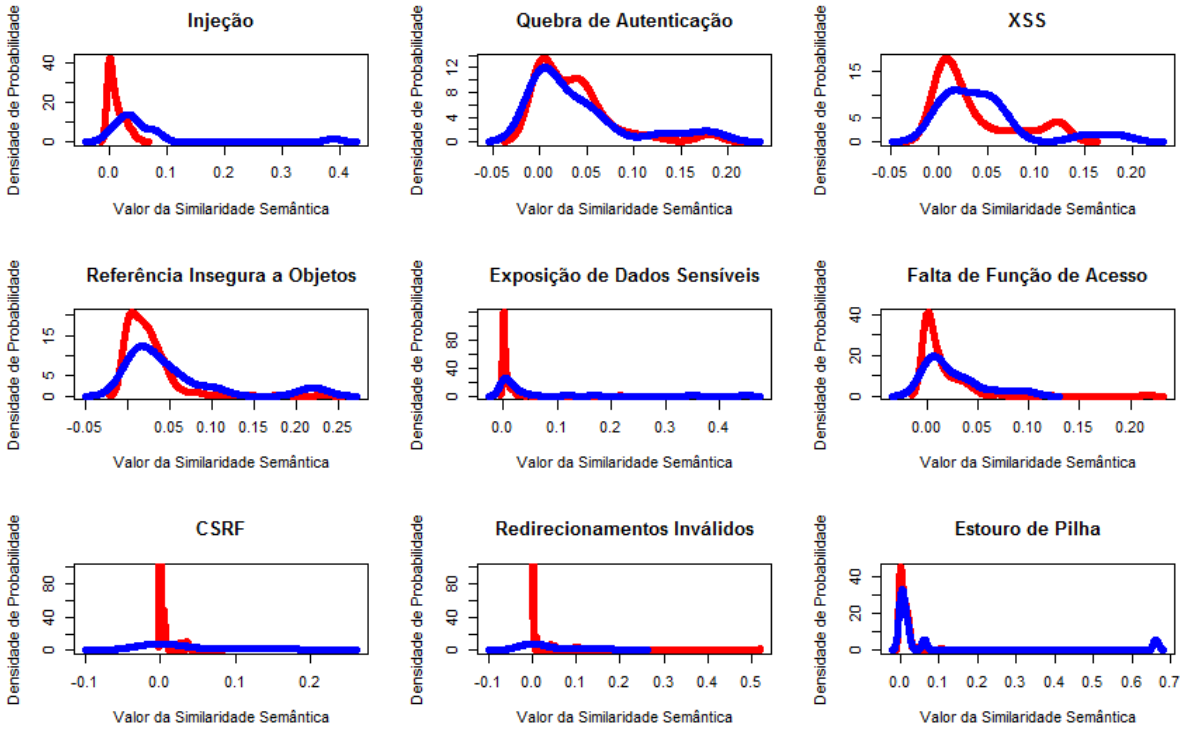

Figura 4.11: Densidades de probabilidade dos verdadeiros positivos e dos verdadeiro negativos das categorias do A2E.

A Figura 4.12 ajuda a explicar esse fenômeno ao mostrar que cerca de 70% das instâncias analisadas têm somente uma ou duas das palavras-chaves extraídas do CWE/SANS *Top 25* por meio do Algoritmo 4. Considerando que, para o sistema de conceitos em questão $P2$, o TF-IDF médio de um termo em relação às categorias em análise é 0.01239695, sentenças com pontuação muito altas tornam-se mais raras. Contudo, cabe observar que, em muitas das classes de riscos apresentadas, com o aumento da pontuação obtido pelos períodos, a densidade de probabilidade dos textos não pertencentes ao respectivo rótulo (vermelho) tende a zero bem antes da densidade daqueles pertencentes (azul). Dessa forma, teoricamente, esse intervalo apresenta menor probabilidade para falsos positivos.

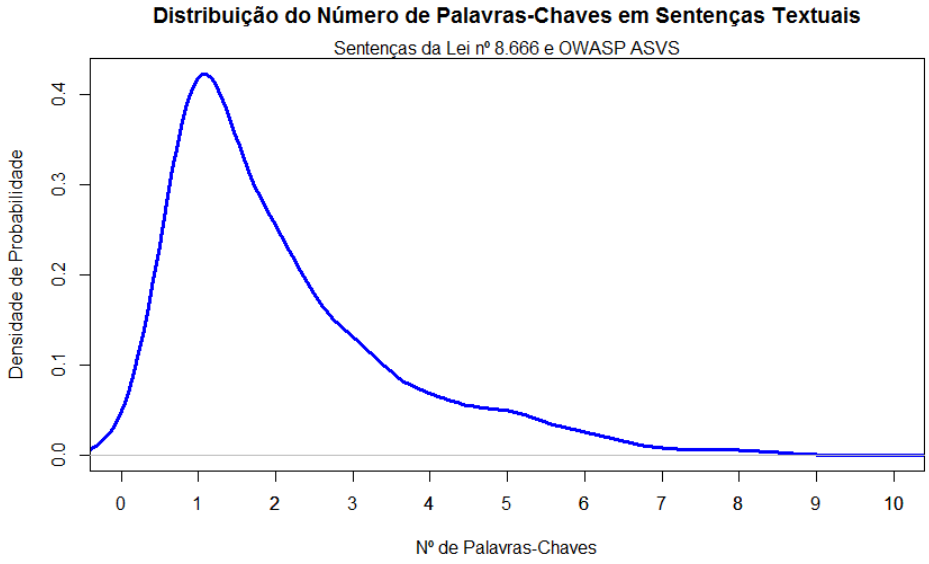

Figura 4.12: Densidades de probabilidade das palavras-chaves do CWE/SANS *Top 25* em sentenças da Lei nº 8.666 e do OWASP ASVS.

Poder-se-ia ainda argumentar que a busca por períodos com pontuação mais alta tem poucos resultados; entretanto, deve-se considerar ao mesmo tempo que restringir a busca à intervalos de pontuação mais baixa tem maior probabilidade para falsos positivos, consequentemente afetando a Precisão da solução. A ideia de atingir um limiar ótimo para cada categoria já foi explorada em estudos sobre classificação multirrótulo, como [54], o qual propõe um algoritmo, utilizado neste trabalho, para obter limiar que maximize o F-*score* nas categorias analisadas.

Diante de todo o exposto, é apresentado na Figura 4.13 visão geral da variação arquitetural discutida nesta Seção.

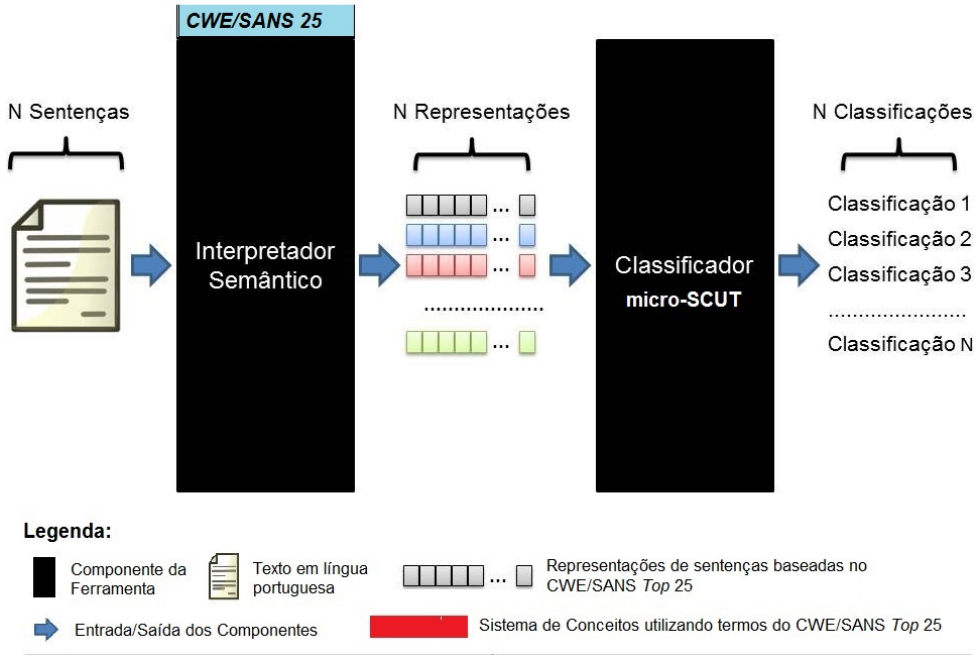

Figura 4.13: Proposta de arquitetura para o A2E com classificador de um único limiar por categoria.

4.4 *Survey*

O A2E, durante as Fases 1 e 2 dos experimentos descritos, foi analisado, seja usando requisitos de segurança, seja usando sentenças em geral extraídas de editais e termos de referência de contratações de fábricas de software, com o objetivo de avaliar o desempenho de uma abordagem semântica de processamento de textos na associação de trechos textuais a riscos de segurança previamente definidos. Até então, a principal medida utilizada para comparação entre diferentes alternativas textuais foi a capacidade de redução do conjunto de instâncias original para um subconjunto que melhor se aproximasse da previsão presente em [22], a qual identificou, ao analisar requisitos de vários projetos de software, que os de segurança não passavam de 10% do total. Dado que as especificações presentes nesses documentos de contratação ainda englobam tópicos de direito administrativo (além de especificações relacionadas à produção de software), espera-se que a proporção de sentenças relacionadas à segurança não ultrapasse esses 10% do total analisado.

Contudo, assim como aconteceu com a proposta apresentada na Fase 1 de utilização combinada de um classificador micro-SCUT sobre uma representação gerada pelo ESA, usando as descrições do OWASP *Top Ten* como sistema de conceitos, embora essa redução

percentual do conjunto de instâncias possa ser um indicador do número de falsos positivos produzido pelo A2E, não é uma medida suficiente para compreensão do desempenho desse protótipo na análise de sentenças, pois, assim como no caso exemplificado, ele pode prover um subconjunto bem menor do conjunto de exemplos, mas de períodos textuais com pouca ou nenhuma afinidade ao domínio de segurança de software.

A fim de analisar as propostas arquiteturais do A2E de forma menos enviesada, propôs-se a realização de um *survey*, com o objetivo de, a partir de um subconjunto das sentenças trabalhadas nas Fases 1 e 2, apresentar uma avaliação indireta de diferentes especialistas em segurança e desenvolvimento de software, comparando a avaliação destes sobre o referido subconjunto com a mesma avaliação provida pelo A2E. Para isso, realizou-se um planejamento para que, nos meses de março e abril de 2015, fosse realizada pela Internet, a coleta de respostas desses profissionais.

Visando minimizar a carga de recursos necessários para a administração do *survey*, conforme o princípio da otimização de custo proposto em [67], optou-se pelo uso de um formulário *online*, Apêndice C deste texto, para a coleta e gerência das respostas junto a esses especialistas, os quais foram convidados por *e-mails* a partir da identificação do envolvimento desses profissionais em grupos de potencial interesse na questão de desenvolvimento, manutenção e aquisição de software seguro. Foram comunidades convidadas para participação:

1. comunidade do SISP[11]: o Sistema de Administração dos Recursos de Tecnologia da Informação (SISP), do Poder Executivo Federal, tem por finalidade, pelo art. 2º, inciso IV do Decreto nº 7.579, de 11 de outubro de 2011 [84], o estímulo do uso racional de recursos de TI. Seu órgão central é responsável, por exemplo, pela instrução normativa dirigida a contratações de bens e serviços de TI no Poder Executivo [30]. A partir de sua comunidade "Núcleo de Segurança da Informação e Comunicações do SISP" foram enviados convites a seus membros.

