Mauro Mascarenhas de Araújo

Heurísticas computacionais para extração de conhecimento em problemas de maratona de programação

Brasil

2019, v-1.0.0

Mauro Mascarenhas de Araújo

Heurísticas computacionais para extração de conhecimento em problemas de maratona de programação

Proposta de trabalho de graduação apresentado ao curso de Ciência da Computação, como parte dos requisitos necessários à obtenção do título de Bacharel.

Universidade Federal do ABC – UFABC

Bacharelado em Ciência da Computação

Centro de Matemática, Computação e Cognição

Orientador: Prof. Dr. Monael Pinheiro Ribeiro

Coorientador: Prof. Dr. Jesús Pascual Mena-Chalco

Brasil

2019, v-1.0.0

Mauro Mascarenhas de Araújo
Heurísticas computacionais para extração de conhecimento em problemas de maratona de programação/ Mauro Mascarenhas de Araújo. – Brasil, 2019, v-1.0.0-17 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Monael Pinheiro Ribeiro

Trabalho de graduação – Universidade Federal do ABC – UFABC
Bacharelado em Ciência da Computação
Centro de Matemática, Computação e Cognição, 2019, v-1.0.0.

1. Análise de enunciados. 2. Sugestão de enunciados. 3. Processamento de Linguagem Natural. 4. URI Online Judge. 5. Maratona de programação.

Resumo

A análise de respostas de problemas, em geral, é um objeto de estudo recorrente no contexto de processamento de linguagem natural. Já o texto do problema, em si, é raramente alvo de estudos um pouco mais aprofundados. A proposta deste projeto é desenvolver um modelo de análise do texto de problemas de maratona de programação utilizando aprendizado de máquina (classificadores) supervisionado. Espera-se que, com o desenvolvimento deste estudo, seja possível fornecer uma base para a confecção de ferramentas de sugestão/recomendação de problemas.

Palavras-chaves: analise de enunciados, sugestão de enunciados, processamento de linguagem natural, URI Online Judge, maratona de programação.

Abstract

On the one hand, the analysis of problem answers is oftenly used as the object of study in the context of natural language processing. On the other hand, the problem description itself is hardly ever target of deeper studies. The purpose of this projet is to develop a textual analysis model for competitive programing problems, using supervised machine learning algorithms (classifiers). It is expected that the generated model to be able to base other tools such as problem suggestion/recommendation.

Key-words: statement analysis. statement suggestion. natural language processing. URI Online Judge. Competitive programming.

Lista de tabelas

Lista de abreviaturas e siglas

JSON	JavaScript Object Notation
K-NN	K-Nearest Neighbours (K-Vizinhos mais próximos)
PCA	Principal Components Analisis (Análise de componentes principais)
PLN	Processamento de Linguagem Natural

Sumário

Introdução

A análise de respostas de problemas, em geral, é um objeto de estudo recorrente, principalmente para buscar relações entre as perguntas e as melhores respostas.

A ideia deste estudo é, no contexto de análise de problemas de maratonas de programação, propor uma análise detalhada dos textos dos desafios, buscando relações entre a formulação dos problemas, a dificuldade deles, bem como as classificações de tópicos para cada um dele (ordenação, busca, etc.).

Diversas técnicas de PLN, tanto as descritas na sessão **Fundamentos**, quanto as omitidas deverão ser empregadas desde o preprocessamento dos dados à geração do modelo de classificação final, uma vez que o texto do problema deverá ser o principal objeto de estudo.

Por fim, busca-se formular um modelo capaz de servir de base para desenvolver ferramentas de recomendação de problemas, considerando suas diversas propriedades (definidas no domínio do modelo).

1 Fundamentos

1.1 Algumas técnicas de PLN

1.1.1 Stemming e Lemmatization

No contexto de PLN, é necessário que duas representações distintas de uma palavra comportem-se de forma similar (uma forma de agrupá-las). Para atingir tal feito, usam-se técnicas que buscam reduzir a palavra à sua forma básica ou literalmente "cortá-las", a fim de que, palavras similares assumam o mesmo formato e/ou comportamento durante o momento do processamento do corpus (JURAFSKY; MARTIN, 2008).

1.1.1.1 Lemmatization

Lemmatization é a técnica que busca reduzir as palavras à sua forma básica, ou seja, é a tarefa de determinar que duas palavras possuem a mesma raiz, independentemente de suas diferenças superficiais (conjungação, por exemplo) (JURAFSKY; MARTIN, 2008). O algoritmo que implementa esta técnica normalmente faz o uso de um dicionário contendo a forma básica das palavras da linguagem a ser analisada.

