HOW GREEK THE WEB IS

S. Athanassopoulos[1], C. Kaklamanis[1], P. Katsikouli[2], E. Papaioannou[1]

[1] University of Patras & CTI "Diophantus" (GREECE)
[2] University of Patras (GREECE)
athanaso@ceid.upatras.gr, kakl@ceid.upatras.gr, katsikouli@ceid.upatras.gr,
papaioan@ceid.upatras.gr

Abstract

Internet, apart from a huge repository of information of any kind, has become the main means of modern communications and World Wide Web has emerged as a new sort of society since it usually reflects almost all aspects of modern societies in terms of their economic, political and social status and structure. Therein, over wired and wireless connections, through ingenious ideas, i.e., algorithms, that exploit the enclosed computational power, a new kind of culture emerges combing elements from existing traditional civilizations/cultures like for instance history, arts, science and technology, education, language… Motivated by the fundamental and influential nature of the Greek language, our paper investigates its influence in written texts hosted in the World Wide Web. Otherwise stated, our work addresses the question: How Greek the Web is?

Our approach lies in automatically detecting and measuring the frequency of words of Greek origin in user-selected URLs; we focused on URLs including English text – but our work can be (easily) extended to URLs containing text in other languages. To this aim, we designed and implemented using python a cultural algorithm which, starting with a small collection of Greek lemmata and exemplars, is able to automatically generate and recognize new lemmata and English words of Greek origin in web texts. In addition, we designed and implemented a python-based application which using our cultural algorithm analyzes user-selected web texts in terms of content of Greek origin and visualizes analysis results. The application has been tested on a collection of web texts coming from education, development, science and technology indicating that, on average, 10% of the English words used is of Greek origin.

Keywords: cultural algorithms, python, culture, language, Greek.

1 INTRODUCTION

During the last decades, we have witnessed the emergence of the World Wide Web as a world-wide repository of information and as one of the main means of modern communication. Internet-connected computing entities - possibly mobile, with varying computational capabilities, connected among them with different communication media - are globally available and able to provide to their users various high-level services that make use of their aggregated computational power, storage space, and information resources.

The explosive growth of the Internet, together with the advances in research and technology, has given rise to the emergence of a new kind of society which reflects the economic, political and social status and structure of modern societies and a new kind of culture which combines and integrates elements from existing traditional civilizations/cultures like for instance history, arts, science and technology, education, language…

The interconnection between language and culture and how language and culture influence each other has been the subject of several studies on linguistic relativity aiming to define how a language affects the ways in which its speakers conceptualize their world (see for example [1]).

Since, information is usually available in the Internet mainly in textual form, an interesting question arises: can linguistic influences be detected in this internet-based emerging culture and society?

Motivated by the fundamental and influential nature of the Greek language (see for example the two speeches of Xenophon Zolotas, a Greek economist who served as an interim non-party Prime Minister of Greece in 1989, in English which are considered to be historic because they contained only terms of Greek origin), our paper investigates the influence of the Greek language in written texts hosted in the World Wide Web.

Our approach mainly lies in automatically detecting (i.e., mining) and measuring the frequency (i.e., the percentage) of words of Greek origin in user-selected web texts; we have focused on URLs including English text – but our work can be (easily) extended to URLs containing text in other languages. The methodology we used consists in (i) the creation of a collection of Greek lemmata and exemplars, (ii) the design and implementation of a cultural algorithm which starting from an initial set of lemmata and words of Greek origin collectively learns and detects such words in URLs and (iii) the design and implementation of a user interface which visualizes and presents obtained results.

Cultural algorithms can be seen as an extension of conventional genetic algorithms and were introduced by Reynolds [2]. They form a branch of evolutionary computation where there is a knowledge component called the belief space, a population component, an acceptance function and an influence function ([2], [3]). The best individuals of the population can update the belief space via the update function. The knowledge categories of the belief space can affect the population component via the influence function; for example, by altering the genome or the actions of the individuals [3]. In our work, cultural algorithms have been used for building a learning system able to generate and recognize words of Greek origin in WWW texts.

Our application has been implemented using python, a free, open-source programming environment that supports a clear and expressive syntax offering implementation efficiency and fast execution.

The experimental evaluation of our application using a collection of URLs from the fields of technology, education and development confirms its learning efficiency and correctness in the qualitative characterization of URLs regarding the Greek flavor of their textual content. Such a system could be used in practice as a tool for educational and/or scientific purposes which could include sociological/cultural studies as well as online evaluation of language skills.

The rest of the paper is structured as follows: In Section 2, we survey previous relevant work. In Section 3, the main algorithm of the paper is presented in detail. Section 4 includes obtained results and discusses experimental evaluation. Conclusion and future work are addressed in Section 5.

