

UNIVERSIDADE FEDERAL DE SANTA CATARINA - CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

**Definição de Modelo de Reconhecimento de Entidade Nomeada
para Detecção Automática de Tópicos de Aplicativos em Português**

Eduardo Dias Gutterres

Florianópolis
2022

Universidade Federal de Santa Catarina
Departamento de Informática e Estatística

Definição de Modelo de Reconhecimento de Entidade Nomeada para Detecção Automática de Tópicos de Aplicativos em Português

Trabalho de Conclusão de Curso de Graduação
em Ciências da Computação, do Departamento de
Informática e Estatística, do Centro Tecnológico da
Universidade Federal de Santa Catarina, requisito
parcial à obtenção do título de Bacharel em
Ciências da Computação.

Autor: Eduardo Dias Gutterres

Orientadora: Prof.^a Dr.^a rer. nat. Christiane Gresse
von Wangenheim, PMP

Coorientadora: Prof.^a Nathalia da Cruz Alves, MSc.

Florianópolis

2022

RESUMO

Com as mudanças sociais e tecnológicas presentes ao longo dos séculos, uma variedade de competências desenvolvidas pelos seres humanos passaram a ser consideradas essenciais. Na era contemporânea, a criatividade é um dos exemplos destas competências. Uma alternativa para o desenvolvimento da criatividade ainda na educação básica se dá por meio do estudo da área da computação, especialmente quando este estudo é realizado a partir do desenvolvimento de artefatos computacionais, por exemplo, aplicativos móveis. O App Inventor é uma ferramenta que proporciona a criação de aplicativos móveis por meio de uma interface de blocos visuais, sendo assim utilizado no ensino da computação na educação básica. Nesse contexto, é importante a avaliação da aprendizagem do aluno também em relação ao desenvolvimento da criatividade. Considerando a originalidade como uma das principais dimensões da criatividade, visa-se assim realizar a avaliação da originalidade dos aplicativos criados com App Inventor em relação aos tópicos que abordam. Usando como entrada para essa análise os elementos textuais extraídos dos apps, a classificação desses tópicos pode ser melhorada com o uso do Reconhecimento de Entidade Nomeada (NER - *Named Entity Recognition*). O NER é uma técnica de computação dentro da área de Processamento de Linguagem Natural (*NLP - Natural Language Processing*) que permite que entidades sejam reconhecidas automaticamente a partir de palavras e frases. Dessa maneira, este trabalho tem como objetivo desenvolver um modelo de NER em português para identificação de entidades a fim de auxiliar na classificação automática de tópicos de aplicativos móveis criados com App Inventor. Com isso, espera-se contribuir para a avaliação da originalidade dos aplicativos criados pelos alunos, contribuindo assim também para o ensino da computação nas escolas do país.

Palavras-chave: Criatividade, Originalidade, Ensino de computação, App Inventor, *Natural Language Processing*, *Named Entity Recognition*

Sumário

1 INTRODUÇÃO	5
1.1 CONTEXTUALIZAÇÃO	5
1.2 OBJETIVOS	8
1.3 METODOLOGIA	9
2 FUNDAMENTAÇÃO TEÓRICA	11
2.1 ELEMENTOS TEXTUAIS DE APPS CRIADOS COM APP INVENTOR	11
2.1.1 APP INVENTOR	11
2.1.2 ELEMENTOS TEXTUAIS DOS APPS	13
2.2 ANÁLISE TEXTUAL AUTOMATIZADA COM NER	18
3 ESTADO DA ARTE	24
3.1 DEFINIÇÃO DO PROTOCOLO DE REVISÃO	24
3.2 EXECUÇÃO DA BUSCA	26
4 ABORDAGEM DE NER PARA CLASSIFICAÇÃO DE TÓPICOS DE APPS CRIADOS COM APP INVENTOR	29
4.1 ANÁLISE DE REQUISITOS	29
4.2 PREPARAÇÃO DO CONJUNTO DE DADOS	30
4.3 TREINAMENTO	32
4.4 TESTE	40
4.5. DISCUSSÃO E TRABALHOS FUTUROS	50
5 CONCLUSÃO	51
REFERÊNCIAS	53

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A constante mudança e evolução da humanidade em diversas áreas fez com que algumas competências e habilidades humanas passassem a ser consideradas essenciais no século XXI, justamente por contribuírem para que as pessoas consigam lidar com situações reais desse período (NRC, 2012). Uma dessas competências essenciais é a criatividade, pois auxilia na criação de ideias inovadoras e na resolução de problemas complexos e atuais (BINKLEY, 2010).

Tipicamente, a criatividade e o seu desenvolvimento estão ligados à área de artes (ALVES et al., 2022). Porém, uma alternativa para o desenvolvimento dessa habilidade na educação básica é por meio do ensino da computação, uma vez que a criação e desenvolvimento de artefatos de software pelas crianças exercita a curiosidade intelectual e estimula a reflexão e a criatividade (ALVES et al., 2022).

Muitas vezes a computação é ensinada na educação básica adotando uma estratégia ativa a fim de tornar o aprendizado mais fácil e dinâmico para os alunos, levando-os a criar artefatos computacionais. Para esse fim, tipicamente, são usados ambientes de programação baseados em blocos visuais (PATTON et al., 2019). Um exemplo é o App Inventor (<https://appinventor.mit.edu>), que permite com que o usuário crie aplicativos para as plataformas Android e IOS por meio de uma interface visual, abstraindo a lógica de programação. Como resultado, podem ser criados aplicativos funcionais (PATTON et al., 2019).

Desta maneira, a criatividade do aluno na educação básica pode ser desenvolvida por meio da aprendizagem de desenvolvimento de aplicativos móveis. Nesse contexto educacional, é também importante avaliar a aprendizagem do aluno com base nos resultados dos apps criados. Assim, similar à avaliação da aprendizagem de conceitos de algoritmos e programação por meio da análise do código do app,

visa-se também avaliar a criatividade de produto baseado nas características do app (ALVES et al., 2020).

Medir a criatividade é uma tarefa complexa (ALVES et al., 2022). Para isso, primeiramente é necessário entender o conceito de criatividade de produto, que se refere aos resultados tangíveis do processo criativo (ou seja, obras de arte, livros, programas de computador, etc) (RANJAN et al., 2018). A avaliação desse tipo de criatividade pode se dar por meio da avaliação das características que a compõem. Embora tais características não sejam consensuais, há um conjunto delas que é considerado o conjunto principal de características de um produto criativo para diversos autores (RANJAN et al., 2018). Seguindo uma definição comum da criatividade de produto, é possível decompô-la em três aspectos (ALVES et al., 2021):

- Originalidade: “Qualidade do que é inusitado, do que não foi ainda imaginado, dito, feito etc.” (Oxford Languages, 2021). No contexto de aplicativos, significa o quanto a aplicação possui aspectos diferentes de outras aplicações já existentes;
- Utilidade: “Qualidade ou caráter do que pode ter ou tem algum uso; que serve ou é necessário para algo.” (Oxford Languages, 2021). No contexto de aplicativos, significa o quanto a aplicação serve a um propósito, seja a resolução de um problema, a criação de uma comodidade ou mesmo entretenimento.
- Qualidade: “O padrão de algo medido em relação a outras coisas de tipo semelhante; o grau de excelência de algo.” (Oxford Languages, 2021). No contexto de aplicativos, significa o grau de excelência da aplicação (utilizando características como usabilidade, acessibilidade, apelo visual, etc) quando comparada a outras aplicações.

Refinando assim o conceito de criatividade de produto, é possível identificar diversas dimensões da avaliação da originalidade de aplicativos móveis, conforme apresentado na Figura 1.

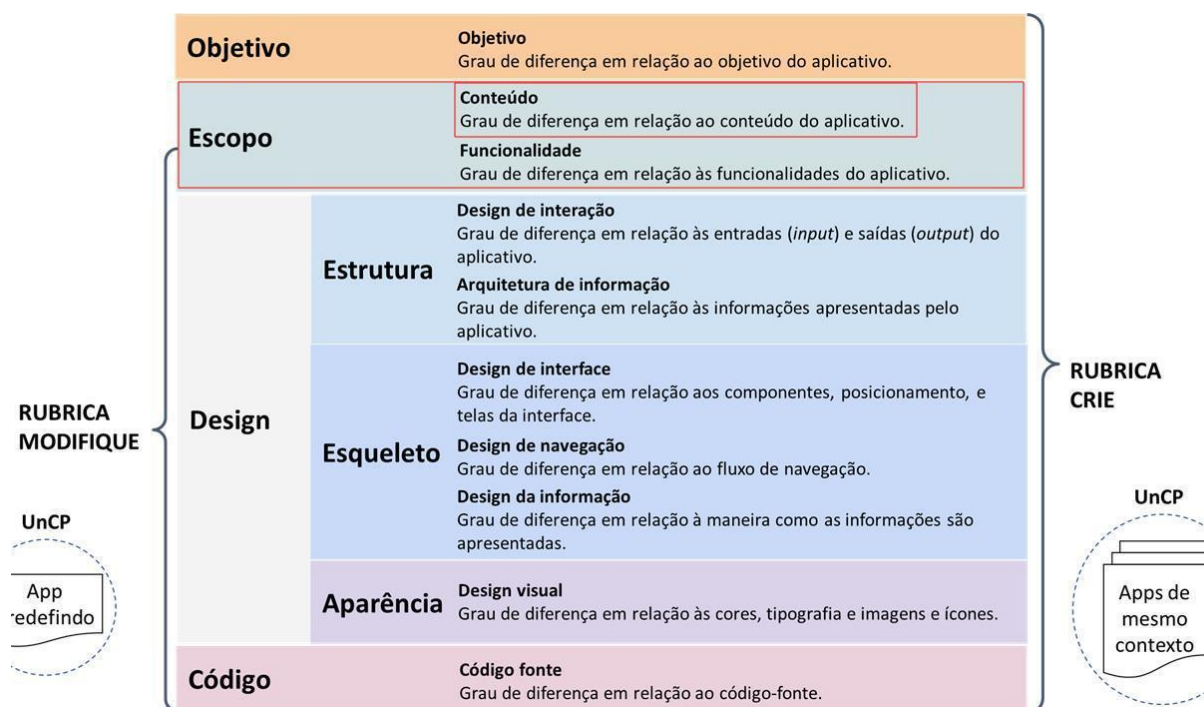


Figura 1. Dimensões da avaliação da originalidade de aplicativos móveis (Alves et al., 2020)

O escopo da originalidade pode ser dividido em conteúdo e funcionalidade. O foco do presente trabalho se dá na originalidade em relação ao conteúdo, como parte do escopo. A originalidade de escopo/conteúdo é definida como o grau de diferença de um tópico de app (receitas ou notícias, por exemplo) em relação aos tópicos de apps apresentados em um universo de referência.

O conteúdo de um aplicativo móvel pode ser identificado por humanos por meio da análise de suas características e funcionalidades, a fim de identificar a categoria e/ou tópico do aplicativo, podendo ser geral ou mais específico. O tópico também é muitas vezes especificado pelos criadores nas lojas de apps, categorizando-os como saúde, educação, entre outros.

O desafio surge quando se tenta realizar a detecção de tais tópicos de forma automatizada. Para este fim, uma das alternativas é aplicar técnicas de Processamento de Linguagem Natural (*NLP - Natural Language Processing*) a partir do texto extraído dos aplicativos.

Nesse contexto, observa-se que muitas vezes são utilizados nomes de objetos específicos e que representam o mesmo tipo de entidade. Caso dentro do texto existam as palavras ‘maçã’, ‘banana’ e ‘melancia’, há uma indicação que o app é relacionado a ‘frutas’, por exemplo. Caso também existam dentro do texto as palavras ‘arroz’ e ‘feijão’, o tópico mais adequado seria ‘alimentos’.

Tipicamente, para o reconhecimento desse tipo de entidades, é adotado o NER (*Named Entity Recognition - Reconhecimento de Entidade Nomeada*). O NER pode ser definido como uma técnica de classificação de texto não-estruturado em entidades nomeadas, sendo utilizado dentro da área de NLP (AMARAL, 2013). Mesmo já existindo pesquisas adotando NER também para textos em português (AMARAL, 2013), ainda há poucas pesquisas voltadas a este tipo de detecção com base em textos curtos extraídos de apps.

Desta forma, o presente trabalho visa contribuir para a criação de um modelo de NER que identifique entidades a partir dos textos extraídos de apps criados com App Inventor, a fim de melhorar uma avaliação de originalidade do escopo/conteúdo de tais apps.

1.2 OBJETIVOS

Objetivo geral

O objetivo geral deste projeto é desenvolver um modelo de Reconhecimento de Entidade Nomeada (NER - *Named Entity Recognition*) para identificação de entidades para auxiliar a modelagem de tópicos de aplicativos móveis criados com App Inventor.

Objetivos Específicos

Objetivo I: Síntese da fundamentação teórica em relação ao conceito de Processamento de Linguagem Natural (*NLP - Natural Language Processing*), NER e apps criados com App Inventor.

Objetivo II: Análise do estado da arte em relação a modelos de NER para a modelagem de tópicos de apps desenvolvidos com App Inventor no contexto da Educação Básica.

Objetivo III: Definir e implementar uma abordagem de NER para identificação de entidades relacionadas à apps de App Inventor com base nos apps compartilhados na comunidade mundial por meio da galeria do App Inventor, além de apps criados no contexto da Computação na Escola/UFSC.

Objetivo IV: Avaliação e testes do desempenho da abordagem NER criada.

1.3. METODOLOGIA

Nessa pesquisa é usada uma abordagem multi-método. A metodologia de pesquisa utilizada neste trabalho é dividida nas seguintes etapas:

Etapa 1 – Fundamentação teórica

Atividade focada em estudar, analisar e sintetizar os conceitos principais e a teoria referente aos temas a serem abordados neste trabalho. Nesta etapa é apresentada a fundamentação teórica utilizando a metodologia de revisão narrativa (CORDEIRO et al., 2007), e são realizadas as seguintes atividades:

A1.1 – Síntese de conceitos sobre NLP e NER;

A1.2 - Análise de características de elementos textuais de apps criados com App Inventor;

Etapa 2 – Estado da arte

Nesta etapa é realizado um mapeamento sistemático da literatura seguindo o processo proposto por Petersen et al. (2015) para identificar e analisar modelos de identificação de entidades. Esta etapa é dividida nas seguintes atividades:

A2.1 – Definição do protocolo de revisão;

A2.2 – Execução da busca e seleção de artigos relevantes;

A2.3 – Extração e análise de informações relevantes;

Etapa 3 – Desenvolvimento

Nesta etapa é desenvolvido um modelo para identificação de entidades de apps criados no App Inventor, adotando métodos de NLP, incluindo principalmente o NER. Esta etapa é dividida nas seguintes atividades, que, com exceção de A3.1, são realizadas por meio de diversas iterações:

A3.1 – Modelar entidades relacionadas às aplicações criadas com App Inventor;

A3.2 - Pesquisar e selecionar abordagens para a identificação de entidades;

A3.3 - Desenvolver um modelo de identificação de entidades a partir de textos extraídos de apps criados com App Inventor;

Etapa 4 – Avaliação do desempenho do modelo criado

Nesta etapa é avaliado o desempenho em relação à qualidade do modelo criado (precisão/*recall* e F1 *score*). Esta etapa é dividida nas seguintes atividades:

A4.1 – Definir o objetivo e métricas de avaliação;

A4.2 – Executar testes;

A4.3 – Analisar os resultados dos testes;

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo analisa os principais conceitos presentes neste trabalho, apresentando o App Inventor e as características textuais de apps criados com ele, bem como conceitos de análise textual automática utilizando NER.

2.1 ELEMENTOS TEXTUAIS DE APPS CRIADOS COM APP INVENTOR

2.1.1 APP INVENTOR

O MIT App Inventor é uma aplicação web *open source* criada pelo Google e atualmente mantida pelo MIT - Massachusetts Institute of Technology (PATTON et al., 2019). Seu intuito é o de fornecer uma maneira simples e prática para o desenvolvimento de aplicações móveis por meio de uma interface de programação baseada em blocos. Esse tipo de interface permite com que o usuário consiga criar aplicativos sem ter conhecimento prévio de nenhuma linguagem de programação textual, podendo assim ser utilizado por crianças ou pessoas de fora da área da computação.

O App Inventor possui recursos para o desenvolvimento dos aplicativos tanto na parte funcional quanto na parte de design de interfaces de usuário.

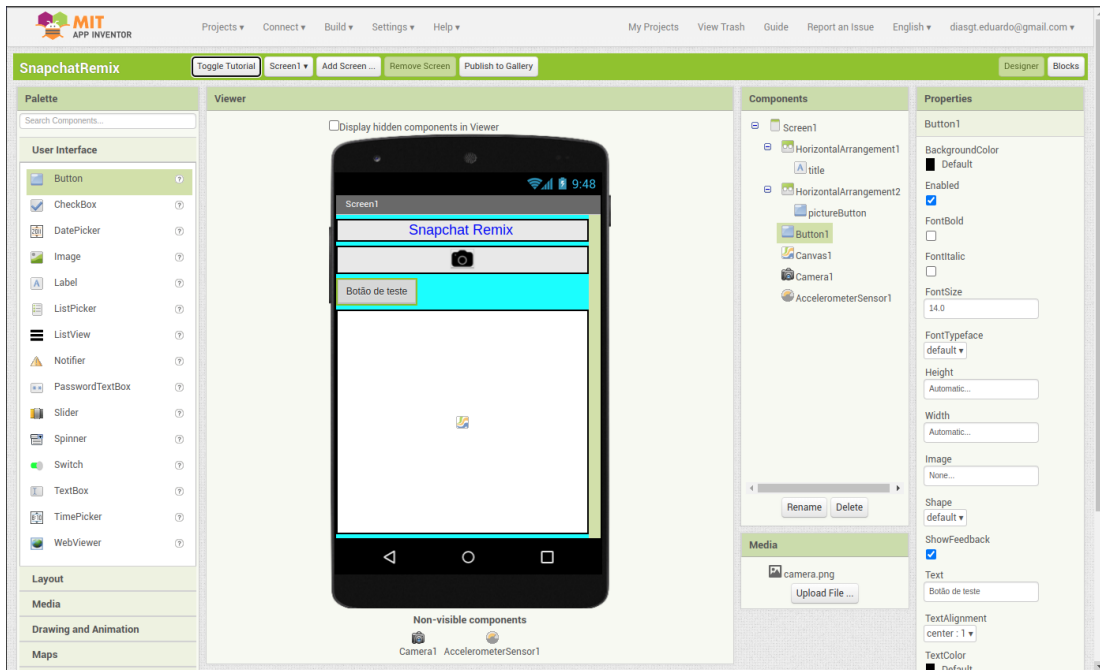


Figura 2. Modo Designer do App Inventor

O modo Designer do App Inventor permite que o usuário realize a criação de interfaces e estruturas visuais de seu aplicativo (Figura 2). Nele, é possível fazer a inserção de elementos visíveis na tela, como botões, textos, imagens, vídeos e mapas. Esse modo também permite incluir elementos invisíveis, como temporizadores e sensores de localização.

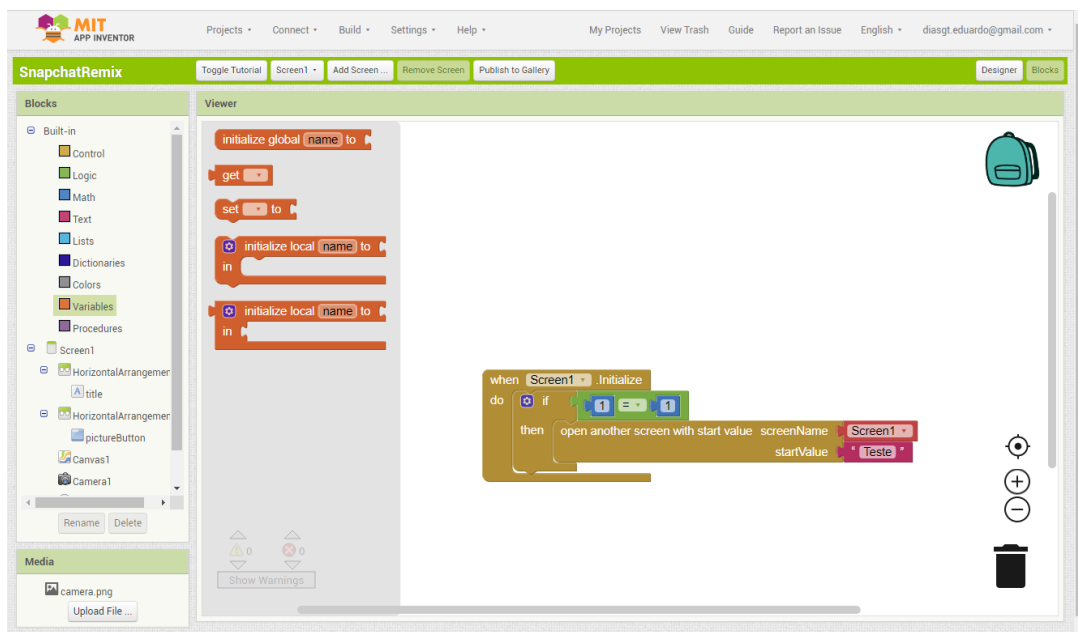


Figura 3. Modo Blocks do App Inventor

O modo funcional é chamado de Blocks pelo App Inventor. Ao acessar este modo, o usuário pode realizar a programação de seu aplicativo arrastando blocos de programação presentes nas abas na parte esquerda da tela e conectando-os uns aos outros, como visto na Figura 3. Cada um dos blocos de programação possui uma funcionalidade diferente, servindo para abstrair os procedimentos realizados por uma linguagem de programação. Exemplos destes procedimentos são estruturas condicionais, atribuições de variáveis, laços de repetição, entre outros.

É possível que o usuário realize o compartilhamento de seu app exportando um arquivo com extensão *.aia* do mesmo. Esse arquivo pode então ser importado para o App Inventor por qualquer usuário que o possua. É possível também exportar um executável *.apk* do app, permitindo sua instalação diretamente em um dispositivo móvel. Além disso, é possível publicar um projeto na galeria do App Inventor, tornando-o disponível para a comunidade.

2.1.2 ELEMENTOS TEXTUAIS DOS APPS

Para identificar os elementos textuais presentes em um app criado com App Inventor e assim tornar possível sua extração, deve-se primeiramente entender a estrutura de um arquivo *.aia*. A Figura 4 mostra a estrutura deste tipo de arquivo, que é em essência a mesma que um arquivo *.zip*, contendo diversos diretórios que por sua vez contém outros diretórios e arquivos dentro deles.

