Análise do artigo “What works better? A study of classifying requirements”

Tomás Ferreira
Universidade da Madeira
tomas.ferr97@gmail.com

Eleutério Mendonça
Universidade da Madeira
mendoncaeleuterio@gmail.com

João Daniel Pereira Jardim
Universidade da Madeira
jd4nnyj@gmail.com

Francisco Sampaio
Universidade da Madeira
FramciscoSampaio@gmail.com

Jorge Miguel Gouveia
Universidade da Madeira
jorgegouveia98@gmail.com

Abstract

No ambito da cadeira de engenharia de requisitos foi proposta a análise de um artigo da IEE à nossa escolha e escolhemos "What works better? A study of classifying requirements".Este artigo consiste numa análise desse artigo ,que inclui resumo desse artigo, discutir os pontos forte e pontos fracos, discussão dos métodos utilizados e sugestões de forma para melhorar o documento.

1. Introdução

Hoje em dia existe um interesse cada vez maior da automatização da transformação de linguagem natural para requisitos funcionais/não funcionais, pois ainda é um desafio. O foco de estudo do artigo é a análise dos diversos métodos de classificação automatizada de requisitos funcionais e não funcionais que existiam na altura. Mais explicitamente, procura arranjar maneiras de melhorar a eficácia dos algoritmos de classificação automáticos para requisitos e também analisa o quão úteis são os métodos de aplicação mais comuns de *machine learning* (e se serão as melhores opções para resolver a questão). As contribuições efetuadas pelos autores do artigo são duas:
1º - investigam se um algoritmo de árvore de decisões atual, quando usado para classificar requisitos, pode ser melhorado através do pré-processamento desses mesmos requisitos, normalizando-os.
2º - analisam o quão bem os métodos atuais de *machine learning* realizam a classificação de requisitos não funcionais em subclasses do tipo disponibilidade, segurança, utilidade, etc.

2. Trabalhos Relacionados

Existem alguns estudos relacionados com este, o de Knauss e Ott sendo um modelo de classificação de requisitos "socio-technical", investigação de Clelande-Huange mining de grandes documentos em requisitos não funcionais e o estudo de Rahimi que apresenta vários métodos de machine learning e data mining.

3. Desafio e Questões de Pesquisa

O desafio apresentado durante a RE’17 envolvia realizar uma tarefa automatizada sob um conjunto de dados, tendo sido escolhida a classificação automática de requisitos para o artigo que se apresenta. O conjunto de dados envolvia 625 requisitos escritos em linguagem natural, não estandardizada, divididos em 255 funcionais e sendo os restantes 370 não funcionais, sendo estes últimos divididos ainda mais uma vez em 11 diferentes subclasses. Para além deste desafio, os autores do artigo sumarizaram o objetivo da sua investigação em duas questões de pesquisa denominadas RQ1 e RQ2. A primeira destas procura descobrir como é que as caraterísticas gramaticais, temporais e sentimentais de uma frase afetam a precisão de classificação de requisitos (em funcionais e não funcionais), enquanto que a segunda questiona até que ponto é que a eficácia de classificação de requisitos não funcionais em subclasses é afetada pelo método selecionado de classificação em *machine learning.*

4. Pré-processamento das Especificações dos Requisitos

Neste segmento é apresentado em extremo detalhe como foi realizado o pré-processamento das especificações de requisitos com o objetivo de reduzir a inconsistência das especificações destes, causadas pelas diversas estruturas de frase e terminologias usadas para descrever os mesmos tipos de requisitos. Este processamento foi feito faseado, dividido em 4 segmentos diferentes: *Part of Speach (PoS) Tagging*, *Entity Tagging*, *Temporal Tagging* e *Co-occurrence and Regular Expressions*. Cada um destes segmentos utilizou um programa/software específico à sua componente de modo a obter os resultados desejados.

O primeiro destes, o *PoS Tagging* envolve o uso do programa Stanford Parser para dividir as frases que descrevem os requisitos nas suas diversas componentes através de *tags*, isto é, as dividir em nomes, verbos, adjetivos, etc. De seguida, o *Entity Tagging* foi responsável por identificar as entidades em cada requisito e as substituir por outras *tags* usando o toolkit LingPipe NLP bem como o Stanford Parser. O *Temporal Tagging* usou o programa SUTIME para realizar o *tagging* de expressões temporais e normalizar todo o conjunto de dados, substituindo essas expressões escritas de diversas maneiras num único estilo. Finalmente, a *Co-occurrence and Regular Expressions* entra em prática depois de ter sido reduzida a complexidade do texto, usando a coocorrência de expressões habituais para determinar a importância de diversas palavras dependendo do tipo de requisito não funcional.

