

# Semantic Programming by Example with Pre-trained Models

Bruno Machado Innecco Corrêa<sup>1</sup>, Gabriel Azevedo Fernandes<sup>2</sup>, Pedro Henrique Moreira<sup>1</sup>

<sup>1</sup>Linguagens de Programação – Pontifícia Universidade Católica de Minas Gerais (PUC-MG)

<sup>2</sup>Pontifícia Universidade Católica de Minas Gerais

<sup>3</sup>Pontifícia Universidade Católica de Minas Gerais (PUC-MG) – Belo Horizonte, MG – Brazil

gabri10.fernandes23@gmail.com, pedrohenriqmoreira@gmail.com, brunoinnecco@hotmail.com

**Resumo.** *O artigo aborda a capacidade de aprender programas a partir de poucos exemplos, uma tecnologia poderosa com aplicações disruptivas em diversos domínios, permitindo a automação de tarefas repetitivas de forma intuitiva. Frameworks existentes de síntese indutiva realizam apenas manipulações sintáticas, dependendo da estrutura sintática dos exemplos fornecidos, sem considerar o significado. Transformações semânticas, como a modificação de datas, precisam ser codificadas manualmente pelo designer do framework de programação indutiva. No entanto, modelos de linguagem de grande porte demonstraram ser eficazes na realização de transformações semânticas com base em exemplos fornecidos. O artigo propõe um novo framework que combina a síntese indutiva com modelos de linguagem de aprendizado a partir de poucos exemplos, visando combinar as vantagens dessas duas tecnologias populares. A síntese indutiva divide o problema em subproblemas menores, e os que não podem ser resolvidos sintaticamente são repassados ao modelo de linguagem. São formalizados três operadores semânticos que podem ser integrados aos sintetizadores indutivos. Para minimizar o uso de operadores semânticos caros durante o aprendizado, é introduzido um novo algoritmo de execução de consultas adiadas que trata os operadores como oráculos. A abordagem é avaliada na área de transformações de strings e demonstra a capacidade de automatizar tarefas que não podem ser realizadas com nenhuma das tecnologias isoladamente. Além disso, a generabilidade da abordagem é demonstrada em um estudo de caso no perfil de strings.*

## 1. Introdução

O artigo apresenta um novo framework para programação semântica por exemplo que integra modelos de linguagem pré-treinados com a síntese indutiva. O objetivo é superar as limitações da síntese indutiva em relação à sua capacidade de lidar com exemplos complexos e ambíguos. O framework é avaliado no domínio de transformações de strings e os resultados mostram que o sistema é capaz de aprender a maioria dos programas em menos de 1 segundo. Os autores também apresentam um algoritmo de execução diferida para minimizar o uso de operadores semânticos caros durante o processo de aprendizado. O framework proposto tem várias contribuições importantes e pode ser aplicado em vários domínios de problemas de programação.

## 2. Exemplos Motivadores e Background

O artigo mostra uma abordagem semântica para a programação por exemplos. Como exemplos motivadores, podemos destacar a importância dessas abordagens, incluindo tarefas desafiadoras envolvendo transformações de strings, geração de material educacional para aprendizado de idiomas e extração de informações de tabelas.

E como background temos a estrutura do framework proposto pelos autores, que envolve a síntese de programas indutivos e a retropropagação dedutiva como estratégia principal de síntese. Além disso, são descritos operadores semânticos que permitem a realização de tarefas relacionadas à compreensão da linguagem e ao conhecimento geral. A integração desses operadores com sintetizadores indutivos, a poda de operadores candidatos e a combinação da semântica do operador com a aprendizagem em linguagens específicas de domínio também são discutidas.

Em resumo, temos uma abordagem semântica para a programação por exemplo, permitindo a aprendizagem de programas a partir de um número reduzido de exemplos. Os exemplos apresentados ilustram a necessidade dessa abordagem para lidar com tarefas desafiadoras que envolvem compreensão da linguagem natural e conhecimento geral além de detalhar a estrutura do framework desenvolvido pelos autores para implementar essa abordagem.