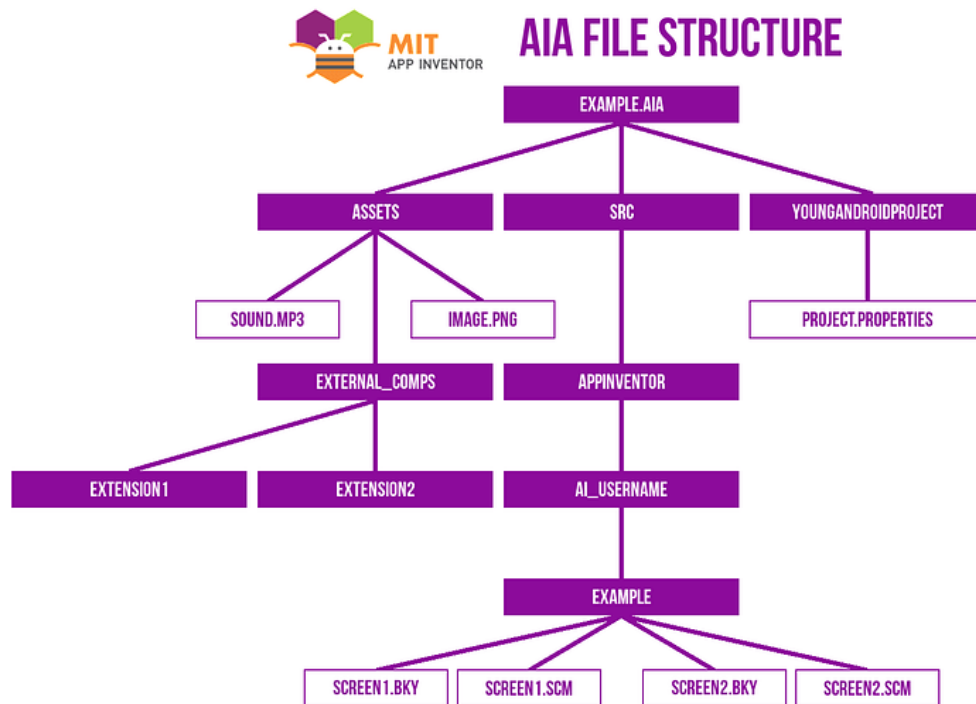


Figura 4. Estrutura de um arquivo .aia (App Inventor Community, 2019).

Dentro do diretório `src/appinventor/[ai_username]/[example]` a existência de uma dupla de arquivos para cada uma das telas do app, um deles com a extensão `.scm` e outro com a extensão `.bky`.

```

"Properties": {
  "$Name": "Screen1",
  "$Type": "Form",
  "$Version": "30",
  "AboutScreen": "Esta \u00e9 a primeira tela depois do login, onde o usu\u00e1rio pode pesquisar os professores",
  "AppName": "UFSC_Review",
  "BackgroundColor": "#FFFFFF",
  "PrimaryColor": "#000000",
  "Title": "Pesquisa_professores",
  "TitleVisible": "False",
  "TutorialURL": "https://www.youtube.com/watch?v=dzYP01CPC6E",
  "Uuid": "0",
  "$Components": [
    {
      "$Name": "Label1",
      "$Type": "Label",
      "$Version": "5",
      "FontBold": "True",
      "FontSize": "20",
      "Width": "-2",
      "Text": "Escolha um professor",
      "TextColor": "#000000",
      "Uuid": "-2072266397"
    },
    {
      "$Name": "ListView1",
      "$Type": "ListView",
      "$Version": "6",
      "BackgroundColor": "#FFFFFF",
      "Height": "-2",
      "ListData": "[]",
      "Uuid": "1985919235"
    }
  ]
}

```

Figura 5. Arquivo .scm com elemento textual destacado

```

<block type="lists_create_with" id="x$]*b?_6w30.WE#715V*">
  <mutation items="4"></mutation>
  <value name="ADD0">
    <block type="text" id="T5v*80w@R|+s]-iac}A7">
      <field name="TEXT">Andre</field>
    </block>
  </value>
  <value name="ADD1">
    <block type="text" id="+JH*Vlm}(GW?q+XZjp7(">
      <field name="TEXT">Bruno</field>
    </block>
  </value>
  <value name="ADD2">
    <block type="text" id="_bahH@cl|(XV*mg4(Wzg">
      <field name="TEXT">Marcelo</field>
    </block>
  </value>
  <value name="ADD3">
    <block type="text" id="7+x_n;(^.a^J@c/4jBnr">
      <field name="TEXT">Victoria</field>
    </block>
  </value>
</block>

```

Figura 6. Arquivo .bky com elementos textuais destacados

O arquivo *.scm* (Figura 5) é um arquivo que contém as informações de layout sobre a tela do app relacionada, escrito no formato de um arquivo JSON. O arquivo *.bky* (Figura 6), por sua vez, possui sua estrutura em formato XML, contendo as informações sobre os blocos lógicos da respectiva tela. Em ambos os arquivos é possível identificar os elementos textuais presentes no app analisando seu conteúdo.

No caso do arquivo *.scm*, é possível observar uma lista de objetos do tipo “\$Components”. Analisando os objetos, percebe-se que o valor presente no campo “Text” do componente representa um elemento textual presente na tela do app referente ao componente identificado, como também pode ser visto na Figura 5. Já no caso do arquivo *.bky*, o elemento textual está presente no valor da tag *field*, já que o atributo *name* contém o valor “Text”, como visto na Figura 6.

Analisando os elementos textuais em apps do App Inventor, identificam-se os campos presentes em cada componente que representam algum elemento textual (Tabela 1). Dessa maneira, pode-se extrair o texto de uma grande gama de apps a partir de seus arquivos *.aia* analisando estes campos, podendo assim ser feita uma análise sobre estes textos e então inferir tópicos de apps a partir deles.

Tabela 1. Propriedades contendo elementos textuais separadas por componente

<i>Tipo</i>	<i>Componente</i>	<i>Propriedade</i>
SCREEN		
		<i>Title</i>
		<i>AboutScreen</i>
DESIGNER		
<i>User Interface</i>	<i>Button</i>	<i>Text</i>
	<i>Checkbox</i>	<i>Text</i>
	<i>DatePicker</i>	<i>Text</i>
	<i>Label</i>	<i>Text</i>
	<i>ListPicker</i>	<i>Text e ElementsFromString</i>
	<i>PasswordTextBox</i>	<i>Text e Hint</i>
	<i>Spinner</i>	<i>ElementsFromString e Prompt</i>
	<i>Switch</i>	<i>Text</i>
	<i>TextBox</i>	<i>Text e Hint</i>
	<i>TimePicker</i>	<i>Text</i>

<i>Media</i>	<i>ImagePicker</i>	<i>Text</i>
<i>Maps</i>	<i>Marker</i>	<i>Description</i>
<i>Social</i>	<i>ContactPicker</i>	<i>Text</i>
	<i>EmailPicker</i>	<i>Text</i>
	<i>PhoneNumberPicker</i>	<i>Text</i>
	<i>Texting</i>	<i>Message</i>
<i>Connectivity</i>	<i>ActivityStarter</i>	<i>ResultName</i>
BLOCKS		
<i>Text</i>	<i>String</i>	<i>Text</i>

Alguns exemplos de textos extraídos desses campos de arquivos *.aia* são apresentados na Tabela 2.

Tabela 2. Texto bruto extraído de arquivos exemplo *.aia*

Nome do App	Texto bruto extraído
MercadoRapido	<p>android.intent.action.VIEW AppMat S_Scan Ir android.intent.action.VIEW https://www.confianca.com.br/ android.intent.action.VIEW http://www.barracaomais.com.br/ android.intent.action.VIEW https://www.dia.com.br/ android.intent.action.VIEW http://www.jauserve.com.br/ AppMat S_mercadonline SUPERMERCADO CONFIANÇA R\$6,29 SUPERMERCADO DIA R\$5,79 SUPERMERCADO BAGARELLI R\$5,99 SUPERMERCADO JAÚ SERVE R\$5,99 SUPERMERCADO SCHIAVON R\$5,69 SUPERMERCADO CONFIANÇA R\$5,39 SUPERMERCADO DIA R\$4,85 SUPERMERCADO BAGARELLI R\$6,09 SUPERMERCADO JAÚ SERVE R\$6,89 SUPERMERCADO SCHIAVON R\$6,59 AppMat Sbebidas Bebidas Coca cola 2L Suco Del Vale (1L) SUPERMERCADO CONFIANÇA R\$14,98 kg (linguiça toscana seara) SUPERMERCADO DIA R\$17,99/600g (linguiça toscana perdigão) SUPERMERCADO BAGARELLI R\$12,90 kg (aurora) SUPERMERCADO JAÚ SERVE R\$12,40 kg (aurora) SUPERMERCADO SCHIAVON R\$13,99 kg (aurora) SUPERMERCADO CONFIANÇA R\$ 7,98 SUPERMERCADO DIA R\$8,79 SUPERMERCADO BAGARELLI R\$10,90 SUPERMERCADO JAÚ SERVE R\$11,50 SUPERMERCADO SCHIAVON R\$6,99 AppMat Scarne Açougue Linguiça (diversa) Peito de frango (bandeja) Smercearia S_Scan Slimpeza Shortifruiti Scarne Sbebidas S_mercadonline Shigiene Spanfleto Mercado Rápido Mercado Rápido Mercado Rápido!! SUPERMERCADO CONFIANÇA R\$ 1,37 SUPERMERCADO DIA R\$ 1,45 SUPERMERCADO BAGARELLI R\$ 1,69 SUPERMERCADO JAÚ SERVE R\$ 1,69 SUPERMERCADO SCHIAVON R\$ 2,30 SUPERMERCADO CONFIANÇA R\$ 6,40 SUPERMERCADO DIA R\$ 6,45 SUPERMERCADO BAGARELLI R\$ 5,99 SUPERMERCADO JAÚ SERVE R\$ 5,59 SUPERMERCADO SCHIAVON R\$ 6,99 AppMat Shigiene Higieine Sabonete nivea Papel higiênico neve (4 rolos) SUPERMERCADO CONFIANÇA R\$ 1,97 SUPERMERCADO DIA R\$ 2,99 SUPERMERCADO BAGARELLI R\$ 3,39 SUPERMERCADO JAÚ SERVE R\$ 1,39 SUPERMERCADO SCHIAVON R\$ 1,39 SUPERMERCADO CONFIANÇA R\$ 0,97 SUPERMERCADO DIA R\$ 1,49 SUPERMERCADO BAGARELLI R\$ 1,97 SUPERMERCADO JAÚ SERVE R\$ 1,39 SUPERMERCADO SCHIAVON R\$ 0,69 AppMat Shortifruiti Hortifruiti Banana Nanica (kg) Laranja Pera (kg) SUPERMERCADO CONFIANÇA R\$ 1,38 SUPERMERCADO DIA R\$ 1,45 SUPERMERCADO BAGARELLI R\$ 1,27 SUPERMERCADO JAÚ SERVE R\$ 1,28 SUPERMERCADO SCHIAVON R\$ 1,69 SUPERMERCADO CONFIANÇA R\$ 6,19 SUPERMERCADO DIA R\$ 6,69 SUPERMERCADO BAGARELLI R\$ 2,95 SUPERMERCADO JAÚ SERVE R\$ 3,99 SUPERMERCADO SCHIAVON R\$ 6,99 AppMat Slimpeza Limpeza Detergente ype Desinfetante veja SUPERMERCADO CONFIANÇA R\$ 6,35 SUPERMERCADO DIA R\$ 11,79 (valor estimado) SUPERMERCADO BAGARELLI R\$ 12,95 SUPERMERCADO JAÚ SERVE R\$ 13,98 SUPERMERCADO SCHIAVON R\$ 14,39 SUPERMERCADO CONFIANÇA R\$ 8,70 (pedretti) SUPERMERCADO DIA R\$ 3,85 (camil) SUPERMERCADO BAGARELLI R\$ 9,98 SUPERMERCADO JAÚ SERVE R\$ 2,92 (denadai) SUPERMERCADO SCHIAVON R\$ 4,99 (pedretti) AppMat Smercearia Mercearia Arroz (Tio João) Feijão (diversos) android.intent.action.VIEW http://issuu.com/confianca/docs/tabloide_090dba0591b11a?e=11976215/53759794 android.intent.action.VIEW https://www.dia.com.br/lojas/sp-pederneiras-jardim-alvorada-avenida-brasil-s-n/ofertas android.intent.action.VIEW http://online.fliphtml5.com/Ingd/ezzx/ android.intent.action.VIEW https://jauserve.websiteseuro.com/?pagina=ofertas AppMat Spanfleto</p>

Tradutor	de es fr nl en it ja la pt ru - Tradutor Tradutor Do: Alemão, Espanhol, Francês, Holandês, Inglês, Italiano, Japonês, Latim, Português, Russo ---> Para: Alemão, Espanhol, Francês, Holandês, Inglês, Italiano, Japonês, Latim, Português, Russo Traduzir... Traduzir Tradução...
FloripaPraias	telainicial Floripa Praias Tela de Login Tela de Login ENTRE COMO CONVIDADO ENTRAR telainicial FloripaPraias Suporte E-MAIL DA EMPRESA: floripapraias.suporte@gmail.com Deixe seu feedback aqui telainicial FloripaPraias mapa Selecionar Praia Ilha do Campeche Morro das Pedras Lagoinha do Leste Matadeiro Lagoa da Conceição Barra da Lagoa Jurere Internacional telainicial descricao descricao None DESCRIÇÃO praias/ descricao None Imagem None Ilha do Campeche IconProprio.png None Morro das Pedras IconProprio.png None Lagoinha do Leste IconProprio.png IconTrilha.png Matadeiro IconProprio.png IconTrilha.png Lagoa da Conceição IconImproprio.png IconTrilha.png Barra da Lagoa IconProprio.png None Jurere Internacional IconProprio.png None IconCarro.png None praias/ Barra da Lagoa descricao A praia fica entre o Campeche e a Armação. Sendo que a faixa de areia fofa se estende até a Joaquina. A praia é longa e larga. As ondas do mar são agitadas, perfeitas para o surfe e um tanto perigosa para os banhistas. A extensão é de 2.450 metros e possui uma largura de 5 a 45 metros. praias/ Lagoinha do Leste descricao Pouco mais de 1,2km de faixa de areia está cercada por costões, matas nativas, mar aberto, lagoa e cachoeira. É um dos últimos redutos de Mata Atlântica ainda preservados em Florianópolis. Para manter esta preservação, a área é protegida por lei e não se pode construir na região. praias/ Matadeiro descricao Matadeiro é uma praia localizada no sul da Ilha de Santa Catarina, em Florianópolis. É voltada para o Oceano Atlântico e separada da praia da Armação por um rio de água salobra. O mar gelado com ondas de medianas a grandes, de boa formação, atraem surfistas. Do final da praia, sai uma das trilhas para a praia da Lagoinha do Leste. praias melhorespraias Selecione uma praia para aparecer a descrição abaixo! LEGENDA melhorespraias mapa Suporte praias telainicial VIP PRAIAS TODAS AS PRAIAS SUPORTE
Calculadora_1	1 2 3 Screen2 calc Calculadora Valor 1 Digite um número Valor 2 Digite um número / * - + RESULTADO : Ir pra tela 2 Screen1 Não existem raízes reais Calc_modelo Screen2 Equação de segundo grau $ax^2 + bx + c = 0$ A= B= C= Raiz1: Raiz2: Delta: Calcular Ir pra tela 1
Livro_de_receita	Screen1 receita2 Livro_de_receita próximo bolinho de bacalhau Hint for TextBox1 ingredientes: 1 cebola picada 1 colher (sopa) de azeite 1/2kg de bacalhau cozido e desfiado 1kg de batata cozida e amassada sal, pimenta do reino e salsinha picada a gosto Hint for TextBox2 preparação: numa frigideira refogue uma cebola picadinha em uma colher (sopa) de azeite modele os bolinhos ,em formato de croquete (9cmde comprimento)e frite em óleo quente. misture em uma tigela 1kg de batata cozida e amassada ,1/2kgde bacalhau cozido e desfiado ,3 gêmeas,sal,pimenta do reino e salsinha picada a gosto e a cebola refogado .. Voltar próximo Livro_de_receita receita pergunta1 receita3 Livro_de_receita receita2 torta salgada Hint for TextBox1 ingredientes: 12 colheres de farinha de trigo 1 copo de óleo 3 ovos inteiro 2copos de leite 1colher de fermento 100g de mussarela ralada recheio a gosto (ex:frango ,sardinha, sal ,etc) modo de preparo: bata todos ingredientes no liquidificador .Depois coloque a metade da massa em uma forma untada e coloque o recheio .depois coloque o resto da massa .leve para assare ate ficar dourado próximo voltar receita2 receita4 Livro_de_receita receita3 panqueca Hint for TextBox1 ingredientes: uma xícara de chá de leite intergral 2 ovos 4 colheres de sopa de óleo 1colher de chá de sal 1 xícara e meia de farinha de trigo Hint for TextBox2 modo de preparo: bata todos os ingredientes líquidos e o sal no liquidificador quando ficar homogêneo, acrescente a farinha aos poucos e bata mais pegue a mistura e despeje em uma frigideira para que fique uma massa fina quando um lado estiver dourada vire e dore o outro.Sirva com qualquer malho voltar próximo receita3 receita5 Livro_de_receita receita4 pão de queijo Hint for TextBox1 ingredientes: 200g de queijo meia cura ralado grosso 200g de polvilho azedo 100 ml de leite 40g de manteiga 1 ovo inteiro sal a gosto Hint for TextBox2 modo preparo: Preeaqueça o forno a 180 graus Aqueça o leite e a manteiga em uma panelina até levantar fervura. jogue a mistura fervente no polvilho e mexa para fazer uma pasta junte o queijo ralado e misture. Coloque o ovo e misture novamente. Ajuste o sal . faça bolinhas com o mão untada com óleo e depois leve ao fogo voltar próximo receita4 Livro_de_receita receita5 COXINHA Hint for TextBox1 INGREDIENTES: 2 COPOS DE ÁGUA 1 copo de leite 1 colher de margarina 1 cubo de caldo de galinha sal a gosto 3 xícaras de trigo Hint for TextBox2 modo de preparo: coloque a água, o leite , a margarina , o caldo e o sal em uma panela e misture até ferver. abaixe o fogo e acrescente o trigo de uma só vez. mexe até obter uma massa lisa e homogenea . recheie de maneira que preferir VOLTAR pergunta1 Livro_de_receita Screen1 livro de receita pratos salgados fácil próximo Livro_de_receita Screen2 Text for Label1 Hint for TextBox1 receita voltar

2.2 ANÁLISE TEXTUAL AUTOMATIZADA COM NER

Processamento de Linguagem Natural (*Natural Language Processing* - NLP), é a área da ciência da computação que une as áreas da inteligência artificial e da linguística (NADKARNI et al., 2011). O NLP visa estudar, compreender e propor técnicas e soluções computacionais para problemas relacionados às linguagens naturais compreendidas pelos seres humanos. Exemplos de aplicações de NLP incluem a tradução automática, assistentes inteligentes, mecanismos de busca, análise textual, entre outros.

O Reconhecimento de Entidade Nomeada (NER) (AMARAL, 2013), é uma técnica amplamente utilizada no NLP cujo objetivo é o reconhecimento e identificação automática de entidades nomeadas presentes em um texto não estruturado. Essas entidades são classificadas dentro de um conjunto de categorias, sendo elas pré-definidas pelo modelo de NER sendo utilizado. Desta maneira, a entrada para um sistema utilizando NER é um texto não estruturado, e sua saída é um conjunto de texto classificado e estruturado entre entidade e categoria (FONSECA et al., 2015).

Usando o texto “João mora em Florianópolis e trabalha no Google” como exemplo, é possível efetuar a extração das entidades nomeadas do mesmo e assim categorizá-las, podendo fazê-lo de maneira genérica ou mais específica. Um possível resultado para esta extração e classificação de categorias do texto de exemplo é apresentado na Tabela 3.

Tabela 3. Exemplo de extração de entidades nomeadas

Entidade	Categoria
João	Pessoa
Florianópolis	Local
Google	Organização

Nota-se que a entidade “Florianópolis” poderia também ser definida como “Cidade” ou “Município”, dependendo do modelo de NER a ser aplicado no texto de entrada.

Algumas das ferramentas mais utilizadas para o uso das técnicas de NER são *frameworks* de NLP acessíveis na linguagem de programação *Python*. Com eles, é possível fornecer um texto de entrada em linha de código e obter a tabela de entidades como saída a partir do modelo de NER utilizado. Dentre os *frameworks* existentes, pode-se destacar os *frameworks* Spacy (EXPLOSION, 2015), NLTK (NLTK PROJECT, 2001) e Flair (FLAIRNLP, 2018).

```
1 import spacy
2
3 nlp = spacy.load("pt_core_news_sm")
4
5 doc = nlp("João mora em Florianópolis e trabalha no Google")
6
7 for X in doc.ents:
8     print("Entidade:", X.text, "| Categoria:", X.label_)
9
```

```
↳ python3 exemploSpacy.py
Entidade: João | Categoria: PER
Entidade: Florianópolis | Categoria: LOC
Entidade: Google | Categoria: ORG
```

Figura 7. Exemplo de código e execução do Spacy

A Figura 7 mostra um exemplo de um código em *Python* utilizando o *framework* Spacy. Nele, é possível observar, na linha 3, que está sendo carregado o modelo “pt_core_news_sm”, um modelo pré-treinado do Spacy que reproduz funcionalidades referentes à língua portuguesa, incluindo NER, com algumas entidades já definidas. Na linha 5, é fornecido o texto de entrada para o modelo, e em seguida é feito um *loop* nas entidades identificadas a fim de realizar a impressão no console das mesmas, assim como em suas categorias. A parte de baixo da Figura 7 mostra o resultado final da aplicação do NER no texto de entrada “João mora em Florianópolis e trabalha no Google”, onde foram identificadas as entidades João - PER (Pessoa), Florianópolis - LOC (Localização) e Google (ORG - Organização).