2 PREVIOUS RELEVANT WORK

Our work falls into the area of text mining, i.e., the discovery by a computer of new, previously unknown information, by automatically extracting information from a usually large amount of different unstructured textual resources. Processing unstructured (textual) information in order to extract meaningful (numeric) indices from the text, and, thus, make the information contained in the text accessible to various statistical and machine learning algorithms has attracted research attention in the area intersecting computational linguistics and natural language processing (NLP). There is a long recent bibliography on theoretical and applied approaches to text mining and its relation to computational linguistics and natural language processing; complete surveys can be found in [4] and [5].

The web, as the biggest pool of unstructured textual information, has served as a very challenging testing platform for automated text mining and categorization (see for example [6], [7], [8], [9]). Mining words of a particular language in URLs has been closely related to automatic translation (see for example [10], [11]) and to corpus analysis of particular languages like Arabic [12], Chinese [13], Turkish [14].

Genetic algorithms have been traditionally used for information extraction and text mining [15]. Cultural algorithms, in particular, have been suggested for the evaluation of classification rules in evolutionary and rule mining systems [16] as well as for more general data mining tasks [17].

To the best of our knowledge, a rich literature on computational linguistics does not contain any work on automatic mining of words of Greek origin (neither of other specific origin) in web texts using cultural algorithms.

Python is a simple yet powerful programming language with excellent functionality for processing linguistic data; it is open source and can be downloaded from http://www.python.org/. Natural Language Toolkit (NLTK) defines an infrastructure that can be used to build NLP programs in python. It provides basic classes for representing data relevant to natural language processing; standard interfaces for performing tasks such as part-of-speech tagging, syntactic parsing, and text classification; and standard implementations for each task that can be combined to solve complex problems. Python programming language together with the NLTK open source library have been recently adopted as an appropriate framework for natural language processing and computational linguistics for text analysis and text mining (see [18] for a complete survey).

3 OUR APPROACH

In order to analyze a URL and determine the percentage of words of Greek origin it contains, we adopted the following approach: first, we created a collection of Greek lemmata; then, we devised a cultural algorithm which collectively learns and detects words of Greek origin in URLs. One could claim that it would be preferable to use, instead, a simple search/match approach in a lexicon containing English words of Greek origin. However, on the one hand, it is rather doubtful whether there is such a complete lexicon; on the other hand, our objective has been to suggest a learning algorithm, which starting with just a *small set* of Greek lemmata and exemplars will be able to automatically generate and recognize new lemmata and English words of Greek origin.

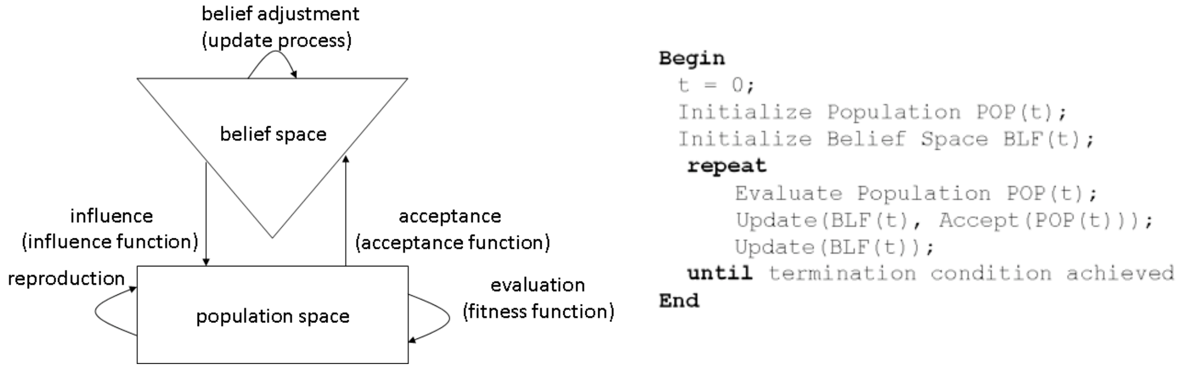
3.1 Collection of Greek lemmata

In order to determine words of Greek origin in English web texts, prefix/suffix analysis has been used. This is because, in vocabularies of foreign languages, many words of Greek origin usually have characteristic prefices and/or suffices. A rich collection of Greek lemmata, called herein GLC, has been produced based on [19]. GLC has been partitioned into three mutually disjoint sets: (i) c_1 which contains characteristic and frequently used Greek prefices (e.g., auto-, tele-, etc), (ii) c_2 which contains characteristic and frequently used Greek suffices (e.g., -logy, -pathy, etc) and (iii) c_3 which contains prefices/suffices that appear in English words which do not necessarily result form words of Greek origin (e.g., in-, -ous, etc).