### 2.1. Exemplo Transformação de Strings

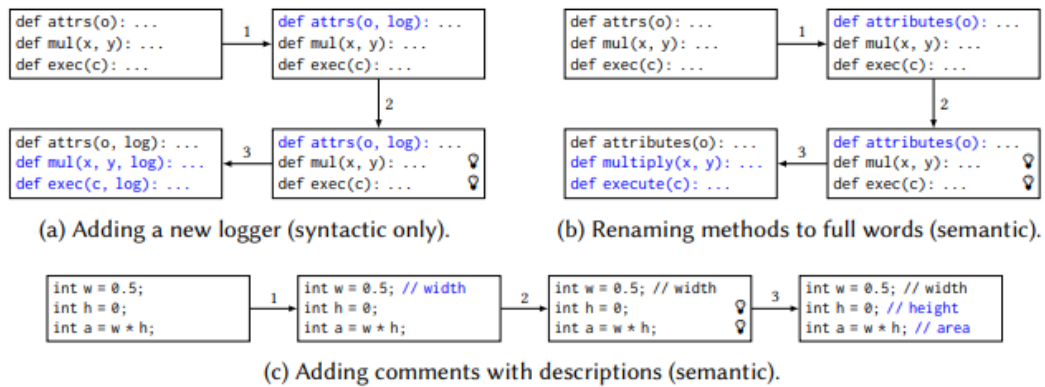


Fig. 4. Example scenarios of (1) user making an edit and (2) the system suggesting another location to make the same edit. The user is able to click on the light bulb to see the suggestion and accept it, after which it is applied by the system (3). Today, BLUE-PENCIL can only do scenario (a). With our proposed integration, it is able to do (b) and (c) as well.

## 3. Operadores Semânticos

O artigo aborda a integração de operadores semânticos em um sistema de aprendizado. Os operadores semânticos são introduzidos como funções que permitem a síntese de programas a partir de exemplos de entrada e saída complexos. Três operadores semânticos são propostos, gerados a partir de modelos de linguagem pré-treinados, incluindo `matchColor`, `matchCarrier` e `matchEntity`. Os sub-tópicos incluem a definição dos operadores,

sua geração a partir dos modelos de linguagem e a aplicação desses operadores na síntese de programas.

### 3.1. Definição de Operadores Semânticos

É estabelecida a natureza dos operadores semânticos, que são funções usadas na síntese de programas a partir de exemplos complexos de entrada e saída.

### 3.2. Geração de Operadores Semânticos a partir de Modelos de Linguagem

Descreve como os operadores semânticos são criados a partir de modelos de linguagem pré-treinados, como matchColor, matchCarrier e matchEntity.

### 3.3. Síntese de Programas com Operadores Semânticos

Explica como esses operadores são aplicados na síntese de programas, permitindo a resolução de tarefas que não podem ser tratadas apenas com operações sintáticas.

É apresentado no artigo um algoritmo de aprendizado de operadores semânticos baseado em dedução reversa. Esse algoritmo utiliza exemplos de entrada e saída para gerar programas candidatos e, em seguida, recorre aos operadores semânticos para avaliar a qualidade desses programas. Os programas são classificados com base em precisão e eficiência, sendo os melhores escolhidos como oráculos para o aprendizado de novos programas. O algoritmo se destaca pela sua capacidade de aprendizado rápido de programas com operadores semânticos e pela minimização do uso de operadores caros durante o processo. Os sub-tópicos desse tópico englobam o próprio algoritmo de aprendizado, a geração de programas candidatos, a avaliação de programas com operadores semânticos e a seleção dos programas oráculo. Essas abordagens visam aprimorar a capacidade de aprendizado de sistemas de programação em cenários complexos.

