Scarlet - Artificial Teaching Assistant

Karić Ilhan{ilhan.karic@edu.fit.ba}, Denis Mušić {denis@edu.fit.ba}; Emina Junuz {emina@edu.fit.ba}; Smajić Mirza{mirza.smajic@edu.fit.ba}
Faculty of Information Technologies

***Abstract*— Scarlet an Artificial Teaching Assistant is a personal digital assistant that has been developed with main aim to assist students in their learning process by ensuring fast and efficiently search of documents and learning materials. Scarlet is able to give an adequate response to a specific question based on knowledge gathered by an unique algorithm which enables her to recognize context during file and web page content search. After finding the most appropriate answer Scarlet seeks for student feedback in order to improve future search. The metric proposed is based on the power law which occurs in natural language, that is the Zipfian distribution[1]. It is designed to work for any spoken language although it might work on some better than other depending on the nature of the language, the structure, grammar and semantics. The method uses this metric to derive context from data and then queries the data source looking for the best match. The whole implementation is rounded off by a learning module which gives the system a learning curve based on users (students) scoring how relevant the output is among other parameters.**

keywords: [artificial intelligence], [machine learning], [pattern recognition], [natural language processing]

I. Introduction

It is impossible to ignore the fact that by every second the amount of data available to the end user experiences exponential increases. Managing large amounts of data and its transformation into knowledge has always occupied minds of researchers. In the early days the problem was in providing sufficient space for data storage. Aforementioned problem can be considered as completely solved thanks to the rapid development of the high-capacity hardware (storage) components. However, some very important challenges still remains. Main challenge is to find adequate methods for searching large amounts of data or data lakes. Although, some of the existing algorithms and indexing techniques made the search process considerably fast, they still require a significant effort in order to improve and customize.

Besides searching, very significant challenge considers the fact that modern computers are still not able to adequately communicate with the user in order to fully understand request and provide proper response.

Education is certainly an area in which effective data search and retrieval have significant impact on its success. Students of new generation are accustomed to find any information within a few minutes, so those expectations are transferred into the process of acquiring knowledge.

Considering all previously presented challenges and expectations this paper describes research result which aimed to develop adequate search algorithm and demonstrate its efficiency by building Artificial Teaching Assistant named Scarlet. Scarlet is able to give an adequate response to a specific question based on knowledge gathered by unique algorithm which enables him to recognize context during file or web page content search.

II. Related Work

Previous research in this area have emphasized the main issues and obstacles which should be solved in order to get noticeable results. In their paper [2] authors discussed an information extraction system in natural language question answering and examined the role of information extraction in question answering application. People have questions and they need answers, not documents. They pointed to urgent need for tools that would reduce the amount of text one might have to read in order to obtain the desired information, or in other words to achieve a special class of information seeking behavior called question answering.

Paper [3] presents LogAnswer - question answering (QA) system for the German language which could providing concise and correct answers to arbitrary questions. LogAnswer system was designed as an embedded artificial intelligence system which integrates methods from several fields of AI, namely natural language processing, machine learning, knowledge representation and automated theorem proving. Authors presented a machine learning solution to the wrong answer avoidance (WAA) problem, applying a meta classifier to the output of simple term-based classifiers and a rich set of other WAA features.

III. The Scarlet System

The Scarlet is designed to work best with a question / answer scheme but is highly adaptable to a vast range of different input schemes. Complete system can be divided into 3 mayor modules: *natural language processing*, *pseudo contextual data analysis* and *trial & error learning*. Complete Scarlet system and relationship between these modules is presented in the Figure 1.

The *natural language processing module* implements different methods trying to model and approximate answer patterns based on the input pattern. This is a very important first step because probability of the search method to find a solution/answer is increased exponentially depending on the quality of the parsed input. This module is in a feedback relation with the trial & error learning module.

978-1-5090-6536-3/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCAIRO.2017.11

The main component of the Scarlet system is the *pseudo contextual data analysis* which makes the same implementation capable of working for almost any spoken language. This module takes the processed input data from the previous module and recursively browses through all the folders and sub folders on a FTP or local server. The currently supported document files are "doc", "docx" and "pdf".

To speed things up the end user (student) can beforehand apply a filter mask by selecting a class subject. In addition, this module supports web search. It allows the user to search the web (up to a certain depth) and process the information from web pages, prompting what is most likely the answer for the given input.