Furthermore, an additional set of English words of Greek origin, called herein EGW, has been produced and used as an auxiliary collection of Greek lemmata. EGW contains a small set of words selected uniformly at random from the collection of [19] to be used as exemplars in our analysis.

3.2 The cultural algorithm

Cultural algorithms [2], [3], [20] are evolutionary algorithms inspired from societal evolution. They involve a belief space, a population space and a communication protocol which provides functions that enable exchange of knowledge between population and belief space (see Fig. 1).



```
Begin
  t = 0;
  Initialize Population POP(t);
  Initialize Belief Space BLF(t);
   repeat
      Evaluate Population POP(t);
      Update(BLF(t), Accept(POP(t)));
      Update(BLF(t));
   until termination condition achieved
End
```

(a) (b)

Fig. 1: Part (a): The cultural algorithm components include a belief space and a population space which interact through a communication protocol. Part (b): basic pseudocode for cultural algorithms.

The belief space of a cultural algorithm is divided into distinct categories which represent different domains of knowledge that the population of the search space has (like for example, *normative knowledge*, i.e., a collection of desirable value ranges for the individuals in the population component, *domain specific knowledge*, i.e., information about the domain of the particular problem the cultural algorithm is applied to, *situational knowledge*, i.e., specific examples of important events like for example successful/unsuccessful solutions, *temporal knowledge*, i.e., history of the search space, *spatial knowledge*, i.e., information about the topography of the search space). The population space consists of the individuals involved in the problem.

Cultural algorithms require an interface, i.e., a communication protocol, between the population and belief space. Each time the algorithm runs, the belief space is updated by the best individuals of the population: a fitness function evaluates the performance of each individual in the population, much like in genetic algorithms. Then, the best individuals are selected by the acceptance function and update the belief space.

The knowledge categories of the belief space can affect the population component via the influence function, which can affect population by altering, for example, the genome or the actions of the individuals. This influence process defines the dual inheritance property of cultural algorithms i.e., the transmission of information between the population and the belief space, that distinguishes them from genetic algorithms, which do not involve a belief space and therefore do not recognize the transmission of information between population and belief space.

In our work, cultural algorithms have been used to build a learning system able to recognize words of Greek origin in web texts. Below, we discuss the functionality of each component of our cultural algorithm as well as relevant implementation details.

3.2.1 Components of our algorithm

In our framework, the initial population consists of the English words contained in a user-selected URL. Each word is an individual, appearing in the initial population as many times as it appears in the URL. The initial belief space involves two knowledge categories: normative and situational knowledge; normative knowledge results from the GLC set, situational knowledge results from words in the set EGW.

Our cultural algorithm works in rounds. During each round, individuals of the population are evaluated and the best of them are selected to influence the belief space, which is then updated accordingly. In particular, after a pair of successive evaluation-selection actions is performed, the population is influenced through the removal of words of Greek origin detected; these words together with detected prefices/suffices indicating Greek origin are then used to update the belief space. The algorithm terminates when all individuals in the population are deleted either because they were moved into the belief space as individuals indicating Greek origin or because they were characterized as words of non-Greek origin.

In each round, four functions are used by our algorithm, namely fitness function, acceptance function, influence function and update function; they work as follows.

The fitness function examines “how Greek” a word in the input web-text may be. In particular, the fitness function is used to assign a score to each word of the population. Scores are defined as points assigned to each word of the population according to the following rule: words with a prefix or suffix in GLC (sets c_1, c_2, respectively) as well as words related (themselves or their derivatives) to the auxiliary set of exemplars, EGW, are assigned 1 point; words with a prefix or suffix in GLC (set c_3) are assigned 0.5 points; all other words receive no points.

Then, based on the score each word of the population received by the fitness function, the acceptance function selects individuals with the best behavior, i.e., it selects words that are most likely to indicate Greek origin. In particular, it classifies words in two groups, A and B, according to the score they received by the fitness function; group A contains words that received 1 point while group B contain words that received 0.5 points. Words in group A are considered to be words of Greek origin and are removed from the population. Further evaluation/test is required for words in group B. All other words (i.e., words with no points) remain in the population for further processing for one more round.

