Detecting Textual Analogies Using Semi-Supervised Learning

Rei Rembeci

Department of Computer Science, Earlham College

Motivation

- Analogy is an integral aspect of human communication, understanding, and knowledge sharing.
- Cognitive systems that can process through textual analogies can learn more from what they read.
- However, strategies for detecting analogy with a machine are largely unexplored.
- There is also no standard corpus of analogy text, an important and expensive tool for developing algorithms for analogy text classification.
- A corpus of analogies is manually created and semi-supervised learning techniques are used to develop a system that can properly identify them.

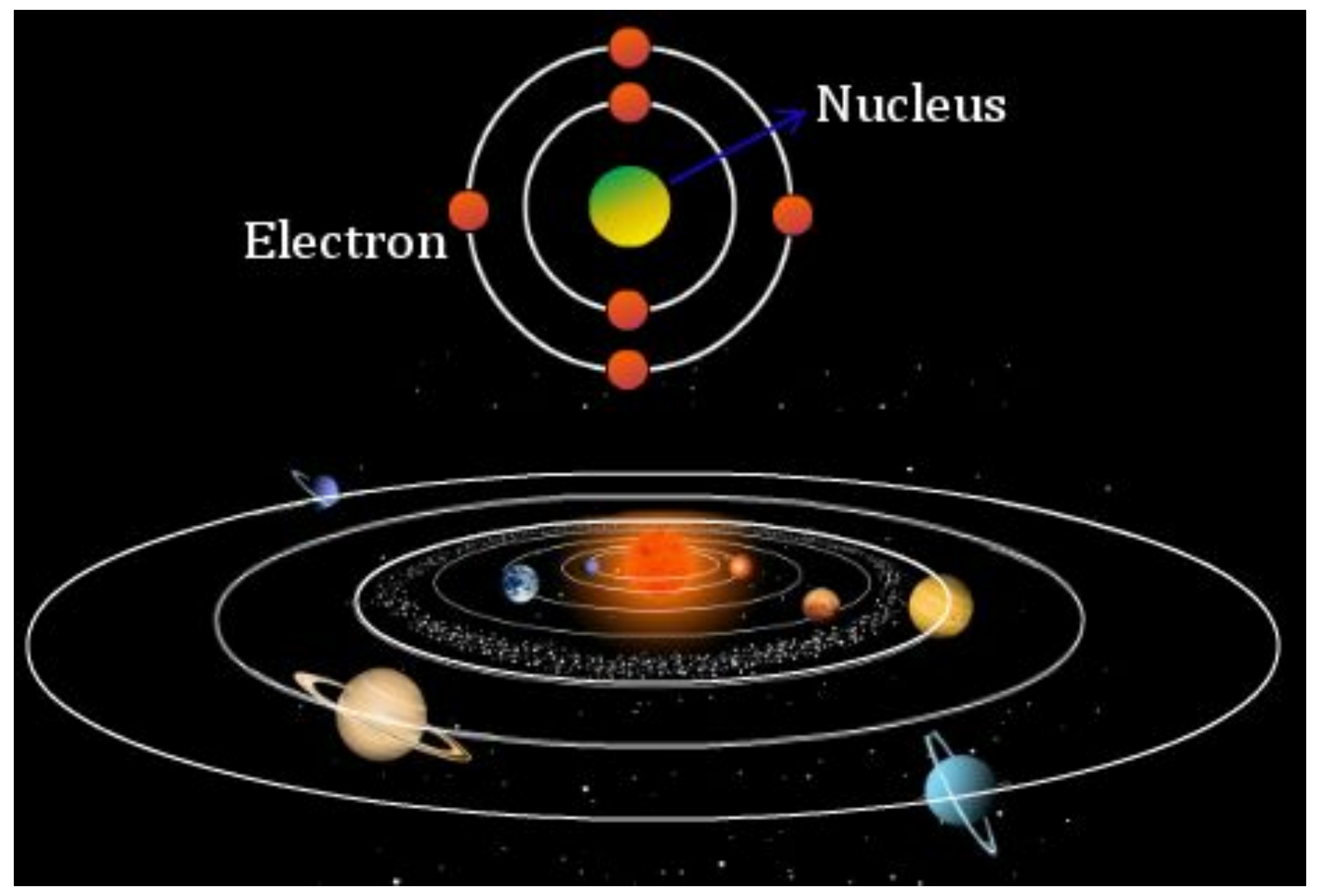
Background

Analogy

- The process of comparing a base case to a structurally similar target case. They share relationships.