The *trial & error learning module* is the last of the three modules. At this point the user is prompted a question if the Scarlets' answer was good or bad. Depending on this user feedback, in case it was positive, the system saves the given input-output pattern into the pattern table. The next time when the user sends some request to the system it will first browse through this table and compare the user input to the known inputs. When the system finds result that is "fit" enough it will look at the saved input-output pattern table and try to find how to morph the input in order to get the desired output and applies this strategy to the new user input. The algorithms used in aforementioned process will be described later in the paper.

IV. THE SCARLET ALGORITHMS

As mentioned in previous part of this paper, there are a few algorithms that were developed for these specific purposes. The two most notable ones are: *the pseudo contextual data analysis algorithm* which is in charge for contextual similarity and *the structure anchoring algorithm*.

A. *Pseudo contextual data analysis*

This algorithm is the main algorithm within the pseudo contextual data analysis module (contextual similarity[1]). The goal of this algorithm is to determinate similarity between two sets (S1 and S2). Similarity determination is done by breaking up set S1 into subsets following a tetrahedral pattern (i.e. for "ABC" the subset pattern would be: "A", "AB", "ABC", "B", "BC", "C"), after which it intersects every subset with S2. In the next step module simply adds the cardinality of each subset to the cardinalities of the previous subset intersections which can be presented by equation (1):

$$\delta(S1, S2) = \sum_{i=0}^{|S1|-1} \sum_{j=i+1}^{|S1|} |S1[i:j) \cap S| \tag{1}$$

Equation (1) represents a function which returns a value of delta representing how "similar" two sets are (S1, S2), i.e. how many characters they have in common. The main disadvantage with this approach is that we get incoherent results for different sets S1. Due to the nature of this function it means that bigger sets are more similar (they score more) just because of their size which is not acceptable. Therefore, in order to solve this issue we have to normalize function value by using Normalization Coefficient presented in equation (2).

$$\mathrm{T}_{\sigma}^{-1} = \binom{n+2}{3} \tag{2}$$

In order to normalize the delta function, it is necessary to multiply by this coefficient which measures the maximum

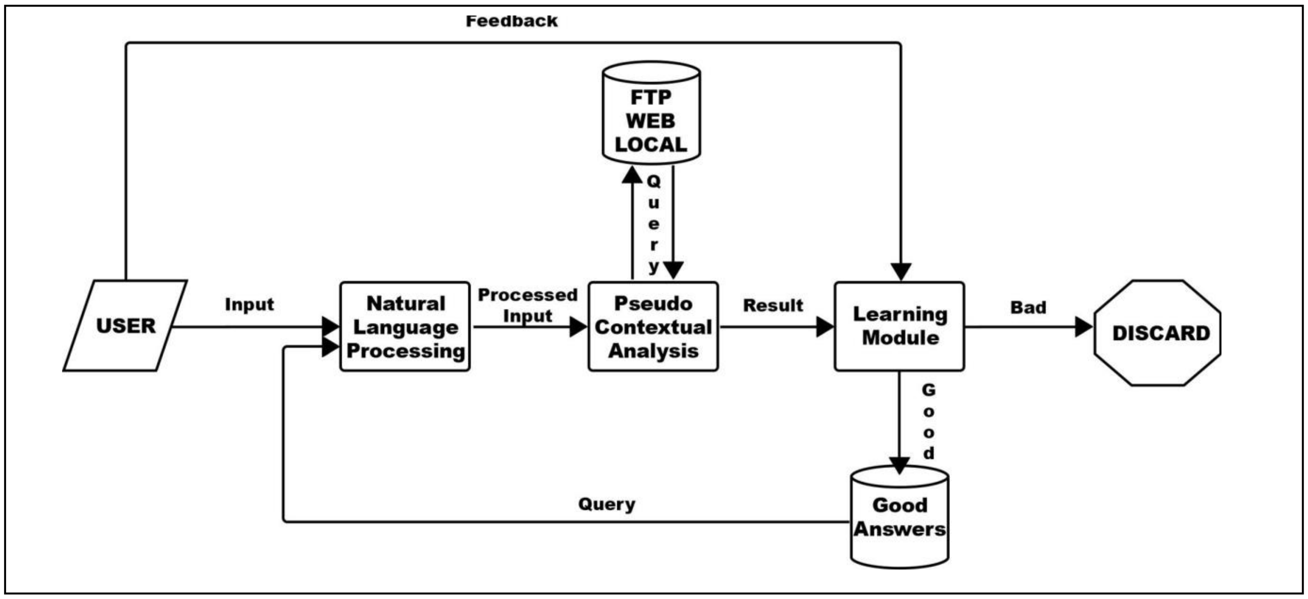


Figure 1. The Scarlet system components

[1]In this paper word *sets* will be considered as synonym for partially ordered sets