2. capítulos brasileiros da OWASP: as comunidades da OWASP de Belo Horizonte, Brasília, São Paulo, Florianópolis, Rio de Janeiro, Porto Alegre e Cuiabá foram convidadas para participação, seja por meio de e-mails dirigido ao seu administrador, seja por envio direto à comunidade, após associação a cada uma delas.

Além dessas comunidades, foi realizada pesquisa de currículos relevantes na Plataforma Lattes[12] e enviado uma série de *e-mails* individualizados para pesquisadores de áreas afins a de riscos de segurança em software. Por último, ainda foi realizado convite presencial em treinamento oferecido pelo Departamento de Segurança da Informação e

[11] `http://www.sisp.gov.br/`
[12] `http://lattes.cnpq.br/`

Comunicações do Gabinete de Segurança Institucional[13] da Presidência da República a gestores de segurança de várias organizações públicas.

Previamente a esses convites, o questionário a ser respondido foi submetido a avaliação externa, por profissional de desenvolvimento de software com conhecimentos básicos dos riscos do OWASP *Top Ten* com o intuito de testar o *design* desse instrumento, conforme recomendado em [67], o qual, para preenchimento completo, levou cerca de 30 minutos. Devido à escassez de recursos, como tempo e pessoas, para a realização de outros testes sobre esse formulário, optou-se por deixar em seu corpo o contato do autor deste estudo para recebimento de dúvidas, sugestões e reclamações, conforme formulário do Apêndice C desta dissertação. Ao abranger a revisão do conteúdo do formulário tanto por especialistas, quanto por respondentes, aproxima-se a sua avaliação da estratégia de "Validação de Conteúdo" apresentada na Seção 2.4.5.

Observa-se ainda que o questionário é composto por 86 questões não obrigatórias, derivadas da amostragem do conjunto $P2$ e do conjunto utilizado na Fase 1 (sentenças do OWASP ASVS e da Lei nº 8.666), para as quais se solicita que o respondente associe a sentença textual informada a uma ou mais das seguintes opções:

1. A1 - Injeção de Código;
2. A2 - Quebra de Autenticação e Gerenciamento de Sessão;
3. A3 - *Cross-Site Scripting*;
4. A6 - Exposição de Dados Sensíveis;
5. A7 - Falta de Função para Controle do Nível de Acesso;
6. "Não se Relaciona com Nenhum desses Riscos".

O quantitativo de sentenças escolhido para a realização do *survey* foi derivado da amostragem aleatória desse conjunto com um grau de confiança (α) de 95% e um erro máximo de estimação igual a 0.06, ainda supondo uma variância de até 0.057392 para as medidas a serem realizadas. Essa amostra foi constituída a partir de dois estratos: o conjunto $P2$ e o utilizado na Fase 1. Essa divisão ocorreu principalmente para garantir a presença de períodos textuais de segurança nesse experimento, bem como para otimizar a participação dos respondentes, permitindo que, de uma só vez, eles pudesse submeter respostas aos conjuntos utilizados nas Fases 1 e 2 deste trabalho.

Ao total, oito foram as respostas recebidas nesse processo, as quais estão sumarizadas no Apêndice C. Devido ao baixo número de respondentes, já esperado pelo exposto na Seção 2.4.5, optou-se por considerar o máximo possível de suas avaliações, isto é, foi

[13] `http://dsic.planalto.gov.br/`

realizado o aceite das respostas sem a aplicação de verificações como a consistência dos julgamentos entre diferentes especialistas. Exceção a esse aceite foi realizada apenas um dos respondentes, por uma questão de simplificação, o qual, ao submeter apenas três respostas, gerou recorrentemente exceções como "divisão por zero" durante o cálculo das métricas utilizadas para análise do desempenho do A2E.

A fim de viabilizar essa avaliação de desempenho do A2E, a partir das respostas coletadas e dos resultados retornados pelas opções arquiteturais apresentadas ao fim da Fase 2, Figura 4.13, elaboraram-se medidas a nível de instância e a nível de categoria. De forma geral, essa medição seguiu-se a seguinte metodologia:

1. apresentação do desempenho de dois atores: desempenho dos especialistas e do A2E micro-SCUT;
2. desempenho dos especialistas: avalia-se a resposta de cada especialista em relação ao demais, obtendo-se seis avaliações distintas, das quais se toma a média. Assim, por exemplo, no caso da Precisão a nível de instância, mede-se o desempenho do profissional 1 tomando-se como referência as respostas do profissional 2 ao profissional 7, calculando-se a média e o desvio padrão desse respondente para, ao final, se identificar a média e o desvio padrão de todo o grupo;
3. desempenho do A2E: para ambas variações, as respostas do A2E são comparadas com as dos especialistas de forma análoga ao descrito no item 2 (desempenho dos especialistas), com a diferença que se considera, neste caso, sete respostas em vez de seis.

4.4.1 Resultados do *Survey*

A Tabela 4.15 e a Tabela 4.16 apresentam métricas por instância no sentido de mensurar o desempenho do A2E sobre períodos de textos submetidos a essa ferramenta [52].

Tabela 4.15: Desempenho do A2E sobre o OWASP ASVS a nível de instância

	Especialistas		**A2E micro – SCUT**	
	μ	σ	μ	σ
$Precisão_i$	0.5799024	0.2089249	**0.7714286**	0.29277
$Recall_i$	0.5789252	0.1072828	0.372381	0.01996452
Perda de Hamming	0.4752193	0.1310067	**0.3771429**	0.2284523
NPV_i	0.6538787	0.1822603	**0.6928571**	0.316792

Observa-se que pelo fato do A2E micro-SCUT ter tido seu limiar elaborado com base nas sentenças do próprio OWASP ASVS (em outras palavras, pelo conjunto de teste estar contido no de validação), admite-se a ocorrência de ressubstituição, a qual promove

Tabela 4.16: Desempenho do A2E sobre editais e termos de referência a nível de instância

	Especialistas		**A2E micro – SCUT**	
	μ	σ	μ	σ
$Precisão_i$	0.3587503	0.1999745	0.1891449	0.08521921
$Recall_i$	0.3653766	0.1754719	0.1013825	0.03990501
Perda de Hamming	0.4477966	0.09635343	**0.2614747**	0.03753133
NPV_i	0.7404464	0.223486	0.5384259	0.1689512

estimativas com viés otimista para uma métrica, conforme apresentado na Subseção 2.4.5. De fato, na Tabela 4.15, o classificador sobrepuja a opinião dos especialistas nas seguintes medições:

1. Precisão: a proposta A2E micro-SCUT superou a opinião dos especialistas em cerca de 30%;
2. Perda de Hamming: apresenta uma perda cerca de 20% menor;
3. NPV: o A2E apresenta uma ligeira superioridade sobre essa opinião especialista em torno de 5%.

Embora a existência de ressubstituição nesse experimento prejudique uma melhor visualização do desempenho do A2E frente à opinião especialista pelo teste ter sido realizado sobre conjunto utilizado para parametrizar essa ferramenta, optou-se por, ainda assim, exibir esses resultados, viabilizando sua utilização como indicadores para algumas conclusões sobre o classificador (como exemplo, se o resultado com erro de ressubstituição fosse pior do que o obtido com a opinião especialista, poderia ser concluído um desempenho inferior do A2E, o que não foi o caso pelas medidas mostradas).

Essa análise a nível de instância foi também realizada sobre as sentenças oriundas de editais e termos de referência da Administração Pública. Os resultados, sem ressubstituição entre os conjuntos de teste e validação, são mostrados na Tabela 4.16 onde se registra uma Perda de Hamming pelo A2E cerca de 40% menor do que a obtida com a opinião especialista, significando que o erro de previsão no A2E (soma de dois conjuntos: rótulos erroneamente atribuídos e rótulos erroneamente não atribuídos) é menor do que o verificado junto aos técnicos consultados.

Ressalta-se a capacidade de previsão da ausência de relacionamento com um risco de segurança específico (NPV), a qual, apesar de apresentar uma média cerca de 20% inferior à obtida a partir da opinião especialista, pode apresentar um desempenho de até 0.7, dependendo do especialista tomado como base para medição (como mostra os desvios padrões σ apresentados). Isso significa uma capacidade do A2E de apresentar um nível de acerto de até 70% na previsão dessa ausência. Ainda sobre os parâmetros

σ identificados nas aferições tabuladas, ressalta-se que o valor até vezes maior do que o utilizado na amostragem (0.057392) foi causado pela heterogeneidade das classificações apresentadas para uma mesma sentença, o que possivelmente evidencia divergências entre os especialistas participantes do *survey*.

Particularmente quanto aos dados dos experimentos realizados sobre os editais e termos de referência, cabe enfatizar a compatibilidade entre os valores do desvio padrão calculados para o A2E referente a métricas como *Recall* e Perda de Hamming, permitindo a previsão de um intervalo de confiança para essa medidas por meio de técnicas como o emprego da distribuição t-Student [70].

A Tabela 4.17 e a Tabela 4.18 apresentam medidas da Precisão do A2E a nível de categoria. Na Tabela 4.17 encontraram-se resultados bem superiores, aos obtidos pelos especialistas em segurança, para a versão micro-SCUT com respeito aos riscos de "Injeção de Código" (A1) e "Quebra de Autenticação e Gerenciamento de Sessão" (A2), o que pode ser consequência da já exposta intersecção entre os conjuntos de teste e validação. Já na Tabela 4.18, que mostra o mesmo experimento da tabela anterior sendo realizado sobre editais e termos de referência, agora sem a citada ressubstituição, é mostrado que a Precisão atingida ficou abaixo da obtida no conjunto das sentenças do OWASP ASVS , com o desempenho variando de acordo com a categoria de risco analisada. Ressalta-se que o risco A7 apresentou menor proporção de perda em relação aos especialistas, com a maior sendo registrada para o risco A2, indicando que existem riscos cuja avaliação do A2E é mais precisa que outros.

Tabela 4.17: Precisão do A2E sobre OWASP ASVS a nível de categorias

Parâmetros da Categoria		**Especialistas**	**A2E micro-SCUT**
A1	μ	0.5451879	**1.0000000**
	σ	0.2870233	0.0000000
A2	μ	0.5922581	**0.7142857**
	σ	0.2646603	0.3659625
A3	μ	0.4829343	-
	σ	0.3811383	-
A6	μ	0.5836713	-
	σ	0.2220134	-
A7	μ	0.6413855	-
	σ	0.1955973	-

A Tabela 4.19 e a Tabela 4.20 voltam-se para o cálculo do NPV. De uma forma geral, tanto nos experimentos realizados sobre o OWASP ASVS quanto sobre os editais e termos de referência foram obtidos para o A2E resultados inferiores aos da opinião especialista.

Tabela 4.18: Precisão do A2E sobre editais e termos de referência a nível de categorias

Parâmetros da Categoria		**Especialistas**	**A2E micro-SCUT**
A1	μ	0.4336168	0.1904762
	σ	0.2347626	0.1533479
A2	μ	0.4617184	0.1984127
	σ	0.2928164	0.1256878
A3	μ	0.2650000	0.1666667
	σ	0.2813500	0.2581989
A6	μ	0.3594841	0.2407407
	σ	0.2605689	0.1181476
A7	μ	0.3099206	0.2301587
	σ	0.3563496	0.2321479

Entretanto, diferentemente do que foi mostrado nas tabelas anteriores, essa inferioridade foi percentualmente bem menor.