Rutherford Atom and Solar System

Atom	Solar System
Nucleus	Sun
Electron	Planet
Electromagnetism	Gravity



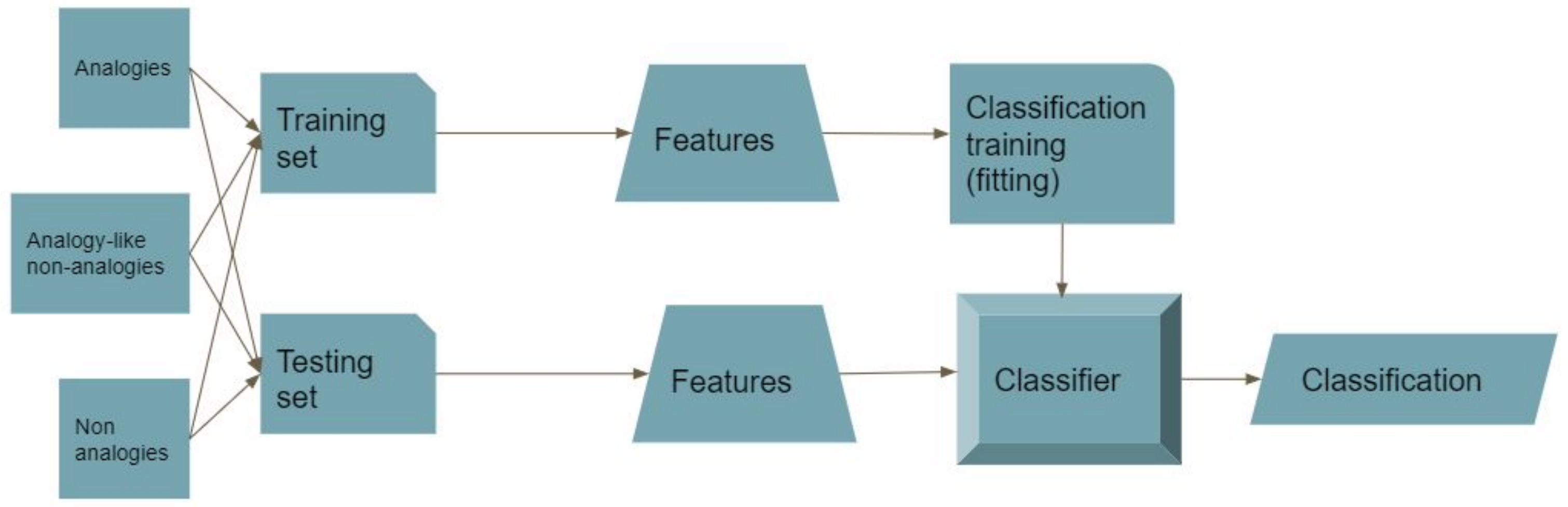
http://images.tutorcircle.com/cms/images/44/rutherfords-atomic-model1.png

Natural Language Processing

- Natural language processing (NLP) is an intersectional field concerning interactions and relationships between computers and natural languages.

Methods

The model is initially trained with 20387 training examples, out of which 386 are labeled. The model was then tested on 69 sentences. Prominent features from the original texts are extracted using NLP techniques. These features are then used to train the classifier.



Semi-supervised learning

- Semi-supervised learning is a machine learning technique which can be used for classification or regression.
- It considers the problem when only a small subset of the training data has corresponding class labels.

Feature extraction

Before training the classifier, feature extractors process raw text and represent it in ways that are suitable for classification. Three different feature extraction techniques were tested namely count vectorization, hash vectorization and tf-idf vectorization.