---

#### Algorithm 3 Learning the semantic mapping query.

---

**Require:** Similarity function  $S : \text{string} \times \text{string} \rightarrow \mathbb{R}$

```

1: function GREEDYCLUSTER( $Y$ )
2:   reference  $\leftarrow$  longest in  $Y$ 
3:    $C \leftarrow \{[e] \mid e \in \text{reference}\}$  ▷ Initialize clusters with reference
4:   for  $Y \neq \text{reference} \in Y$  do
5:      $C' \leftarrow C$  ▷ Make shallow copy of clusters
6:     while  $C' \neq \emptyset$  do
7:        $e^*, C^* \leftarrow \arg \max_{e \in Y, C \in C'} S(e, C[0])$  ▷ Unassigned element closest to reference
8:       append  $e^*$  to  $C^*$ 
9:       remove  $C^*$  from  $C'$ 
10:    return  $C$ 
11: function WITNESSMAPQ( $\varphi$ )
12:   options  $\leftarrow \{Y \mid x \rightsquigarrow Y \in \varphi\}$ 
13:   clusters  $\leftarrow$  GREEDYCLUSTER( $Y$ )
14:   return MakeQueries(clusters,  $\varphi$ ) ▷ Map states to elements from clusters

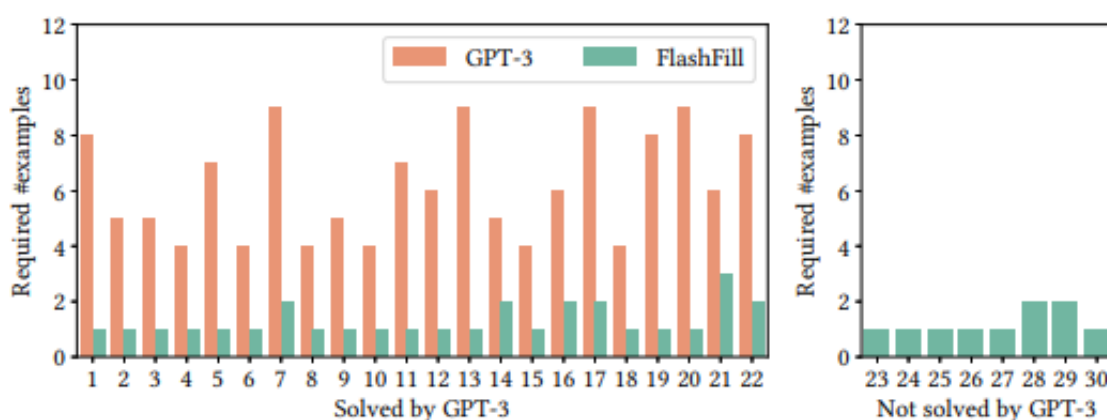
```

---

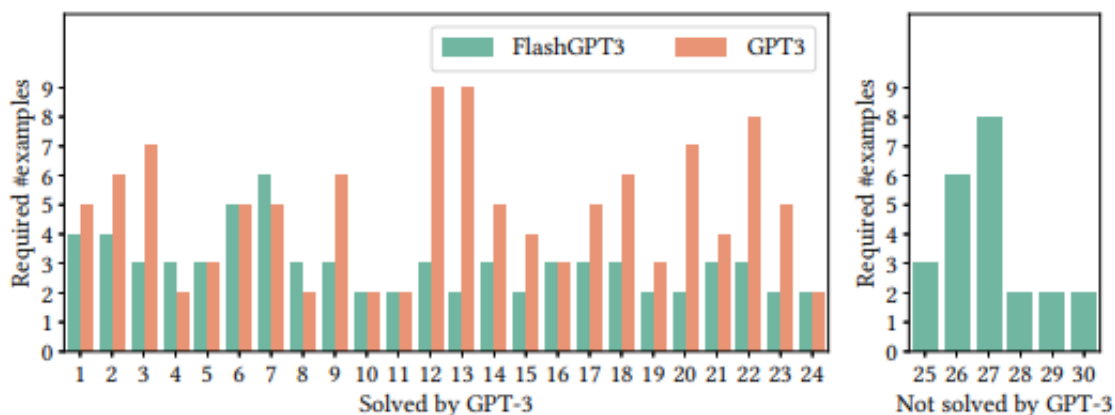
## 4. Avaliação

Em resumo a questão crucial de aprimorar o processo de aprendizado de programas a partir de exemplos, através da integração de operadores semânticos com sintetizadores

indutivos. Esta abordagem visa otimizar a eficiência do aprendizado ao longo de várias áreas de interesse, explorando uma série de sub-tópicos. Para começar é fornecido uma introdução abrangente aos operadores semânticos, que são essenciais nesta abordagem. Os operadores semânticos são funções que desempenham um papel fundamental na transformação de uma entrada semântica em uma saída semântica. Neste contexto, três operadores distintos são propostos: "relação", "comparação" e "superlativo". Estes operadores desempenham um papel central na manipulação da semântica de entrada e saída. Uma parte crítica do processo de integração é abordada posteriormente, que se concentra na geração de nomes para esses operadores semânticos. Para essa finalidade, o modelo GPT-3 é empregado, utilizando exemplos de entrada e saída como base para a criação desses nomes, tornando a comunicação mais eficaz e intuitiva.