5. Análise de Resultados

RQ1- Influência de características gramaticais, temporais e sentimentais das frases na classificação de requisitos funcionais ou não funcionais.
Na classificação de requisitos funcionais e não funcionais foi usada a abordagem de Hussain et al. Esta abordagem foi aplicada nos dados não processados e processados (resultantes do pré processamento) dos requisitos.

Classificação de processo:

Primeiro, removemos os respetivos conjuntos de dados, retirando de forma alternada erros de formatação e codificação para assegurar o processamento. De seguida aplicamos etiquetas de parte de discurso do Stanford Parser para atribuir partes do discurso de cada palavra em cada requisito. Baseado em marcação de todos requisitos, extraímos 5 características sintáticas, número de adjetivos, número de advérbios, número de advérbios que modifiquem verbos, números de cardinais e números de grau adjetivo/adverbio. Para cada caraterística determinamos a sua classificação baseada na probabilidade da ocorrência da caraterística nos requisitos do conjunto de dados.

Segundo Hussain et al selecionamos um limite de corte de 0.8. Portanto determinamos o número de cardinais e o número de graus de adjetivos/advérbios como características válidas entre todos os 5 para o conjunto de dados não processados. Para o conjunto de dados processados identificamos o número de cardinais e o número de advérbios como características válidas.
Posteriormente, extraímos as características das palavras chaves necessárias para os 9 grupos de palavras chave definidas em parte do discurso: adjetivo, adverbio, modal, determinador, verbo, preposição, substantivo singular e plural. Para cada grupo de palavras chaves calculamos a medida de probabilidade suavizada e selecionamos o respetivo limite de corte manualmente para determinar as palavras chave mais discriminantes para cada conjunto de dados.
A lista final de características para o conjunto de dados processados consiste em 10 características, número de cardinais, números de advérbios, adjetivo, adverbio, modal, determinador, verbo, preposição, substantivo singular e substantivo plural.
Para classificar cada requisito do respetivo conjunto de dados foi implementado um protótipo de extração baseado em JAVA que analisa todos os requisitos do conjunto de dados e extrai todos valores das características mencionadas acima. Depois foi usado o Weka para treinar um algoritmo de árvore de decisão C4.5.
Como o conjunto de dados tinha 625 requisitos, foi realizado uma validação cruzada 10 vezes.

Resultados:

A abordagem utilizada no artigo de pré-processamento impactou positivamente o desempenho da classificação aplicada de requisitos funcionais e não funcionais. Podemos melhorar a precisão de 89,92% para 95,04%, havendo uma melhoria de 4.48%, no total conseguiu-se classificar corretamente mais 28 requisitos, 9 funcionais e 19 não funcionais.
Quando se classifica requisitos não funcionais em subcategorias, a influencia do pré processamento é muito forte. Para todos os algoritmos, os resultados são muito melhores quando usado o pré processamento de

dados comparado com o uso de dados não tratados.

RQ2 - Requisitos não funcionais de classificação é feito em subcategorias a através da modelagem de tópicos, agrupamento e classificação Naïve Bayes.
A modelagem de tópicos é uma técnica de análise de texto não supervisionada que agrupa um pequeno número de palavras altamente correlacionadas, em tópicos. São usados dois algoritmos: o algoritmo de *Alocação Dirichlet Latente* (LDA) que classifica os documentos com base na frequência de coocorrências de palavras e o método Biterm Topic Model (BTM), que modela os tópicos com base nos padrões de coocorrência de palavras e aprende os tópicos explorando os padrões de palavra-palavra. A exemplo, o LDA produz o conjunto de palavras {usuário, acesso, permissão, prévia e detalhe} para o tópico que descreve a subcategoria Tolerância a Falhas, enquanto o BTM produz o conjunto {falha, tolerância, caso, uso e dados}. Assim, o BTM tem desempenho melhor para modelagem e geração de tópicos, mas pior que o LDA para subclassificar os RNFs.
O **agrupamento** é uma técnica de classificação não supervisionada que categoriza os documentos em grupos com base na semelhança, que pode ser definida como a distância numérica entre dois documentos. Utiliza um algoritmo hierárquico que atribui a cada documento o seu próprio cluster e mescla iterativamente os clusters mais próximos e …. K-means

Os algoritmos de agrupamento tiveram um mau desempenho na classificação de RNFs. Isso pode implicar que o conjunto de dados em estudo seja bastante desestruturado e as subcategorias de RNFs não sejam bem separadas.
A Classificação Naïve Bayes que é um método de aprendizado supervisionado que prevê dados não vistos com base no teorema de Bayes. Foi utilizado uma variação do algoritmo multinomial Naïve Bayes (BNB) conhecido como Binarized Naïve Bayes. Nesse método, o termo frequências é substituído por recursos de presença/ausência de booleanos. A lógica por trás disso é a maior importância da ocorrência de palavras do que a frequência das palavras na classificação de sentimentos.