1.1.1.2 Stemming

Diferentemente da *Lemmatização*, o processo de *Stemming* consiste em "cortar" uma palavra de acordo com um conjunto de regras preestabelecidas (normalmente elaboradas com a ajuda de, ou por, um linguista). As palavras resultantes não devem, necessariamente, ter algum significado aparente, pois, o objetivo é simplesmente normalizar palavras similares (para a língua inglesa, normalmente usa-se o Porter Stemmer, já para a língua portuguesa, usa-se o Orengo). Por ser mais simples, os algoritmos que o implementam costumam ser bem menos custosos que os de *Lemmatização*.

1.1.2 N-grama

Resumidamente, um N-grama "é uma sequência contígua de N elementos (e.g., caracteres, palavras, sílabas, fonemas, pares-base)", normalmente obtidas através da análise de um corpus (MENA-CHALCO, 2019).

1.1.3 Bag Of Words

Bag Of Words nada mais é do que a representação vetorial (ou matricial) da frequência de uso das palavras de um documento. Normalmente, é utilizada a análise da

frequência dos unigramas, caso sejam palavras, presentes no documento.

1.2 Aprendizado supervisionado

Algoritmos de aprendizado de máquina, em geral, buscam encontrar relações entre eventos para que, dado um histórico de acontecimentos e um dado (ou conjunto de dados) pertencentes ao elemento de domínio do modelo, eles possam estimar o elemento correspondente no contradomínio (como uma "predição").

Enquanto os algoritmos de aprendizado de máquina não supervisionados buscam encontrar uma relação entre as variáveis ou observações presentes no modelo, sem qualquer relação de implicação explícita, os supervisionados têm, como característica, o fato de que, para cada elemento preditor X_i (i = 1, 2, ..., n) , haver uma resposta de medição Y_i associada. (JAMES et al., 2013).

1.3 Mineração de texto

A ideia é que estes algoritmos de aprendizado sejam aplicados, em conjunto com técnicas de processamento de linguagem natural (Lemmatização ou Stemmer, remoção de *stopwords*, etc.), sobre um conjunto de dados textual, com o objetivo de, dado um histórico (elementos preditores), no caso, problemas típicos de maratona de programação já existentes, juntamente com suas características adicionais (tipo de problema, quantidade de acertos, número de submissões, etc.) que possam vir a ser importantes (também deverá haver um estudo sobre quais componentes têm mais peso sobre o modelo elaborado).

Já com o modelo desenvolvido e testado, é possível tanto analisar os resultados obtidos do conjunto de treinamento, quanto elaborar sistemas de recomendação, por exemplo, baseado em elementos do domínio do problema (que devem ser devidamente tratados a fim de obter um melhor desempenho nos resultados de busca).

2 Objetivos

2.1 Objetivo geral

Implementar heurísticas computacionais e desenvolver modelos de regressão que possam permitir a caracterização de grupos de problemas (agrupá-los por características relacionadas) utilizando técnicas de PLN e reconhecimento de padrões em geral.

2.2 Objetivos específicos

Os objetivos específicos referem-se às etapas principais do desenvolvimento do projeto:

- Adquirir o conjunto de problemas, juntamente com seus atributos, disponíveis na plataforma URI Online Judge.
- Construir um modelo de análise com base nos dados obtidos, identificando seus principais atributos.
- Implementar um algoritmo de sugestão de problemas com base nos atributos fornecidos como entrada.

3 Método

3.1 Coleta e preparação dos dados

Inicialmente será necessário desenvolver uma ferramenta para realizar a coleta e pré-processamento dos dados. Para isso, será elaborado um web scraper para adquirir os dados tidos como relevantes para a análise dos problemas.

Inicialmente, os dados serão obtidos da plataforma URI Online Judge e convertidos para um documento JSON, que deverá conter todas as informações consideradas relevantes para a análise do problema.

Casos de dados inexistentes, inconsistentes ou que não seja possível serem interpretados no momento da extração (tais como imagens), também estarão indicados, de forma que será mais fácil identificá-los e tratá-los na próxima etapa.

3.2 Aplicação dos algoritmos de mineração

Após a obtenção dos dados, dever-se-á elaborar uma estratégia para lidar com os dados inconsistentes e com a normalização deles, a fim de evitar possíveis *outliers*, o que prejudicaria o desempenho da análise produzida.

Logo após, deverá ser proposto quais atributos coletados dos problemas serão importantes para a realização da análise e, caso haja a necessidade de incluir mais de dois atributos para realizar a mineração, também deverá ser aplicado algum método de redução de dimensão do espaço amostral.

Por fim, deverá ser aplicado algum algoritmo de aprendizado de máquina supervisionado, como o K-NN, Árvore de Decisão, Regressão Linear/Logística, entre outros. Como parte deste processo, também deverá ser feita a validação cruzada para verificar se os parâmetros utilizados nos algoritmos foram eficazes e os mais eficientes possíveis.