Tabela 4.19: NPV do A2E sobre o OWASP ASVS a nível de categorias

Parâmetros da Categoria		**Especialistas**	**A2E micro-SCUT**
A1	μ	0.7269183	0.6974790
	σ	0.3243845	0.3163406
A2	μ	0.6522527	0.5863095
	σ	0.2602510	0.2930099
A3	μ	0.7491625	0.717033
	σ	0.3038383	0.3306191
A6	μ	0.6662225	0.5906593
	σ	0.2592426	0.2875443
A7	μ	0.7187294	0.6291209
	σ	0.2707363	0.2801614

A Tabela 4.19 mostra a opinião especialista superando o A2E por uma porcentagem em torno de 5% a 10%. Com relação à Tabela 4.20, essa diferença percentual é ainda menor. Essa baixa diferença motivou a aplicação de testes para avaliar se os resultados apresentados pelo A2E são, de fato, estatisticamente inferiores aos decorrentes da avaliação humana especializada. Para isso, adotou-se a suposição (pelo Teorema Central do Limite) de que a distribuição de NPVs apresentados na Tabela 4.19 e na Tabela 4.20 seguem uma distribuição Normal, depois do teste de normalidade de Shapiro-Wilk [85] não ter contrariado essa hipótese, para então poder comparar esses resultados por meio de testes t-Student pareados [70], os quais comparam médias de duas populações.

A aplicação deste teste, nos dados da Tabela 4.19, tendo como hipótese alternativa que a opinião especialista seria estatisticamente superior, a um nível de significância

Tabela 4.20: NPV do A2E sobre editais e termos de referência a nível de categorias

Parâmetros da Categoria		**Especialistas**	**A2E micro-SCUT**
A1	μ	0.9094486	0.9028571
	σ	0.07661698	0.07609518
A2	μ	0.8882157	0.8624339
	σ	0.10764395	0.122305
A3	μ	0.9335012	0.9285714
	σ	0.07421506	0.07805056
A6	μ	0.8904647	**0.8928571**
	σ	0.06904703	0.07926581
A7	μ	0.9313670	**0.9365079**
	σ	0.06580604	0.07624679

$\alpha = 0.05$, apresentou um p-valor $= 0.004022$ em relação ao A2E micro-SCUT, confirmando a superioridade dos especialistas.

Resultado diferente foi encontrado na análise dos dados da Tabela 4.20. Aplicação do teste t-Student entre os resultados da opinião especialista e os do A2E micro-SCUT não foi suficiente para descartar a hipótese de igualdade entre essas distribuições (pelo retorno de um p-valor superior a 0.05).

Em termos práticos, essa análise da Tabela 4.20 com o t-Student permite a conclusão de que, sobre os dados analisados, a identificação da ausência de relacionamento entre as sentenças analisadas em editais e termos de referência e os riscos de segurança selecionados para o presente *survey* realizada pelo A2E micro-SCUT é, estatisticamente, tão boa quanto a provida por especialistas em segurança de software, viabilizando, portanto, a equipes da APF carentes desse perfil profissional, envolvidas com a elaboração e revisão desses documentos, a utilização do A2E como ferramenta de apoio aumentando, assim, sua capacidade de realizar avaliações da integração de gestão de riscos de segurança aos processos de software descritos nessas especificações.

Esses dados de Precisão e NPV a nível de categoria podem ser sumarizados pela macro-Precisão e macro-NPV apresentados na Tabela 4.21 e na Tabela 4.22. Junto a eles, também são apresentadas as micro-Precisão e micro-NPV, tanto sobre o OWASP ASVS, como sobre editais e termos de referência. Notar que a Tabela 4.21 complementa a discussão anterior sobre o desempenho superior do A2E, em termos de Precisão, sobre a opinião especializada. Nessa tabela, essa superioridade também é identificada, embora tenha de se considerar os efeitos do erro de ressubstituição ao se ter testado esse classificador sobre um subconjunto de requisitos do OWASP ASVS utilizado previamente para ajuste de seus parâmetros (validação). De forma análoga, a Tabela 4.22 também ratifica o que já foi dito sobre a diferença dos NPVs do A2E e da avaliação humana ao mostrar a

pequena diferença entre as medidas apresentadas.

Tabela 4.21: Micro e Macro Precisão a nível de categorias

Resultados		**Especialistas**	**A2E micro-SCUT**
Editais e TR's	macro-Precisão	0.36594798	0.20529100
	σ	0.0823444	0.0301401
	micro-Precisão	0.360433	0.2310709
	σ	0.1973661	0.08601912
OWASP ASVS	macro-Precisão	0.56908742	**0.85714285**
	σ	0.0590976	0.2020305
	micro-Precisão	0.5673912	**0.7714286**
	σ	0.2195687	0.2927700

Tabela 4.22: Micro e Macro NPV a nível de categorias

Resultados		**Especialistas**	**A2E micro-SCUT**
Editais e TR's	macro-Precisão	0.91059944	0.90464548
	σ	0.0215844	0.0296295
	micro-Precisão	0.8875141	0.866636
	σ	0.09166375	0.06534825
OWASP ASVS	macro-Precisão	0.70265708	0.64412034
	σ	0.0414661	0.0603915
	micro-Precisão	0.596395	0.5258177
	σ	0.2585277	0.2591955

Assim, conclui-se a análise das métricas geradas a partir dos dados coletados pelo *survey*, com o resultado de que o A2E pode apresentar um desempenho tão bom na previsão da ausência de sentenças de segurança em categorias de risco pré-definidas quanto especialistas em software seguro. Esse fato é ainda reforçado pelo menor número de erros com classificações errôneas, como pode ser visto pelos valores obtidos na comparação entre as Perdas de Hamming do A2E e das opiniões de profissionais da área.

4.5 Procedimento para Utilização do A2E

Os resultados apresentados pelo A2E em termos da previsão de ausência de integração de categorias específicas de riscos de segurança a especificações de fábricas de software (NPV por categoria), conforme Tabela 4.20, permitem o planejamento de procedimentos de revisão ou auditoria desses documentos explorando seu desempenho compatível a de especialistas em segurança da informação (ou engenharia de software). No contexto

estudado - contratações de serviços de desenvolvimento e manutenção de aplicações - destacam-se dois grupos específicos de usuários:

1. integrante de equipes de planejamento de contratação: responsáveis pela elaboração ou pela verificação das especificações de fábricas de software; e
2. auditores: responsáveis pela consultoria e avaliação independente em relação a esses documentos.

Pelo primeiro grupo de usuários, pode ser utilizado para verificar a presença de integração entre certos tipos de riscos de segurança, como os do OWASP *Top Ten*, aos planejamentos de fábricas de software produzidos. Um procedimento de verificação poderia abranger os passos presentes no Procedimento 1:

Procedimento 1 Verificação da integração da gestão de riscos de segurança nas especificações de fábricas de software

1: Submeter especificação ao A2E gerando o relatório do número de sentenças associadas a cada categoria de risco
2: **Enquanto** "houver categorias de riscos com nenhuma sentença associada" **Fazer**
3: Identificar junto às áreas responsáveis se há análise de riscos, exigências normativas ou objetivos organizacionais que exijam o tratamento desses riscos
4: **Se** "houver necessidade" **Então**
5: Identificar na especificação os trechos em que as categorias de riscos de segurança consideradas de tratamento necessário são abordadas
6: Reescrever esses trechos de forma a explicitar esses riscos e os controles esperados da contratada para tratá-los
7: Submeter a especificação modificada ao A2E
8: **Fim do Se**
9: **Fim do Enquanto**

Já para os auditores, duas situações de uso são basicamente identificadas: as referentes ao planejamento das suas avaliações e as referentes às avaliações propriamente ditas. No primeiro caso, abordado pelo Procedimento 2, o A2E pode ser utilizado para subsidiar a amostragem dos contratos de fábrica de software a serem auditados quanto à aderência com os normativos que exigem ou recomendam a produção e a aquisição de software seguro, uma vez que ele é capaz de fornecer uma medida de relevância desse tema (número de categorias de riscos de segurança com nenhuma sentença associada) nos documentos de contratações a serem analisados. No segundo caso, Procedimento 3, o A2E pode ser empregado como uma ferramenta de análise não apenas dessas especificações, mas

também das respostas apresentadas pelos gestores às solicitações de informação oriundas da equipe de auditoria.

Procedimento 2 Amostragem de conjunto de editais de fábricas de software

1: Submeter conjunto de editais/termos de referência ao A2E
2: Agrupar por edital o relatório de resultados quanto à presença de categorias de riscos com nenhuma sentença associada
3: Para cada edital, identificar o total de categorias de riscos com nenhuma sentença associada
4: Ordenar de forma decrescente quanto ao total de categorias com nenhuma sentença associada os editais
5: Usar essa informação juntamente com os critérios de materialidade e criticidade para obter uma amostra de contratos a serem auditados

Procedimento 3 Auditoria de especificações de fábricas de software

1: Submeter edital/termo de referência ao A2E
2: **Se** "houver categorias de riscos com nenhuma sentença associada" **Então**
3: Solicitar ao gestor da organização esclarecimentos sobre o tratamento desses riscos no âmbito do contrato analisado, bem como os trechos desses ajustes que embasam esse tratamento
4: Solicitar ao gestor da organização a disponibilização de análises de riscos de segurança realizadas sobre seu portfólio de aplicativos
5: **Se** "Gestor afirma que essas categorias estão contempladas no edital ou no termo de referência e que são de tratamento necessário" **Então**
6: Submeter as respostas do gestor ao A2E
7: **Se** "categorias de riscos apresentam sentenças associadas" **Então**
8: Verificar, usando o OWASP *Top Ten*, se os trechos informados como base desses tratamentos não foram escritos de forma genérica
9: Entrevistar o preposto da contratada, ou sua equipe, questionando a base contratual para a exigência do tratamento desses riscos de segurança
10: **Se** "houve descrição genérica" **Então**
11: Constatar a dificuldade no estabelecimento do preço justo do contrato pela falta de clareza dos métodos necessários para sua realização e riscos não abrangidos pelo contrato
12: **Fim do Se**
13: **Fim do Se**
14: **Fim do Se**
15: **Se** "Gestor afirma que essas categorias não são de tratamento necessário" **Então**
16: Verificar as análises de riscos disponibilizadas para identificar o tratamento organizacional dos riscos de segurança indicados pelo A2E com nenhuma categoria de risco associada
17: **Se** "o conjunto de riscos a ser tratado não foi definido pela organização" **Então**
18: Constatar a ausência de definição organizacional sobre riscos de segurança a serem tratados
19: **Fim do Se**
20: **Fim do Se**
21: **Fim do Se**

Cabe observar que, embora fora do escopo do presente trabalho dissertativo, o A2E poderia ser disponibilizado futuramente a integrantes de equipes de contratação e a auditores com outras funcionalidades presentes nos procedimentos descritos, como agrupamento de resultados por especificação submetida à ferramenta e disponibilização de informações téc-

nicas acerca do risco com nenhuma sentença associada de forma a subsidiar seus usuários em eventuais correções de problemas desses documentos de contratação.