As Figuras 8 e 9 mostram exemplo de código dos *frameworks* NLTK e Flair, respectivamente. Esses *frameworks*, ainda que possuam pacotes de NLP em português, não possuem a funcionalidade de NER neste idioma. Sendo assim, a frase de exemplo foi traduzida para o inglês, com a cidade de Florianópolis sendo trocada para Los Angeles pelo contexto do idioma, e foram utilizados modelos em inglês. É possível observar no terminal de execução que as mesmas entidades foram identificadas para a frase traduzida. No caso do Flair (Figura 9), é possível observar um número ao lado da identificação da entidade. Esse número representa o grau de confiança da ferramenta na classificação da entidade, indo de zero a um.

```
1 import nltk
2 from nltk.corpus import floresta
3
4 nltk.download('punkt')
5 nltk.download('averaged_perceptron_tagger')
6 nltk.download('maxent_ne_chunker')
7 nltk.download('words')
8
9 text = "John lives in Los Angeles and works at Google"
10
11 for sent in nltk.sent_tokenize(text):
12     for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(sent))):
13         if hasattr(chunk, 'label'):
14             print(chunk.label(), ' '.join(c[0] for c in chunk))

```

```
↳ python3 exemploNLTK.py
PERSON John
GPE Los Angeles
ORGANIZATION Google
```

Figura 8. Exemplo de código e execução do NLTK

```

1  from flair.data import Sentence
2  from flair.models import SequenceTagger
3  from segtok.segmenter import split_single
4
5  tagger = SequenceTagger.load('ner-ontonotes')
6  text = "John lives in Los Angeles and works at Google"
7
8  sentence = [Sentence(sent, use_tokenizer=True) for sent in split_single(text)]
9  tagger.predict(sentence)
10
11 for sent in sentence:
12     for entity in sent.get_spans('ner'):
13         print(entity)

```

↳ python3 exemploFlair.py

```

Span[0:1]: "John" → PERSON (0.9554)
Span[3:5]: "Los Angeles" → GPE (0.9995)
Span[8:9]: "Google" → ORG (0.9357)

```

Figura 9. Exemplo de código e execução do Flair

As categorias atualmente reconhecidas pelos modelos base de NER dos frameworks analisados são comparados na Tabela 4, tanto em português quanto em inglês. Observa-se que esses modelos base possuem uma quantidade baixa de categorias, especialmente o modelo em português do Spacy, que é o que mais se encaixa no âmbito deste trabalho.

Tabela 4. Categorias reconhecidas pelos frameworks

	Spacy	NLTK	Flair
Português	LOC, MISC, ORG, PER	-	-
Inglês	CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART	ORGANIZATION, PERSON, LOCATION, DATE, TIME, MONEY, PERCENT, FACILITY, GPE	CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART

Visto a pouca quantidade de categorias, é necessário treinar estes modelos base para que eles comportem também as categorias referentes aos tópicos de aplicativos móveis definidos.

Para treinar os modelos do *Spacy*, deve-se primeiramente utilizar a função `get_pipe('ner')` do modelo carregado, a fim de recuperar a pipeline de NER do modelo. Em seguida, pode-se utilizar a função `add_label()` a partir deste elemento, passando como parâmetro uma nova categoria não presente no modelo, como IDIOMA, por exemplo, visando incluí-la e assim possibilitando identificação de entidades da nova categoria. Por fim, deve-se utilizar a função `update()` para treinar o modelo, fornecendo como dado de entrada da função uma lista de tuplas, como visto na Figura 10.

```
[  
  ("maçã, banana, melancia",{"entities":[(0,3,"FRUTA"),(6,11,"FRUTA"),(14,21,"FRUTA")]})  
  ("português, inglês, espanhol",{"entities":[(0,8,"IDIOMA"),(11,16,"IDIOMA"),(19,26,"IDIOMA")]})  
]
```

Figura 10. Formato de dados para treinamento dos modelos do Spacy

O primeiro parâmetro da tupla é uma frase contendo exemplos de instâncias de entidades que o usuário pretende reconhecer. O segundo parâmetro é um objeto contendo o campo “entities”, cujo valor é uma nova lista de tuplas. Cada valor desta nova lista de tuplas identifica a posição inicial, posição final e por fim a categoria a qual pertence tal entidade identificada. No segundo exemplo da Figura 10, a frase de entrada é “português, inglês, espanhol”. A palavra “português” é em seguida identificada pelo trecho (0,8,“IDIOMA”), sendo 0 e 8 as posições iniciais e finais da palavra na frase, e “IDIOMA” a categoria identificada.

3. ESTADO DA ARTE

3.1 DEFINIÇÃO DO PROTOCOLO DE REVISÃO

O objetivo desta revisão do estado da arte é responder a seguinte pergunta de pesquisa: Quais abordagens existem para automaticamente analisar o texto em português de aplicativos (criados com App Inventor) usando NER?

Esta pergunta de pesquisa é refinada nas seguintes questões de análise:

AQ1. Quais abordagens existem e quais suas características?

AQ2. Quais técnicas/bibliotecas são adotadas na análise?

AQ3. Qual a qualidade (precisão) das abordagens?

Critérios de inclusão/exclusão: Em conformidade com o foco da pesquisa, são definidos os seguintes critérios de inclusão e exclusão de artigos:

- São considerados somente artigos que apresentam abordagens para a detecção automatizada de tópicos utilizando NER com base no código de aplicativos para dispositivos móveis;
- São excluídos artigos que não sejam baseados em código, como descrição do aplicativo, resenhas, dados de uso, etc, pelo fato que no contexto do App inventor não existem essas informações;
- São considerados apenas pesquisas publicadas nos últimos 10 anos desde 2012 levando em consideração o avanço recente especificamente em relação a aplicativos móveis;
- São considerados apenas artigos que apresentam informações substanciais, permitindo a extração de informações relevantes sobre as questões de análise. Portanto, artigos com resumo ou apenas uma página são excluídos;
- São considerados somente artigos escritos em português ou inglês.

Fontes: Foram pesquisados nos principais bancos de dados e bibliotecas digitais no

campo da computação, incluindo as Bibliotecas Digitais ACM, a IEEE Xplore e Scopus com acesso por meio do Portal Capes. A pesquisa também foi feita dentro do site Google Scholar, para complementar a busca.

Critérios de qualidade: São considerados apenas artigos que apresentem informações substanciais para se extrair informações referente às perguntas de análise.

Termos de busca: Com base na questão de pesquisa, várias pesquisas informais foram realizadas para calibrar o *string* de busca, identificando termos de pesquisa relevantes e seus sinônimos (Tabela 5). Foram utilizados sinônimos para minimizar o risco de omitir trabalhos relevantes.

Tabela 5. Termos de busca

Termo	Sinônimos	Tradução (inglês)
NER	Reconhecimento de entidade nomeada, Reconhecimento de entidades nomeadas	Named entity recognition, NER
Aplicativo móvel	App, iOS, Android, App Inventor, aplicativo móvel	Mobile application, App, iOS, Android, App Inventor

String de busca: Após a definição dos termos e seus sinônimos, definiu-se o *string* de busca padrão a ser aplicado nas bases de dados:

("NER" OR "Named entity recognition" OR "Reconhecimento de entidade nomeada" OR "Reconhecimento de entidades nomeadas") AND ("mobile application" OR "app" OR "ios" OR "android" OR "app inventor" OR "Aplicativo móvel")

Os *strings* de busca conforme a formatação de cada repositório estão apresentados na Tabela 6. As buscas na ACM Digital Library, IEEE e Scopus foram realizadas em relação ao título e *abstract* dos artigos. Para o Google Scholar, foi definido um *string* em especial para atender as limitações de opções de busca, realizando a busca também no texto na íntegra.

Tabela 6. Strings de busca por repositório.

Repositório	Search string
ACM Digital Library	ContentGroupTitle:(("NER" OR "Named entity recognition" OR "Reconhecimento de entidade nomeada" OR "Reconhecimento de entidades nomeadas") AND ("mobile application" OR "app" OR "ios" OR "android" OR "app inventor" OR "Aplicativo móvel")) OR Abstract:(("NER" OR "Named entity recognition" OR "Reconhecimento de entidade nomeada" OR "Reconhecimento de entidades nomeadas") AND ("mobile application" OR "app" OR "ios" OR "android" OR "app inventor" OR "Aplicativo móvel")) Filter: 2012-2022
IEEE Xplore Digital Library	((("Publication Title": "NER" OR "Publication Title": "Named Entity Recognition" OR "Publication Title": "Reconhecimento de entidade nomeada" OR "Publication Title": "Reconhecimento de entidades nomeadas") AND ("Publication Title": "mobile application" OR "Publication Title": "app" OR "Publication Title": "ios" OR "Publication Title": "android" OR "Publication Title": "app inventor" OR "Publication Title": "aplicativo móvel")) OR (("Abstract": "NER" OR "Abstract": "Named Entity Recognition" OR "Abstract": "Reconhecimento de entidade nomeada" OR "Abstract": "Reconhecimento de entidades nomeadas") AND ("Abstract": "mobile application" OR "Abstract": "app" OR "Abstract": "ios" OR "Abstract": "android" OR "Abstract": "app inventor" OR "Abstract": "aplicativo móvel")))) Filter: 2012-2022
Scopus	TITLE-ABS-KEY (("NER" OR "Named entity recognition" OR "Reconhecimento de entidade nomeada" OR "Reconhecimento de entidades nomeadas") AND ("mobile application" OR "app" OR "ios" OR "android" OR "app inventor" OR "Aplicativo móvel")) AND PUBYEAR > 2011 AND PUBYEAR > 2011
Google Scholar	"mobile app" "named entity recognition" classification

3.2 EXECUÇÃO DA BUSCA

A busca foi realizada em Maio de 2022 pelo autor do trabalho e revisada pelas (co-)orientadoras. A busca inicial resultou em 506 artigos (Tabela 7).

Tabela 7. Número de artigos identificados por repositório e por fase de seleção.

Fonte	Nº de resultados da busca	Nº de resultados analisados	Nº de documentos potencialmente relevantes	Nº de documentos relevantes
ACM	0	0	0	0
IEEE	4	4	2	0
SCOPUS	36	36	5	0
Google Scholar	466	200	9	0
Total				0

Uma vez executadas as buscas, foram selecionados artigos potencialmente relevantes por meio da análise do título e do resumo dos trabalhos, a fim de confirmar a relevância dos mesmos levando em consideração os critérios de

inclusão e exclusão definidos previamente. Esta etapa foi executada com todos os trabalhos encontrados, exceto os resultados do *Google Scholar*, dos quais foram analisados os 200 resultados mais relevantes.

Em seguida, foram analisados os textos na íntegra dos trabalhos potencialmente relevantes. Mesmo tendo encontrado artigos relevantes referentes à classificação de tópicos de aplicativos, como por exemplo *Classifying Mobile Applications Using Word Embeddings* (EBRAHIMI et al., 2022) e *Enriching iTunes App Store Categories via Topic-Modeling* (VAKULENKO et al., 2014), estes artigos não foram considerados relevantes no contexto deste trabalho, tanto por não fazerem uso do NER quanto por não se basearem no código dos apps, mas em suas descrições nas lojas de aplicativos.

Outros trabalhos, como *Named-Entity Recognition Using Automatic Construction of Training Data From Social Media Messaging Apps* (LEE et al., 2020) fazem o uso do NER para identificação de tópicos relevantes, mas não classificam tópicos de apps em si. Esses trabalhos utilizam informações inerentes aos apps (mensagens trocadas em apps de mensagens, resenhas de usuários na loja, entre outros). Portanto, estes trabalhos também não foram considerados como relevantes no escopo dessa revisão.

Há, ainda, trabalhos como *Code and Named Entity Recognition in StackOverflow* (TABASSUM et al., 2020), que utilizam o NER para classificação de tópicos baseados em código, porém os trechos de código analisados são provenientes de comentários no fórum online *StackOverflow*, e são utilizados para a classificação de tópicos mais técnicos referentes ao código em si, sendo portanto também excluídos da análise.

Desta maneira, após análise de todos os trabalhos potencialmente relevantes, não foi identificado nenhum trabalho que atendesse a todos os critérios de inclusão ou exclusão definidos previamente. Assim, como resultado desse mapeamento sistemático, não foi encontrado nenhum trabalho existente atualmente voltado à

análise automática do texto de aplicativos usando NER, tampouco abordagens voltadas a apps criados com App Inventor.

4. ABORDAGEM DE NER PARA CLASSIFICAÇÃO DE TÓPICOS DE APPS CRIADOS COM APP INVENTOR

4.1 ANÁLISE DE REQUISITOS

O objetivo do presente trabalho é contribuir para a classificação de tópicos de apps a partir da análise de texto extraído dos aplicativos, focando na classificação de entidades utilizando NER.

A análise dos requisitos do modelo de NER a ser construído começa pela identificação de entidades relevantes. Analisando o nome do app e seu conteúdo textual extraído de um universo de 2.806 aplicativos criados com App Inventor, foram identificadas entidades relevantes a serem extraídas usando NER (Tabela 8).

Tabela 8. Identificação de entidades relevantes no contexto de apps criados com App Inventor.

Tópico	Entidade	Exemplos de palavras
Comida	FOOD	leite massas óleo café miúdos manga enlatados sal cogumelo rosca vinagre açúcar maçã limão couve aveia alface pão aspargos pimenta azeite carne morango mostarda ovos manteiga ave banana alho carnes arroz massa biscoito pastel verduras detergente embutidos batata linguiça abacaxi sardinha frutas legumes tapioca bacalhau milho pescados maionese trigo tomate macarrão ervilha paio queijo feijão pêsego beterraba tangerina aves molho água soja atum farinha pimentão uva salsicha chocolate melão bolo brócolis cebola espinafre cenoura
Animal	ANIMAL	urso cachorro dog elefante animais hipopótamo cavalo galinha leopardo zebra vaca animais gato leão hipopótamo peru
Localização	LOC	belo horizonte salvador fortaleza porto velho mapa vitória palmas brasil equador argentina peru
Nome de pessoa	PER	joão maria pedro ana
Disciplina escolar/acadêmica	COURSE	matérias matemática física programação ciências inglês geografia arquitetura história artes
Planta	PLANT	arruda planta alecrim louro árvore palmeira flor salsa plantas violeta cactus lírios

Idioma	LANG	inglês alemão português espanhol francês italiano
Número	NUM	1 2 3 4 5 6 7 8 9 0
Operação Matemática	OP	+ - / * ^ =
Organização/ empresa	ORG	google facebook apple microsoft amazon

Dessa maneira, o requisito referente ao presente trabalho é que o modelo de NER a ser desenvolvido deve ser capaz de identificar as entidades especificadas na Tabela 8 a partir do texto extraído de arquivos .aia de apps criados com App Inventor.

4.2 PREPARAÇÃO DO CONJUNTO DE DADOS

A fim de preparar um conjunto de dados a ser utilizado no treinamento do modelo, foi necessário coletar uma quantidade de conteúdo textual para ser anotado com entidades. Assim, foram selecionados 100 aplicativos criados com App Inventor, e os elementos textuais presentes nesses aplicativos foram recuperados e anotados com entidades. Os elementos textuais foram obtidos realizando uma extração diretamente do código dos aplicativos a partir dos seus arquivos .aia, conforme apresentado na seção 2.1.2, em que os componentes levados em consideração para a extração foram aqueles apresentados na Tabela 1, sendo esses os componentes que de fato apresentam conteúdo textual em seu código. Assim, o conteúdo desses componentes foi extraído em sua totalidade, ou seja, sem passar por um pré-processamento de filtragem de termos repetitivos ou irrelevantes.

Os 100 apps utilizados no processo foram selecionados pelo autor a partir de uma planilha contendo um universo de 2.806 apps criados com App Inventor. Seus códigos foram obtidos tanto da galeria de aplicativos, disponível diretamente no site do App Inventor (<https://gallery.appinventor.mit.edu/featured/>), quanto da iniciativa Computação na Escola, em que os apps foram desenvolvidos por alunos do ensino básico e seus códigos foram recuperados por educadores da iniciativa. O critério de seleção dos apps utilizados foi a variedade e quantidade de entidades definidas na

Tabela 8 visualmente presentes nos elementos textuais dos mesmos, a fim de realizar um balanceamento coerente de entidades presentes no conjunto de dados e assim obter melhores resultados no treinamento do modelo.

Além disso, em adição aos textos extraídos de apps, foram adicionados dois textos fabricados pelo autor a fim de treinar as entidades “Plant” e “Lang”. Isso foi feito com o intuito de fornecer ao modelo exemplos mais variados de tais entidades, no caso uma quantidade maior de idiomas e tipos de plantas, pois dentro do texto dos apps essa variedade não estava presente em tão grande número.

A etapa seguinte da preparação dos dados foi a anotação das entidades dos textos extraídos. Para isso, utilizou-se a ferramenta *NER Annotator* (TECOHOLIC, 2022). Dentre as diversas ferramentas gratuitas testadas para esta etapa, ela mostrou uma boa eficiência nos quesitos praticidade e usabilidade. Essa ferramenta recebe um arquivo *.txt* como dado de entrada e apresenta uma interface gráfica para que o usuário selecione cada termo presente no texto e o classifique como uma entidade (Figura 11). Dessa maneira, foram separados dois arquivos *.txt* para serem anotados, sendo um para o conjunto de dados de treinamento e outro para os dados de validação.

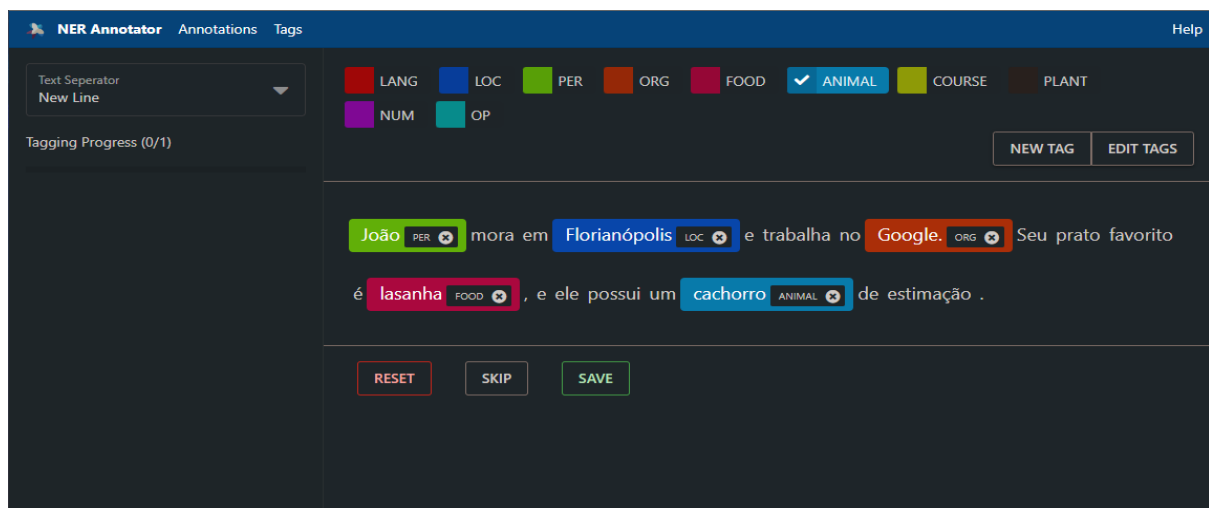


Figura 11. Exemplo de anotações com o NER Annotator

É importante ressaltar que as anotações dos termos com entidades foram feitas integralmente pelo autor do trabalho, estando portanto limitadas ao viés do mesmo

sobre a classificação de tais termos. Por consequência, uma vez que o modelo foi treinado com um conjunto de dados contendo esse viés, é esperado que o modelo também o contenha quando executado para automaticamente reconhecer entidades em textos.

Ao final da anotação das entidades dos arquivos, o texto anotado de cada um dos arquivos foi exportado para um arquivo `.json` (Figura 12). Posteriormente, ambos os arquivos `.json` foram fornecidos à biblioteca de treinamento dos modelos de NER do Spacy.

```
1  {
2    "classes": [
3      "LANG",
4      "LOC",
5      "PER",
6      "ORG",
7      "FOOD",
8      "ANIMAL",
9      "COURSE",
10     "PLANT",
11     "NUM",
12     "OP"
13   ],
14   "annotations": [
15     [
16       "João mora em Florianópolis e trabalha no Google. Seu prato favorito é lasanha, e ele possui um cachorro de estimação.",
17       {
18         "entities": [
19           [0, 4, "PER"],
20           [13, 26, "LOC"],
21           [41, 48, "ORG"],
22           [70, 77, "FOOD"],
23           [95, 103, "ANIMAL"]
24         ]
25       }
26     ]
27   ]
28 }
```

Figura 12. Exemplo de arquivo `.json` gerado com o NER Annotator

Deste conjunto de dados, 80 aplicativos foram utilizados no conjunto de treinamento e 20 no conjunto de validação.

4.3 TREINAMENTO

Para a execução do treinamento, foi utilizado o ambiente baseado em nuvem *Google Colab*, que permite a execução de *jupyter notebooks* com o código de treinamento de forma gratuita.

O pacote escolhido para executar o treinamento foi o Spacy, que fornece a possibilidade de treinar entidades customizadas. Durante os testes realizados, observou-se que o modelo padrão do Spacy em português “pt_core_news_sm” sofria um processo de “desaprendizagem” de suas entidades base (LOC, PER e ORG) caso os dados de entrada fornecidos para o treinamento do modelo não possuíssem tais entidades em suas anotações. Desse modo, optou-se por realizar o treinamento do modelo contendo anotações de todas as entidades especificadas na Tabela 8 no conjunto de dados preparado.