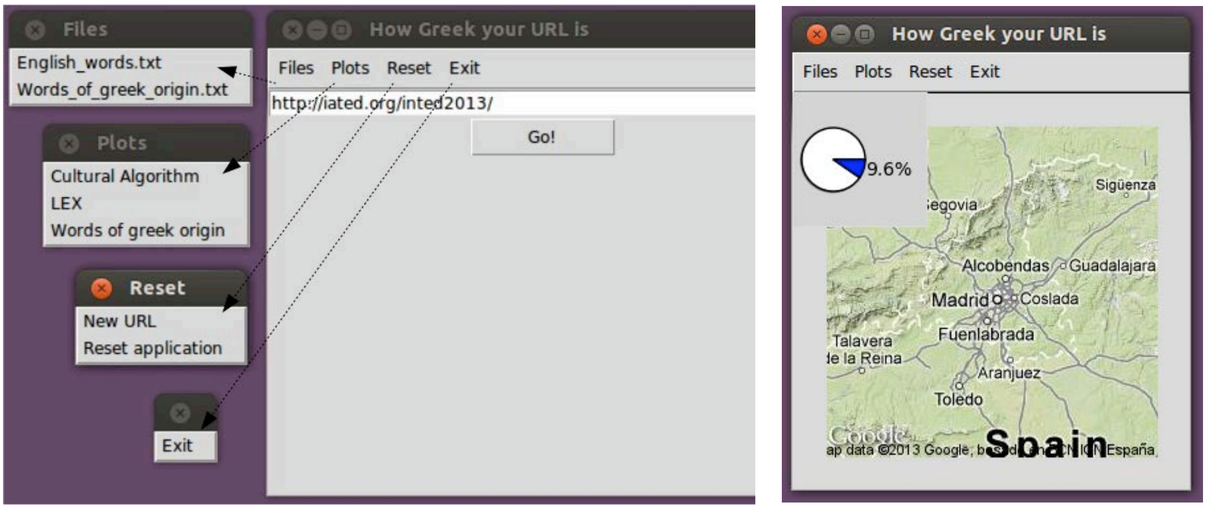
The influence and update functions are used to enrich the normative and situational knowledge of the belief space (i.e., the GLC and the auxiliary set EGW, respectively) and, thus, influence the next generation of the population which will be re-examined during subsequent rounds against an updated collection of words of Greek origin. In particular, the auxiliary set EGW is extended to include words of group A; the GLC is extended through the addition of a new set, c_4, containing newly discovered prefices and suffices. Then, words in group B are re-examined for determining whether they have a prefix or suffix in c_4 or match a word in the extended set EGW; if a word in group B meets at least one of these conditions, it is moved in group A (and therefore it is removed from the population); otherwise, it remains in the population for further processing for one more round.

The solution returned by our algorithm, i.e., the English words of Greek origin contained in a given web text, essentially contains words in group A. Notice that, words that have not been characterized as words of Greek origin after two rounds of processing are removed from the population. This condition is rather heuristic and has been adopted based on experimental evidence; this evidence indicates that additional processing rounds increase execution time but do not increase performance of the algorithm accordingly, i.e., only a small number of additional words of Greek origin is further mined.

4 RESULTS AND EXPERIMENTAL EVALUATION

For running the application and the experiments, we have used an Intel(R) Core(TM) i7 CPU at 2.80GHz (with 3.24GB RAM) and Ubuntu 12.04 Linux distribution Operating System (open source). Our application has been implemented using python ([18], [21]), a free, open-source programming environment designed and implemented in the late 1980s by the Dutch Guido van Rossum. Python is essentially a scripting language which is interpreted, interactive (i.e., it facilitates interactive exploration) and object-oriented (i.e., it permits data and methods to be encapsulated and re-used easily). It supports a clear and expressive syntax, it offers fast execution and implementation efficiency allowing programmers to develop software more quickly through the high-level data types [18]. Python supports all traditional variable types and provides modules and libraries for the string type, which is highly used in our work that involves processing of text and words. We chose python because it is a simple yet powerful programming language with excellent functionality for linguistic analysis [18] as it supports the Natural Language Toolkit (NLTK) i.e., toolkit with modules and functions for text and linguistic analysis. In our work, we used python version 2.7 and NLTK version 2.0.

The interface has been implemented using the Tkinter (version 2.7.3) basic python graphical library. As shown in Fig. 2, the interface consists of a menu, an input box and a canvas. The menu provides four options: 1) Option *Files* provides access to the file containing English words of the input URL as well as to the file containing words of Greek origin detected by our algorithm, 2) option *Plots* can be used for graphical presentation (through pie-charts) of the results of the linguistic analysis of a URL in terms of the words of Greek and non-Greek origin it contains, 3) option *Reset* allows the analysis of a new URL; the existing belief space can be either maintained or reset and 4) option *Exit* allows termination of the application. The input box is used for entering the input, i.e., the URL to be analyzed in terms of words of Greek origin. Results of the analysis are presented in the canvas through (i) a map showing the country hosting the IP address of the input URL and (ii) a pie-chart showing in blue the percentage of words of Greek origin detected by our algorithm in the input URL.