Capítulo 5

Conclusões

Na Introdução desta dissertação, foi descrita a problemática que atinge a Administração Pública brasileira a qual consiste da necessidade de um meio mais eficiente para realizar avaliações da integração de gestão de riscos de segurança às especificações de fábrica de software apresentadas em licitações públicas. Juntamente a essa problema, questões sobre a escassez de recursos humanos especializados e de ferramentas automatizadas nesse domínio foram discutidas. Nos capítulos 3 e 4 deste trabalho, uma solução computacional (A2E) para processamento dessas especificações foi apresentada, visando o tratamento do problema estabelecido, a qual foi submetida a uma série de experimentos em busca da melhor de suas versões dentre todas as propostas nesses capítulos. Por fim, uma configuração final do A2E foi identificada, sendo capaz de prever a ausência da integração da gestão de riscos em categorias pré-estabelecidas com resultados tão bons quanto de especialistas em software seguro participantes de um *survey* ao final dessas experiências. A seguir, nesta conclusão, serão abordadas as considerações finais deste trabalho, bem como suas limitações e trabalhos futuros.

5.1 Considerações Finais

No Capítulo 1, expôs-se o seguinte objetivo geral:

Analisar o desempenho de uma implementação de processamento semântico de textos escritos em português que utilize conhecimento enciclopédico sobre segurança de software (como o CWE e o OWASP Top Ten*) na identificação dos trechos de editais de contratação de fábricas de software da Administração Pública brasileira relacionados à gestão de riscos de segurança considerados relevantes.*

Adicionando-se os seguintes objetivos específicos:

1. estudar a adaptação do A2E ao domínio de segurança de software por meio de experimentos junto a períodos textuais reconhecidamente de segurança de software;

2. estudar a adaptação do A2E ao domínio de segurança de software por meio de experimentos junto a termos de referências elaborados por órgãos e entidades públicas visando a contratação de fábricas de software;

3. compreender alternativas de utilização simples e eficazes do A2E que possam ser acessíveis a equipes carentes de especialista em segurança de software.

De fato, ao longo deste texto, avaliou-se o desempenho de uma proposta, denominada A2E, a qual é voltada para análise de editais de contratação de serviços de desenvolvimento e manutenção de software. A partir de uma proposta inicial de solução expressa no Capítulo 3, realizou-se uma série de experimentos, seja junto a requisitos de segurança de software retirados do OWASP ASVS, seja junto a sentenças textuais de uma amostra de cerca de 100 editais e termos de referência empregados em licitações reais da Administração Pública brasileira, chegando-se a uma configuração arquitetural da proposta que obteve-se um índice acima de 90%, conforme Tabela 4.22, na comparação com avaliações de especialistas humanos quanto à previsão de que um conjunto de riscos previamente definido não está sendo tratado num fragmento de texto.

Dessa forma, a fim de facilitar a equipes de elaboração e de revisão de especificações de fábricas de software a análise quanto à ausência de gestão de riscos de segurança previamente definidos, foi proposto, conforme minuta de Nota Técnica presente no Apêndice B, o encaminhamento do presente trabalho às áreas responsáveis da Controladoria-Geral da União por avaliar a conveniência e a oportunidade de sua integração a seus sistemas atuais ou futuros, esperando-se com isso que tanto o protótipo aqui proposto, quanto seu procedimento de utilização sejam evoluídos pela CGU para permitirem que sejam empregados por essas equipes da APF em suas atividades.

5.2 Limitações

Algumas poucas limitações foram identificadas ao longo deste trabalho, as quais são aqui registradas:

1. limitação de *hardware*: cada conceito trabalhado neste trabalho pode ter um número de termos da ordem de milhares, podendo provocar um alto consumo de memória durante os experimentos. Os algoritmos aqui utilizados foram reescritos para lidar com essa limitação, na maioria das vezes aplicando uma estratégia de dividir para conquistar, em vez de utilizar diretamente as funções já disponibilizadas pelo R;

2. escassez de *frameworks* voltados para a mineração de texto: Adotou-se no âmbito deste trabalho o CRISP-DM, com adaptações, para suportar o trabalho específico

de mineração de textos, mas ele se mostrou limitado, principalmente, nas discussões acerca de qualidade desses documentos, para os quais se teve de recorrer a fontes complementares, como as diretrizes presentes em [60];

3. baixo número de trabalhos anteriores voltados para a associação automatizada a riscos de segurança: embora iniciativas variadas para a identificação de requisitos tenham sido identificadas, como o trabalho registrado em [86], ressentiu-se de pesquisas anteriores que tivessem como objeto explorar o conhecimento enciclopédico de repositórios como o da OWASP e o do MITRE para desenvolvimento de soluções de análise de risco e desenvolvimento de requisitos de software.

4. questões em aberto de sobre a estratificação de conjuntos de exemplos multirrótulos: a estratificação de conjuntos multirrótulos trouxe dificuldade adicional ao trabalho, pois uma divisão problemática pode levar a resultados enviesados, diminuindo a eficácia e a eficiência da solução em problemas reais. Adotou-se uma estratégia neste trabalho, mas a escassez da literatura voltada ao assunto levanta dúvidas acerca de sua adequação, uma vez que comparações entre diferentes métodos de particionamento de conjuntos desse tipo não foram encontradas.

5.3 Contribuições

São contribuições oriundas do presente trabalho:

1. alternativa computacional para tratamento do problema de avaliação da integração de gestão de riscos de segurança à especificações de fábricas de software - de fato, é proposto o Analisador Automático de Editais (A2E) o qual se destina a apoiar equipes de elaboração e revisão de especificações de fábricas de software na integração da gestão de riscos relevantes de segurança de software aos métodos típicos da área de engenharia de software;

2. disponibilização dos códigos em R utilizados ao longo dos experimentos para reutilização no âmbito do desenvolvimento de uma versão do A2E para ambientes de produção - um conjunto de arquivos escritos na linguagem R é disponibilizado para os interessados em compreender as atividades e os algoritmos propostos no âmbito dessa trabalho;

3. base de sentenças rotuladas por diferentes especialistas em segurança e desenvolvimento de software que pode servir para futuras comparações entre o desempenho de ferramentas de classificação de sentenças em português - um conjunto de 86 sentenças extraídas de editais, termos de referência, OWASP ASVS e Lei nº 8.666

foi analisado por 8 diferentes especialistas em segurança e desenvolvimento de software, proporcionando um meio para comparação de novas soluções computacionais de classificação de períodos textuais em riscos de segurança;

4. proposição de conjuntos de experimentos em mineração de textos que podem ser customizados para outras características da qualidade, como desempenho, usabilidade, etc. - o A2E foi desenvolvido para tratar a necessidade de avaliação da integração da gestão de riscos de segurança em especiicações de fábricas de software. Contudo, um processo análogo poderia ser aplicado caso desejássemos avaliar a integração de métodos, como os de acessibilidade do *Modelo de Acessibilidade em Governo Eletrônico* (eMAG);

5. identificação de uma forma de emprego do A2E cujo desempenho é compatível com o de especialistas em segurança e desenvolvimento de software - ao final dos experimentos, conseguiu-se apresentar resultados relativos ao NPV do A2E tão bons quanto os alcançados por especialistas em segurança e desenvolvimento de software, permitindo a elaboração de procedimentos de revisão ou elaboração de especificações de fábricas de software visando prover razoável garantia de que riscos de segurança relevantes estão sendo tratados no âmbito desses documentos;

6. submissão do trabalho à Controladoria-Geral da União para aproveitamento em seus sistemas de informação futuros ou atuais - conforme pode ser encontrado no Apêndice B desta dissertação, uma nota técnica foi elaborada e encaminhada ao setor competente da Controladoria-Geral da União para avaliação e uso posterior da pesquisa realizada neste trabalho.

5.4 Trabalhos Futuros

O primeiro conjunto de trabalhos futuros identificado é sobre a estratificação do conjunto de instâncias utilizadas para validação e teste do A2E. Esse conjunto de sentenças foi particionado de forma que seus subconjuntos estivessem o mais balanceados possíveis com respeito às diferentes categorias (rótulos) existentes. Entretanto, a representatividade desses subconjuntos fica ameaçada por essa forma de estratificação não considerar a combinação entre os rótulos. Um trabalho que considerasse essas combinações, ou pelo menos as combinações mais frequentes dessas combinações, poderia ter como consequência um novo classificador com desempenho superior ao produzido no presente trabalho.

O interpretador semântico do A2E também motiva novas pesquisas. Entre elas, está o efeito do detalhamento dos conceitos utilizados. Descrições conceituais com um certo número de termos poderiam ser descartadas se essa quantidade não atingisse um de-

terminado valor. Essa limitação não foi utilizada aqui. Adotá-la poderia de melhorar o desempenho do classificador aqui utilizado pelo aperfeiçoamento desse interpretador semântico (devido a um novo sistema de conceitos melhor detalhado). Além disso, relacionado a essa questão da descrição dos conceitos, não foi analisado o efeito da adoção do CWE como sistema de conceitos, o qual possivelmente pode atingir bons resultados em especificações de fábricas de software escritas em língua inglesa.

Ainda sobre o interpretador semântico utilizado ao longo deste trabalho, há potencial para seu aperfeiçoamento ao considerar o relacionamento entre as descrições associadas às categorias de risco e as expressões distribuídas ao longo dos editais e termos de referência (como a relação entre o risco A1 e a expressão "injeção de parâmetros"). Em [11] é apresentada ainda a alternativa de emprego de um *parser* semântico que pode ser utilizado nas futuras pesquisas.

O classificador do A2E talvez seja a opção mais óbvia de realização de trabalhos futuros devido ao potencial, não explorado no presente trabalho, de incorporação de técnicas de aprendizagem de máquina. Mesmo a dificuldade inicial desse trabalho da ausência de sentenças classificadas por especialistas foi superada com a realização do *survey* descrito no Capítulo 4. Classificadores probabilísticos, como redes bayesianas [87] e *Naive Bayes* [42], poderão ser explorados devido aos relacionamentos já identificados entre riscos e fraquezas no CWE (o qual pode servir como sistema de conceitos). A aplicação de algoritmos de associação, como o C 4.5 [42], também pode ser promissor na medida em que identifica associações comuns entre rótulos para uma instância.

O efeito que o aumento da granularidade dos trechos de texto submetidos ao A2E (como parágrafos ou conjuntos de parágrafos) tem sobre seu desempenho não foi investigado. Futuras pesquisas poderão utilizar os códigos já escritos em R e verificar se, em vez de uma única sentença, o classificador, ao lidar com um conjunto maior, tem uma melhoria significativa em relação aos resultados aqui expressos.