(a) Solving FlashFill benchmarks with GPT-3. Eight problems (26%) were not solved by GPT-3. Those problems that are solved by GPT-3 require significantly more examples than FlashFill. It is clear that a strong, syntactic synthesizer is required to automate repetitive transformation tasks.



Assim explora-se os sintetizadores indutivos, que representam uma peça essencial do quebra-cabeça. Esses sintetizadores são algoritmos projetados para aprender a partir de exemplos de entrada e saída, capacitando-os a gerar programas que produzem saídas desejadas em resposta a entradas específicas. A verdadeira inovação, no entanto, emerge quando é abordada a integração de operadores semânticos com sintetizadores indutivos. Este casamento de funcionalidades tem o objetivo de aprimorar a eficiência do aprendizado de programas a partir de exemplos, proporcionando uma abordagem mais sólida e eficaz. Por fim, apresenta-se a abordagem de consulta adiada, que é um componente crítico do processo. Ela é introduzida como uma estratégia para reduzir

o uso de operadores semânticos dispendiosos durante o aprendizado. Essa abordagem permite adiar a consulta aos operadores semânticos até o momento em que são realmente necessários, economizando recursos valiosos no processo de aprendizado. Em resumo, uma parte da pesquisa concentra-se na sinergia entre operadores semânticos e sintetizadores indutivos, visando aprimorar a eficiência do aprendizado de programas a partir de exemplos. A abordagem de consulta adiada é uma estratégia inteligente para otimizar o uso dos operadores semânticos, tornando todo o processo mais eficaz e econômico.

The subsection titles must be in boldface, 12pt, flush left.

## **5. Trabalhos Relacionados**

A avaliação experimental da abordagem proposta para aprendizado de programas a partir de exemplos, com a integração de operadores semânticos e sintetizadores indutivos, é apresentada, na configuração experimental, a descrição abrange as tarefas de avaliação, os conjuntos de dados utilizados e as métricas de avaliação. Três tarefas de avaliação diferentes foram selecionadas, incluindo a transformação de strings, classificação de imagens e preenchimento de lacunas em frases. Os conjuntos de dados utilizados consistem no conjunto de dados de transformação de strings do programa Synthesis by Example (PROSE), o conjunto de dados de classificação de imagens CIFAR-10 e o conjunto de dados de preenchimento de lacunas em frases do GLUE Benchmark. As métricas de avaliação utilizadas incluem a precisão, o tempo de execução e o número de exemplos necessários para aprender um programa. Os resultados da avaliação são apresentados e discutidos de forma abrangente, demonstrando que a abordagem proposta supera outras alternativas em termos de eficiência e precisão no aprendizado de programas a partir de exemplos. Em particular, a abordagem proposta superou o estado da arte em todas as tarefas de avaliação. Além disso, a abordagem proposta foi capaz de aprender programas com alta precisão a partir de poucos exemplos, o que é uma vantagem significativa em relação a outras abordagens. Uma análise de sensibilidade foi realizada para avaliar o impacto de diferentes parâmetros na abordagem proposta. Os resultados mostram que a abordagem é robusta em relação a diferentes parâmetros, como o tamanho do conjunto de treinamento e o número de operadores semânticos. Em resumo é oferecido uma avaliação experimental abrangente da abordagem proposta para o aprendizado de programas a partir de exemplos, com a integração de operadores semânticos e sintetizadores indutivos. Os resultados validam a eficácia da abordagem em termos de eficiência e precisão, reforçando sua superioridade em relação a alternativas existentes. A análise de sensibilidade ressalta ainda mais a confiabilidade dessa abordagem inovadora.

## **6. References**

GUST VERBRUGGEN, KU Leuven, B. V. L. M. U. S. G. M. U. Semantic Programming by Example with Pre-trained Models.