Com os conjuntos de dados já construído nos arquivos *.json*, como descrito na seção 4.2, foi necessário realizar a conversão dos arquivos para a extensão *.spacy*, formato reconhecido para treinamento pelo pacote Spacy. Para isso, foi construída a função *convert* (Figura 13), que foi chamada para ambos os arquivos (Figura 14), gerando os arquivos *train.spacy* e *valid.spacy*.

```
from tqdm import tqdm

def convert(nlp, TRAIN_DATA, output_path: Path):
    for text, annot in tqdm(TRAIN_DATA):
        doc = nlp.make_doc(text)
        ents = []
        for start, end, label in annot["entities"]:
            span = doc.char_span(start, end, label=label, alignment_mode="contract")
            if span is None:
                msg = f"Skipping entity [{start}, {end}, {label}] in the following" \
                    f" text because the character span '{doc.text[start:end]}' " \
                    f"does not align with token boundaries:\n\n{repr(text)}\n"
                warnings.warn(msg)
            else:
                ents.append(span)
        doc.ents = ents
        db.add(doc)
    db.to_disk(output_path)
```

Figura 13. Função de conversão de arquivo *.json* para *.spacy*

```

import json
from spacy.tokens import DocBin

db = DocBin()

t_data = open('/content/drive/MyDrive/TCC/train_data.json')
TRAIN_DATA = json.load(t_data)

v_data = open('/content/drive/MyDrive/TCC/valid_data.json')
VALID_DATA = json.load(v_data)

[ ] convert(nlp, TRAIN_DATA['annotations'], "/content/train.spacy")
convert(nlp, VALID_DATA['annotations'], "/content/valid.spacy")

```

Figura 14. Leitura e conversão dos arquivos de treinamento e validação

Em seguida, foi construído um arquivo de configuração *config.cfg*, que define os parâmetros utilizados para o treinamento do modelo. Os parâmetros utilizados, que foram os valores padrão já fornecidos pelo pacote Spacy, podem ser observados na Tabela 9, assim como detalhes de execução e avaliação do modelo. Em código, o comando de execução do treinamento pode ser observado na Figura 15, bem como detalhes de cada iteração.

```

!python -m spacy train /content/config.cfg --output /content/models/output --paths.train /content/train.spacy --paths.dev /content/valid.spacy
2022-10-25 21:04:12.959342: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
✓ Created output directory: /content/models/output
! Saving to output directory: /content/models/output
! Using CPU

===== Initializing pipeline =====
[2022-10-25 21:04:14,265] [INFO] Set up nlp object from config
INFO:spacy:Set up nlp object from config
[2022-10-25 21:04:14,280] [INFO] Pipeline: ['tok2vec', 'ner']
INFO:spacy:Pipeline: ['tok2vec', 'ner']
[2022-10-25 21:04:14,280] [INFO] Resuming training for: ['ner', 'tok2vec']
INFO:spacy:Resuming training for: ['ner', 'tok2vec']
[2022-10-25 21:04:14,292] [INFO] Created vocabulary
INFO:spacy:Created vocabulary
[2022-10-25 21:04:14,293] [INFO] Finished initializing nlp object
INFO:spacy:Finished initializing nlp object
[2022-10-25 21:04:14,293] [INFO] Initialized pipeline components: []
INFO:spacy:Initialized pipeline components: []
✓ Initialized pipeline

===== Training pipeline =====
! Pipeline: ['tok2vec', 'ner']
! Initial learn rate: 0.001
E # LOSS TOK2VEC LOSS NER ENTS_F ENTS_P ENTS_R SCORE
-- --
0 0 0.00 88.22 9.88 10.65 9.22 0.10
2 200 0.00 13430.76 52.07 60.84 45.50 0.52
4 400 0.00 7518.37 62.99 69.15 57.83 0.63
7 600 0.00 6542.38 70.98 80.31 63.59 0.71
10 800 0.00 4176.18 79.46 81.62 77.41 0.79
13 1000 0.00 2866.07 81.37 87.41 76.11 0.81
16 1200 0.00 2299.16 85.95 91.96 80.67 0.86
20 1400 0.00 1744.57 86.76 91.07 82.84 0.87
24 1600 0.00 1179.10 87.21 91.33 83.44 0.87
27 1800 0.00 1108.33 87.56 90.96 84.41 0.88
32 2000 0.00 744.87 87.09 91.08 83.44 0.87
36 2200 0.00 816.22 89.09 93.06 85.45 0.89
41 2400 0.00 852.28 87.94 90.91 85.16 0.88
45 2600 0.00 624.74 87.69 90.09 85.40 0.88
50 2800 0.00 513.14 88.15 91.73 84.83 0.88
56 3000 0.00 557.32 87.89 90.85 85.11 0.88
61 3200 0.00 552.05 88.77 93.64 84.38 0.89
67 3400 0.00 570.89 88.86 93.03 85.06 0.89
73 3600 0.00 497.40 88.15 91.14 85.35 0.88
80 3800 0.00 537.49 88.48 91.69 85.48 0.88
✓ Saved pipeline to output directory
/content/models/output/model-last

```

Figura 15. Execução do treinamento do modelo

Tabela 9. Dados gerais do modelo de NER

Modelo de ML	
Nome do modelo	Classificação de entidades presentes no texto de apps
Data	23/10/2022
Versão	0.1
Objetivo do modelo de ML	
Tarefa	Classificar entidades a partir de uma entrada textual
Contexto de uso	O modelo é utilizado no contexto de ensino na Educação Básica para realizar a classificação de entidades presentes no texto de apps criados com App Inventor, a fim de auxiliar na avaliação de criatividade destes apps.
Público alvo	Estudantes do ensino fundamental e médio (13+ anos)
Riscos	O risco de não classificar corretamente uma ou mais entidades dentro do texto de um app é um possível equívoco na avaliação da criatividade do app.
Tipo da tarefa	Single-label classificação de entidades
Entidades	Food: Tipos de comidas Animal: Tipos de animais Plant: Tipos de plantas Loc: Endereços e localizações Per: Nomes de pessoas Org: Nomes de empresas e organizações Op: Operações matemáticas Num: Números Lang: Idiomas Course: Disciplinas escolares e acadêmicas
Conjunto de dados	
Descrição dos dados	Conteúdo textual presente em apps criados com App Inventor, contendo todos os tipos das entidades previamente mapeadas, prontas para classificação.
Origem dos dados	Texto extraído do código (arquivo <i>.aia</i>) de aplicativos a partir dos elementos da tela e blocos lógicos, conforme especificado na seção 2.
Quantidade total de dados	Textos extraídos de 100 apps
Distribuição dos dados por entidade	Food: 516 Animal: 301 Plant: 162

	Loc: 776 Per: 442 Org: 598 Op: 110 Num: 671 Lang: 323 Course: 118
Rotulação	Estudante do curso de graduação em Ciências da Computação
Tipos de aumento de dados aplicados	Inclusão de elementos textuais das entidades sub-representadas Lang e Plant
Tamanho do batch	1.000
Separação do conjunto de dados	80% para treinamento (80) 20% para validação (20)
Treinamento - Transfer learning	
Tipo de modelo	spacy.Corpus.v1
Pipeline	tok2vec, ner
Modelo fonte	pt_core_news_sm
Quantidade de épocas	80
Taxa de aprendizagem	1e-3
Loss/taxa de erro por época	Melhor modelo encontrado na epoch 36
Avaliação - Transfer learning (resultados da validação)	
Precisão	0.9305668565
Recall	0.8545454545
F1 score	0.8909374189
Valores por entidade	Ver Tabela 11
Limitações e considerações éticas	

Limitações	O modelo é limitado a reconhecer apenas as entidades propostas neste trabalho
Implantação	
Formato da exportação do modelo	Diretório: ner —cfg —model —moves tok2vec —cfg —model vocab —key2row —lookups.bin —strings.json —vectors —vectors.cfg config.cfg meta.json tokenizer
Referências	
Conjunto de dados	https://codigos.ufsc.br/gqs/app-inventor-ner/-/tree/main/dataset
Modelo	https://codigos.ufsc.br/gqs/app-inventor-ner/-/tree/main/models/output/model-best

Para avaliar o desempenho do modelo, tipicamente analisa-se o resultados das métricas de classificação de *machine learning* chamadas de precisão, *recall* e *F1 score* (BROWNLEE, 2020). No caso do presente modelo, essas métricas são calculadas automaticamente pelo pacote Spacy. Para isso, toma-se como base os textos do conjunto de dados de validação e o modelo realiza a classificação automática desses textos. Em seguida, compara-se a classificação de entidades

obtidas pelo modelo com a classificação feita previamente pelo usuário, categorizando cada palavra dentro do seguinte conjunto:

- TP (True Positive) - Entidade corretamente classificada.
- TN (True Negative) - Entidade corretamente não classificada.
- FP (False Positive) - Entidade falsamente classificada.
- FN (False Negative) - Entidade falsamente não classificada.

A partir dessas classificações dos resultados as medidas de desempenho são calculadas:

- $\text{Precisão} = \text{TP} / (\text{TP} + \text{FP})$ - Indica o quanto o modelo é capaz de classificar entidades com o mínimo possível de falsos positivos.
- $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$ - Indica o quanto o modelo é capaz de classificar entidades com o mínimo possível de falsos negativos.
- $\text{F1 Score} = 2 * \text{Precisão} * \text{Recall} / (\text{Precisão} + \text{Recall})$ - Média harmônica entre Precisão e Recall.

O resultado dessas medidas de desempenho é apresentado na Tabela 9, no tópico *Avaliação - Transfer learning (resultados da validação)*, e na Tabela 11, onde os resultados são apresentados para cada uma das entidades separadamente.

A fim de efetivamente determinar o grau de desempenho do modelo, foram definidas interpretações a partir de uma escala ordenada de 4 níveis (ALLWRIGHT, 2022), definidas de acordo com o resultado do *F1 score* obtido pelo modelo. Nessa escala, o maior nível possível a ser alcançado por um modelo é interpretado como “Muito bom” e ocorre se o modelo obtiver um *F1 score* maior que 0.9. Da mesma maneira, o menor nível possível é interpretado como “Ruim” e ocorre se o modelo obtiver um *F1 score* menor que 0.5. Essa escala de avaliação de modelos é apresentada na Tabela 10.

*Tabela 10. Escala de avaliação de modelos de acordo com o F1 Score
(ALLWRIGHT, 2022)*

F1 score	Interpretação
----------	---------------

> 0.9	Muito bom
0.8 - 0.9	Bom
0.5 - 0.8	OK
< 0.5	Ruim

Assim, a interpretação foi calculada tanto para o modelo como um todo como para cada uma das entidades propostas, e então agregada na Tabela 11 juntamente às métricas de precisão, *recall* e *F1 score*.

Tabela 11. Resultados do treinamento do modelo por entidade

Entidade	Precisão	Recall	F1 score	Avaliação
FOOD	0.967032967	0.8527131783	0.9062821833	Muito bom
ANIMAL	0.8754716981	0.7707641196	0.8197879859	Bom
LOC	0.9718670077	0.9793814433	0.9756097561	Muito bom
PER	0.7421875	0.4298642534	0.5444126074	OK
COURSE	0.950617284	0.6525423729	0.7738693467	Bom
PLANT	0.961038961	0.9135802469	0.9367088608	Muito bom
LANG	0.9813664596	0.9783281734	0.9798449612	Muito bom
NUM	0.9074585635	0.9791356185	0.9419354839	Muito bom
OP	0.9081632653	0.8240740741	0.8640776699	Bom
ORG	0.9490909091	0.872909699	0.9094076655	Muito bom
Total	0.9214294615	0.8253293179	0.8651936521	Bom

É possível observar que o modelo (total) foi definido com “Bom”, de acordo com seu F1 Score de ≈ 0.86 . A entidade com o melhor resultado de *F1 Score* foi a entidade LANG, representando entidades de nomes de idiomas, avaliada como “Muito bom” e com um F1 Score de ≈ 0.98 . A entidade que obteve o pior resultado foi a entidade PER, representando nomes de pessoas, avaliada como “OK” e com um F1 Score de ≈ 0.54 . O que mais afetou o resultado do F1 Score da entidade PER foi o baixo resultado de seu *recall* (≈ 0.43), ou seja, o modelo estaria falhando em reconhecer

algumas entidades PER que deveriam ser reconhecidas. Assume-se que o desempenho inferior dessa entidade em relação às demais poderia ser aprimorado caso o modelo conhecesse uma quantidade maior de dados de treinamento e validação desta entidade.

Ademais, a avaliação do restante das entidades foi positiva, interpretando os resultados das medidas como “Bom” ou “Muito bom”.

4.4 TESTE

A fim de realizar um teste visual da capacidade e acurácia do modelo, foram extraídos conteúdos textuais de 10 aplicativos a partir de seus arquivos *.aia*, assim como feito na seção 4.2. O texto extraído foi integralmente retirado do código dos aplicativos, sem passar por nenhum tipo de pré-processamento para remover caracteres especiais ou palavras consideradas irrelevantes no contexto de aplicativos. Os 10 apps escolhidos para essa etapa não pertenciam aos conjuntos de dados de treinamento e validação construídos na seção 4.2, a fim de se realizar o teste com conteúdos textuais totalmente alheios ao modelo. O texto completo extraído dos 10 apps de teste está documentado no Anexo A.

O teste foi realizado visando calcular manualmente a precisão, *recall* e *F1 score* (BROWNLEE, 2020) de cada tipo de entidade. Para isso, o texto dos apps foi analisado pelo autor e a cada termo encontrado foi alocado um valor (TP, TN, FP e FN) de acordo com o resultado de sua classificação.

O resultado dessa classificação de entidades é apresentado na Tabela 12, mostrando para cada um dos 10 apps de teste o texto presente no aplicativo e a classificação feita automaticamente pelo modelo, onde cada entidade reconhecida é destacada com uma cor e com uma identificação ao lado do termo. É mostrado também, abaixo do texto do app, a quantidade de palavras identificadas pelo autor

como TP, TN, FP e FN para cada uma das entidades do modelo, com os valores totais por entidade sendo mostrados na última linha da tabela.

Tabela 12. Análise de desempenho do modelo

App	Texto	FOOD				ANIMAL				LOC				PER				COURSE				PLANT				LANG				NUM				OP				ORG							
		TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
Sotaques	<p>Sotaques</p> <p>None Spelling Spelling by mauro.lima@seduc.net Apagar Sair Use os Emojis e ouça a descrição no idioma LANG selecionado. 1 NUM Português LANG Alemão LANG Inglês LANG Espanhol LANG</p> <p>Francês LANG Italiano LANG Russo LANG Chinês LANG Japonês LANG Árabe LANG Grego LANG Hebraico</p>	0	30	0	0	0	30	0	0	0	30	0	0	0	30	0	0	0	30	0	0	0	30	0	0	12	17	0	1	1	29	0	0	0	30	0	0	0	30	0	0	0	30	0	0
App_Reator_Biodiesel	<p>App_Reator_Biodiesel Tela_inicial Tela Inicial Bem-vindo ao aplicativo do reator de biodiesel! O aplicativo tem como objetivo realizar o controle à distância, de forma segura, do reator de biodiesel projetado pela UNESP</p> <p>ORG - Universidade Estadual Paulista de Tupã. ORG Iniciar Créditos App_Reator_Biodiesel Tela_Creditos Agradecimentos Projeto de pesquisa executado pelo Programa Institucional de Bolsas de Iniciação Científica</p> <p>- Ensino Médio (PIBIC-EM ORG) Edição/Vigência da bolsa: 2020/2021 NUM Bolsa fornecida pelo Conselho Nacional de Desenvolvimento ORG Científico e Tecnológico (CNPq) Instituições vinculadas: -</p> <p>Instituto Federal de São Paulo ORG - Câmpus Avançado Tupã (IFSP/TUP) - Universidade Estadual Paulista ORG - Faculdade de Ciências e Engenharia ORG (UNESP/FCE ORG) Nome do projeto:</p> <p>Desenvolvimento de Aplicativo de Celular, com MIT App Inventor, para Automação e Controle de um Reator de Biodiesel usando Arduino Bolsista: Lucas Satoshi Cipriano PER Oikawa Orientador: Paulo Sérgio</p> <p>Barbosa dos Santos PER Voltar 000 EMERGENCIA NUM None None 000 NUM 0 NUM http:// /MODO= NUM 060 NUM 60 NUM - Mensagem enviada Erro enviando mensagem 100 NUM 100</p> <p>NUM ON OFF App_Reator_Biodiesel Tela_painel Configurações do painel de controle do reator de biodiesel IP Arduino: Voltar 000 NUM O</p>	0	169	0	0	0	169	0	0	0	168	0	1	8	160	0	1	0	169	0	0	0	169	0	0	9	158	2	0	0	169	0	0	25	129	0	15								

BoxTest
e

BOXTESTE

Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Acre INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Alagoas INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Amapá INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Amazonas INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Distrito Federal Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência na Bahia ORG INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Ceará Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Espírito Santo INICIO Screen1 BOXTESTE Facebook INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Goiás Loc INICIO ParA Paraíba Pernambuco Loc Piaui parana RiodeJaneiro RioGrandedoNorte Screen1 BOXTESTE MAIS ESTADOS BRASILEIROS TELEFONES ÚTEIS DO BRASIL Loc Pará Paraíba Pernambuco Loc Piauí Loc Paraná Loc Rio de Janeiro Loc Rio Grande do Norte Loc INICIO MinasGerais Ceara MatoGrosso EspiritoSanto mts Goias MAISESTADO3 Maranhao BOXTESTE MAIS ESTADOS BRASILEIRO TELEFONES ÚTEIS DO BRASIL Ceará Loc Espírito Santo Goiás Loc Maranhão Loc Minas Gerais Loc Mato Grosso Loc Mato Grosso do Sul Loc MAIS ESTADOS RioGrandedoSul Rondonia Roraima SantaCatarina Sergipe Loc Tocantins BOXTESTE MAIS ESTADOS BRASILEIRO TELEFONES ÚTEIS DO BRASIL Rio Grande do Sul Loc Rondonia Roraima Santa Catarina Sergipe Tocantins Loc Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Maranhão Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Mato Grosso INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Minas Gerais Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Pará Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência na Paraíba ORG INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Pernambuco Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Piauí INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Rio Grande do Norte Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Rio Grande do Sul Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Rio de Janeiro Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência de Rondonia COURSE INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Roraima INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Santa Catarina Loc INICIO TELE3 BR ACRE Loc Alagoas Amazonas Loc Bahia Amapa Facebook MAISESTADOS2 BOXTESTE TELEFONES ÚTEIS DO BRASIL Hint for TextBox1 TELEFONE ÚTEIS DE TODO BRASIL São Paulo Loc Brasília Loc Acre Loc Alagoas Amazonas Loc Amapá Bahia MAIS ESTADOS Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Sergipe Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em São Paulo Loc Polícia Militar SAMU - Ambulâncias ORG Corpo de Bombeiros ORG Polícia Civil ORG Polícia Federal Defesa ORG Civil Polícia Rodoviária Federal ORG Secretaria dos Direitos Humanos ORG AACD Aeroporto Internacional de Cumbica Água e Esgoto AI-Anon Alcoólicos Anônimos Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Tocantins INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Mato Grosso do Sul Loc INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Paraná ORG INICIO

FOOD	ANIMAL	LOC	PER	COURSE	PLANT	LANG	NUM	OP	ORG
------	--------	-----	-----	--------	-------	------	-----	----	-----

TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN		
0	493	0	0	0	493	0	0	72	388	6	27	0	493	0	0	0	490	3	0	0	0	493	0	0	0	493	0	0	0	493	0	0	0	493	0	0	0	17	459	9	8

MamiferosRepteis

mamiferoserepteis

mamiferoserepteis Screen1 a TARTARUGA ANIMAL é um memifero reptil o LEÃO ANIMAL é um: mamifero reptil o CACHORRO ANIMAL é um mamifero reptil a IGUANA ORG é um mamifero reptil

FOOD				ANIMAL				LOC				PER				COURSE				PLANT				LANG				NUM				OP				ORG							
TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
0	26	0	0	3	22	0	1	0	26	0	0	0	26	0	0	0	26	0	0	0	26	0	0	0	26	0	0	0	26	0	0	0	26	0	0	0	26	0	0	0	25	1	0

Maratona_prog

maratona_prog

Funções Matemática Screen1 O que vamos estudar? Função do 1º grau Função do 2º grau maratona_prog Screen2 Definição Chama-se função polinomial do 1º grau, ou função afim, a qualquer função f de IR em IR dada por uma lei da forma $f(x) = ax + b$, onde a e b são números reais dados e $a \neq 0$. Na função $f(x) = ax + b$, o número a é chamado de coeficiente de x e o número b é chamado termo constante. Veja alguns exemplos de funções polinomiais do 1º grau: $f(x) = -2x - 7$ NUM, onde $a = -2$ e $b = -7$ NUM $f(x) = 5x - 3$ NUM, onde $a = 5$ e $b = -3$ NUM $f(x) = 11x$, onde $a = 11$ e $b = 0$ OP VAMOS PRATICAR! maratona_prog Screen3 $x + 2 = 7$ OP 0 NUM 2 NUM 3 NUM 5 NUM VERIFICAR RESPOSTA! RESOLUÇÃO maratona_prog Screen4 RESOLUÇÃO $x + 2 - 2 = 7$ NUM - 2 NUM $x + 0 = 5$ Dica para CaixaDeTexto3 $x = 5$ OP maratona_prog Screen5 Chamamos de equação do segundo grau as equações do tipo $ax^2 + bx + c = 0$ OP com a, b e c $\in \mathbb{R}$, onde $a \neq 0$. OP Os parâmetros da equação são: a – coeficiente principal b – coeficiente secundário c – termo independente Exemplo: $2x^2 + 5x + 3 = 0$ OP (essa é uma equação do segundo grau, veja o grau 2 NUM na primeira incógnita) Chamamos a, b e c de coeficientes, a é sempre coeficiente de x^2 , b é sempre coeficiente de x, e c é sempre coeficiente do termo independente. Exemplos: $3x^2 + 4x + 1 = 0$ OP : é uma equação do segundo grau, com $a = 3$, $b = 4$, $c = 1$. $x^2 - x - 1 = 0$ OP : é uma equação com grau 2 NUM, com $a = 1$, $b = -1$ OP, $c = -1$. $9x^2 - 5x = 0$ OP : também é uma equação de grau 2 NUM, com $a = 9$, $b = -5$, $c = 0$. $5x^2 - 4 = 0$ OP : equação do segundo grau, com $a = 5$, $b = 0$ OP, $c = -4$. OP VAMOS PRATICAR! maratona_prog Screen6 Dica para CaixaDeTexto1 $4x^2 + 8x + 6 = 0$ $\Delta = -32$ $\Delta = -0$ OP $\Delta = -5$ $\Delta = 1$ VERIFICAR RESPOSTA! RESOLUÇÃO maratona_prog Screen7 RESOLUÇÃO $\Delta = 8^2 - 4 \cdot 4 \cdot 6$ $\Delta = 64 - 96$ NUM $\Delta = -32$ NUM

FOOD				ANIMAL				LOC				PER				COURSE				PLANT				LANG				NUM				OP				ORG			
------	--	--	--	--------	--	--	--	-----	--	--	--	-----	--	--	--	--------	--	--	--	-------	--	--	--	------	--	--	--	-----	--	--	--	----	--	--	--	-----	--	--	--

	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
	0	398	0	0	0	398	0	0	0	398	0	0	0	398	0	0	0	397	0	1	0	398	0	0	0	398	0	0	14	384	0	0	86	239	5	68	0	398	0	0				