(a) (b)

Fig. 2: User interface: menu options and input box (a) and canvas (b).

The analysis of a URL is conducted in terms of a sequence of four actions: pre-processing, Greek-word mining, evaluation and visualization of obtained results. During the pre-processing phase, the IP of a URL provided as input to the application is syntactically validated and the URL is decomposed into elementary strings through the use of regular expressions and string functions. Textual content of the URL is extracted through removal of html special characters via NLTK functions. English words of the obtained text are separated from words of other languages through the use of string functions, python embedded English dictionaries and regular expressions; they are then provided as input to our cultural algorithm which essentially performs the mining of words of Greek origin.

For the evaluation of our cultural algorithm, the following framework has been used: first, a lexicon of English words of Greek origin has been produced mainly based on the collection of [19] with the addition of extra words and derivatives. Derivative words have been produced mainly through word stemmers i.e., algorithms that reduce inflected or derived words to their base or root form, from the NLTK library. In particular, we have used the lancaster stemming algorithm (http://nltk.org/_modules/nltk/stem/lancaster.html) and one of its significant variations: the porter stemming algorithm (http://nltk.org/_modules/nltk/stem/porter.html). Then, English words contained in the input URL are examined against the lexicon in order to get the percentage, denoted by LEX, of English words of Greek origin in the input URL. The percentage of English words of Greek origin detected in the input URL by our algorithm is denoted by CA. The evaluation is then performed based on the comparison of CA and LEX.

Evaluation results are visualized through pie-charts produced using the matplotlib.pyplot mathematical library (http://matplotlib.org/), which include matlab plot functions and can be manually imported to the python environment. Three types of pie-charts can be plotted: (1) CA, i.e. the percentage of English words of Greek origin mined by our algorithm in the input URL, (2) LEX, i.e. the percentage of English words of Greek origin detected in the input URL through the use of the lexicon and (3) the ratio CA/LEX showing how our algorithm performed in detecting words of Greek origin (Fig. 3).

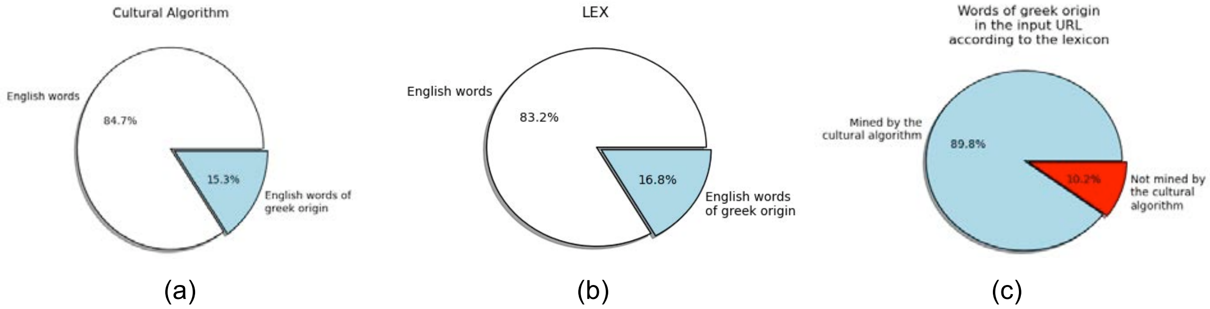


Fig 3: Visualization of evaluation results: (a) percentage of English words of Greek origin mined by our algorithm in the input URL, (b) percentage of English words of Greek origin detected in the input URL through the use of the lexicon and (c) how well our algorithm performed.

The performance of our cultural algorithm actually reflects its ability to mine English words of Greek origin from web texts through learning. Evaluation of the performance of the algorithm is done through the comparison of CA, which is the percentage of English words of Greek origin detected in the input URL by our algorithm, and LEX, which is the percentage of English words of Greek origin in the input URL determined using a predefined lexicon. Our algorithm has been used to analyze a collection of 80 URLs from the fields of technology, education, science and development. Fig. 4, shows to which extend words of Greek origin appear in our URL collection, according to both the lexicon and our culture algorithm.

