CS461 – ARTIFICIAL INTELLIGENCE

Term Project Proposal

Nickname of the Program and Group

SWAPLIANO

Names of the Group Members

1. Ayhan Okuyan (CONTACT)
2. Barış Akçin
3. Berkan Özdamar
4. Mustafa Bay
5. Deniz

Description of the Project

In this project, we will be building a program called SWAPLIANO. We will be using Python 3 to build the project. We will be extracting John Fagliano's daily 5x5 puzzle from the HTML (https://www.nytimes.com/crosswords/game/mini) source code by using Selenium. We will be taking the puzzle structure, the answers and the clues that is given for each question. Then we will create an algorithm that uses various sources online to change the puzzle clues in a way that their answers would still be the same as before. For that we will be using each word or phrase's description from the WordNet 3.1, Urban Dictionary, the Merriam Webster dictionary, or other sources that will be found while working on that part of the project to come up with new descriptions. For phrases that should contain a space in between, we probably will be querying the words on Google to see if we get any "Did you mean" clause at the beginning of the HTML page.

Paper Review

The paper "A Fully Automatic Crossword Generator" is a paper that proposes a new system that is used to develop new crossword puzzles using state-of-the-art Natural Language Processing (NLP) algorithms. The program composes a new crossword puzzle with the topography, and the clues. This project is different from other crossword generator projects in the sense that it does not use a human-generated vocabulary. Instead, it uses a definition extracting module that finds new definitions of a word by processing information from the internet sources. It uses SVM-based chunking methods to form the puzzle. Overall, the algorithm can perform high quality, medium sized puzzles in a short amount of time. This paper consists of a complicated scheme to create a crossword; however, I think we will be able to grasp some parts that will be relevant within our project.

Rigutini et. al., 2009, "A Fully Automatic Crossword Generator", IEEE Xplore, 10.1109/ICMLA.2008.104

A Fully Automatic Crossword Generator

Leonardo Rigutini Michelangelo Diligenti Marco Maggini
Marco Gori
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Siena,
Via Roma 56, Siena, Italy
{rigutini,diligmic,maggini,marco}@dii.unisi.it

Abstract

This paper presents a software system that is able to generate crosswords with no human intervention including definition generation and crossword compilation. In particular, the proposed system crawls relevant sources of the Web, extracts definitions from the downloaded pages using state-of-the-art Natural Language Processing *(NLP) techniques and, finally, attempts at compiling a crossword schema with the extracted definitions using a* Constrain Satisfaction Programming *(CSP) solver. The crossword generator has relevant applications in entertainment, educational and rehabilitation contexts.*

1. Introduction

This paper introduces a software system that generates crosswords with no human intervention, including definition/clue generation and crossword compilation. The system crawls a predefined set of information sources (wiki pages, dictionaries, etc.) and extracts definitions from the downloaded pages using state-of-the-art *Natural Language Processing* (NLP) techniques. Finally, the system compiles a given crossword layout, positioning the extracted definitions on the crossword slots by using a *Constraint Satisfaction Programming* (CSP) solver [10]. Up to our knowledge, this is the first system that is able to fully automatically generate crosswords. Other works attempted at automatically solving crosswords given a human edited vocabulary of definitions [1]. For example, *Proverb* [7] solves the clues using *Natural Language Processing* (NLP) techniques working over a pre-compiled knowledge base. [4] attempts at overtaking the limited flexibility of Proverb by acquiring his knowledge directly from the Web: each clue is sent to a Web search engine, the results are analyzed using NLP techniques and a list of possible answers are selected. *Constraint Satisfaction Programming* (CSP) techniques are used to refine the selected answers in the attempt of finding a combination of candidates that solves the biggest possible portion of the schema. The outline of the paper is the following: section 2 introduces the general architecture of the system. Some experimental result are shown in section 3. Finally, section 4 draws some conclusions.

2. System architecture

The architecture of the system is composed by a definition extraction and a scheme generation module. The definition extractor crawls a set of relevant data sources from the Web and discovers new definitions by applying NLP techniques on the downloaded pages. The step is performed offline, and the resulting definitions can be used to generate an arbitrary number of crosswords.

The extracted definitions are then used by the crossword compiler. This module is executed every time that a user requires the generation of a new crossword. Since this module works online, it should provide the result in a reasonably short time. The overall architecture of the system is sketched in fig:1-(a). The system is able to generate high-quality medium sized crosswords with no human interaction. See in fig:1-(b) for an example of a (15×15) crossword generated using our system.

2.1. The Definition Extraction module

The definition extraction module has three components: the information gatherer, the Natural Language analyzer and the definition extractor (see fig:2). The gatherer downloads the pages from the information sources and processes the content of each page discarding the formatting information and retaining the raw text. The text is then passed to the *Natural Language Processing* (NLP) analyzer. The NLP analysis is divided into layers as sketched in the fig:2. Each layer works on the output of the previous layer, with the exception of the first layer that directly works on the input