Guiade Estudos

GuiadeEstudos

Geografia COURSE Brasil Geografia Geral Frente História Brasil ORG História Geral Frente Gramática Literatura Redação Frente Biologia A Biologia COURSE B Biologia COURSE C Frente Química A Química COURSE B Química COURSE C Frente Física A Física B Física C Frente Matemática A Matemática B Matemática C Frente GuiadeEstudos Developed by Pablo Frentes Frente, Matemática A, Matemática B, Matemática C Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, Física A, Física B, Física C Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, Química COURSE A, Química B, Química C Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, Biologia COURSE A, Biologia COURSE B, Biologia C Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, Gramática, Literatura, Redação Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, História Brasil, História COURSE Geral Salvar Verificar Salvar Verificar Frentes Frente, Geografia Brasil LOC , Geografia Geral Salvar ORG

Verificar Salvar Verificar Aplicativo ainda em fase de desenvolvimento. Para entrar em contato harbar2705@gmail.com

FOOD				ANIMAL				LOC				PER				COURSE				PLANT				LANG				NUM				OP				ORG							
TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
0	144	0	0	0	144	0	0	1	140	0	3	0	144	0	0	9	106	0	29	0	144	0	0	0	144	0	0	0	144	0	0	0	144	0	0	0	144	0	0	0	136	8	0

Catalogo_de_arvores

Projeto_FINAL_Catalogo_de_arvores_1

http://servicos.cptec.inpe.br/XML/cidade/ /previsao.xml None nome cidade não encontrou "cidade" não encontrou "nome" previsao cidade não encontrou "cidade" não encontrou "previsao" \n Projeto_FINAL Screen1 CATÁLOGO DE ÁRVORES DO PARQUE MUNICIPAL NOVO CADASTRO BANCO DE DADOS Previsão do tempo: Digite o código da sua cidade. Ex: São Paulo LOC é 244 NUM (Consultar antes de iniciar visita periódica) Buscar previsão do tempo None None None None None latitude None longitude None , árvore Altura Data latitude longitude árvore n/a Altura n/a Data n/a latitude n/a longitude n/a SALVO! Projeto_FINAL Screen2 Espécie da árvore PLANT : Digite a espécie da árvore Altura da árvore PLANT : Digite a altura da árvore Localização: Data: DD/MM/AAAA VOLTAR SALVAR árvore PLANT None Altura None Data None latitude None , longitude None http://servicos.cptec.inpe.br/XML/cidade/ /previsao.xml None nome cidade não encontrou "cidade" não encontrou "nome" previsao cidade não encontrou "cidade" não encontrou "previsao" \n Projeto_FINAL Screen3 BANCO DE DADOS Altura Árvore Data Localização VOLTAR PARA INÍCIO

FOOD				ANIMAL				LOC				PER				COURSE				PLANT				LANG				NUM				OP				ORG			
------	--	--	--	--------	--	--	--	-----	--	--	--	-----	--	--	--	--------	--	--	--	-------	--	--	--	------	--	--	--	-----	--	--	--	----	--	--	--	-----	--	--	--

TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
0	149	0	0	0	149	0	0	1	148	0	0	0	149	0	0	0	149	0	0	3	140	0	6	0	149	0	0	1	148	0	0	0	149	0	0	0	149	0	0

Tribo

Tribo

endereço Lista de endereço Digite o local e clique em enviar! Clique em Adicionar endereço e digite parte ou o endereço completo Digite o CEP em adicionar endereço Adicionado, cliquem em seleccionar endereço e seleccione o endereço Você pode ver no mapa Alerta Ok Endereço inválido. Por favor, digite um novo endereço! Alerta Ok None Endereço adicionado com sucesso android.intent.action.VIEW geo:0,0?q= Nenhum endereço seleccionado Alerta Ok android.intent.action.VIEW geo:0,0?q= Tribo Eletronica Veja onde está sua tribo: Selecione sua locação Adicionar endereço Ajuda Digite o endereço: Hint for TextBox1 Enviar Cancelar Endereço seleccionado: Hint for TextBox1 Ver no mapa Onde estou? None None usuario 0 NUM senha 0 NUM 0 NUM Salvar Para utilizar o aplicativo é necessário ter um login e senha Alerta Ok Nenhum campo pode estar vazio usuario senha None None Entrar usuario 0 NUM senha 0 NUM Login salvo com Sucesso Parabéns Ok Nenhum campo pode estar vazio Tribos Usuário e/ou senha incorretos! None None Tribo Login Desenvolvido por: Ana Kelly PER, Frederico Kappaun ORG, Leonardo Cabral PER, Mariana Mendes e Susane ORG Bonifácio. Digite o usuário Digite a Senha Entrar Cancelar Tribo Pop Tribo Rock Login Tribos Screen1 Desenvolvido por: Ana Kelly PER, Frederico Kappaun ORG, Leonardo Cabral PER, Mariana Mendes e Susane ORG Bonifácio. Tribo Sertanejo Eletronica Sertanejo Rock Pop Tribo Tribos

FOOD				ANIMAL				LOC				PER				COURSE				PLANT				LANG				NUM				OP				ORG							
TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
0	203	0	0	0	203	0	0	0	203	0	0	8	183	0	12	0	203	0	0	0	203	0	0	0	203	0	0	5	198	0	0	0	203	0	0	0	199	4	0				

English Translator

EnglishTranslator

TelaInicial Screen2 English Translator Menu Welcome Inglês LANG - Português LANG Como você deseja traduzir o seu texto? Português LANG - Inglês LANG Criado por: Douglas Raynner da Silva Santos PER Giovanni Alvarenga Vieira Soares Mateus Coelho Santos PER portuguese EnglishTranslator Inglês LANG - Português LANG Típe your text here: English: Portuguese: Translate english EnglishTranslator Português LANG - Inglês LANG Digite aqui; Português LANG : Inglês LANG : Traduzir

FOOD				ANIMAL				LOC				PER				COURSE				PLANT				LANG				NUM				OP				ORG			
------	--	--	--	--------	--	--	--	-----	--	--	--	-----	--	--	--	--------	--	--	--	-------	--	--	--	------	--	--	--	-----	--	--	--	----	--	--	--	-----	--	--	--

A partir do resultado das classificações, as medidas de desempenho foram calculadas e classificadas de acordo com seus resultados. Esses resultados são apresentados na Tabela 13.

Tabela 13. Avaliação de precisão, recall e F1 score

Entidade	Precisão	Recall	F1 score	Avaliação
FOOD	1	0.5064935065	0.6724137931	OK
ANIMAL	0.6	0.75	0.6666666667	OK
LOC	0.925	0.7047619048	0.8	Bom
PER	1	0.6829268293	0.8115942029	Bom
COURSE	0.75	0.2307692308	0.3529411765	Ruim
PLANT	1	0.4545454545	0.625	OK
LANG	0.9166666667	0.9565217391	0.9361702128	Muito bom
NUM	0.9459459459	1	0.9722222222	Muito bom
OP	0.9450549451	0.5584415584	0.7020408163	OK
ORG	0.65625	0.6461538462	0.6511627907	OK
Total	0.8738917558	0.649061407	0.7190211881	OK

É possível observar comparando os resultados da Tabela 11 e da Tabela 13 que o *F1 score* do modelo no teste executado se mostrou inferior ao resultado da validação calculado pelo Spacy. No teste, o modelo obteve um *F1 score* de ≈ 0.72 , sendo esse considerado um grau de desempenho “OK”, enquanto a validação do Spacy resultou em ≈ 0.86 , sendo esse um grau de desempenho “Bom”.

Essa diferença se deu pelos resultados de *F1 score* obtidos nas entidades FOOD, ANIMAL, COURSE, PLANT, OP e ORG, que foram inferiores aos resultados da validação do Spacy, principalmente por conta dos resultados de *recall*, que também foram inferiores. Ou seja, no teste realizado com os 10 apps selecionados, o modelo teve mais dificuldade para reconhecer essas entidades do que nos apps do conjunto de validação fornecido ao Spacy. No teste, a entidade COURSE, que representa

disciplinas escolares e acadêmicas, foi a única classificada como um desempenho “Ruim”, obtendo um F1 score de ≈ 0.35 .

Nota-se, entretanto, que as entidades LANG e NUM não obtiveram diferenças em suas avaliações, continuando com a classificação de desempenho “Muito bom” em relação à validação do Spacy. Houve, ainda, o caso da entidade PER, que no teste executado foi a única entidade a obter uma classificação superior em relação à validação do Spacy, obtendo um F1 score de ≈ 0.81 e portanto indicando um grau de desempenho “Bom”.

É possível concluir a partir dos testes executados que o modelo possui uma variação de desempenho ao realizar a classificação automática de entidades em textos retirados de apps, podendo ser mais ou menos efetivo dependendo do texto fornecido ao mesmo.

Assume-se que essa volatilidade nos resultados de desempenho, principalmente o baixo desempenho da entidade COURSE, tenha sido afetado pela diferença dos elementos textuais a serem analisados pelo modelo. Ou seja, o modelo pode ser mais efetivo em reconhecer entidades em alguns casos, como no caso dos elementos textuais dos apps de validação, e menos efetivo em outros, como no caso dos elementos textuais dos apps de teste. Assume-se que caso o modelo seja treinado com uma maior quantidade de elementos textuais de apps, os resultados de suas métricas se tornem mais precisos e consistentes.

Há, ainda, a questão do balanceamento das entidades do conjunto de dados utilizado no modelo. Esse balanceamento é apresentado no tópico *Distribuição dos dados por entidade* na Tabela 9, e nele é possível observar que as entidades PLANT, OP e COURSE se apresentam em uma quantidade consideravelmente menor em relação ao restante das entidades do modelo. Nesse sentido, é possível que o desempenho inferior da entidade COURSE tenha sido causado por esta razão, uma vez que é recomendado que o modelo possua um balanceamento igualitário entre todas as entidades.

Apesar dos problemas analisados, os resultados gerais apresentados foram positivos, uma vez que o desempenho do modelo em sua totalidade foi classificado como “Bom” e “OK” nas duas ocasiões em que suas medidas de desempenho foram calculadas. Portanto, é possível afirmar que o modelo possui efetividade em classificar as entidades propostas neste trabalho, ainda que essa efetividade seja volátil.

4.5 DISCUSSÃO E TRABALHOS FUTUROS

A partir dos resultados de treinamento e testes do modelo, e conseqüentemente do objetivo geral do trabalho, surgem os seguintes desafios que podem ser desvendados em trabalhos futuros:

Aprimoramento e extensão do modelo. Como visto na seção 4.4, o modelo apresenta uma certa volatilidade na detecção de algumas entidades. Assim, sugere-se que seja feita uma análise mais aprofundada sobre os hiperparâmetros utilizados no modelo a fim de melhorar sua performance. Além disso, é possível aprimorar o modelo com um conjunto de dados de treinamento em maior quantidade e com um melhor balanceamento de entidades, realizando uma busca maior nos elementos textuais dos aplicativos e então realizando mais anotações manuais no conjunto de dados. Essa busca pode ainda ser aproveitada para a identificação de novas entidades relevantes no contexto de aplicativos móveis que não foram incluídas no presente modelo, estendendo-o para o conhecimento de mais entidades e posteriormente auxiliando ainda mais na avaliação da originalidade.

Utilização do modelo na avaliação de originalidade. Com o modelo pronto para ser utilizado na prática, é possível estudar seu uso efetivo na avaliação de originalidade de aplicativos dentro de um contexto educacional. Um exemplo seria a coleta da taxa de entidades presentes em um certo aplicativo individualmente e a comparação desta taxa com outros aplicativos dentro de um universo de referência.

Outras abordagens. Outro assunto interessante a ser discutido é a análise de abordagens alternativas ao NER para a identificação de tópicos em elementos textuais. Um exemplo de uma abordagem alternativa que poderia ser adotada nesse contexto poderia ser por meio do uso e análise de ontologias, dentro do conceito de web semântica.

5. CONCLUSÃO

Neste trabalho foi desenvolvida uma abordagem para a identificação de entidades do conteúdo textual de apps criados com App Inventor. Como parte do trabalho foi realizada uma síntese da fundamentação teórica em relação a aplicativos criados com App Inventor e a extração de elementos textuais dos mesmos. Além disso, foram apresentados conceitos de NLP e NER e como utilizar conteúdo textual para criar e treinar modelos de NER.

Como resultado de uma análise sistemática do estado da arte em relação a modelos de NER para a classificação de tópicos de apps desenvolvidos observou-se que ainda não existem pesquisas com esse foco.

Assim foi definido e implementado um modelo de NER para identificação de entidades do conteúdo textual de apps criados com App Inventor. Após levantados os requisitos e preparação do conjunto de dados, foi executado o treinamento de um modelo do pacote Spacy em português visando reconhecer as 10 entidades definidas na análise de requisitos. Analisando as métricas de precisão, *recall* e *F1 score* do modelo a partir do conjunto de dados de validação observou-se que o mesmo obteve níveis de desempenho aceitáveis.

Em seguida, foi executado um teste com um novo conjunto de de 10 aplicativos comparando um reconhecimento manual de entidades à identificação automática pelo modelo treinado. Analisando as métricas, observou-se que o resultado da precisão, *recall* e *F1 score* obteve uma variação para a maioria das entidades propostas em relação aos resultados obtidos na validação. Ainda assim, os

resultados do teste indicaram no mínimo um grau de desempenho aceitável para todas as entidades, com exceção da entidade COURSE. O desempenho do modelo como um todo também obteve valores aceitáveis, indicando portanto uma relativa eficácia na classificação das entidades propostas.

Assim, este trabalho pode contribuir para a avaliação da criatividade de escolha de tópicos de aplicativos criados com App Inventor, auxiliando no ensino da computação na educação básica e conseqüentemente contribuindo para o desenvolvimento de uma competência essencial no século XXI.

Como trabalhos futuros, sugere-se a aprimoração da eficácia do modelo, bem como uma análise mais aprofundada e ajustes nos hiperparâmetros do algoritmo. Sugere-se, ainda, a adição de mais entidades ao modelo que possam ser relevantes no contexto de aplicativos móveis, e então seu uso efetivo na avaliação de originalidade. Também é possível realizar um estudo sobre a classificação de tópicos e avaliação da criatividade adotando técnicas alternativas, como por exemplo a abordagem de ontologias e web semântica no lugar de NER.

REFERÊNCIAS

ALLWRIGHT, S. 2022. What is a good F1 score and how do I interpret it? Disponível em: <<https://stephenallwright.com/good-f1-score/>>. Acesso em Novembro de 2022.

ALVES, N. d. C.; BARBOSA, H.; AZEVEDO, L. F. CodeMaster – Automatic Assessment and Grading of App Inventor and Snap! Programs. Informatics in Education, 17(1), 2018, 117-150.

ALVES, N. da C., GRESSE VON WANGENHEIM, C., ALBERTO, M., MARTINS-PACHECO, L. H. Uma Proposta de Avaliação da Originalidade do Produto no Ensino de Algoritmos e Programação na Educação Básica. In: Anais do Simpósio Brasileiro de Informática na Educação, Natal, Brasil, 2020.

ALVES, N. da C., GRESSE VON WANGENHEIM, C., MARTINS-PACHECO, L. H., BORGATTO, A. F. Artefatos computacionais são considerados criativos? In: Anais do Simpósio Brasileiro de Educação em Computação, Feira de Santana, Bahia, 2022.

ALVES, N. da C., GRESSE VON WANGENHEIM, C., & MARTINS-PACHECO, L. H. (2021). Assessing Product Creativity in Computing Education: A Systematic Mapping Study. Informatics in Education, 20(1), 19-45. doi:10.15388/infedu.2021.02

AMARAL, D. O. F. D. “O reconhecimento de entidades nomeadas por meio de conditional random fields para a língua portuguesa,”. Dissertação do programa de Mestrado em Ciência da Computação da PUCRS, Porto Alegre, 2013.

APP INVENTOR COMMUNITY. AIA File Structure, 2019. Disponível em: <<https://community.appinventor.mit.edu/t/aia-file-structure/219>>.

BINKLEY, M., et al., Defining 21st Century Skills. Draft white paper 1. Melbourne: The University of Melbourne, Melbourne, Australia. 2010.

BROWNLEE, J. (2020). How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification. Disponível em: <<https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced>>

-classification/>. Acesso em Julho de 2022.

EBRAHIMI, F., TUSHEV, M., MAHMOUD, A. 2021. Classifying Mobile Applications Using Word Embeddings. ACM Trans. Softw. Eng. Methodol. 31(2), Artigo 20 , 2022.

Explosion. Spacy, 2015. Disponível em: <<https://spacy.io/>>. Acesso em Julho de 2022.

FlairNLP. Flair, 2018. Disponível em: <<https://github.com/flairNLP>>. Acesso em Julho de 2022.

GRESSE VON WANGENHEIM,C.; HAUCK, J. C. R.; DEMETRIO, M. F.; PELLE, R. CodeMaster - Automatic Assessment and Grading of App Inventor and Snap! Programs. Informatics in Education, 17(1), 2018, p. 117-150.

LEE, S.; KO, Y. Named-Entity Recognition Using Automatic Construction of Training Data From Social Media Messaging Apps. IEEE Access, 8, 2020.

MIT App Inventor, About Us, 2012. Disponível em: <<http://appinventor.mit.edu/about-us>>. Acesso em: Maio de 2020.

MUSTAFARAJ, E., TURBAK, F., SVANBERG, M. Identifying Original Projects in App Inventor. In: Proc. of the 30th Int. Florida Artificial Intelligence Research Society Conference, Marco Island, FL, USA, 2017, 567-572.

NADKARNI, P.M; OHNO-MACHADO, L.; CHAPMAN, W.W. Natural language processing: an introduction, Journal of the American Medical Informatics Association, 18(5), 2011. National Research Council . Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century. Disponível em: <<https://doi.org/10.17226/13398>>. Washington, DC, USA. 2012.

NLTK Project. NLTK, 2001. Disponível em: <<https://www.nltk.org>>. Acesso em Julho de 2022.

OXFORD LANGUAGES, Dicionário Online de Português. Disponível em: <<https://languages.oup.com/google-dictionary-pt/>>. Acesso em: 09/12/2021.

PATTON, E.W.; TISSENBAUM, M.; HARUNANI, F., MIT App Inventor: Objectives,

Design and Development. In: KONG, S.; ABELSON, H. Computational Thinking Education. Springer, Singapore, 2019.

QUEIROZ, R. L.; SAMPAIO, F. F. DuinoBlocks for Kids: um ambiente de programação em blocos para o ensino de conceitos básicos de programação a crianças do Ensino Fundamental I por meio da Robótica Educacional. In: Anais do Workshop sobre Educação em Computação, Porto Alegre, RS, Brasil, 2016.

RANJAN B.S.C.; SIDDHARTH, L.; CHAKRABARTI, A. A systematic approach to assessing novelty, requirement satisfaction, and creativity. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 32(4), p. 390-414, 2018.

SCAICO, P. D. LIMA, A. A., SILVA, J. B. B., AZEVEDO, S., PAIVA, L. F., RAPOSO, E. H., ALENCAR, Y., MENDES, J. P., SCAICO, A. Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch. Revista Brasileira de Informática na Educação, 21(2), 2013.

TABASSUM, J.; MADDELA, M.; XU, W.; RITTER, A. Code and Named Entity Recognition in StackOverflow.” ArXiv abs/2005.01634 (2020).

NER Annotator. Tecoholic, 2022. Disponível em: <<https://tecoholic.github.io/ner-annotator/>>. Acesso em outubro de 2022.

TISSENBAUM, M.; SHELDON, J.; ABELSON, H. From Computational Thinking to Computational Action. Communication of the ACM, 62(3), 34-36, 2019 .

Training Custom NER models in Spacy to auto-detect named entities [Complete Guide], 2020. Disponível em: <<https://www.machinelearningplus.com/nlp/training-custom-ner-model-in-spacy/>>.

VAGARINHO, J P. Como identificar a originalidade num artigo científico ou numa tese de doutoramento? Educar em Revista, 2019.

VAKULENKO, S.; MÜLLER, O.; BROCKE, J. V. Enriching iTunes App Store Categories via Topic Modeling. In: International Conference on Information Systems, Auckland, New Zealand, 2014.

YADAV, V.; BETHARD, S. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. In: Proc. of the 27th International Conference on Computational Linguistics, Santa Fe, USA, 2018.