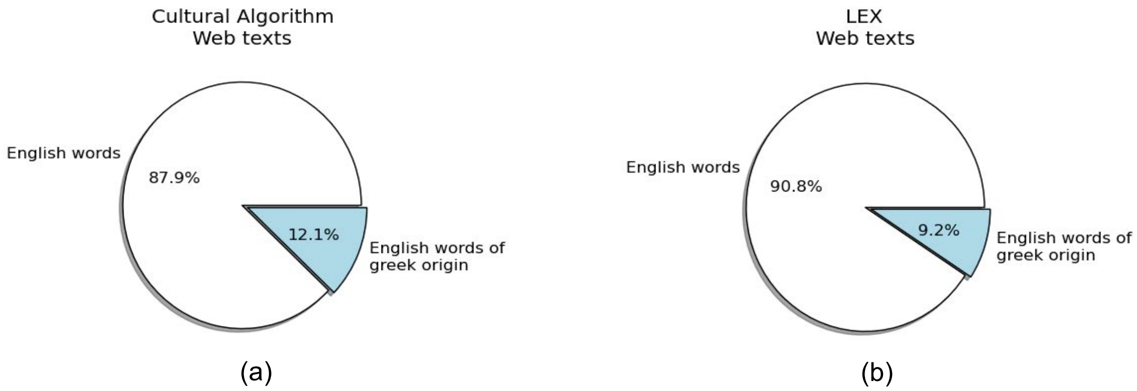


Fig. 4: “How Greek our URL collection is” according to our Cultural Algorithm (a) and the LEXicon (b).

Fig. 5 shows the distribution of words of Greek origin per URL category; each category contains results for texts included in 20 URLs. So, regarding our initial question on the presence of Greek language in the web, our study shows that, on average, one out of ten words appearing in modern, contemporary English web texts is of Greek origin: this makes indeed an impressive percentage.