Entre os algoritmos de machine learning LDA, BTM, Hierárquico, K-means, Híbrido e Naive Bayes Binarizado (BNB), o BNB teve o desempenho mais alto para sub-classificar RNFs. Embora o BTM geralmente funcione melhor que o LDA para explorar os temas e tópicos gerais de um corpus de textos curtos, ele não teve um bom desempenho ao subclassificar RNFs.

Existe uma clara necessidade de padrões sentimentais/estruturas de sentenças adicionais para diferenciar os requisitos de usabilidade de outros tipos de RNFs.

6. Limitações e ameaças à validade

Limitação da análise dos dados - A maior limitação é que o modelo de preprocessamento foi desenvolvido baseando-se no conjunto de dados (data set) fornecido pelo RE Data Challenge e tiveram de usar o mesmo data set para avaliar de que maneira respondem a este desafio. Tentaram mitigar esta limitação usando estruturas de frases que são aplicáveis a frases com frases com estruturas diferentes e com contextos diferentes.
Outro fator limitador é que o trabalho dependa da escolha de sub-categorias de RNF usados pelos criadores do data set. Mas o preprocessamento dos autores é adaptável, e têm como objetivo expandir as regras utilizadas ao criar mais sub-categorias de RNF.

Limitação do dataset - Devido à natureza do desafio proposto tiveram de usar o data set sem alterações, que tem 4 grandes falhas.
- Alguns requisitos estão incorretamente definidos.
-A distinção importante entre a qualidade dos requisitos e as restrições não são correntamente refletidas na sua classificação.
-A seleção de requisitos do data set não é imparcial.
- Apenas um único requisito é classificado como PO (portability), o que torna esta subcategoria quase inútil para este estudo.
Apesar de que talvez usar um data set não balanceado pode afetar os resultados deste estudo, existem vários estudos que a importância de ser um data set não balanceado é mínima ou nula.
Estes estudos são: Feitos por Xue e Titterington e o estudo de López et al.

7. Pontos fortes

1) Excelente representação de dados na secção "Analysis and Results", extremamente detalhada e bem explicada e com tabelas e gráficos para todos os diferentes valores obtidos.
Apesar de ser um grande volume de informação, achamos que foram apresentados de maneira concisa os resultados e explicadas as conclusões a que os autores chegaram de maneira clara e fácil de entender.
2) As contribuições feitas pelos autores (nomeadamente a técnica de pré-processamento do dataset de modo a normalizar os requisitos) levaram a um aumento dramático na precisão dos algoritmos responsáveis por dividir os RNF em

subclasses. Registaram-se melhoramentos em todos os métodos usados, havendo até alguns cuja precisão duplicou.
3) Foram ultrapassadas algumas das limitações do dataset (apesar de se continuar a afirmar que os resultados seriam melhores ainda sem estas limitações), tendo sido corrigidas classificações incorretas de requisitos e desprezadas as subclasses inutilizadas por falta de requisitos. Isto foi feito através das features estruturais das frases extraídas dos requisitos e que são depois aplicáveis a frases com contexto e estrutura diferentes.
4) O pré-processamento dos requisitos e a sua normalização foi feita em profundidade e com muita precisão, tendo sido todos os 625 passados de linguagem natural, complicada de classificar, para um conjunto de dados simplificados e sem inconsistências. Isto é comprovado devido aos excelentes resultados obtidos nos testes finais de subclassificação de RNF.

8. Pontos fracos

1) Existem alguma falhas que podem ocorrer durante o processo de classificação, sendo requisitos de diversas subclasses classificados incorretamente como sendo do tipo Usability, provavelmente devido a estas subclasses estarem indiretamente ligadas a esta. Isto acontece devido a uma falta de padrões/estruturas de frases sentimentais que permitam distinguir melhor os requisitos pertencentes a estas subclasses relacionadas.
2) O conjunto de dados usado no estudo faz com que os resultados sejam limitados sendo que o uso de um mais abrangente e mais bem classificado poderia retornar melhores resultados, como o próprio artigo afirma em "Only one single requirement is classified as PO which makes this sub-category useless for our study. The repetition of our study on a data set of higher data quality is subject to future work."