ANEXO A - DADOS TEXTUAIS DOS 10 APLICATIVOS UTILIZADOS NA ETAPA DE TESTE

Nome do app	Conteúdo textual
Sotaques	None Spelling Spelling by mauro.lima@seduc.net Apagar Sair Use os Emojis e ouça a descrição no idioma selecionado. 1 Português Alemão Inglês Espanhol Francês Italiano Russo Chinês Japonês Árabe Grego Hebraico
App_Reator_Biodiesel	App_Reator_Biodiesel Tela_inicial Tela Inicial Bem-vindo ao aplicativo do reator de biodiesel! O aplicativo tem como objetivo realizar o controle à distância, de forma segura, do reator de biodiesel projetado pela UNESP - Universidade Estadual Paulista de Tupã. Iniciar Créditos App_Reator_Biodiesel Tela_Creditos Agradecimentos Projeto de pesquisa executado pelo Programa Institucional de Bolsas de Iniciação Científica - Ensino Médio (PIBIC-EM) Edição/Vigência da bolsa: 2020/2021 Bolsa fornecida pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) Instituições vinculadas: · Instituto Federal de São Paulo - Câmpus Avançado Tupã (IFSP/TUP) · Universidade Estadual Paulista - Faculdade de Ciências e Engenharia (UNESP/FCE) Nome do projeto: Desenvolvimento de Aplicativo de Celular, com MIT App Inventor, para Automação e Controle de um Reator de Biodiesel usando Arduino Bolsista: Lucas Satoshi Cipriano Oikawa Orientador: Paulo Sérgio Barbosa dos Santos Voltar 000 EMERGENCIA None None 000 0 http:// /MOD0= 060 60 - Mensagem enviada Erro enviando mensagem 100 100 ON OFF App_Reator_Biodiesel Tela_painel Configurações do painel de controle do reator de biodiesel IP Arduino: Voltar 000 O
BoxTeste	Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Acre INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Alagoas INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Amapá INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Amazonas INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Distrito Federal INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência na Bahia INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Ceará INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Espírito Santo INICIO Screen1 BOXTESTE Facebook INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Goiás INICIO ParA Paraíba Pernambuco Piauí parana RiodeJaneiro RioGrandedoNorte Screen1 BOXTESTE MAIS ESTADOS BRASILEIROS TELEFONES ÚTEIS DO BRASIL Pará Paraíba Pernambuco Piauí Paraná Rio de Janeiro Rio Grande do Norte INICIO MinasGerais Ceara MatoGrosso EspiritoSanto mts Goiás MAISESTADO3 Maranhao BOXTESTE MAIS ESTADOS BRASILEIRO TELEFONES ÚTEIS DO BRASIL Ceará Espírito Santo Goiás Maranhão Minas Gerais Mato Grosso Mato Grosso do Sul MAIS ESTADOS RioGrandedoSul Rondonia Roraima SantaCatarina Sergipe Tocantins BOXTESTE MAIS ESTADOS BRASILEIRO TELEFONES ÚTEIS DO BRASIL Rio Grande do Sul Rondonia Roraima Santa Catarina Sergipe Tocantins Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Maranhão INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Mato Grosso INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Minas Gerais INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Pará INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência na Paraíba INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Pernambuco INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Piauí INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Rio Grande do Norte INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Rio Grande do Sul INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Rio de Janeiro INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência de Rondonia INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Roraima INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Santa Catarina INICIO TELE3 BR ACRE Alagoas Amazonas Bahia Amapa Facebook MAISESTADOS2 BOXTESTE TELEFONES ÚTEIS DO BRASIL Hint for TextBox1 TELEFONE ÚTEIS DE TODO BRASIL São Paulo Brasília Acre Alagoas Amazonas Amapá Bahia MAIS ESTADOS Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Sergipe INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em São Paulo Polícia Militar SAMU - Ambulâncias Corpo de Bombeiros Polícia Civil Polícia Federal Defesa Civil Polícia Rodoviária Federal Secretaria dos Direitos Humanos AACD Aeroporto Internacional de Cumbica Água e Esgoto Al-Anon Alcoólicos Anônimos INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência em Tocantins INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Mato Grosso do Sul INICIO Screen1 BOXTESTE Lista de Telefones Úteis e de Emergência no Paraná INICIO
MamiferosRepteis	mamiferoserepteis Screen1 a TARTARUGA é um memifero reptil o LEÃO é um: mamifero reptil o CACHORRO é um mamifero reptil a IGUANA é um mamifero reptil

Maratona_prog	<p>Funções Matemática Screen1 O que vamos estudar? Função do 1º grau Função do 2º grau maratona_prog Screen2 Definição Chama-se função polinomial do 1º grau, ou função afim, a qualquer função f de \mathbb{R} em \mathbb{R} dada por uma lei da forma $f(x) = ax + b$, onde a e b são números reais dados e $a \neq 0$. Na função $f(x) = ax + b$, o número a é chamado de coeficiente de x e o número b é chamado termo constante. Veja alguns exemplos de funções polinomiais do 1º grau: $f(x) = -2x - 7$, onde $a = -2$ e $b = -7$ $f(x) = 5x - 3$, onde $a = 5$ e $b = -3$ $f(x) = 11x$, onde $a = 11$ e $b = 0$ VAMOS PRATICAR! maratona_prog Screen3 $x + 2 = 7$ 0 2 3 5 VERIFICAR RESPOSTA! RESOLUÇÃO maratona_prog Screen4 RESOLUÇÃO $x + 2 - 2 = 7 - 2$ $x + 0 = 5$ Dica para CaixaDeTexto3 $x = 5$ maratona_prog Screen5 Chamamos de equação do segundo grau as equações do tipo $ax^2 + bx + c = 0$ com a, b e $c \in \mathbb{R}$, onde $a \neq 0$. Os parâmetros da equação são: a – coeficiente principal b – coeficiente secundário c – termo independente Exemplo: $2x^2 + 5x + 3 = 0$ (essa é uma equação do segundo grau, veja o grau 2 na primeira incógnita) Chamamos a, b e c de coeficientes, a é sempre coeficiente de x^2, b é sempre coeficiente de x, e c é sempre coeficiente do termo independente. Exemplos: $3x^2 + 4x + 1 = 0$: é uma equação do segundo grau, com $a = 3, b = 4, c = 1$. $x^2 - x - 1 = 0$: é uma equação com grau 2, com $a = 1, b = -1, c = -1$. $9x^2 - 5x = 0$: também é uma equação de grau 2, com $a = 9, b = -5, c = 0$. $5x^2 - 4 = 0$: equação do segundo grau, com $a = 5, b = 0, c = -4$. VAMOS PRATICAR! maratona_prog Screen6 Dica para CaixaDeTexto1 $4x^2 + 8x + 6 = 0$ $\Delta = -32$ $\Delta = -0$ $\Delta = -5$ $\Delta = 1$ VERIFICAR RESPOSTA! RESOLUÇÃO maratona_prog Screen7 RESOLUÇÃO $\Delta = 8^2 - 4 \cdot 4 \cdot 6$ $\Delta = 64 - 96$ $\Delta = -32$</p>
GuiadeEstudos	<p>Geografia Brasil Geografia Geral Frente História Brasil História Geral Frente Gramática Literatura Redação Frente Biologia A Biologia B Biologia C Frente Química A Química B Química C Frente Física A Física B Física C Frente Matemática A Matemática B Matemática C Frente GuiadeEstudos Developed by Pablo Frentes Frente, Matemática A, Matemática B, Matemática C Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, Física A, Física B, Física C Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, Química A, Química B, Química C Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, Biologia A, Biologia B, Biologia C Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, Gramática, Literatura, Redação Salvar Verificar Salvar Verificar Salvar Verificar Frentes Frente, História Brasil, História Geral Salvar Verificar Salvar Verificar Frentes Frente, Geografia Brasil, Geografia Geral Salvar Verificar Salvar Verificar Aplicativo ainda em fase de desenvolvimento. Para entrar em contato harbar2705@gmail.com</p>
Catalogo_de_arvore_s	<p>http://servicos.cptec.inpe.br/XML/cidade/ /previsao.xml None nome cidade não encontrou "cidade" não encontrou "nome" previsao cidade não encontrou "cidade" não encontrou "previsao" \n Projeto_FINAL Screen1 CATÁLOGO DE ÁRVORES DO PARQUE MUNICIPAL NOVO CADASTRO BANCO DE DADOS Previsão do tempo: Digite o código da sua cidade. Ex: São Paulo é 244 (Consultar antes de iniciar visita periódica) Buscar previsão do tempo None None None None None latitude None longitude None , árvore Altura Data latitude longitude árvore n/a Altura n/a Data n/a latitude n/a longitude n/a SALVO! Projeto_FINAL Screen2 Espécie da árvore: Digite a espécie da árvore Altura da árvore: Digite a altura da árvore Localização: Data: DD/MM/AAAA VOLTAR SALVAR árvore None Altura None Data None latitude None , longitude None http://servicos.cptec.inpe.br/XML/cidade/ /previsao.xml None nome cidade não encontrou "cidade" não encontrou "nome" previsao cidade não encontrou "cidade" não encontrou "previsao" \n Projeto_FINAL Screen3 BANCO DE DADOS Altura Árvore Data Localização VOLTAR PARA INÍCIO</p>
Tribo	<p>endereço Lista de endereço Digite o local e clique em enviar! Clique em Adicionar endereço e digite parte ou o endereço completo Digite o CEP em adicionar endereço Adicionado, cliquem em selecionar endereço e seleccione o endereço Você pode ver no mapa Alerta Ok Endereço inválido. Por favor, digite um novo endereço! Alerta Ok None Endereço adicionado com sucesso android.intent.action.VIEW geo:0,0?q= Nenhum endereço selecionado Alerta Ok android.intent.action.VIEW geo:0,0?q= Tribo Eletronica Veja onde está sua tribo: Selecione sua locação Adicionar endereço Ajuda Digite o endereço: Hint for TextBox1 Enviar Cancelar Endereço selecionado: Hint for TextBox1 Ver no mapa Onde estou? None None usuario 0 senha 0 0 Salvar Para utilizar o aplicativo é necessário ter um login e senha Alerta Ok Nenhum campo pode estar vazio usuario senha None None Entrar usuario 0 senha 0 Login salvo com Sucesso Parabéns Ok Nenhum campo pode estar vazio Tribos Usuário e/ou senha incorretos! None None Tribo Login Desenvolvido por: Ana Kelly, Frederico Kappaun, Leonardo Cabral, Mariana Mendes e Susane Bonifácio. Digite o usuário Digite a Senha Entrar Cancelar Tribo Pop Tribo Rock Login Tribos Screen1 Desenvolvido por: Ana Kelly, Frederico Kappaun, Leonardo Cabral, Mariana Mendes e Susane Bonifácio. Tribo Sertanejo Eletronica Sertanejo Rock Pop Tribo Tribos</p>
EnglishTranslator	<p>TelaInicial Screen2 English Translator Menu Welcome Inglês - Português Como você deseja traduzir o seu texto? Português - Inglês Criado por: Douglas Raynner da Silva Santos Giovanni Alvarenga Vieira Soares Mateus Coelho Santos portuguese EnglishTranslator Inglês - Português Tipe your text here: English: Portuguese: Translate english EnglishTranslator Português - Inglês Digite aqui; Português: Inglês: Traduzir</p>
Projeto_um_para_u	<p>Screen2 teste Screen1 QUER VER AS RECEITAS? QUEROO! Screen5 teste Screen10 INGREDIENTES -4 ovos -2 xícaras (chá) de açúcar -1 xícara (chá) de óleo</p>

m	<p>-suco de 2 laranjas -casca de 1 laranja -2 xícaras (chá) de farinha de trigo -1 colher (sopa) de fermento VOLTAR Screen5 teste Screen11 MODO DE PREPARO 1- Bata no liquidificador os ovos, o açúcar, o óleo, o suco e a casca da laranja. 2- Passe para uma tigela e acrescente a farinha de trigo e o fermento. 3- Leve para assar em uma forma com furo central, untada e enfarinhada, por mais ou menos 30 minutos. 5- Desenforme o bolo e molhe com suco de laranja. VOLTAR Screen3 Screen4 Screen5 Screen1 teste Screen2 RECEITAS bolo de chocolate bolo de cenoura bolo de laranja VOLTAR Screen2 Screen6 Screen7 teste Screen3 BOLO DE CHOCOLATE! INGREDIENTES MODO DE PREPARO VOLTAR Screen2 Screen8 Screen9 teste Screen4 BOLO DE CENOURA! INGREDIENTES MODO DE PREPARO VOLTAR Screen2 Screen10 Screen11 teste Screen5 BOLO DE LARANJA! INGREDIENTES MODO DE PREPARO VOLTAR Screen3 teste Screen6 INGREDIENTES -5 ovos -1 xícara (chá) de açúcar -1 xícara (chá) de Leite Integral -2 xícaras (chá) de farinha de trigo -meia xícara (chá) de Chocolate em Pó -1 colher (sopa) de fermento em pó VOLTAR Screen3 teste Screen7 MODO DE PREPARO 1- Em uma batedeira, bata as claras em neve. 2- Junte as gemas, uma a uma, e acrescente o açúcar. 3- Despeje o Leite NINHO aos poucos, sem parar de bater. Incorpore delicadamente a farinha peneirada com o Chocolate em pó e o fermento. 4-Despeje em uma forma redonda (28 cm de diâmetro) untada com manteiga e polvilhada com farinha de trigo e leve para assar em forno médio-alto (200°C), preaquecido, por cerca de 40 minutos. 5- Desenforme, deixe esfriar e corte-o ao meio. VOLTAR Screen4 teste Screen8 INGREDIENTES -3 cenouras médias cortadas em cubinhos -2 xícaras (chá) de açúcar -2 xícaras (chá) de farinha de trigo -1 xícara de óleo -4 ovos -1 pitada de sal -1 colher (sopa) de fermento em pó VOLTAR Screen4 teste Screen9 MODO DE PREPARO 1- Bata no liquidificador as cenouras e o óleo até que a mistura fique em creme; 2- Acrescente os ovos, açúcar e o sal e bata até misturar bem; 3- Coloque o creme na batedeira e vá acrescentando a farinha aos poucos até que a massa fique homogênea; 4- Seguidamente, coloque o fermento em pó e misture a massa, calmamente, com uma colher ou fuê. VOLTAR</p>
---	---

Apêndice 1: Artigo científico originado do presente estudo.

Definição de Modelo de Reconhecimento de Entidade Nomeada para Detecção Automática de Tópicos de Aplicativos em Português

Eduardo Dias Gutterres

Departamento de Informática e Estatística, Universidade Federal de Santa Catarina,
Florianópolis, SC, Brasil

***Abstract.** With the social and technological changes, creativity has come to be considered essential. An alternative for the development of creativity even in K-12 is through the study of computing by developing computational artifacts, for example, mobile applications with App Inventor. In this context, it is important to evaluate the student's learning also in relation to the development of creativity. Considering originality one of the main dimensions of creativity, the aim is to evaluate the originality of applications created with App Inventor in relation to the topics they address. Using as input for this analysis the textual elements extracted from the apps, the classification of these topics can be improved with the use of Named Entity Recognition (NER). NER is a computing technique within the field of Natural Language Processing (NLP) that allows entities to be automatically recognized from words and phrases. Thus, this work aims to develop a NER model in Portuguese for entity identification in order to assist in the automatic classification of mobile application topics created with App Inventor. With this, it is expected to contribute to the evaluation of the originality of the applications created by the students, thus also contributing to the teaching of computing in schools in the country.*

***Resumo.** Com as mudanças sociais e tecnológicas presentes ao longo dos séculos, a criatividade passou a ser considerada essencial. Uma alternativa para o desenvolvimento da criatividade ainda na educação básica se dá por meio do estudo da área da computação com o desenvolvimento de artefatos computacionais, como por exemplo aplicativos móveis com App Inventor. Nesse contexto, é importante a avaliação da aprendizagem do aluno também em relação ao desenvolvimento da criatividade. Considerando a originalidade como uma das principais dimensões da criatividade, visa-se assim realizar a avaliação da originalidade dos aplicativos criados com App Inventor em relação aos tópicos que abordam. Usando como entrada para essa análise os elementos textuais extraídos dos apps, a classificação desses tópicos pode ser melhorada com o uso do Reconhecimento de Entidade Nomeada (NER - Named Entity Recognition). O NER é uma técnica de computação dentro da área de Processamento de Linguagem Natural (NLP - Natural Language Processing)*

que permite que entidades sejam reconhecidas automaticamente a partir de palavras e frases. Dessa maneira, este trabalho tem como objetivo desenvolver um modelo de NER em português para identificação de entidades a fim de auxiliar na classificação automática de tópicos de aplicativos móveis criados com App Inventor. Com isso, espera-se contribuir para a avaliação da originalidade dos aplicativos criados pelos alunos, contribuindo assim também para o ensino da computação nas escolas do país.

1. Introdução

A constante mudança e evolução da humanidade em diversas áreas fez com que algumas competências e habilidades humanas passassem a ser consideradas essenciais no século XXI, justamente por contribuírem para que as pessoas consigam lidar com situações reais desse período (NRC, 2012). Uma dessas competências essenciais é a criatividade, pois auxilia na criação de ideias inovadoras e na resolução de problemas complexos e atuais (BINKLEY, 2010).

Tipicamente, a criatividade e o seu desenvolvimento estão ligados à área de artes (ALVES et al., 2022). Porém, uma alternativa para o desenvolvimento dessa habilidade na educação básica é por meio do ensino da computação, uma vez que a criação e desenvolvimento de artefatos de software pelas crianças exercita a curiosidade intelectual e estimula a reflexão e a criatividade (ALVES et al., 2022).

Muitas vezes a computação é ensinada na educação básica adotando uma estratégia ativa a fim de tornar o aprendizado mais fácil e dinâmico para os alunos, levando-os a criar artefatos computacionais. Para esse fim, tipicamente, são usados ambientes de programação baseados em blocos visuais (PATTON et al., 2019). Um exemplo é o App Inventor (<https://appinventor.mit.edu>), que permite com que o usuário crie aplicativos para as plataformas Android e IOS por meio de uma interface visual, abstraindo a lógica de programação. Como resultado, podem ser criados aplicativos funcionais (PATTON et al., 2019).

Desta maneira, a criatividade do aluno na educação básica pode ser desenvolvida por meio da aprendizagem de desenvolvimento de aplicativos móveis. Nesse contexto educacional, é também importante avaliar a aprendizagem do aluno com base nos resultados dos apps criados. Assim, similar à avaliação da aprendizagem de conceitos de algoritmos e programação por meio da análise do código do app, visa-se também avaliar a criatividade de produto baseado nas características do app (ALVES et al., 2020).

Medir a criatividade é uma tarefa complexa (ALVES et al., 2022). Para isso, primeiramente é necessário entender o conceito de criatividade de produto, que se refere aos resultados tangíveis do processo criativo (ou seja, obras de arte, livros, programas de computador, etc) (RANJAN et al., 2018). A avaliação desse tipo de criatividade pode se dar por meio da avaliação das características que a compõem. Embora tais características não sejam consensuais, há um conjunto delas que é considerado o conjunto principal de características de

um produto criativo para diversos autores (RANJAN et al., 2018). Seguindo uma definição comum da criatividade de produto, é possível decompô-la em três aspectos: originalidade, utilidade e qualidade.

Refinando assim o conceito de criatividade de produto, é possível identificar diversas dimensões da avaliação da originalidade de aplicativos móveis. O escopo da originalidade pode ser dividido em conteúdo e funcionalidade. O foco do presente trabalho se dá na originalidade em relação ao conteúdo, como parte do escopo. A originalidade de escopo/conteúdo é definida como o grau de diferença de um tópico de app (receitas ou notícias, por exemplo) em relação aos tópicos de apps apresentados em um universo de referência.

O conteúdo de um aplicativo móvel pode ser identificado por humanos por meio da análise de suas características e funcionalidades, a fim de identificar a categoria e/ou tópico do aplicativo, podendo ser geral ou mais específico. O tópico também é muitas vezes especificado pelos criadores nas lojas de apps, categorizando-os como saúde, educação, entre outros.

O desafio surge quando se tenta realizar a detecção de tais tópicos de forma automatizada. Para este fim, uma das alternativas é aplicar técnicas de Processamento de Linguagem Natural (NLP - Natural Language Processing) a partir do texto extraído dos aplicativos. Nesse contexto, observa-se que muitas vezes são utilizados nomes de objetos específicos e que representam o mesmo tipo de entidade. Caso dentro do texto existam as palavras ‘maçã’, ‘banana’ e ‘melancia’, há uma indicação que o app é relacionado a ‘frutas’, por exemplo. Caso também existam dentro do texto as palavras ‘arroz’ e ‘feijão’, o tópico mais adequado seria ‘alimentos’.

Tipicamente, para o reconhecimento desse tipo de entidades, é adotado o NER (Named Entity Recognition - Reconhecimento de Entidade Nomeada). O NER pode ser definido como uma técnica de classificação de texto não-estruturado em entidades nomeadas, sendo utilizado dentro da área de NLP (AMARAL, 2013). Mesmo já existindo pesquisas adotando NER também para textos em português (AMARAL, 2013), ainda há poucas pesquisas voltadas a este tipo de detecção com base em textos curtos extraídos de apps.

Desta forma, o presente trabalho visa contribuir para a criação de um modelo de NER que identifique entidades a partir dos textos extraídos de apps criados com App Inventor, a fim de melhorar uma avaliação de originalidade do escopo/conteúdo de tais apps.

2. Fundamentação Teórica

2.1 Elementos textuais de apps criados com App Inventor

2.1.1 App Inventor

O MIT App Inventor é uma aplicação web open source criada pelo Google e atualmente mantida pelo MIT - Massachusetts Institute of Technology (PATTON et al., 2019). Seu intuito é o de fornecer uma maneira simples e prática para o desenvolvimento de aplicações móveis por meio de uma interface de programação baseada em blocos. Esse tipo de interface permite com que o usuário consiga criar aplicativos sem ter conhecimento prévio de nenhuma

linguagem de programação textual, podendo assim ser utilizado por crianças ou pessoas de fora da área da computação.

O App Inventor possui recursos para o desenvolvimento dos aplicativos tanto na parte funcional quanto na parte de design de interfaces de usuário. É possível que o usuário realize o compartilhamento de seu app exportando um arquivo com extensão .aia do mesmo. Esse arquivo pode então ser importado para o App Inventor por qualquer usuário que o possua. É possível também exportar um executável .apk do app, permitindo sua instalação diretamente em um dispositivo móvel. Além disso, é possível publicar um projeto na galeria do App Inventor, tornando-o disponível para a comunidade.

2.1.2 Elementos textuais dos apps

Para identificar os elementos textuais presentes em um app criado com App Inventor e assim tornar possível sua extração, deve-se primeiramente entender a estrutura de um arquivo .aia. A Figura 1 mostra a estrutura deste tipo de arquivo, que é em essência a mesma que um arquivo .zip, contendo diversos diretórios que por sua vez contém outros diretórios e arquivos dentro deles.

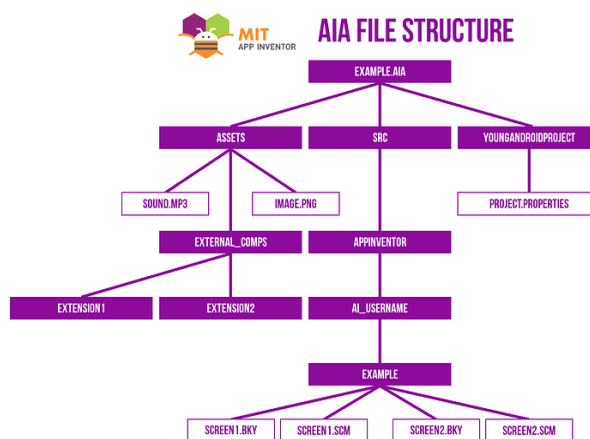


Figura 1. Estrutura de um arquivo .aia (App Inventor Community, 2019).

Dentro do diretório src/appinventor/[ai_username]/[example] a existência de uma dupla de arquivos para cada uma das telas do app, um deles com a extensão .scm e outro com a extensão .bky.

O arquivo .scm é um arquivo que contém as informações de layout sobre a tela do app relacionada, escrito no formato de um arquivo JSON. O arquivo .bky, por sua vez, possui sua estrutura em formato XML, contendo as informações sobre os blocos lógicos da respectiva tela. Em ambos os arquivos é possível identificar os elementos textuais presentes no app analisando seu conteúdo.