Referências

[1] ISACA, *COBIT 5: A Business Framework for the Governance and Management of Enterprise IT*. Information Systems Audit and Control Association, 2012. 1, 6, 8

[2] G. McGraw, *Software security: building security in*, vol. 1. Addison-Wesley Professional, 2006. 1, 14

[3] D. LeBlanc and M. Howard, *Writing secure code*. Pearson Education, 2002. 1

[4] J. H. Allen, S. Barnum, R. Ellison, G. McGraw, and N. Mead, *Software security engineering*. Addison-Wesley Professional, 2009. 2, 3, 6

[5] ABNT, "NBR ISO/IEC 27005/2011. Tecnologia da informação — Técnicas de segurança — Gestão de riscos de segurança da informação," *Rio de Janeiro: ABNT*, 2011. 2, 14

[6] G. McGraw, B. Chess, and S. Migues, "Building security in maturity model bsimm v5. 0," 2013. 3, 10, 15

[7] P. Chandra and S. Deleersnyder, "OWASP SAMM: Software Assurance Maturity Model," 2012. 3

[8] TCU, "Acórdão 1200/2014 p - Diagnóstico da situação da estrutura de recursos humanos alocadas na Área de tecnologia da informação das instituições públicas federais," 05 2014. `http://portal2.tcu.gov.br/portal/page/portal/TCU`. 3, 62

[9] B. M. Shuaibu, N. M. Norwawi, M. H. Selamat, and A. Al-Alwani, "Systematic review of web application security development model," *Artificial Intelligence Review*, vol. 43, no. 2, pp. 259–276, 2013. 3

[10] H. Meth, M. Brhel, and A. Maedche, "The state of the art in automated requirements elicitation," *Information and Software Technology*, vol. 55, no. 10, pp. 1695–1709, 2013. 4, 11

[11] E. Cambria and B. White, "Jumping nlp curves: A review of natural language processing research," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, 2014. 4, 12, 18, 34, 35, 90

[12] E. Gabrilovich and S. Markovitch, "Wikipedia-based semantic interpretation for natural language processing," *Journal of Artificial Intelligence Research*, pp. 443–498, 2009. 4

[13] F. G. Tenório and R. Valle, *Fábrica de Software*. Fundação Getúlio Vargas (FGV), 2012. 4, 60

[14] BRASIL, "Lei nº 8.666. Regulamenta o art. 37, inciso XXI, da Constituição Federal, institui normas para licitações e contratos da Administração Pública e dá outras providências." `http://www.planalto.gov.br/ccivil_03/leis/l8666cons.htm`, 06 1993. [Online; accessado 26-junho-2015]. 5, 12, 13, 14, 46, 62

[15] BRASIL, "Decreto nº 5.450. Regulamenta o pregão, na forma eletrônica, para aquisição de bens e serviços comuns, e dá outras providências." `http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/decreto/d5450.htm`, 05 2005. [Online; accessado 26-junho-2015]. 5

[16] DSIC/GSIPR, "Norma Complementar nº 10/IN01/DSIC/GSIPR, Rev 00. Inventário e mapeamento de ativos de informação nos aspectos relativos à segurança da informação e comunicações nos órgãos e entidades da Administração Pública Federal." `http://dsic.planalto.gov.br/documentos/nc_10_ativos.pdf`, 01 2012. [Online; accessado 26-junho-2015]. 5

[17] DSIC/GSIPR, "Norma Complementar nº 02/IN01/DSIC/GSIPR, Rev 00. Metodologia de gestão de segurança da informação e comunicações." `http://dsic.planalto.gov.br/documentos/nc_2_metodologia.pdf`, 10 2008. [Online; accessado 26-junho-2015]. 5

[18] DSIC/GSIPR, "Norma Complementar nº 16/IN01/DSIC/GSIPR, Rev 00. Diretrizes para desenvolvimento e obtenção de software seguro nos órgãos e entidades da Administração Pública Federal." `http://dsic.planalto.gov.br/documentos/nc_16_software_seguro.pdf`, 11 2012. [Online; accessado 26-junho-2015]. 6, 14

[19] D. Ameller, C. P. Ayala, J. Cabot, and X. Franch, "Non-functional requirements in architectural decision making," *IEEE Software*, vol. 30, no. 2, pp. 61–67, 2013. 10

[20] S. Taubenberger, J. Jürjens, Y. Yu, and B. Nuseibeh, "Resolving vulnerability identification errors using security requirements on business process models," *Information Management & Computer Security*, vol. 21, no. 3, pp. 202–223, 2013. 11

[21] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "The detection and classification of non-functional requirements with application to early aspects," in *Requirements Engineering, 14th IEEE International Conference*, pp. 39–48, Sept 2006. 11

[22] A. Casamayor, D. Godoy, and M. Campo, "Identification of non-functional requirements in textual specifications: A semi-supervised learning approach," *Information and Software Technology*, vol. 52, no. 4, pp. 436–445, 2010. 11, 28, 53, 59, 73

[23] J. Slankas and L. Williams, "Automated extraction of non-functional requirements in available documentation," in *Natural Language Analysis in Software Engineering (NaturaLiSE), 2013 1st International Workshop on*, pp. 9–16, IEEE, 2013. 11

[24] L. Huang, *Concept-based text clustering*. PhD thesis, The University of Waikato, 2011. 12, 18, 19, 21, 35

[25] F. Rahutomo and M. Aritsugi, "Econo-esa in semantic text similarity," *SpringerPlus*, vol. 3, no. 1, p. 149, 2014. 12

[26] V. Paulo and M. Alexandrino, "Direito constitucional," *3ª edição. São Paulo: Impetus*, 2005. 12

[27] BRASIL, "Constituição da República Federativa do Brasil." `http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm`, out 1988. [Online; accessado 26-junho-2015]. 12

[28] M. Alexandrino and V. Paulo, *Direito administrativo descomplicado*. Método, 2008. 12

[29] BRASIL, "Decreto nº 7.174 - Regulamenta a contratação de bens e serviços de informática e automação pela administração pública federal, direta ou indireta, pelas fundações instituídas ou mantidas pelo Poder Público e pelas demais organizações sob o controle direto ou indireto da União." `http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2010/Decreto/D7174.htm`, mai 2010. [Online; accessado 26-junho-2015]. 13

[30] SLTI/MP, "Instrução Normativa nº 04. Dispõe sobre o processo de contratação de Soluções de Tecnologia da Informação pelos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP) do Poder Executivo Federal." `http://www.governoeletronico.gov.br/sisp-conteudo/nucleo-de-contratacoes-de-ti/modelo-de-contratacoes-normativos-e-documentos-de-referencia/instrucao-normativa-mp-slti-no04`, 11 2010. [Online; accessado 26-junho-2015]. 13, 44, 74

[31] DSIC/GSIPR, "04/in01/dsic/gsipr, rev 01: Gestão de riscos de segurança da informação e comunicações - grsic," 02 2013. `http://dsic.planalto.gov.br/documentos/nc_04_grsic.pdf`. 14

[32] ISO, "ISO/IEC 25010:2011 - Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models," *Switzerland*, 2011. 14, 59

[33] ABNT, "NBR ISO/IEC 27002/2013. Tecnologia da informação — Técnicas de segurança — Código de prática para controles de segurança da informação," *Rio de Janeiro: ABNT*, 2013. 14

[34] S. Christey, C. Harris, and B. Heinbockel, "Introduction to vulnerability theory," 2009. 15

[35] IEEE, "IEEE Standard Classification for Software Anomalies," *IEEE Std 1044-2009 (Revision of IEEE Std 1044-1993)*, pp. 1–23, Jan 2010. 15

[36] OWASP, "OWASP Top Ten 2013: The Ten Most Critical Web Application Security Risks." `https://www.owasp.org/index.php/Top_10_2013-Top_10`, 2013. [Online; accessado 26-junho-2015]. 15, 16

[37] Veracode, "State of software security report: The intractable problem of insecure software," Tech. Rep. 5, Veracode, Inc, 65 Network Drive, Burlington, MA 01803, EUA, 4 2013. 16

[38] J. Feiman and N. MacDonald, "Magic quadrant for application security testing," Tech. Rep. 1, Gartner, Inc, 56 Top Gallant Road, Stamford, CT 06902, EUA, 7 2014. 16

[39] C. Jones and O. Bonsignour, *The economics of software quality*. Addison-Wesley Professional, 2011. 17

[40] D. Linday, "What does owasp top 10 coverage mean to you...and do you have it?." `http://blog.coverity.com/2014/03/24/owasp-top-10-coverage/`. [Online; accessado 26-junho-2015]. 17

[41] R. Feldman and J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, 2007. 18, 19, 35, 65

[42] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA, EUA: Morgan Kaufmann Publishers Inc., 3rd ed., 2011. 18, 26, 28, 47, 90

[43] P. Norvig and S. Russell, *Inteligência Artificial, 3ª Edição*, vol. 1. Elsevier Brasil, 2014. 18

[44] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis.," in *IJCAI*, vol. 7, pp. 1606–1611, 2007. 18, 19, 20, 21, 35, 36, 38

[45] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JAsIs*, vol. 41, no. 6, pp. 391–407, 1990. 19, 35

[46] E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa, "Wikiwalk: random walks on wikipedia for semantic relatedness," in *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pp. 41–49, Association for Computational Linguistics, 2009. 21

[47] E. Agirre and A. Soroa, "Personalizing pagerank for word sense disambiguation," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 33–41, Association for Computational Linguistics, 2009. 21

[48] A. N. Langville and C. D. Meyer, "Deeper inside pagerank," *Internet Mathematics*, vol. 1, no. 3, pp. 335–380, 2004. 21

[49] M.-W. Chang, L.-A. Ratinov, D. Roth, and V. Srikumar, "Importance of semantic representation: Dataless classification.," in *AAAI*, pp. 830–835, 2008. 21, 37

[50] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. 21, 31

[51] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data mining and knowledge discovery handbook*, pp. 667–685, Springer, 2010. 22, 23, 28, 30

[52] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 8, pp. 1819–1837, 2014. 22, 23, 28, 29, 76

[53] Y. Yang, "A study of thresholding strategies for text categorization," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 137–145, ACM, 2001. 22, 37

[54] I. Pillai, G. Fumera, and F. Roli, "Threshold optimisation for multi-label classifiers," *Pattern Recognition*, vol. 46, no. 7, pp. 2055–2065, 2013. 23, 24, 29, 58, 72

[55] OWASP, "OWASP ASVS - Application Security Verification Standard." `https://www.owasp.org/index.php/Top_10_2013-Top_10`, 2009. [Online; accessado 26-junho-2015]. 24

[56] J. M. Moine, *Metodologías para el descubrimiento de conocimiento en bases de datos: un estudio comparativo*. PhD thesis, Facultad de Informática, 2013. 25

[57] G. Mariscal, O. Marbán, and C. Fernández, "A survey of data mining and knowledge discovery process models and methodologies," *The Knowledge Engineering Review*, vol. 25, pp. 137–166, 6 2010. 25

[58] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0 step-by-step data mining guide," 2000. 25, 26, 41

[59] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for data quality assessment and improvement," *ACM Computing Surveys*, vol. 41, pp. 16:1–16:52, July 2009. 26

[60] ISACA, *COBIT 5: Enabling Information*. Information Systems Audit and Control Association, 2013. 26, 88

[61] P. E. N. Lutu, "The importance of data quality assurance to the data analysis activities of the data mining process," *Knowledge Discovery Process and Methods to Enhance Organizational Performance*, p. 143, 2015. 26

[62] K. Sechidis, G. Tsoumakas, and I. Vlahavas, "On the stratification of multi-label data," in *Machine Learning and Knowledge Discovery in Databases*, pp. 145–158, Springer, 2011. 28

[63] E. Spyromitros-Xioufis, *Dealing with concept drift and class imbalance in multi-label stream classification*. PhD thesis, Department of Computer Science, Aristotle University of Thessaloniki, 2011. 28