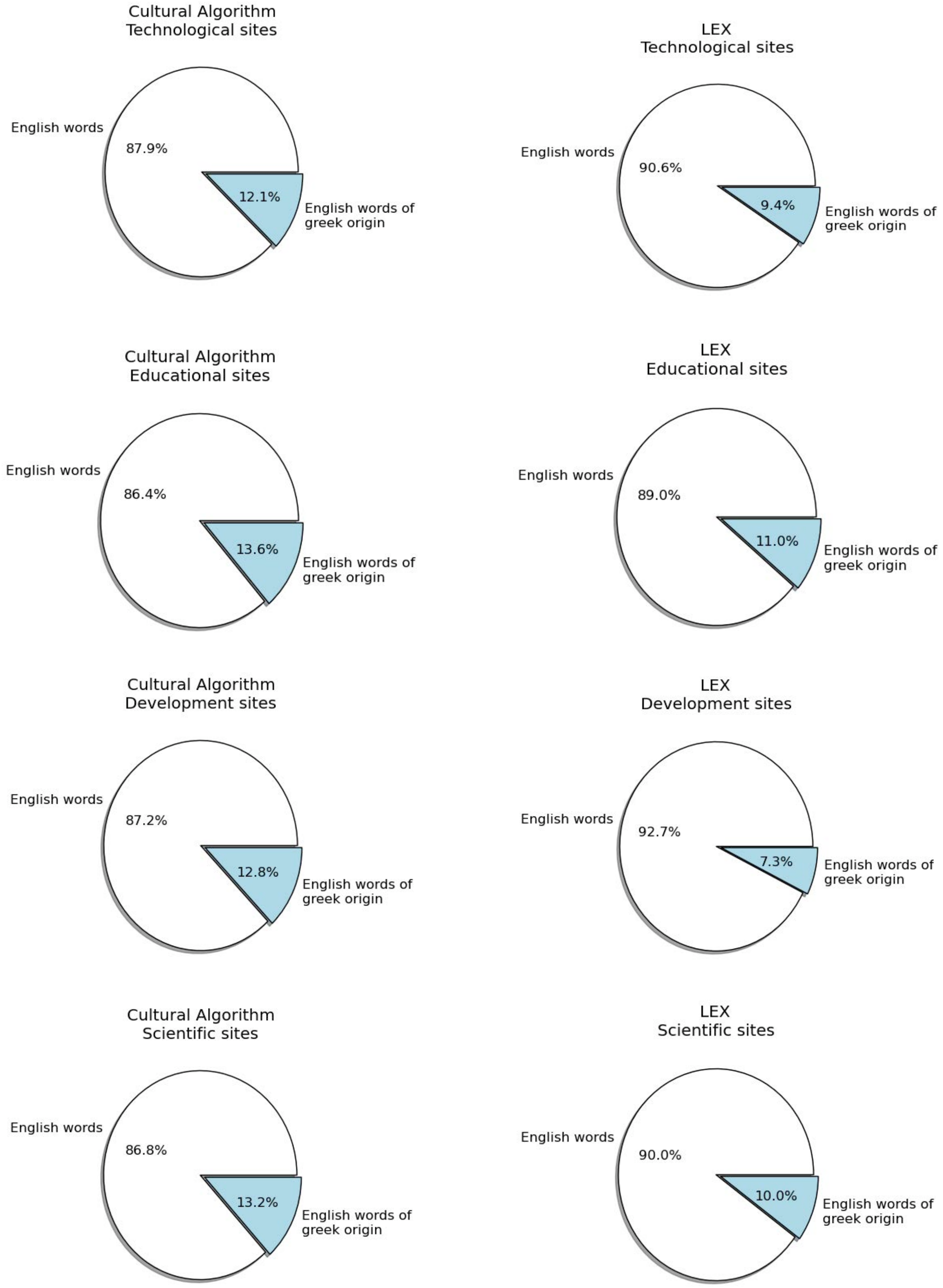


Fig. 5: The distribution of words of Greek origin per URL-category (20 URLs/category).

In general, our algorithm performs well with regard to the actual number of words of Greek origin contained in a given web text. However, there are cases where our algorithm erroneously reports words as ones of Greek origin. As observed, this is mainly due to the following reasons: (i) misleading presence of a Greek-like lemma, e.g. because of the "-ous" suffix that usually leads to words of Greek origin, "previous" is erroneously reported as a word of Greek origin, (ii) the prefix/suffix analysis sometimes fails to detect words of Greek origin that do not have a characteristic lemma but can only be mined through composite words or through the set of exemplars, e.g. the prefix/suffix analysis fails to detect "artery" as a word of Greek origin, (iii) not all words of Greek origin are included in the lexicon we used for evaluation purposes; as a consequence words of Greek origin are not always reported as such, e.g., we observed that while "maths", shortcut for "mathematics", was not included in this lexicon despite that it is obviously a word of Greek origin.

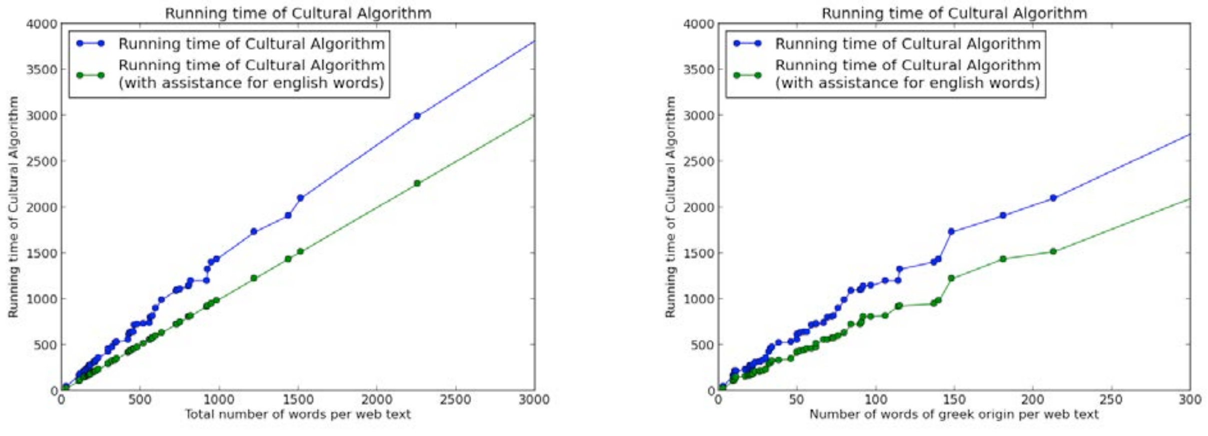
Fig. 6 illustrates the execution time of our algorithm. In Fig. 6(a), x axis shows the web texts used for experimentation (80 in total) ordered according to the number of words they contain, while y axis presents corresponding running times. In Fig. 6(b), the running time of our algorithm is addressed on the basis of the total words of Greek origin included in each web text of our collection. It can be easily observed (as indicated by the blue lines in Fig. 6) that an increased number of words (or words of Greek origin) in a web text increases the execution time of the algorithm.

The execution time of the algorithm depends on (i) the initial size of the population and (ii) number of words that must be processed more than once before they are removed from the population as words of non-Greek origin. A possible decrease in the execution time can, thus, be achieved by decreasing processing rounds per word or by eliminating unnecessary processing of words of non-Greek origin.

Our algorithm allows at most two processing rounds per word before it decides to characterize it as a word of non-Greek origin; this is due to strong experimental evidence according to which more than two processing rounds per word result in a significant increase in execution time without a corresponding increase in the number of words of Greek origin detected.