No caso do arquivo .scm, é possível observar uma lista de objetos do tipo “\$Components”. Analisando os objetos, percebe-se que o valor presente no campo “Text” do componente representa um elemento textual presente na tela do app referente ao componente identificado,

como também pode ser visto na Figura 5. Já no caso do arquivo .bky, o elemento textual está presente no valor da tag field, já que o atributo name contém o valor “Text”, como visto na Figura 6.

Analisando os elementos textuais em apps do App Inventor, identificam-se os campos presentes em cada componente que representam algum elemento textual (Tabela 1). Dessa maneira, pode-se extrair o texto de uma grande gama de apps a partir de seus arquivos .aia analisando estes campos, podendo assim ser feita uma análise sobre estes textos e então inferir tópicos de apps a partir deles.

Tabela 1. Propriedades contendo elementos textuais separadas por componente

Tipo	Componente	Propriedade
SCREEN		
		<i>Title</i>
		<i>AboutScreen</i>
DESIGNER		
<i>User Interface</i>	<i>Button</i>	<i>Text</i>
	<i>Checkbox</i>	<i>Text</i>
	<i>DatePicker</i>	<i>Text</i>
	<i>Label</i>	<i>Text</i>
	<i>ListPicker</i>	<i>Text e ElementsFromString</i>
	<i>PasswordTextBox</i>	<i>Text e Hint</i>
	<i>Spinner</i>	<i>ElementsFromString e Prompt</i>
	<i>Switch</i>	<i>Text</i>
	<i>TextBox</i>	<i>Text e Hint</i>
	<i>TimePicker</i>	<i>Text</i>
<i>Media</i>	<i>ImagePicker</i>	<i>Text</i>
<i>Maps</i>	<i>Marker</i>	<i>Description</i>
<i>Social</i>	<i>ContactPicker</i>	<i>Text</i>
	<i>EmailPicker</i>	<i>Text</i>
	<i>PhoneNumberPicker</i>	<i>Text</i>
	<i>Texting</i>	<i>Message</i>
<i>Connectivity</i>	<i>ActivityStarter</i>	<i>ResultName</i>
BLOCKS		
<i>Text</i>	<i>String</i>	<i>Text</i>

2.2 Análise textual automatizada com NER

Processamento de Linguagem Natural (Natural Language Processing - NLP), é a área da ciência da computação que une as áreas da inteligência artificial e da linguística (NADKARNI et al., 2011). O NLP visa estudar, compreender e propor técnicas e soluções computacionais para problemas relacionados às linguagens naturais compreendidas pelos seres humanos. Exemplos de aplicações de NLP incluem a tradução automática, assistentes inteligentes, mecanismos de busca, análise textual, entre outros.

O Reconhecimento de Entidade Nomeada (NER) (AMARAL, 2013), é uma técnica amplamente utilizada no NLP cujo objetivo é o reconhecimento e identificação automática de entidades nomeadas presentes em um texto não estruturado. Essas entidades são classificadas dentro de um conjunto de categorias, sendo elas pré-definidas pelo modelo de NER sendo utilizado. Desta maneira, a entrada para um sistema utilizando NER é um texto não estruturado, e sua saída é um conjunto de texto classificado e estruturado entre entidade e categoria (FONSECA et al., 2015).

Usando o texto “João mora em Florianópolis e trabalha no Google” como exemplo, é possível efetuar a extração das entidades nomeadas do mesmo e assim categorizá-las, podendo fazê-lo de maneira genérica ou mais específica. Um possível resultado para esta extração e classificação de categorias do texto de exemplo é apresentado na Tabela 2.

Tabela 2. Exemplo de extração de entidades nomeadas

Entidade	Categoria
João	Pessoa
Florianópolis	Local
Google	Organização

Nota-se que a entidade “Florianópolis” poderia também ser definida como “Cidade” ou “Município”, dependendo do modelo de NER a ser aplicado no texto de entrada.

Algumas das ferramentas mais utilizadas para o uso das técnicas de NER são frameworks de NLP acessíveis na linguagem de programação Python. Com eles, é possível fornecer um texto de entrada em linha de código e obter a tabela de entidades como saída a partir do modelo de NER utilizado. Dentre os frameworks existentes, pode-se destacar os frameworks Spacy (EXPLOSION, 2015), NLTK (NLTK PROJECT, 2001) e Flair (FLAIRNLP, 2018).

```
1 import spacy
2
3 nlp = spacy.load("pt_core_news_sm")
4
5 doc = nlp("João mora em Florianópolis e trabalha no Google")
6
7 for X in doc.ents:
8     print("Entidade:", X.text, "| Categoria:", X.label_)
9
↳ python3 exemploSpacy.py
Entidade: João | Categoria: PER
Entidade: Florianópolis | Categoria: LOC
Entidade: Google | Categoria: ORG
```

Figura 2. Exemplo de código e execução do Spacy

A Figura 2 mostra um exemplo de um código em Python utilizando o framework Spacy. Nele, é possível observar, na linha 3, que está sendo carregado o modelo “pt_core_news_sm”, um modelo pré-treinado do Spacy que reproduz funcionalidades referentes à língua portuguesa, incluindo NER, com algumas entidades já definidas. Na linha 5, é fornecido o texto de entrada para o modelo, e em seguida é feito um loop nas entidades identificadas a fim de realizar a impressão no console das mesmas, assim como em suas categorias. A parte de baixo da Figura 2 mostra o resultado final da aplicação do NER no texto de entrada “João mora em Florianópolis e trabalha no Google”, onde foram identificadas as entidades João - PER (Pessoa), Florianópolis - LOC (Localização) e Google (ORG - Organização).

As categorias atualmente reconhecidas pelos modelos base de NER dos frameworks analisados por este trabalho são comparados na Tabela 3, tanto em português quanto em inglês. Observa-se que esses modelos base possuem uma quantidade baixa de categorias, especialmente o modelo em português do Spacy, que é o que mais se encaixa no âmbito deste trabalho.

Tabela 3. Categorias reconhecidas pelos frameworks

	Spacy	NLTK	Flair
Português	LOC, MISC, ORG, PER	-	-
Inglês	CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART	ORGANIZATION, PERSON, LOCATION, DATE, TIME, MONEY, PERCENT, FACILITY, GPE	CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON, PRODUCT, QUANTITY, TIME, WORK_OF_ART

Visto a pouca quantidade de categorias, é necessário treinar estes modelos base para que eles comportem também as categorias referentes aos tópicos de aplicativos móveis definidos.

Para treinar os modelos do Spacy, deve-se primeiramente utilizar a função `get_pipe('ner')` do modelo carregado, a fim de recuperar a pipeline de NER do modelo. Em seguida, pode-se utilizar a função `add_label()` a partir deste elemento, passando como parâmetro uma nova categoria não presente no modelo, como `IDIOMA`, por exemplo, visando incluí-la e assim possibilitando identificação de entidades da nova categoria. Por fim, deve-se utilizar a função `update()` para treinar o modelo, fornecendo como dado de entrada da função uma lista de tuplas. O primeiro parâmetro da tupla é uma frase contendo exemplos de instâncias de entidades que o usuário pretende reconhecer. O segundo parâmetro é um objeto contendo o campo “`entities`”, cujo valor é uma nova lista de tuplas. Cada valor desta nova lista de tuplas identifica a posição inicial, posição final e por fim a categoria a qual pertence tal entidade identificada. No segundo exemplo da Figura 10, a frase de entrada é “português, inglês, espanhol”. A palavra “português” é em seguida identificada pelo trecho `(0,8,'IDIOMA')`,

sendo 0 e 8 as posições iniciais e finais da palavra na frase, e “IDIOMA” a categoria identificada.

3. Estado da Arte

3.1 Definição do protocolo de revisão

O objetivo desta revisão do estado da arte é responder a seguinte pergunta de pesquisa: Quais abordagens existem para automaticamente analisar o texto em português de aplicativos (criados com App Inventor) usando NER?

Esta pergunta de pesquisa é refinada nas seguintes questões de análise:

- AQ1. Quais abordagens existem e quais suas características?
- AQ2. Quais técnicas/bibliotecas são adotadas na análise?
- AQ3. Qual a qualidade (precisão) das abordagens?

Critérios de inclusão/exclusão: Em conformidade com o foco da pesquisa, são definidos os seguintes critérios de inclusão e exclusão de artigos:

- São considerados somente artigos que apresentam abordagens para a detecção automatizada de tópicos utilizando NER com base no código de aplicativos para dispositivos móveis;
- São excluídos artigos que não sejam baseados em código, como descrição do aplicativo, resenhas, dados de uso, etc, pelo fato que no contexto do App inventor não existem essas informações;
- São considerados apenas pesquisas publicadas nos últimos 10 anos desde 2012 levando em consideração o avanço recente especificamente em relação a aplicativos móveis;
- São considerados apenas artigos que apresentam informações substanciais, permitindo a extração de informações relevantes sobre as questões de análise. Portanto, artigos com resumo ou apenas uma página são excluídos;
- São considerados somente artigos escritos em português ou inglês.

Fontes: Foram pesquisados nos principais bancos de dados e bibliotecas digitais no campo da computação, incluindo as Bibliotecas Digitais ACM, a IEEE Xplore e Scopus com acesso por meio do Portal Capes. A pesquisa também foi feita dentro do site Google Scholar, para complementar a busca.

Critérios de qualidade: São considerados apenas artigos que apresentem informações substanciais para se extrair informações referente às perguntas de análise.

Termos de busca: Com base na questão de pesquisa, várias pesquisas informais foram realizadas para calibrar o string de busca, identificando termos de pesquisa relevantes e seus sinônimos. Foram utilizados sinônimos para minimizar o risco de omitir trabalhos relevantes.

String de busca: Após a definição dos termos e seus sinônimos, definiu-se o string de busca padrão a ser aplicado nas bases de dados:

("NER" OR "Named entity recognition" OR "Reconhecimento de entidade nomeada" OR "Reconhecimento de entidades nomeadas") AND ("mobile application" OR "app" OR "ios" OR "android" OR "app inventor" OR "Aplicativo móvel")

3.2 Execução da busca

A busca foi realizada em Maio de 2022 pelo autor do trabalho e revisada pelas (co-)orientadoras. A busca inicial resultou em 506 artigos (Tabela 4).

Tabela 4. Número de artigos identificados por repositório e por fase de seleção.

Fonte	Nº de resultados da busca	Nº de resultados analisados	Nº de documentos potencialmente relevantes	Nº de documentos relevantes
ACM	0	0	0	0
IEEE	4	4	2	0
SCOPUS	36	36	5	0
Google Scholar	466	200	9	0
Total				0

Uma vez executadas as buscas, foram selecionados artigos potencialmente relevantes por meio da análise do título e do resumo dos trabalhos, a fim de confirmar a relevância dos mesmos levando em consideração os critérios de inclusão e exclusão definidos previamente. Esta etapa foi executada com todos os trabalhos encontrados, exceto os resultados do Google Scholar, dos quais foram analisados os 200 resultados mais relevantes.

Em seguida, foram analisados os textos na íntegra dos trabalhos potencialmente relevantes. Mesmo tendo encontrado artigos relevantes referentes à classificação de tópicos de aplicativos, como por exemplo *Classifying Mobile Applications Using Word Embeddings* (EBRAHIMI et al., 2022) e *Enriching iTunes App Store Categories via Topic-Modeling* (VAKULENKO et al., 2014), estes artigos não foram considerados relevantes no contexto deste trabalho, tanto por não fazerem uso do NER quanto por não se basearem no código dos apps, mas em suas descrições nas lojas de aplicativos.

Desta maneira, após análise de todos os trabalhos potencialmente relevantes, não foi identificado nenhum trabalho que atendesse a todos os critérios de inclusão ou exclusão definidos previamente. Assim, como resultado desse mapeamento sistemático, não foi encontrado nenhum trabalho existente atualmente voltado à análise automática do texto de aplicativos usando NER, tampouco abordagens voltadas a apps criados com App Inventor.

4. Abordagem de NER para classificação de tópicos de apps criados com App Inventor

4.1 Análise de requisitos

O objetivo do presente trabalho é contribuir para a classificação de tópicos de apps a partir da análise de texto extraído dos aplicativos, focando na classificação de entidades utilizando NER.

A análise dos requisitos do modelo de NER a ser construído começa pela identificação de entidades relevantes. Analisando o nome do app e seu conteúdo textual extraído de um universo de 2.806 aplicativos criados com App Inventor, foram identificadas entidades relevantes a serem extraídas usando NER (Tabela 5).

Tabela 5. Identificação de entidades relevantes no contexto de apps

Tópico	Entidade	Exemplos de palavras
Comida	FOOD	leite massas óleo café miúdos manga
Animal	ANIMAL	urso cachorro dog elefante animais hipopótamo cavalo
Localização	LOC	belo horizonte salvador fortaleza porto velho
Nome de pessoa	PER	joão maria pedro ana
Disciplina escolar/acadêmica	COURSE	matérias matemática física programação ciências inglês geografia arquitetura história artes
Planta	PLANT	arruda planta alecrim louro árvore palmeira flor
Idioma	LANG	inglês alemão português espanhol francês italiano
Número	NUM	1 2 3 4 5 6 7 8 9 0
Operação Matemática	OP	+ - / * ^ =
Organização	ORG	google facebook apple microsoft amazon

Dessa maneira, o requisito referente ao presente trabalho é que o modelo de NER a ser desenvolvido deve ser capaz de identificar as entidades especificadas na Tabela 5 a partir do texto extraído de arquivos .aia de apps criados com App Inventor.

4.1 Preparação do conjunto de dados

A fim de preparar um conjunto de dados a ser utilizado no treinamento do modelo, foi necessário coletar uma quantidade de conteúdo textual para ser anotado com entidades. Assim, foram selecionados 100 aplicativos criados com App Inventor, e os elementos textuais presentes nesses aplicativos foram recuperados e anotados com entidades. Os elementos textuais foram obtidos realizando uma extração diretamente do código dos aplicativos a partir dos seus arquivos .aia, conforme apresentado na seção 2.1.2, em que os componentes levados em consideração para a extração foram aqueles apresentados na Tabela 1, sendo esses os componentes que de fato apresentam conteúdo textual em seu código. Assim, o conteúdo

desses componentes foi extraído em sua totalidade, ou seja, sem passar por um pré-processamento de filtragem de termos repetitivos ou irrelevantes.

Os 100 apps utilizados no processo foram selecionados pelo autor a partir de uma planilha contendo um universo de 2.806 apps criados com App Inventor. Seus códigos foram obtidos tanto da galeria de aplicativos, disponível diretamente no site do App Inventor (<https://gallery.appinventor.mit.edu/featured/>), quanto da iniciativa Computação na Escola, em que os apps foram desenvolvidos por alunos do ensino básico e seus códigos foram recuperados por educadores da iniciativa. O critério de seleção dos apps utilizados foi a variedade e quantidade de entidades definidas na Tabela 5 visualmente presentes nos elementos textuais dos mesmos, a fim de realizar um balanceamento coerente de entidades presentes no conjunto de dados e assim obter melhores resultados no treinamento do modelo.

Além disso, em adição aos textos extraídos de apps, foram adicionados dois textos fabricados pelo autor a fim de treinar as entidades “Plant” e “Lang”. Isso foi feito com o intuito de fornecer ao modelo exemplos mais variados de tais entidades, no caso uma quantidade maior de idiomas e tipos de plantas, pois dentro do texto dos apps essa variedade não estava presente em tão grande número.

A etapa seguinte da preparação dos dados foi a anotação das entidades dos textos extraídos. Para isso, utilizou-se a ferramenta NER Annotator (TECOHOLIC, 2022). Dentre as diversas ferramentas gratuitas testadas para esta etapa, ela mostrou uma boa eficiência nos quesitos praticidade e usabilidade. Essa ferramenta recebe um arquivo .txt como dado de entrada e apresenta uma interface gráfica para que o usuário selecione cada termo presente no texto e o classifique como uma entidade (Figura 3). Dessa maneira, foram separados dois arquivos .txt para serem anotados, sendo um para o conjunto de dados de treinamento e outro para os dados de validação.

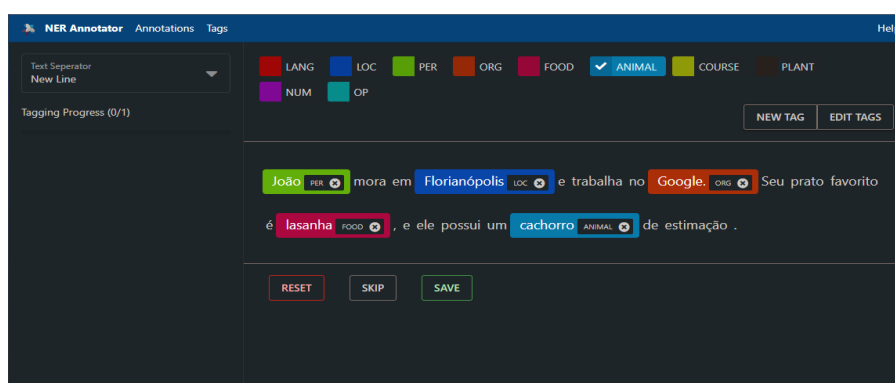


Figura 3. Exemplo de anotações com o NER Annotator

É importante ressaltar que as anotações dos termos com entidades foram feitas integralmente pelo autor do trabalho, estando portanto limitadas ao viés do mesmo sobre a classificação de tais termos. Por consequência, uma vez que o modelo foi treinado com um conjunto de dados

contendo esse viés, é esperado que o modelo também o contenha quando executado para automaticamente reconhecer entidades em textos.

Ao final da anotação das entidades dos arquivos, o texto anotado de cada um dos arquivos foi exportado para um arquivo .json. Posteriormente, ambos os arquivos .json foram fornecidos à biblioteca de treinamento dos modelos de NER do Spacy. Deste conjunto de dados, 80 aplicativos foram utilizados no conjunto de treinamento e 20 no conjunto de validação.

4.3 Treinamento

Para a execução do treinamento, foi utilizado o ambiente baseado em nuvem Google Colab, que permite a execução de jupyter notebooks com o código de treinamento de forma gratuita.

O pacote escolhido para executar o treinamento foi o Spacy, que fornece a possibilidade de treinar entidades customizadas. Durante os testes realizados, observou-se que o modelo padrão do Spacy em português “pt_core_news_sm” sofria um processo de “desaprendizagem” de suas entidades base (LOC, PER e ORG) caso os dados de entrada fornecidos para o treinamento do modelo não possuíssem tais entidades em suas anotações. Desse modo, optou-se por realizar o treinamento do modelo contendo anotações de todas as entidades especificadas na Tabela 5 no conjunto de dados preparado.

Com os conjuntos de dados já construído nos arquivos .json, como descrito na seção 4.2, foi necessário realizar a conversão dos arquivos para a extensão .spacy, formato reconhecido para treinamento pelo pacote Spacy. Para isso, foi construída uma função de conversão que foi chamada para ambos os arquivos, gerando os arquivos train.spacy e valid.spacy.

Em seguida, foi construído um arquivo de configuração config.cfg, que define os parâmetros utilizados para o treinamento do modelo. Os parâmetros utilizados, que foram os valores padrão já fornecidos pelo pacote Spacy, podem ser observados na Tabela 6, assim como detalhes de execução e avaliação do modelo. Em código, o comando de execução do treinamento pode ser observado na Figura 4, bem como detalhes de cada iteração.

```

python -m spacy train /content/config.cfg --output /content/models/output --paths.train /content/train.spacy --paths.dev /content/valid.spacy

2022-10-25 21:04:12.959342: E tensorflow/stream_executor/cuda/cuda_driver.cc:271] failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
✓ Created output directory: /content/models/output
i Saving to output directory: /content/models/output
i Using CPU

===== Initializing pipeline =====
[2022-10-25 21:04:14,265] [INFO] Set up nlp object from config
INFO:spacy:Set up nlp object from config
[2022-10-25 21:04:14,280] [INFO] Pipeline: ['tok2vec', 'ner']
INFO:spacy:Pipeline: ['tok2vec', 'ner']
[2022-10-25 21:04:14,280] [INFO] Resuming training for: ['ner', 'tok2vec']
INFO:spacy:Resuming training for: ['ner', 'tok2vec']
[2022-10-25 21:04:14,292] [INFO] Created vocabulary
INFO:spacy:Created vocabulary
[2022-10-25 21:04:14,293] [INFO] Finished initializing nlp object
INFO:spacy:Finished initializing nlp object
[2022-10-25 21:04:14,293] [INFO] Initialized pipeline components: []
INFO:spacy:Initialized pipeline components: []
✓ Initialized pipeline

===== Training pipeline =====
i Pipeline: ['tok2vec', 'ner']
i Initial learn rate: 0.001
E # LOSS TOK2VEC LOSS NER ENTS_F ENTS_P ENTS_R SCORE
---
0 0 0.00 88.22 9.88 10.65 9.22 0.10
2 200 0.00 13430.76 52.07 60.84 45.50 0.52
4 400 0.00 7518.37 62.99 69.15 57.83 0.63
7 600 0.00 6542.38 70.98 80.31 63.59 0.71
10 800 0.00 4176.18 79.46 81.62 77.41 0.79
13 1000 0.00 2866.07 81.37 87.41 76.11 0.81
16 1200 0.00 2299.16 85.95 91.96 80.67 0.86
20 1400 0.00 1744.57 86.76 91.07 82.84 0.87
24 1600 0.00 1179.10 87.21 91.33 83.44 0.87
27 1800 0.00 1108.33 87.56 90.96 84.41 0.88
32 2000 0.00 744.87 87.09 91.08 83.44 0.87
36 2200 0.00 816.22 89.09 93.06 85.45 0.89
41 2400 0.00 852.28 87.94 90.91 85.16 0.88
45 2600 0.00 624.74 87.69 90.09 85.40 0.88
50 2800 0.00 513.14 88.15 91.73 84.83 0.88
56 3000 0.00 557.32 87.89 90.85 85.11 0.88
61 3200 0.00 552.05 88.77 93.64 84.38 0.89
67 3400 0.00 570.89 88.86 93.03 85.06 0.89
73 3600 0.00 407.40 88.15 91.14 85.35 0.88
80 3800 0.00 537.49 88.48 91.69 85.48 0.88
✓ Saved pipeline to output directory
/content/models/output/model-last