[64] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 687–719, 2009. 30

[65] S. L. Pfleeger and B. A. Kitchenham, "Principles of survey research: part 1: turning lemons into lemonade," *ACM SIGSOFT Software Engineering Notes*, vol. 26, no. 6, pp. 16–18, 2001. 32, 70

[66] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to advanced empirical software engineering*, pp. 285–311, Springer, 2008. 32

[67] S. L. Pfleeger and B. A. Kitchenham, "Principles of survey research part 2: designing a survey," *Software Engineering Notes*, vol. 27, no. 1, pp. 18–20, 2002. 32, 33, 74, 75

[68] B. Kitchenham and S. L. Pfleeger, "Principles of survey research part 4: questionnaire evaluation," *ACM SIGSOFT Software Engineering Notes*, vol. 27, no. 3, pp. 20–23, 2002. 32

[69] T. Gottron, M. Anderka, and B. Stein, "Insights into explicit semantic analysis," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, (New York, NY, EUA), pp. 1961–1964, ACM, 2011. 35

[70] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers*. John Wiley & Sons, 2010. 35, 78, 79

[71] M. Lee, B. Pincombe, and M. Welsh, "An empirical evaluation of models of text document similarity," *Cognitive Science*, 2005. 35

[72] H. Anton, *Elementary linear algebra*. John Wiley & Sons, 2010. 39

[73] P. Clements, D. Garlan, L. Bass, J. Stafford, R. Nord, J. Ivers, and R. Little, *Documenting software architectures: views and beyond*. Pearson Education, 2002. 39

[74] PMI, *PMBoK: A guide to the project management body of knowledge*. Project Management Institute, Pennsylvania, EUA, 2013. 41

[75] C. Canongia, A. Gonçalves Júnior, and R. Mandarino Junior, *Guia de Referência para a Segurança das Infraestruturas Críticas da Informação*. Gabinete de Segurança Institucional da Presidência da República, novembro 2010. 43, 46

[76] TCU, "Licitações e Contratos: orientações e jurisprudência do TCU. rev., atual. e ampl." `http://portal2.tcu.gov.br/portal/pls/portal/docs/2057620.PDF`, 2010. [Online; accessado 26-junho-2015]. 45

[77] CGU, "Licitações e Contratos Administrativos: Perguntas e Respostas." `http://www.cgu.gov.br/Publicacoes/auditoria-e-fiscalizacao/arquivos/licitacoescontratos.pdf`, 2011. [Online; accessado 26-junho-2015]. 45

[78] G. Travassos and M. Kalinowski, "IMPS 2012: evidências sobre o desempenho das empresas que adotaram o modelo MPS-SW desde 2008," *SOFTEX, Campinas. OpenURL*, 2013. 46

[79] I. Feinerer, "Introduction to the TM package text mining in R." `http://cran.r-project.org/web/packages/tm/vignettes/tm.pdf`, 2015. [Online; accessado 26-junho-2015]. 50

[80] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, vol. 1. Cambridge university press Cambridge, 2008. 51

[81] S. D. P. Torrano, *Produtividade e Criatividade do Léxico: Os Neologismos na área da informática*. PhD thesis, Universidade de São Paulo, 2010. 51

[82] MITRE, "Building CWE and Consensus." `http://cwe.mitre.org/about/images/lg_consensus.jpg`. [Online; accessado 26-junho-2015]. x, 64

[83] H. Bolfarine and W. O. Bussab, *Elementos de amostragem*. Blucher, 2005. 69

[84] BRASIL, "Decreto nº 7.579. Dispõe sobre o Sistema de Administração dos Recursos de Tecnologia da Informação - SISP, do Poder Executivo Federal," *Diário Oficial da União*, out 2011. 74

[85] N. M. Razali and Y. B. Wah, "Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, lilliefors and Anderson-Darling tests," *Journal of Statistical Modeling and Analytics*, vol. 2, no. 1, pp. 21–33, 2011. 79

[86] A. Casamayor, D. Godoy, and M. Campo, "Mining textual requirements to assist architectural software design: a state of the art review," *Artificial Intelligence Review*, vol. 38, no. 3, pp. 173–191, 2012. 88

[87] Y. Guo and S. Gu, "Multi-label classification using conditional dependency networks," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, p. 1300, 2011. 90

Apêndice A

Descrição Arquitetural do A2E

Este apêndice apresenta a documentaçãodiferentes visões arquiteturais da proposta de solução computacional (A2E), até então desenvolvida, para obter a classificação de sentenças presentes em editais e termos de referência de contratações de fábricas software na Administração Pública brasileira.

A.1 Visão Geral da Arquitetura do A2E

Conforme Figura A.1, a ferramenta desenvolvida nesse trabalho estrutura-se basicamente em duas partes: um interpretador semântico que, a partir de um texto escrito em língua portuguesa, produz a representação vetorial de cada uma de suas sentenças em um espaço de conceitos previamente definido; e um classificador, que processa essa representação associando a sentença analisada, segundo critérios estabelecidos, possivelmente a uma ou mais categorias de riscos de segurança pré-estabelecidos.

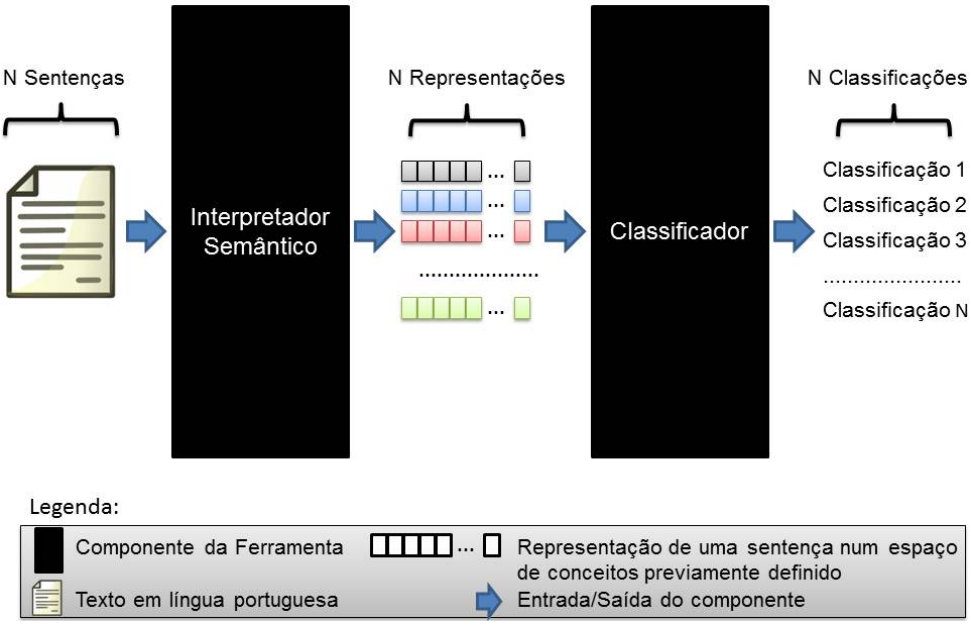

Figura A.1: Visão geral da arquitetura do A2E.

A.2 Visão Modular

A.2.1 Apresentação

Um módulo é uma unidade de implementação que provê um conjunto coerente de responsabilidades. O A2E apresenta dois módulos principais: um de interpretação semântica e outro de classificação de sentenças. A Figura A.2 apresenta uma visão geral não apenas dessas estruturas, como também dos pacotes necessários ao funcionamento correto delas. A seguir, os módulos são detalhados.

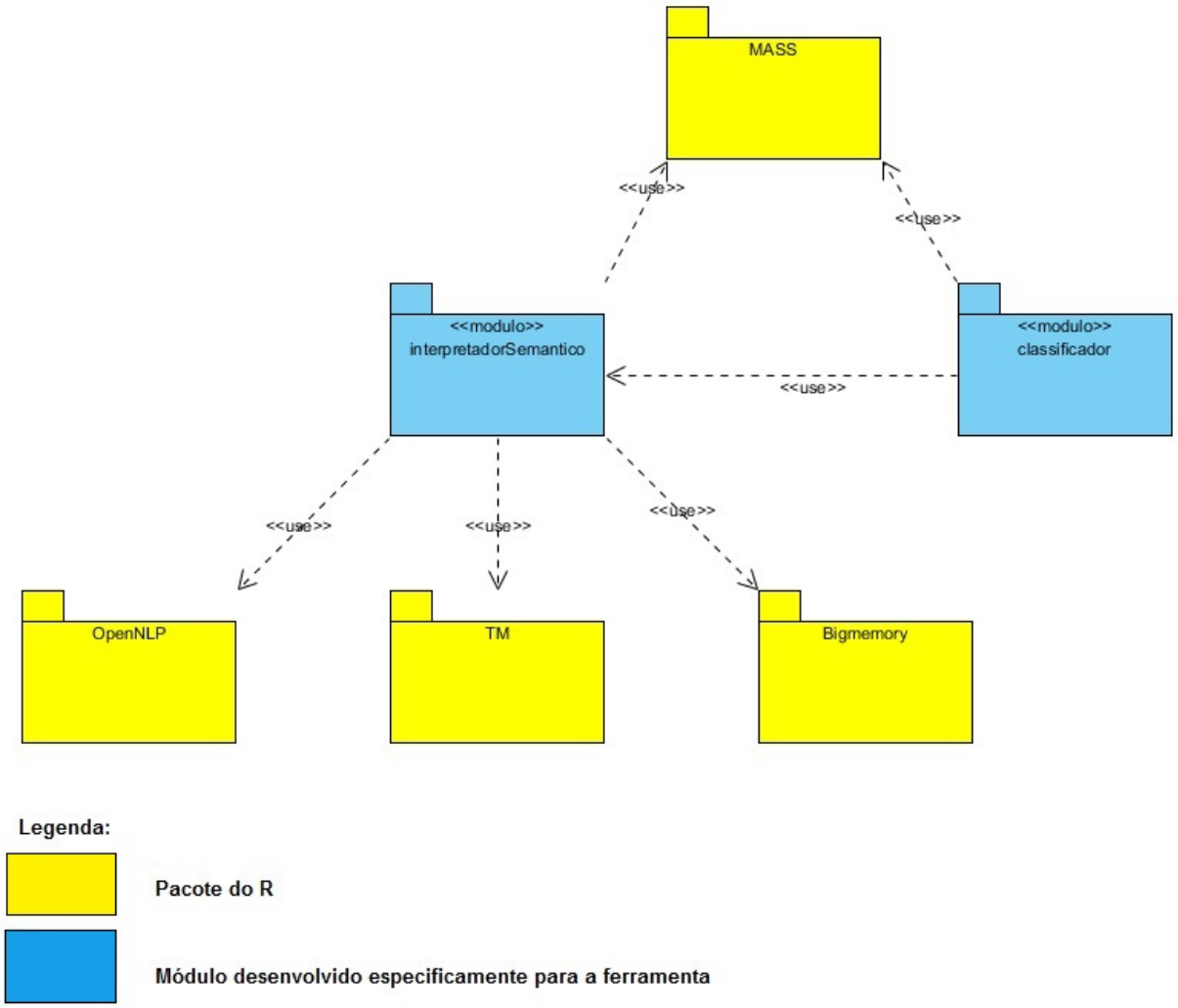

Figura A.2: Diagrama dos módulos do A2E.