score possible (if S1 <=> S2). Normalization coefficient can be evaluated as shown in (3):

$$\Sigma(\mathrm{n}) = \sum_{\mathrm{k}=1}^{\mathrm{n}} \frac{\mathrm{k}(\mathrm{k}+1)}{2}$$
$$= \sum_{\mathrm{k}=1}^{\mathrm{n}} \frac{1}{2}\mathrm{k}(\mathrm{k}+1) = \frac{1}{2}\sum_{\mathrm{k}=1}^{\mathrm{n}} \mathrm{k}(\mathrm{k}+1)$$
$$= \frac{1}{2}\sum_{\mathrm{k}=1}^{\mathrm{n}} (\mathrm{k}^2+\mathrm{k})$$
$$=> \mathrm{T}_\delta = \binom{\mathrm{n}+2}{3}$$
$$T_\delta^{-1} = \binom{\mathrm{n}+2}{3}^{-1} \tag{3}$$

We can finally express the coefficient as the inverse binomial $\binom{\mathrm{n}+2}{3}^{-1}$ where *n* would be the cardinality of set S1. We arrive at the final expression for the contextual similarity (the metric) by multiplying the delta function with the normalization coefficient to get equation (4).

$$\bar{\delta}(S1, S2) = \frac{1}{T_\sigma} \sum_{i=0}^{|S1|-1} \sum_{j=i+1}^{|S1|} |S1[i{:}j) \cap S2| \tag{4}$$

Notice that this function is only defined for $|\mathrm{S}_1| > 0$. In the opposite case (where $|\mathrm{S}_1| = 0$) $\bar{\delta}$ simply becomes 0. This function will return a value in the range $0 \leq \bar{\delta} \leq 1$, where 1 means "complete" match ($S_1 = S_2$) and 0 means no match (sets have nothing in common).

We can use this delta similarity function to score strings (sets of characters). The closer the result is to 1, the more likely it is that the given snippet of text contains the information we want to retrieve. Due to the nature of the algorithm it can differentiate between not just keywords but also their context. Note that there are different ways this to implement this scoring. The method used had been proven to run in $\log n * \log n) = O(n \log^2 n)$ complexity assuming randomly generated strings were compared with randomly generated text. The reason why this asumption would work with languages in general is due to the fact that both randomly generated and coherent text follow a Zipfian distribution[1]

Previously described algorithm requires two steps. The first step is to run and compare the entire document file to the given input string. We do this for all documents and only take the one with the biggest score (most likely containing what we need) and prompt the student which document probably contains his answer and should be appropriate for studying.

During second step algorithm parses all sentences from the text as individual strings and repeat the algorithm for each sentence. We keep only the one sentences that scores the most thus most likely contain the definition or answer that user is looking for.

An additional (optional) third step can be applied in terms of a selection filter which allows user to narrow down the searching area by defining subject to his question belongs to.

B. *Structure Anchoring Algorithm*

Structure Anchoring Algorithm is the main algorithm in the *natural language processing module*. It is also used in the *trial & error learning module*. For three given strings (S1, S2, S3) it will try to "morph" S1 based on how you can construct S3 from S2. It does this by finding the biggest common substring found in S1, S2, and S3. A suffix tree can be used to find substrings fast and efficiently. This depends on how large the string is (i.e. if it pays off to construct a tree first) and then the transversal is in O(n) linear time.

During The next step system anchors this common information and observes everything left and right of the anchor point. First it anchors the sample input S2 and the sample output S3. Then it looks for words (left of the anchor) that have been moved right of the anchor. If it finds only one word that does this, it anchors string S1 and moves everything left from S1's anchor point to the right. The same algorithm is applied for words that have been right of the anchor point and which migrated to the left.

Here is an example situation where Structure Anchoring is applied.

S1 = "Who **is** Andrew?"

S2 = "Who **is** Robert?"