```

Figura 4. Execução do treinamento do modelo

Tabela 6. Dados gerais do modelo de NER

Modelo de ML	
Nome do modelo	Classificação de entidades presentes no texto de apps
Data	23/10/2022
Versão	0.1
Objetivo do modelo de ML	
Tarefa	Classificar entidades a partir de uma entrada textual
Contexto de uso	O modelo é utilizado no contexto de ensino na Educação Básica para realizar a classificação de entidades presentes no texto de apps criados com App Inventor, a fim de auxiliar na avaliação de criatividade destes apps.
Público alvo	Estudantes do ensino fundamental e médio (13+ anos)
Riscos	O risco de não classificar corretamente uma ou mais entidades dentro do texto de um app é um possível equívoco na avaliação da criatividade do app.
Tipo da tarefa	Single-label classificação de entidades
Entidades	Food: Tipos de comidas Animal: Tipos de animais Plant: Tipos de plantas Loc: Endereços e localizações Per: Nomes de pessoas Org: Nomes de empresas e organizações Op: Operações matemáticas Num: Números Lang: Idiomas Course: Disciplinas escolares e acadêmicas

Conjunto de dados	
Descrição dos dados	Conteúdo textual presente em apps criados com App Inventor, contendo todos os tipos das entidades previamente mapeadas, prontas para classificação.
Origem dos dados	Texto extraído do código (arquivo <i>.aia</i>) de aplicativos a partir dos dos elementos da tela e blocos lógicos, conforme especificado na seção 2.
Quantidade total de dados	Textos extraídos de 100 apps
Distribuição dos dados por entidade	Food: 516 Animal: 301 Plant: 162 Loc: 776 Per: 442 Org: 598 Op: 110 Num: 671 Lang: 323 Course: 118
Rotulação	Estudante do curso de graduação em Ciências da Computação.
Tipos de aumento de dados aplicados	Inclusão de elementos textuais das entidades sub-representadas Lang e Plant
Tamanho do batch	1.000
Separação do conjunto de dados	80% para treinamento (80) 20% para validação (20)
Treinamento - Transfer learning	
Tipo de modelo	spacy.Corpus.v1
Pipeline	tok2vec, ner
Modelo fonte	pt_core_news_sm
Quantidade de épocas	80
Taxa de aprendizagem	1e-3
Loss/taxa de erro por época	Melhor modelo encontrado na epoch 36
Avaliação - Transfer learning (resultados da validação)	
Precisão	0.9305668565
Recall	0.8545454545
F1 score	0.8909374189
Valores por entidade	Ver Tabela 8
Limitações e considerações éticas	

Limitações	O modelo é limitado a reconhecer apenas as entidades propostas neste trabalho
Implantação	
Formato da exportação do modelo	Diretório: ner —cfg —model —moves tok2vec —cfg —model vocab —key2row —lookups.bin —strings.json —vectors —vectors.cfg config.cfg meta.json tokenizer
Referências	
Conjunto de dados	https://codigos.ufsc.br/gqs/app-inventor-ner/-/tree/main/dataset
Modelo	https://codigos.ufsc.br/gqs/app-inventor-ner/-/tree/main/models/output/model-best

Para avaliar o desempenho do modelo, tipicamente analisa-se o resultados das métricas de classificação de machine learning chamadas de precisão, recall e F1 score (BROWNLEE, 2020). No caso do presente modelo, essas métricas são calculadas automaticamente pelo pacote Spacy. Para isso, toma-se como base os textos do conjunto de dados de validação e o modelo realiza a classificação automática desses textos. Em seguida, compara-se a classificação de entidades obtidas pelo modelo com a classificação feita previamente pelo usuário, categorizando cada palavra dentro do seguinte conjunto:

- TP (True Positive) - Entidade corretamente classificada.
- TN (True Negative) - Entidade corretamente não classificada.
- FP (False Positive) - Entidade falsamente classificada.
- FN (False Negative) - Entidade falsamente não classificada.

A partir dessas classificações dos resultados as medidas de desempenho são calculadas:

- $\text{Precisão} = \text{TP} / (\text{TP} + \text{FP})$ - Indica o quanto o modelo é capaz de classificar entidades com o mínimo possível de falsos positivos.
- $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$ - Indica o quanto o modelo é capaz de classificar entidades com o mínimo possível de falsos negativos.
- $\text{F1 Score} = 2 * \text{Precisão} * \text{Recall} / (\text{Precisão} + \text{Recall})$ - Média harmônica entre Precisão e Recall.

O resultado dessas medidas de desempenho é apresentado na Tabela 6, no tópico Avaliação - Transfer learning (resultados da validação), e na Tabela 8, onde os resultados são apresentados para cada uma das entidades separadamente.

A fim de efetivamente determinar o grau de desempenho do modelo, foram definidas interpretações a partir de uma escala ordenada de 4 níveis (ALLWRIGHT, 2022), definidas de acordo com o resultado do F1 score obtido pelo modelo. Nessa escala, o maior nível possível a ser alcançado por um modelo é interpretado como “Muito bom” e ocorre se o modelo obtiver um F1 score maior que 0.9. Da mesma maneira, o menor nível possível é interpretado como “Ruim” e ocorre se o modelo obtiver um F1 score menor que 0.5. Essa escala de avaliação de modelos é apresentada na Tabela 7.

Tabela 7. Escala de avaliação de modelos de acordo com o F1 Score (ALLWRIGHT, 2022)

F1 score	Interpretação
> 0.9	Muito bom
0.8 - 0.9	Bom
0.5 - 0.8	OK
< 0.5	Ruim

Tabela 8. Resultados do treinamento do modelo por entidade

Entidade	Precisão	Recall	F1 score	Avaliação
FOOD	0.967032967	0.8527131783	0.9062821833	Muito bom
ANIMAL	0.8754716981	0.7707641196	0.8197879859	Bom
LOC	0.9718670077	0.9793814433	0.9756097561	Muito bom
PER	0.7421875	0.4298642534	0.5444126074	OK
COURSE	0.950617284	0.6525423729	0.7738693467	Bom
PLANT	0.961038961	0.9135802469	0.9367088608	Muito bom
LANG	0.9813664596	0.9783281734	0.9798449612	Muito bom
NUM	0.9074585635	0.9791356185	0.9419354839	Muito bom
OP	0.9081632653	0.8240740741	0.8640776699	Bom
ORG	0.9490909091	0.872909699	0.9094076655	Muito bom
Total	0.9214294615	0.8253293179	0.8651936521	Bom

É possível observar que o modelo (total) foi definido com “Bom”, de acordo com seu F1 Score de ≈ 0.86 . A entidade com o melhor resultado de F1 Score foi a entidade LANG, representando entidades de nomes de idiomas, avaliada como “Muito bom” e com um F1 Score de ≈ 0.98 . A entidade que obteve o pior resultado foi a entidade PER, representando nomes de pessoas, avaliada como “OK” e com um F1 Score de ≈ 0.54 . O que mais afetou o resultado do F1 Score da entidade PER foi o baixo resultado de seu recall (≈ 0.43), ou seja, o

modelo estaria falhando em reconhecer algumas entidades PER que deveriam ser reconhecidas. Assume-se que o desempenho inferior dessa entidade em relação às demais poderia ser aprimorado caso o modelo conhecesse uma quantidade maior de dados de treinamento e validação desta entidade.

Ademais, a avaliação do restante das entidades foi positiva, interpretando os resultados das medidas como “Bom” ou “Muito bom”.

4.4 Teste

A fim de realizar um teste visual da capacidade e acurácia do modelo, foram extraídos conteúdos textuais de 10 aplicativos a partir de seus arquivos .aia, assim como feito na seção 4.2. O texto extraído foi integralmente retirado do código dos aplicativos, sem passar por nenhum tipo de pré-processamento para remover caracteres especiais ou palavras consideradas irrelevantes no contexto de aplicativos. Os 10 apps escolhidos para essa etapa não pertenciam aos conjuntos de dados de treinamento e validação construídos na seção 4.2, a fim de se realizar o teste com conteúdos textuais totalmente alheios ao modelo.

O teste foi realizado visando calcular manualmente a precisão, recall e F1 score (BROWNLEE, 2020) de cada tipo de entidade. Para isso, o texto dos apps foi analisado pelo autor e a cada termo encontrado foi alocado um valor (TP, TN, FP e FN) de acordo com o resultado de sua classificação.

O resultado dessa classificação de entidades é apresentado na Tabela 9, mostrando a quantidade de palavras identificadas pelo autor como TP, TN, FP e FN para cada uma das entidades do modelo. A partir do resultado das classificações, as medidas de desempenho foram calculadas e classificadas de acordo com seus resultados. Esses resultados são apresentados na Tabela 10.

Tabela 9. Análise de desempenho do modelo

FOOD				ANIMAL				LOC				PER				COURSE				PLANT				LANG				NUM				OP				ORG							
TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
39	2077	0	38	3	2071	2	1	74	1966	6	31	28	2036	0	13	9	2035	3	30	5	2066	0	6	22	2052	2	1	35	2040	2	0	86	1918	5	68	42	1990	22	23				

Tabela 10. Avaliação de precisão, recall e F1 score

Entidade	Precisão	Recall	F1 score	Avaliação
FOOD	1	0.5064935065	0.6724137931	OK
ANIMAL	0.6	0.75	0.6666666667	OK
LOC	0.925	0.7047619048	0.8	Bom
PER	1	0.6829268293	0.8115942029	Bom
COURSE	0.75	0.2307692308	0.3529411765	Ruim

PLANT	1	0.4545454545	0.625	OK
LANG	0.9166666667	0.9565217391	0.9361702128	Muito bom
NUM	0.9459459459	1	0.9722222222	Muito bom
OP	0.9450549451	0.5584415584	0.7020408163	OK
ORG	0.65625	0.6461538462	0.6511627907	OK
Total	0.8738917558	0.649061407	0.7190211881	OK

É possível observar comparando os resultados da Tabela 8 e da Tabela 10 que o F1 score do modelo no teste executado se mostrou inferior ao resultado da validação calculado pelo Spacy. No teste, o modelo obteve um F1 score de ≈ 0.72 , sendo esse considerado um grau de desempenho “OK”, enquanto a validação do Spacy resultou em ≈ 0.86 , sendo esse um grau de desempenho “Bom”.

Essa diferença se deu pelos resultados de F1 score obtidos nas entidades FOOD, ANIMAL, COURSE, PLANT, OP e ORG, que foram inferiores aos resultados da validação do Spacy, principalmente por conta dos resultados de recall, que também foram inferiores. Ou seja, no teste realizado com os 10 apps selecionados, o modelo teve mais dificuldade para reconhecer essas entidades do que nos apps do conjunto de validação fornecido ao Spacy. No teste, a entidade COURSE, que representa disciplinas escolares e acadêmicas, foi a única classificada como um desempenho “Ruim”, obtendo um F1 score de ≈ 0.35 .

Nota-se, entretanto, que as entidades LANG e NUM não obtiveram diferenças em suas avaliações, continuando com a classificação de desempenho “Muito bom” em relação à validação do Spacy. Houve, ainda, o caso da entidade PER, que no teste executado foi a única entidade a obter uma classificação superior em relação à validação do Spacy, obtendo um F1 score de ≈ 0.81 e portanto indicando um grau de desempenho “Bom”.

É possível concluir a partir dos testes executados que o modelo possui uma variação de desempenho ao realizar a classificação automática de entidades em textos retirados de apps, podendo ser mais ou menos efetivo dependendo do texto fornecido ao mesmo.

Assume-se que essa volatilidade nos resultados de desempenho, principalmente o baixo desempenho da entidade COURSE, tenha sido afetado pela diferença dos elementos textuais a serem analisados pelo modelo. Ou seja, o modelo pode ser mais efetivo em reconhecer entidades em alguns casos, como no caso dos elementos textuais dos apps de validação, e menos efetivo em outros, como no caso dos elementos textuais dos apps de teste. Assume-se que caso o modelo seja treinado com uma maior quantidade de elementos textuais de apps, os resultados de suas métricas se tornem mais precisos e consistentes.

Há, ainda, a questão do balanceamento das entidades do conjunto de dados utilizado no modelo. Esse balanceamento é apresentado no tópico Distribuição dos dados por entidade na Tabela 6, e nele é possível observar que as entidades PLANT, OP e COURSE se apresentam

em uma quantidade consideravelmente menor em relação ao restante das entidades do modelo. Nesse sentido, é possível que o desempenho inferior da entidade COURSE tenha sido causado por esta razão, uma vez que é recomendado que o modelo possua um balanceamento igualitário entre todas as entidades.

Apesar dos problemas analisados, os resultados gerais apresentados foram positivos, uma vez que o desempenho do modelo em sua totalidade foi classificado como “Bom” e “OK” nas duas ocasiões em que suas medidas de desempenho foram calculadas. Portanto, é possível afirmar que o modelo possui efetividade em classificar as entidades propostas neste trabalho, ainda que essa efetividade seja volátil.

4.5 Discussão e trabalhos futuros

A partir dos resultados de treinamento e testes do modelo, e conseqüentemente do objetivo geral do trabalho, surgem os seguintes desafios que podem ser desvendados em trabalhos futuros:

Como visto na seção 4.4, o modelo apresenta uma certa volatilidade na detecção de algumas entidades. Assim, sugere-se que seja feita uma análise mais aprofundada sobre os hiperparâmetros utilizados no modelo a fim de melhorar sua performance. Além disso, é possível aprimorar o modelo com um conjunto de dados de treinamento em maior quantidade e com um melhor balanceamento de entidades, realizando uma busca maior nos elementos textuais dos aplicativos e então realizando mais anotações manuais no conjunto de dados. Essa busca pode ainda ser aproveitada para a identificação de novas entidades relevantes no contexto de aplicativos móveis que não foram incluídas no presente modelo, estendendo-o para o conhecimento de mais entidades e posteriormente auxiliando ainda mais na avaliação da originalidade.

Com o modelo pronto para ser utilizado na prática, é possível estudar seu uso efetivo na avaliação de originalidade de aplicativos. Um exemplo seria a coleta da taxa de entidades presentes em um certo aplicativo individualmente e a comparação desta taxa com outros aplicativos dentro de um universo de referência.

Outro assunto interessante a ser discutido é a análise de abordagens alternativas ao NER para a identificação de tópicos em elementos textuais. Um exemplo de uma abordagem alternativa que poderia ser adotada nesse contexto poderia ser por meio do uso e análise de ontologias, dentro do conceito de web semântica.

5. Conclusão

Neste trabalho foi desenvolvida uma abordagem para a identificação de entidades do conteúdo textual de apps criados com App Inventor. Como parte do trabalho foi realizada uma síntese da fundamentação teórica em relação a aplicativos criados com App Inventor e a

extração de elementos textuais dos mesmos. Além disso, foram apresentados conceitos de NLP e NER e como utilizar conteúdo textual para criar e treinar modelos de NER.

Como resultado de uma análise sistemática do estado da arte em relação a modelos de NER para a classificação de tópicos de apps desenvolvidos observou-se que ainda não existem pesquisas com esse foco.

Assim foi definido e implementado um modelo de NER para identificação de entidades do conteúdo textual de apps criados com App Inventor. Após levantados os requisitos e preparação do conjunto de dados, foi executado o treinamento de um modelo do pacote Spacy em português visando reconhecer as 10 entidades definidas na análise de requisitos. Analisando as métricas de precisão, recall e F1 score do modelo a partir do conjunto de dados de validação observou-se que o mesmo obteve níveis de desempenho aceitáveis.

Em seguida, foi executado um teste com um novo conjunto de de 10 aplicativos comparando um reconhecimento manual de entidades à identificação automática pelo modelo treinado. Analisando as métricas, observou-se que o resultado da precisão, recall e F1 score obteve uma variação para a maioria das entidades propostas em relação aos resultados obtidos na validação. Ainda assim, os resultados do teste indicaram no mínimo um grau de desempenho aceitável para todas as entidades, com exceção da entidade COURSE. O desempenho do modelo como um todo também obteve valores aceitáveis, indicando portanto uma relativa eficácia na classificação das entidades propostas.

Assim, este trabalho pode contribuir para a avaliação da criatividade de escolha de tópicos de aplicativos criados com App Inventor, auxiliando no ensino da computação na educação básica e conseqüentemente contribuindo para o desenvolvimento de uma competência essencial no século XXI.

Como trabalhos futuros, sugere-se a aprimoração da eficácia do modelo, bem como uma análise mais aprofundada e ajustes nos hiperparâmetros do algoritmo. Sugere-se, ainda, a adição de mais entidades ao modelo que possam ser relevantes no contexto de aplicativos móveis, e então seu uso efetivo na avaliação de originalidade. Também é possível realizar um estudo sobre a classificação de tópicos e avaliação da criatividade adotando técnicas alternativas, como por exemplo a abordagem de ontologias e web semântica no lugar de NER.

Referências

ALLWRIGHT, S. 2022. What is a good F1 score and how do I interpret it? Disponível em: <<https://stephenallwright.com/good-f1-score/>>. Acesso em Novembro de 2022.

ALVES, N. d. C.; BARBOSA, H.; AZEVEDO, L. F. CodeMaster – Automatic Assessment and Grading of App Inventor and Snap! Programs. *Informatics in Education*, 17(1), 2018, 117-150.

ALVES, N. da C., GRESSE VON WANGENHEIM, C., ALBERTO, M., MARTINS-PACHECO, L. H. Uma Proposta de Avaliação da Originalidade do Produto no Ensino de Algoritmos e Programação na Educação Básica. In: *Anais do Simpósio Brasileiro de Informática na Educação*, Natal, Brasil, 2020.

ALVES, N. da C., GRESSE VON WANGENHEIM, C., MARTINS-PACHECO, L. H., BORGATTO, A. F. Artefatos computacionais são considerados criativos? In: *Anais do Simpósio Brasileiro de Educação em Computação*, Feira de Santana, Bahia, 2022.

ALVES, N. da C., GRESSE VON WANGENHEIM, C., & MARTINS-PACHECO, L. H. (2021). Assessing Product Creativity in Computing Education: A Systematic Mapping Study. *Informatics in Education*, 20(1), 19-45. doi:10.15388/infedu.2021.02

AMARAL, D. O. F. D. “O reconhecimento de entidades nomeadas por meio de conditional random fields para a língua portuguesa.”. Dissertação do programa de Mestrado em Ciência da Computação da PUCRS, Porto Alegre, 2013.

APP INVENTOR COMMUNITY. AIA File Structure, 2019. Disponível em: <<https://community.appinventor.mit.edu/t/aia-file-structure/219>>.

BINKLEY, M., et al., Defining 21st Century Skills. Draft white paper 1. Melbourne: The University of Melbourne, Melbourne, Australia. 2010.

BROWNLEE, J. (2020). How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification. Disponível em: <<https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/>>. Acesso em Julho de 2022.

EBRAHIMI, F., TUSHEV, M., MAHMOUD, A. 2021. Classifying Mobile Applications Using Word Embeddings. *ACM Trans. Softw. Eng. Methodol.* 31(2), Artigo 20 , 2022.

Explosion. Spacy, 2015. Disponível em: <<https://spacy.io/>>. Acesso em Julho de 2022.

FlairNLP. Flair, 2018. Disponível em: <<https://github.com/flairNLP>>. Acesso em Julho de 2022.

GRESSE VON WANGENHEIM,C.; HAUCK, J. C. R.; DEMETRIO, M. F.; PELLE, R. CodeMaster - Automatic Assessment and Grading of App Inventor and Snap! Programs. *Informatics in Education*, 17(1), 2018, p. 117-150.

LEE, S.; KO, Y. Named-Entity Recognition Using Automatic Construction of Training Data From Social Media Messaging Apps. *IEEE Access*, 8, 2020.

MIT App Inventor, About Us, 2012. Disponível em: <<http://appinventor.mit.edu/about-us>>. Acesso em: Maio de 2020.

MUSTAFARAJ, E., TURBAK, F., SVANBERG, M. Identifying Original Projects in App Inventor. In: Proc. of the 30th Int. Florida Artificial Intelligence Research Society Conference, Marco Island, FL, USA, 2017, 567-572.

NADKARNI, P.M; OHNO-MACHADO, L.; CHAPMAN, W.W. Natural language processing: an introduction, Journal of the American Medical Informatics Association, 18(5), 2011. National Research Council . Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century. Disponível em: <<https://doi.org/10.17226/13398>>. Washington, DC, USA. 2012.

NLTK Project. NLTK, 2001. Disponível em: <<https://www.nltk.org>>. Acesso em Julho de 2022.

OXFORD LANGUAGES, Dicionário Online de Português. Disponível em: <<https://languages.oup.com/google-dictionary-pt/>>. Acesso em: 09/12/2021.

PATTON, E.W.; TISSENBAUM, M.; HARUNANI, F., MIT App Inventor: Objectives, Design and Development. In: KONG, S.; ABELSON, H. Computational Thinking Education. Springer, Singapore, 2019.

QUEIROZ, R. L.; SAMPAIO, F. F. DuinoBlocks for Kids: um ambiente de programação em blocos para o ensino de conceitos básicos de programação a crianças do Ensino Fundamental I por meio da Robótica Educacional. In: Anais do Workshop sobre Educação em Computação, Porto Alegre, RS, Brasil, 2016.

RANJAN B.S.C.; SIDDHARTH, L.; CHAKRABARTI, A. A systematic approach to assessing novelty, requirement satisfaction, and creativity. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 32(4), p. 390-414, 2018.

SCAICO, P. D. LIMA, A. A., SILVA, J. B. B., AZEVEDO, S., PAIVA, L. F., RAPOSO, E. H., ALENCAR, Y., MENDES, J. P., SCAICO, A. Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch. Revista Brasileira de Informática na Educação, 21(2), 2013.

TABASSUM, J.; MADDELA, M.; XU, W.; RITTER, A. Code and Named Entity Recognition in StackOverflow.” ArXiv abs/2005.01634 (2020).

NER Annotator. Tecoholic, 2022. Disponível em: <<https://tecoholic.github.io/ner-annotator/>>. Acesso em outubro de 2022.

TISSENBAUM, M.; SHELDON, J.; ABELSON, H. From Computational Thinking to Computational Action. Communication of the ACM, 62(3), 34-36, 2019 .

Training Custom NER models in Spacy to auto-detect named entities [Complete Guide], 2020. Disponível em: <<https://www.machinelearningplus.com/nlp/training-custom-ner-model-in-spacy/>>.

VAGARINHO, J P. Como identificar a originalidade num artigo científico ou numa tese de doutoramento? Educar em Revista, 2019.

VAKULENKO, S.; MÜLLER, O.; BROCKE, J. V. Enriching iTunes App Store Categories via Topic Modeling. In: International Conference on Information Systems, Auckland, New Zealand, 2014.

YADAV, V.; BETHARD, S. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. In: Proc. of the 27th International Conference on Computational Linguistics, Santa Fe, USA, 2018.