A.2.2 Descrição dos Elementos e Relações

Conforme mostra a Figura A.2, são elementos dessa visão:

- Módulo interpretadorSemântico: Responsável não apenas pelo pré-processamento dos textos submetidos ao A2E, como também pela representação das sentenças presentes nesses textos em espaços de conceitos.
- Módulo classificador: Responsável pela associação das sentenças previamente processadas pelo módulo de interpretação semântica a riscos de segurança previamente definidos e associados ao espaço de conceitos pretendido.

Esses são os módulos nativos do A2E. Além deles, os seguintes módulos externos[1], oriundos do R, são utilizados para a construção da ferramenta:

- Pacote OpenNLP: Pacote nativo do R responsável pelo fornecimento de funcionalidades associadas ao processamento de textos em língua portuguesa.
- Pacote TM: Pacote nativo do R responsável pelo fornecimento de funcionalidades associadas à mineração de texto. Como exemplo, está a manipulação das sentenças em corpos.
- Pacote Bigmemory: Pacote nativo do R responsável por permitir o processamento e a manipulação de volumes de dados acima do limite estabelecido pela memória principal, usando a memória secundária de cada máquina.
- Pacote MASS: Pacote responsável por permitir a simplificação da funcionalidade de inversão de matrizes utilizada tanto pelo módulo de inerpretação semântica, quanto pelo módulo de classificação.

Cabe ressaltar que os relacionamentos são todos do tipo "utilização". Dessa forma, pela Figura A.2, é possível observar que o funcionamento correto do sistema é dependente, principalmente, do bom funcionamento dos pacotes discriminados. Além disso, dessa figura, ainda pode ser inferido a dependência do classificador quanto à representação adequada por sentença a ser provida pelo interpretador semântico. Eventuais problemas nessa representação, pela forma como esta ferramente apresenta-se estruturada, potencialmente provocará problemas na classificação obtida.

A.2.3 Justificativas do *Design*

A incorporação de conhecimento externo à ferramenta para a classificação dos textos a ela submetidos trouxe consigo a necessidade de atribuição de dois grandes conjuntos de responsabilidades: um relacionado à representação desses textos e outro relacionado à categorização de sentenças. Dessa forma, torna-se mais clara a opção pela criação dos dois módulos descritos.

O pacote MASS foi escolhido por oferecer procedimento para o cálculo de uma matriz pseudoinversa, facilitando a aplicação do método dos mínimos quadrados para o classificador determinar limiares a nível de linha. De forma similar, optou-se, por uma questão de manutenibilidade, em utilizar o mesmo procedimento para a inversão de matrizes necessário para o módulo de interpretação semântica.

[1] `http://cran.r-project.org/web/packages/`

Já o pacote TM foi escolhido por oferecer, dentro do R, uma API ampla para as atividades de mineração de texto existentes no módulo de interpretação semântica. Essa API foi particularmente importante na manipulação de corpos para a formação de matrizes de sentenças por termos. Ressalta-se que, também por manutebilidade, esse pacote foi escolhido para a construção do A2E, dado que apresenta uma quantidade de documentações, tutoriais e discussões que facilita o aprendizado de seu funcionamento e, assim, da consequente utilizaçao em novas aplicações.

O pacote OpenNLP foi utilizado devido à necessidade de extração de sentenças pelo interpretador semântico da ferramenta. Esse pacote traz consigo um classificador previamente treinado que reconhece em um texto escrito em português o início e o fim de seus períodos. Essa funcionalidade é importante ao se diminuir o nível de granularidade das análises realizadas pelo A2E. Contudo, cabe ressaltar a possibilidade de se trabalhar em maiores porções textuais, como conjuntos de parágrafos. Todavia, no âmbito da presente solução, o objeto de análise são sentenças.

O pacote Bigmemory foi utilizado pelo interpretador semântico como um viabilizador das manipulações matriciais necessárias à solução. Editais e termos de referência apresentam, em média, um número de sentenças da ordem de milhar. Em muitos experimentos realizados durante o desenvolvimento do A2E, 8GB de memória, disponíveis para as computações realizadas, mostraram-se insuficientes para os cálculos exigidos, principalmente aqueles referentes à manipulação de matrizes extensas. Como solução de contorno, a utilização do Bigmemory permitiu que essas computações fossem realizadas com auxílio da memória secundária, viabilizando a adoção, ao longo da solução, de uma estratégia "dividir para conquistar" sempre que problemas de limitação da memória principal apareceram.

A.3 Visão de *runtime*

A.3.1 Apresentação

Uma visão de *runtime* objetiva mostrar os elementos significativos para a aplicação em tempo de execução, os quais se classificam basicamente em componentes e conectores. Enquanto um um componente pode ser caracterizado como uma unidade de processamento, um conector é o mecanismo pelo qual os componentes interagem. A Figura A.3 traz, de forma geral, uma representação dessas estruturas.

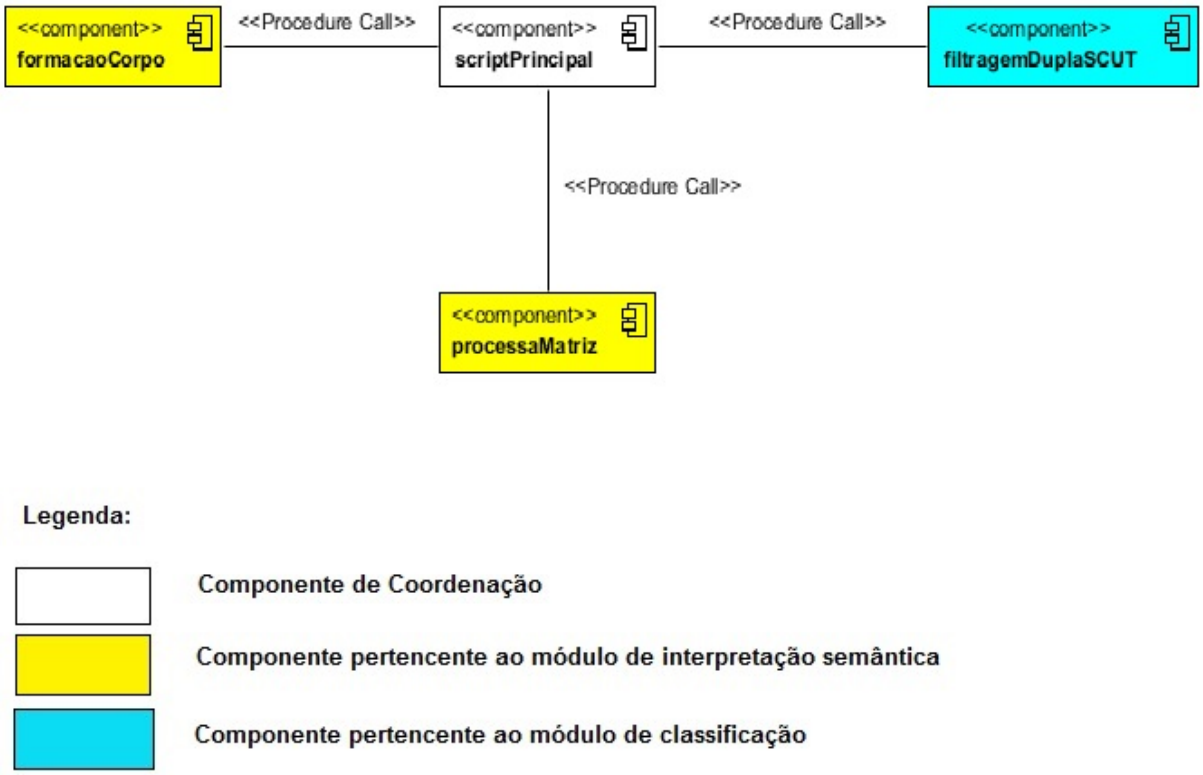

Figura A.3: Diagrama de *runtime* do A2E.

A.3.2 Descrição dos Elementos e Relações

Conforme mostra a Figura A.3, são elementos dessa visão:

- Componente scriptPrincipal: Script responsável pela coordenação das funções oferecidas pelos módulos de interpretação semântica e de classificação com a finalidade de, a partir de um texto escrito em português, associar suas sentenças a riscos de segurança de software.
- Componente formacaoCorpo: Função responsável pela formação de uma matriz sentença por termos a partir de um texto. Os termos utilizados para a construção dessa matriz correspondem à tradução em português daqueles de maior TF-IDF da versão inglesa das descrições de fraquezas do CWE, relacionados a um subconjunto dos riscos de segurança presentes no OWASP Top *Ten*.
- Componente processaMatriz: Função responsável por transformar uma matriz sentença por termos em uma matriz sentença por conceitos. De uma forma geral, essa transformação ocorre a partir da multiplicação matricial entre a matriz sentença por termos utilizada como entrada para esta função e uma matriz termos por conceitos,

a qual representa a contribuição, em termos de TF-IDF, de uma palavra específica para a uma descrição de conceitos, os quais, no caso desta ferramenta, correspondem a um subconjunto dos riscos de segurança presentes no OWASP Top *Ten*.

- Componente filtragemSCUT: Função responsável por realizar, a partir da representação vetorial de cada sentença no espaço de conceitos alvo, uma avaliação de quais de suas coordenadas se encontram num intervalo numérico previamente definido para um determinado risco. Em caso positivo, a sentença é considerada estar associada àquele risco.

Observa-se ainda que esses componentes estão relacionados a partir de chamadas de procedimento (*procedure calls*) realizadas a partir do componente scriptPrincipal.

A.3.3 Justificativas do *Design*

Essa composição é voltada principalmente ao uso do pacote TM, já citado, o qual apresenta como entrada para a maior parte de suas funções um corpo de textos. Dessa forma, a submissão de textos ao A2E provoca a interação entre os componentes scriptPrincipal e formacaoCorpo para estabelecer um corpo a ser manipulado pelo R.

A manipulação desse corpo por meio de diversas funções contribui para a limpeza e a formatação de seus dados, os quais em seguida necessitam ser classificados. A fim de viabilizar essa classificação, a relação entre os termos (palavras) presentes nesse conjunto de textos e os conceitos para os quais se deseja categorizá-los é proposta em uma matriz que calcula o TF-IDF entre os elementos desses grupos (termos e conjuntos). Assim, o scriptPrincipal interage com o processaMatriz buscando a representar esse relacionamento.

Uma vez obtida essa classificação, a filtragem das sentenças ocorre calculando sua pontuação em cada um dos conceitos (a partir do TF-IDF obtido pelos seus termos) comparando com seus limiares previamente definidos. Caso a pontuação da sentença em um conceito seja superior ao correspondente limiar, ela é associada à categoria respectiva. Buscando esse resultado, interage o scriptPrincipal com o filtragemSCUT.

Apêndice B

Nota Técnica de Submissão do A2E à Controladoria-Geral da União

PRESIDÊNCIA DA REPÚBLICA
CONTROLADORIA-GERAL DA UNIÃO
SECRETARIA FEDERAL DE CONTROLE INTERNO
DIRETORIA DE AUDITORIA DA ÁREA DE INFRAESTRUTURA

NOTA TÉCNICA Nº /DICIT/DI/SFC/CGU-PR

Assunto: Proposta de solução computacional para a avaliação da integração de gestão de riscos de segurança aos processos de software descritos em documentos do planejamento da contratação de serviços de desenvolvimento e manutenção de aplicativos.