Caso não seja obtido resultados relevantes, uma nova análise deverá ser realizada, desde o espaço amostral até o algoritmo utilizado para realizar a análise. Resumidamente, a etapa de aplicação dos algoritmos de mineração deverá ser refeita.

4 Descrição de atividades

4.1 Descrição das fases e metas do projeto

- **Discussão e definição do escopo final do projeto** : Através de diversas com os professores orientadores, será definido o escopo final do projeto, levando em consideração os atributos a serem inicialmente considerados para a análise, a forma como esses atributos serão extraídos e processados, além das possíveis aplicações adjacentes.
- **Implementar web scraper (definição do modelo)** : Dada a definição do modelo no passo anterior, dever-se-á ser definida/construída a ferramenta que será utilizada para realizar a aquisição dos dados através das páginas disponíveis na web.
- **Pré-processamento dos dados obtidos** : Como parte de qualquer projeto de mineração de dados, dever-se-á fazer o pré-processamento de dados, a fim de estabelecer um conjunto de treinamento e de testes suficientemente bons para que a análise apresente bons resultados.
- **Aplicar o algoritmo de agrupamento nos dados obtidos** : Aplicar, efetivamente, os algoritmos que farão a transformação/agrupamento dos itens analisados. Estes algoritmos serão definidos durante os passos anteriores, através de reuniões realizadas com os orientadores.
- **Verificação de correlação entre propriedades e possível redefinição do conjunto de atributos** : É possível que o conjunto de testes e treinamento sejam insuficientes, ou que os atributos utilizados não possuam forte correlação. Portanto, será necessário um tempo para reanalisar os resultados e refazer o modelo.
- **Possíveis otimizações no algoritmo** : Como inicialmente será feita uma prototipação do algoritmo a ser utilizado, também será necessário que correções e melhorias para ganho de desempenho sejam feitas no algoritmo final.
- **Elaboração do modelo final e apresentação dos resultados aos orientadores** : Dever-se-á elaborar um modelo final para que o relatório possa ser gerado com base nele. Aqui os atributos permanentes serão definidos e os parâmetros utilizados nos modelos fixados.
- **Elaboração de possíveis documentações necessárias** : Elaboração de documentação dos softwares desenvolvidos durante o processo.

- **Formulação do artigo científico** : Passo necessário para a formalização do trabalho em formato de artigo científico, projeto acadêmico (de acordo com as normas ABNT), além de possíveis apresentações.

4.2 Cronograma

A Tabela 1 a seguir, descreve o planejamento de atividades a serem desempenhadas mensalmente.

Tabela 1 – Cronograma de atividades

Atividade/Mês	**1º**	**2º**	**3º**	**4º**	**5º**	**6º**	**7º**	**8º**	**9º**	**10º**	**11º**	**12º**
Discussão e definição do escopo final do projeto												
Implementar web scraper (definição do modelo)												
Pré-processamento dos dados obtidos												
Aplicar o algoritmo de agrupamento nos dados obtidos												
Verificação de correlação entre propriedades e possível redefinição do conjunto de atributos												
Possíveis otimizações no algoritmo												
Elaboração do modelo final e apresentação dos resultados aos orientadores												
Elaboração de possíveis documentações necessárias												
Formulação do artigo científico												

Devido a possíveis erros não previstos, o cronograma poderá ser alterado.

5 Resultados esperados

A partir do melhor modelo de regressão obtido, espera-se poder, além de classificar (agrupar) melhor os tipos de problemas, compreendendo melhor os aspectos que os relacionam (a partir a aplicação de técnicas de análise de PLN), criar um sistema de recomendação de problemas, cuja entrada possa ser outro problema, ou ainda alguma característica específica do problema (descrição, dificuldade (que também deverá ser ponderada), tipo de resolução esperada, etc.).

Ferramentas adicionais também poderão ser estudadas e elaboradas, caso não haja excedimento dos prazos descritos no cronograma preestabelecido.

Referências

JAMES, G. et al. *An introduction to statistical learning*. [S.l.]: Springer, 2013. v. 112. Citado na página 10.

JURAFSKY, D.; MARTIN, J. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. [S.l.: s.n.], 2008. v. 2. Citado na página 9.

MENA-CHALCO, J. P. *Modelando a linguagem com N-gramas*. 2019. Http://professor.ufabc.edu.br/ jesus.mena/courses/pln-2q-2019/PLN-aula05.pdf. Online. Acessado em 23 de Agosto de 2019. Disponível em: <http://professor.ufabc.edu.br/~jesus.mena/courses/pln-2q-2019/PLN-aula05.pdf>. Citado na página 9.

Índice