S3 = "Robert **is** a person"

The anchor in this example would be the word **is** because that's the only word common in all 3 strings at once. Next it looks if any word left of the anchor point in S2 is contained right of the anchor point in S3. This is not the case so it moves on checking for the opposite case (if any word right of the anchor point in S2 is contained in S3).

The word "Robert" fulfills these criteria thus everything that is right of the anchor in S1 is moved to the left. Finally, system joins these strings as follows {newLeft} {anchor} {new Right} to get the newly morphed S1: "Andrew is".

It is important to note that there is no way to turn the string into "Andrew is a person" because system at this point is unable to conclude that the word Andrew has any contextual similarity with the word Robert. However, its content is good enough for the pseudo contextual search algorithm which will most probably produce better result.

Although the algorithm seems rather simple and inefficient, it is able to increases the rate at which the system finds an appropriate response. A neural network would probably be a better solution and produce much better learning curve.

V. IMPLEMENTATION

In order to demonstrate the practical application of the developed algorithm we implemented basic version of Scarlet agent shown in Figure 2. Current version contains three main operating modes: Web, FTP and Utility.

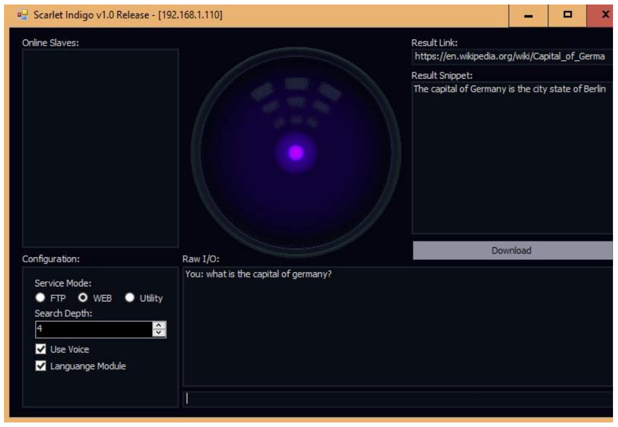
In the first operating mode (Web) agent is configured to search the Internet. After the user enters a specific question agent permutes the input string so that it can recognize the form of the appropriate answer, for example:

>> Input: What is the capital of France?

>> Output: The capital of France is

This is followed by a second step where newly transformed string is entered into Google's search engine which returns the first N websites. Result set is then parsed by the pseudo contextual analysis algorithm which calculates how similar the transformed string is to the text that is received from Google. When the best source of information is determined, the same algorithm is applied to all sentences within the best results and return a sentence that have the highest score.

Figure 2. The Scarlet interface



Result Link: https://en.wikipedia.org/wiki/Capital_of_Germa

Result Snippet: The capital of Germany is the city state of Berlin

Raw I/O: You: what is the capital of germany?

Configuration:

Service Mode: FTP [✓ WEB] Utility

Search Depth: 4

✓ Use Voice

✓ Languange Module

FTP operating mode uses the same principle but with a difference that files represent source of information and are placed on the FTP server. It is important to emphasis that the used algorithm works the same for any language.

Utility operating mode is basically the same algorithm that compares the input string with fixed template string and based on "context" execute various commands like opening or closing programs, searching YouTube etc.

VI. CONCLUSION

This Scarlet system proved to be very effective in contextual information analysis thus in returning the proper document that is adequate to the subject. It can differentiate between different subjects (Databases, Programming, Communication Technologies etc.) due to the different context and keywords.

The case where the system has found not only the right subject, but the right lecture is also very high. It mostly returns the exact definition the user asked for (not using the learning module which would yield even better results).

In addition to this the system can also search the web applying a slightly transformed algorithm (get list of best matching pages, get best matching page, get best matching content)

REFERENCES

[1] S. T. Piantadosi, "Zipf's word frequency law in natural language: A critical review and future directions", Psychonomic Bulletin & Review, vol. 21, 2014, pp.1112-1130.

[2] Rohini Srihari, Wei Li., A Question Answering System Supported by Information Extraction, ANLC '00 Proceedings of the sixth conference on Applied natural language processing, (2000), pp. 166-172

[3] Morrissey J., Zhao R., R/quest: A Question Answering System. In: Larsen H.L., Martin-Bautista M.J., Vila M.A., Andreasen T., Christiansen H. (eds) Flexible Query Answering Systems. FQAS 2013. Lecture Notes in Computer Science, vol 8132. (2013), Springer, Berlin, Heidelberg