Senhor Coordenador,

1. Trata-se de proposta de solução computacional para a avaliação da integração de gestão de riscos de segurança aos processos de software descritos em documentos do planejamento da contratação de serviços de desenvolvimento e manutenção de aplicativos em órgãos e entidades da Administração Pública Federal (APF).

2. Essa proposta é decorrente de estudo realizado junto à Universidade de Brasília (UnB), em seu Mestrado Profissional em Computação Aplicada (MPCA), o qual, em sua linha de pesquisa em Engenharia de Software, permitiu a abordagem da problemática da necessidade existente em instituições da APF de produzir software mais seguro e aderente à Norma Complementar DSIC/GSI nº 16, de 21 de novembro de 2012, no cenário atual de carência de recursos humanos e computacionais especializados nessa área.

3. Sobre a relevância estratégica dessa proposta para a Controladoria-Geral da União, recorre-se ao Plano de Integridade Institucional 2012-2015 desse Órgão, o qual inclui entre seus objetivos estratégicos: fortalecer os controles internos e a capacidade de gerir riscos das instituições públicas; fomentar a melhoria contínua da gestão e da prestação de serviços públicos e aprimorar a governança de TI, mediante o alinhamento das ações aos objetivos estratégicos do órgão. Além disso, faz-se referência ao recém lançado sistema de Análise de Licitações e Editais (ALICE) e ao Portal do Observatório de Despesa Pública (ODP), os quais poderiam incorporar a referida proposta, subsidiando seus usuários com informações técnicas a respeito de planejamentos de contratação de serviços de desenvolvimento e manutenção de software (fábricas de software), favorecendo, assim, não apenas o controle posterior desses ajustes, mas também seu controle prévio e concomitante.

4. Conforme registrado na dissertação intitulada “Avaliação Semântica da Integração da Gestão de Riscos de Segurança em Documentos de Software da Administração Pública”, a

DOCUMENTO ASSINADO DIGITALMENTE – VIDE FOLHA DE ASSINATURAS

qual embasa tecnicamente a solução computacional referida nesta Nota Técnica, de 2009 a 2014 foram gastos pela Administração Pública Federal, segundo dados do Portal da Transparência, 3,5 bilhões de reais somente com esse tipo de contratação (fábricas de software), gastos estes de tecnologia da informação apenas superados, segundo informações do mesmo Portal, pelos gastos com consultorias de TI. Esse fato aliado a fatores tais como exigência normativa específica para produção e aquisição de software seguro presente na Norma Complementar DSIC/GSI nº 16 (determinações estas já presentes de forma genérica na Norma Complementar DSIC/GSI nº 2, de 13 de outubro de 2008), carência de recursos humanos especializados nas áreas de tecnologia da informação da APF identificada no Acórdão TCU 1200/2014 - Plenário e problemas na gestão de riscos de segurança e nos processos de software abordados no Acórdão TCU 3117/2014-2 -Plenário motivam a submissão da presente proposta de ferramenta computacional, a qual, como dito, pode impactar positivamente sobre o controle realizado pelos gestores e pela CGU em contratações de fábricas de software, viabilizando aplicações desenvolvidas ou mantidas, no âmbito desses contratos, com maiores níveis de segurança.

5. Acerca do custo para provimento dessa solução, ressalta-se que, para o desenvolvimento da referida proposta, o esforço estimado em homem-hora foi de 720 HH (correspondendo a um único técnico trabalhando cerca de 3 horas por dia, em 2/3 do mês, durante 12 meses), utilizando-se, basicamente, de ferramentas e bibliotecas encontradas de forma gratuita na Internet. Contudo, devido à existência de uma proposta arquitetural já presente na citada dissertação, à identificação, por meio de experimentos, de sua configuração de melhor desempenho na identificação da ausência de gestão de riscos nesses planejamentos de fábrica de software e à existência de material e de analista com conhecimento técnico detalhado sobre o funcionamento dessa ferramenta, aumenta-se a oportunidade de que um novo desenvolvimento dessa solução pela CGU seja realizado com esforço significativamente menor.

6. Dessa forma, sugiro o encaminhamento de cópias desta Nota Técnica ao Núcleo de Coordenação de Gestão de Sistemas e de Informação para Ações de Controle (DCINF) e à Diretoria de Informações Estratégias (DIE), juntamente com a referida dissertação que embasa a proposta em questão, para avaliação da conveniência e da oportunidade de aproveitamento dessa solução em seus sistemas atuais ou futuros.

Assinado digitalmente
RODRIGO NUNES PECLAT
Analista de Finanças e Controle

Apêndice C

Detalhes do *Survey*

Com o objetivo de viabilizar a realização de um *survey* com especialistas em segurança da informação e em engenharia de software, utilizou-se um serviço *online* de elaboração de questionários eletrônicos[1] para disponibilizar a esses profissionais um meio de coleta de suas classificações quanto a 86 sentenças extraídas do OWASP ASVS e de editais e termos de referência de licitações para contratação de fábricas de software pela Administração Pública brasileira, principalmente, mas não de forma limitada, a sua esfera federal. O formulário, as questões, as respostas apresentadas e a sumarização delas se encontram em `https://github.com/rnpeclat/A2E`.

Especificamente quanto ao formulário, ele é composto de 86 questões não obrigatórias, para as quais cada respondente pode associar uma ou mais das seguintes opções:

1. A1 - Injeção;
2. A2 - Quebra de Autenticação e Gerenciamento de Sessão;
3. A3 - *Cross-Site Scripting* (XSS);
4. A6 - Exposição de Dados Sensíveis;
5. A7 - Falta de Função para Controle de Nível de Acesso;
6. Não se Aplica.

A Figura C.1 sumariza a distribuição das respostas apresentadas pelos participantes do *survey*. Ressalta-se que por ela não é possível observar o predomínio de alguma das categorias, embora se possa afirmar que a categoria A3 - XSS foi a percentualmente menos assinalada pelos respondentes, apesar de ser constatado pela Veracode no volume quinto do seu estudo *State of Software Security Report* que esse risco é o mais frequente em aplicações *web*, atingindo até 67% delas.

[1] `https://www.google.com`

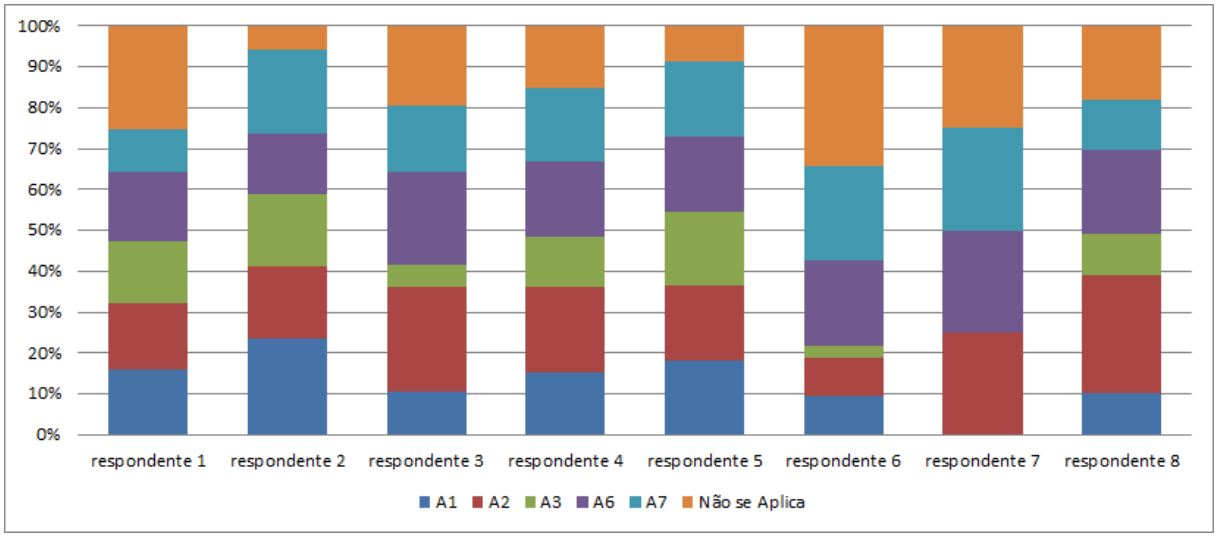

Figura C.1: Distribuição das respostas do *survey* por respondente.

As respostas disponibilizadas no referido endereço eletrônico e sua sumarização mostram que a sentença que obteve menos respondida teve quatro participantes, enquanto a que obteve mais chegou a sete. Em média, houve 5.5 resultados por item, com um desvio padrão de 0.66. A Figura C.2 apresenta o gráfico do número de respondentes por questão do *survey*.

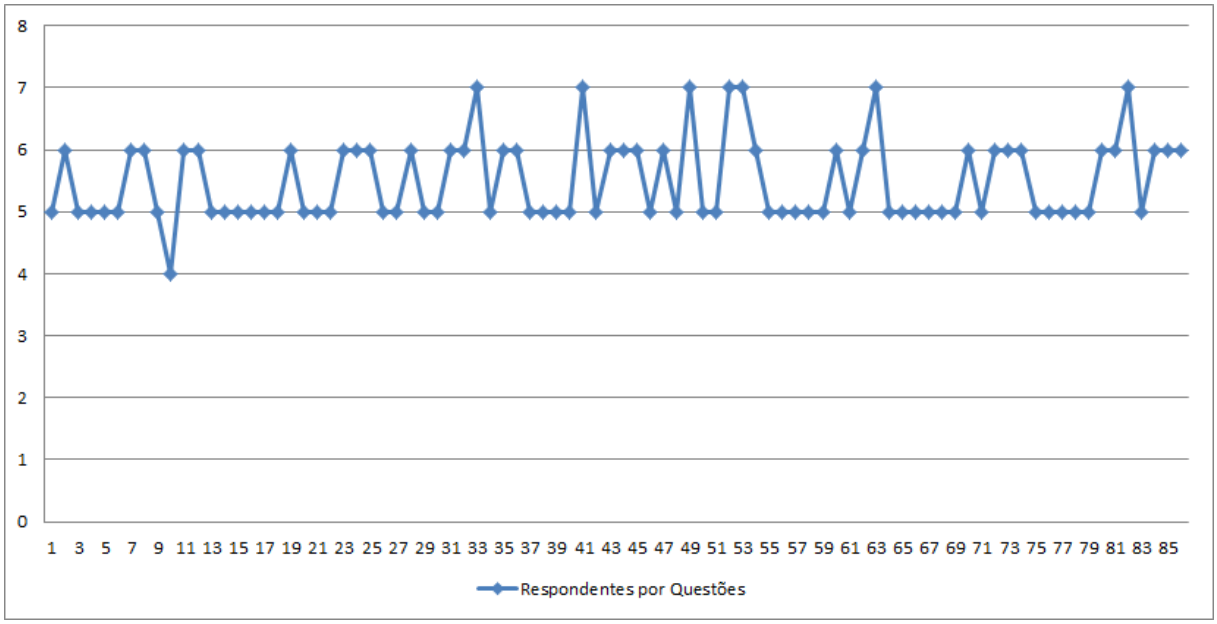

Figura C.2: Número de respondentes por questões.

Ressalta-se que essas respostas permitiram não apenas a realização da comparação entre o desempenho do A2E e de especialistas em segurança e desenvolvimento de software na classificação dessas sentenças, como também permitiram a formação de um repositório que pode ser utilizado em futuros experimentos com novas ferramentas de mineração de texto no domínio de segurança de software.