Classifiers

The model uses two semi-supervised classifiers:

- Transductive Support Vector Machines
- Label Propagation

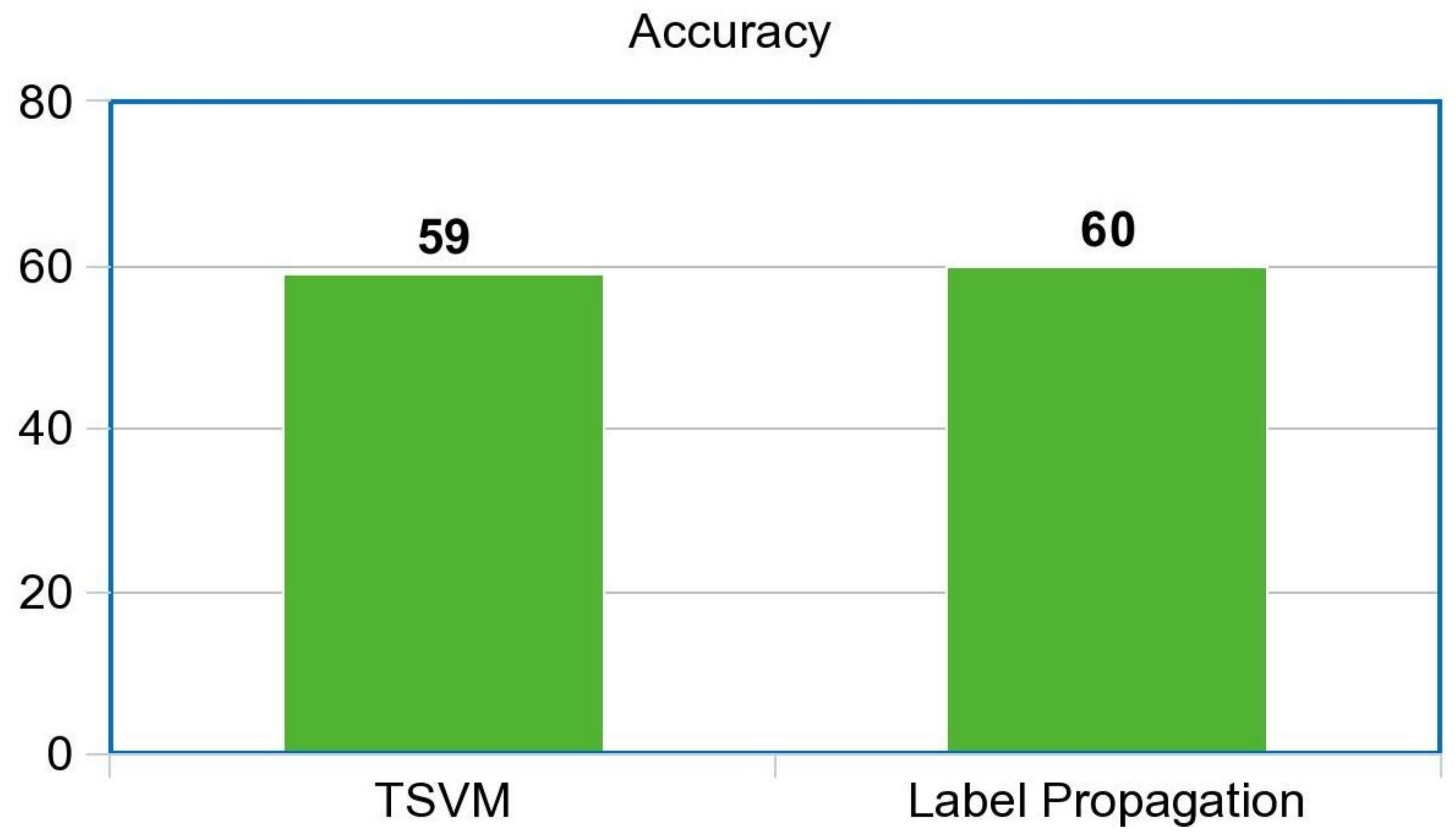
Exhaustive parameter search

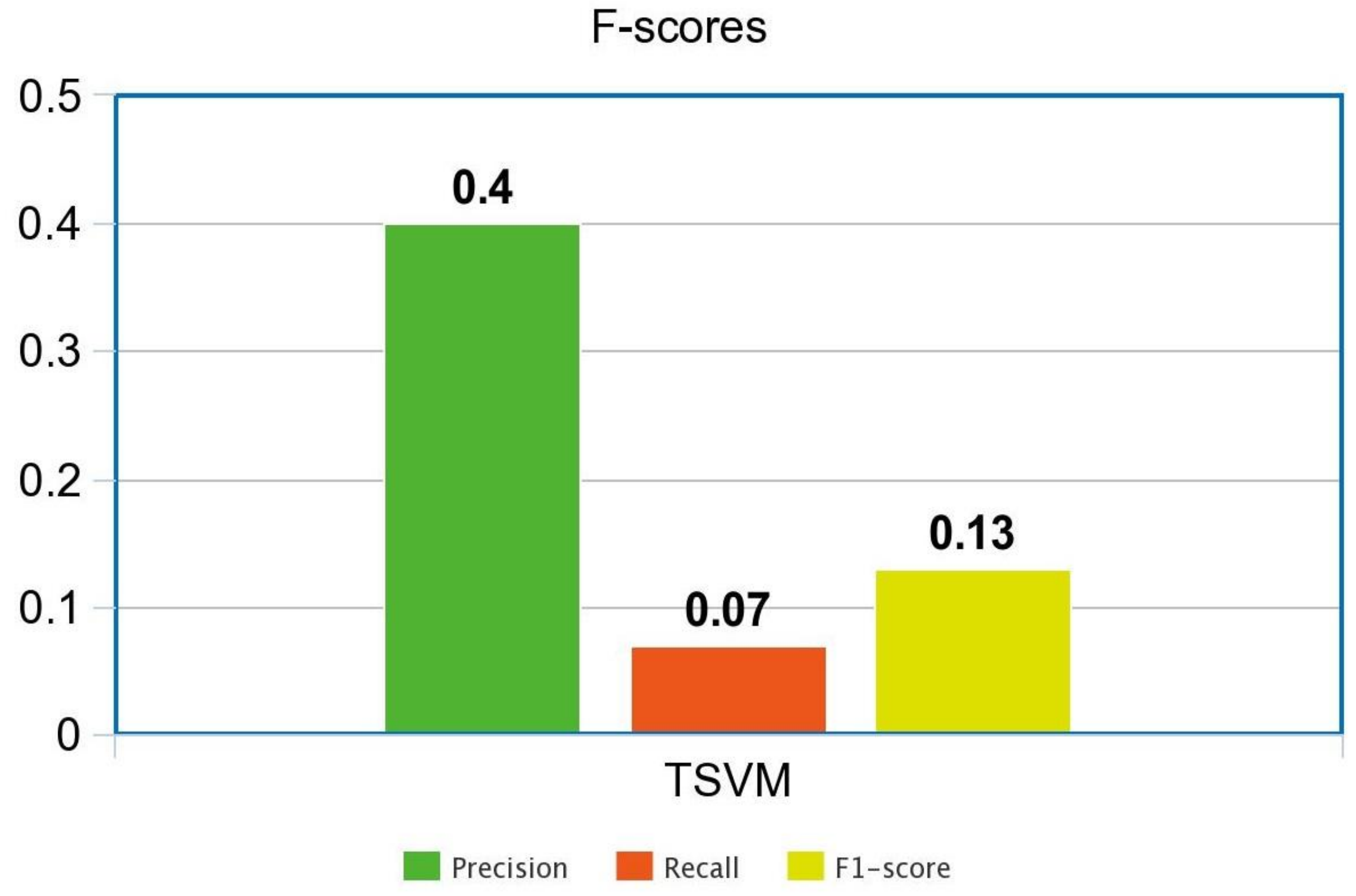
To boost the performance of the classifiers, an exhaustive search over specified parameters values for each combination of feature extraction tools and classifier is computed.

References

Lars Buitinck et al. API design for machine learning software: experiences from the scikit-learn project.

Preliminary Findings





- TSVM and Label Propagation had similar results when considering the accuracy of the model.
- However, the f-scores were significantly different. Label Propagation classified everything as non-analogies, leading to a precision, recall, and f1-score equal to 0.

Future work

- Instead of considering the full sentence, I would like to use a parser to extract the base and target from each sentence.
- Expand the corpus to have a bigger training and testing set.

Acknowledgements

This project was supported by the Earlham College Computer Science Department as part of the Senior Capstone. I would like to thank David Barbella, Xunfei Jiang, Charles Peck, Ajit Chavan, and Phi Nguyen.