9. Discutir métodos usados

Para reduzir a inconsistência da especificação de requisitos é feito o pré-processamento do conjunto de dados.

O pré-processamento consiste em três partes: Part of speech tagging,Entity tagging e temporal tagging e a coocorrência de expressões.
Part of speech consiste em atribuir partes de palavras a cada requisito. Estas são necessárias pois os autores estão a usar o método de Hussain.
Entity tagging consiste em entidades que são simplificadas usando o toolkit LingPipe e o dicionário SRS, que simplesmente transforma todos as entidades do subgrupo de user (utilizador) em user, faz o mesmo para System (Sistema) e para Product (Produto).
O temporal tagging, que consiste na identificação temporal, por exemplo se o sistema tem que estar funcional num período de tempo então o sistema deteta a parte da frase referente a esse período, por exemplo, da 8 da manhã até às 6 da tarde.
Depois do texto estar simplificado foi usado o método de coocorrência e expressões regulares. Este consiste em explorar 6 a 10 requisitos de cada RNF e colocar diferentes componentes do Stanford Pases tal como o part-of-speech, nomes de entidades, sentimento e relações.
Através deste processamento houve um aumento de 89.92% para 94.40%, ou seja, uma melhoria de 4.48% de requisitos corretamente especificados.
Após este pré-processamento é necessário aplicar algoritmos aos dados antes e após serem pré-processados para testar se a teoria dos autores é correta, respondendo à RQ2 até que ponto é que o performance de classificar RNF em subcategorias é influenciado pelo método de machine learning.
Os algoritmos aplicados são:
BTM- Baseia-se na aprendizagem da coocorrência e padrões de tópicos através de uma exploração de padrões de palavra por palavra. Este método é geralmente melhor para documentos pequenos.
Hierarchical – clusters?
BNB – A lógica deste algoritmo é dar mais importância à coocorrência das palavras do que a classificação de sentimentos. É o melhor algoritmo para especificar subcategorias de RNF's.

10. Melhoramentos para o documento

Data set melhorado:
- Necessita de maior variedade de requisitos, 3 das subclasses não têm nenhum e apenas uma inclui um requisito
- Melhor classificação, há requisitos classificados incorretamente (entre NF e F)
- Não está bem definida a diferença entre requisitos de qualidade e restrições (constraints)
- São necessários mais requisitos em geral
- Usar vários data sets, de modo a estudar os resultados dos algoritmos em diferentes conjuntos de dados com diferentes variações de requisitos.
- Melhoramentos a nível de padrões sentimentais, um dos parâmetros usados para fazer a

divisão de RNF em subclasses. Estes melhoramentos podem ser implementados através de um maior número de padrões e/ou melhorando a qualidade destes.

11. Relacionamento com a disciplina

A classificação de requisitos feita de maneira correta é crucial para um bom processo de desenvolvimento do trabalho. No entanto ainda se fazem bastantes erros humanos durante este procedimento, o que resulta em software mal desenvolvido que não corresponde aos requisitos apresentados pelos stakeholders. Uma maneira de solucionar este problema é a automatização da classificação de requisitos que, seguindo um standard, poderia levar a menos erros e, consequentemente, a melhores resultados no projeto final.

12. Conclusão

A fase de elicitação dos requisitos é considerada a mais complexa e a mais estratégica para obtenção de bons resultados. Por isso erros precisam ser detetados durante esta fase inicial, antes de começarmos a implementação, evitando-se assim o desperdício de tempo e recursos.

Assim, os algoritmos de classificação automática para esse fim, estudados para apoiar a análise de requisitos funcionam melhor com o pré-processamento, melhora o desempenho da classificação de RF e RNFs, e da classificação de RNFs em subcategorias. O algoritmo Binarized Naïve Bayes (BNB) foi o que teve o melhor desempenho na tarefa de classificar a RNF em subcategorias e o algoritmo LDA apresenta bom desempenho ao classificar os RNFs em subcategorias. Finalmente, há uma clara necessidade padrões sentimentais adicionais e estruturas de sentenças que são necessários para diferenciar os requisitos de usabilidade de outros tipos de RNFs.

13. Referências

[1] « What works better? A study of classifying requirements ». [Online]. Disponível em: https://ieeexplore.ieee.org/document/8049172

[2] Slides das aulas de Engenharia de requisitos