Technische Universität Berlin
Fakultät IV - Fakultät Elektrotechnik und Informatik
Fachgebiet Datenbanksysteme und Informationsmanagement

Project Report
Generating Acrostics via Paraphrasing and Heuristic Search
DBPRO - Database Projects (WS 2014/2015)

Supervisor:
Johannes Kirschnick

Authors:
Bruno Soares Fillmann ()
Fernando Bombardelli da Silva (bombardelli.f@mailbox.tu-berlin.de)
Jürgen Bauer ()
William Bombardelli da Silva (wbombardellis@mailbox.tu-berlin.de)

February 2, 2015

Contents

Abstract

ascasdas

1 Introduction and Motivation

lnmkm

2 Generating Acrostics via Paraphrasing and Heursitic Search

2.1 Problem Definition

2.2 Modeling as Search Problem

2.3 Cost Measure

2.4 Operators

2.4.1 Word Insertion or Deletion

The idea around this operator is to insert or delete words in the middle in order to insert new letter and accomplish the goal acrostic or to remove words and change the position of words inside the text.

//Example here?

The Word Insertion or Deletion operator takes as input a text and first tries to insert a new word in each space and second tries also to remove each word of the text. The condition to insert a new word w in the *i-th* space of the text is that w has to fit the context around the *i-th* space. It means that from the set of all possible words of the language, only a restricted subset can be inserted in this place. More specifically, the algorithm starts by taking for each space in the text n words around it as context. This is a so called n-gram, a kind of window to the text with length n or an array of words. After this, the n-gram just taken is sent to the context database (which is in this implementation the NetSpeak API [2]), that returns the possible words that could be inserted in the required space. For each of these ppssible words a new version of the text is created with it inside. Analogously, for each word w in text a n-gram with the words around is created, w is then taken out of the n-gram, which is tested against the context database to check whether this ngram is frequent enough in the language. If the answer is positive a new version of the text without w is created.

The queries to the context database are made in form of HTTP requests to the netspeak web service using the NetSpeak API.

// example here

2.4.2 Synonyms

The synonym operator has the goal of changing words in the text for other words, which have similar meaning. In general the operator takes a text as input and generates a set of new texts, in which each text has a word replaced by a synonym that may eventually contain more than just one word.

In order to perform the replacements it is required a synonym dictionary, which is know as thesaurus. In our implementation we used Open Thesaurus [3], which is available for download for free. This data source is available as a plain text file, but as the dictionary is accessed many times during the execution of the algorithm, it easily becomes intractable to handle a text file as a database.

To solve this problem we decided to use a NoSQL database server [4], namely, Redis. Redis is an open source advanced key-value pair cache and store [5]. Into the database server we load once the data from the thesaurus in a structured way where, every word is added as a key that points to a set of synonyms. Thus can the application easily and efficiently find similar terms for a given word only by accessing this key.

Naturally it is then required that the Redis server is running and listening to requests when the application runs, and that it has been once loaded by our script with the data from the dictionary.

[EXAMPLE]

PROS

CONS

3 Evaluation of the Results

4 Summary of Findings

to better: size of ngram, other context databas, degenerating into breadth first

References

[1] Benno Stein, Matthias Hagen, and Christof Bräutigam. *Generating Acrostics via Paraphrasing and Heuristic Search.*
In Junichi Tsujii and Jan Hajic, editors, 25th International Conference on Computational Linguistics (COLING 14), pages 2018-2029, August 2014. Association for Computational Linguistics.

[2] Martin Potthast, Martin Trenkmann, and Benno Stein. *Netspeak: Assisting Writers in Choosing Words.* In Cathal Gurrin et al, editors, Advances in Information Retrieval. 32nd European Conference on Information Retrieval (ECIR 10) volume 5993 of Lecture Notes in Computer Science, pages 672, Berlin Heidelberg New York, March 2010. Springer.

[3] *Synonyme - OpenThesaurus - Deutscher Thesaurus.* Available on: <`https://www.openthesaurus.de`>. Accessed in: Januar 2015.

[4] Jing Han; Haihong, E.; Guan Le; Jian Du. *Survey on NoSQL database.* Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on, pp.363,366, 26-28 October 2011.

[5] *Redis.* Available on: <`http://redis.io`>. Accessed in: Januar 2015.

A Appendix