Assuming an initial population consisting of n words, the execution time of our cultural algorithm ranges from n (when all words of the initial population are of Greek origin) to 2n (when every word of the initial population is a word of non-Greek origin and, therefore, is examined at most twice).

In order to eliminate unnecessary processing rounds per word, we implemented the following trick: in addition to "teaching" the algorithm which words to consider as words of Greek origin, we provided it with further information about words that should be directly classified as words of non-Greek origin. In particular, a list containing words of non-Greek origin has been created so that before a word is examined for a second time by our algorithm, it is first checked against words in this list: if there is a match, the word in question is immediately removed from the population (as a word of non-Greek origin) and no further processing is required. Indeed, this idea results in a decrease in execution time, as shown by the green lines in Fig. 6, which is given by the formula f(n)=1.01n, assuming an initial population of n words.



(a)

(b)

Fig 6.: Execution time of our Cultural Algorithm.

5 CONCLUSION AND FUTURE WORK

The paper presents a python-based application which automatically learns and mines words of Greek origin in textual information contained in user-selected web texts written in English. Mining is performed by a cultural algorithm whose learning ability is based on prefix/suffix analysis and a small initial set of exemplars. Texts written in other languages could also be analyzed using appropriate lexicon and set of exemplars.

Our experimentation with a collection of 80 URLs from education, development, science and technology confirms its learning efficiency and correctness in the qualitative characterization of URLs regarding the Greek flavor of their textual content. Regarding the presence of Greek language in the web, our study shows that, on average, one out of ten words appearing in modern, contemporary English web texts is of Greek origin, which makes indeed an impressive percentage. Our application could be used in practice as a tool for educational and/or scientific purposes which could include sociological/cultural studies as well as online evaluation of language skills.

Issues to be addressed in future work include (i) extension of the prefix/suffix collection (GLC) and refinement of the set of exemplars (EGW) in order to improve the initial belief space of our cultural algorithm as well as (ii) investigation of more involved analysis techniques to be used in combination with prefix/suffix analysis in order to improve the learning efficiency of our algorithm. Furthermore, in order to achieve a more accurate evaluation of results, a sophisticated lexicon produced by experts in linguistics should be used.

REFERENCES

[1] Leavitt, John (2011). Linguistic Relativities: Language Diversity and Modern Thought. Cambridge, UK: Cambridge University Press, ISBN 978-0-521-76782-8.

[2] Reynolds R.G. (1979). An adaptive computer model of the evolution of agriculture for hunter-gatherers in the valley of Oaxaca, Mexico. Ph.D. Thesis, University of Michigan, Ann Arbor.

[3] Best C. Xiangdong Che, Reynolds R.G. and Dapeng Liu (2010). Multi-objective cultural algorithms. Congress on Evolutionary Computation (CEC), IEEE, pp.1-9.

[4] Manning C. and Schütze H. (1999). Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA, USA.

[5] Indurkhya, N., and Damerau, F. (2010). *Handbook Of Natural Language Processing*, 2nd Edition. Boca Raton, FL: CRC Press. ISBN 978-1-4200-8592-1

[6] Sebastiani F. (2006). Classification of text, automatic. In Keith Brown (ed.), The Encyclopedia of Language and Linguistics, Volume 14, 2nd Edition, Elsevier Science Publishers, Amsterdam, NL, 2006, pp. 457-462.

[7] Suchanek F.M., Ifrim G. and Weikum G. (2006). Combining linguistic and statistical analysis to extract relations from Web documents. In Proc. of the 12th International Conference on Knowledge discovery and Data mining, ACM, pp.712-717.

[8] Abbasi A., Chen H. and Salem A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. ACM transactions on Information Systems (TOIS), Vol. 26, Issue 3, no.12.

[9] Sebastiani F. (2002). Machine learning in automated text categorization. Journal of Computing Systems (CSUR), ACM, Vol.34, Issue 1, pp.1-46.

[10] Fortuna B., Shawe-Taylor J. (2005). The use of machine translation tools for cross-lingual text mining. In Proceedings of the Workshop on Learning with Multiple Views, 22nd ICML, Bonn, Germany.

[11] Latiri C., Smaili K., Lavecchia C. Nasri C., Langlois D. (2011). Phrase-based Machine Translation based on Text Mining and Statistical Language Modeling Techniques. International Journal of Computational Linguistics and Applications, Vol. 2, No. 1–2, pp. 193–208.

[12] Al-Diabat M. (2012). Arabic Text Categorization Using Classification Rule Mining. Applied Mathematical Sciences, Vol. 6, no. 81, pp. 4033 - 4046.

[13] Jia Z. (2010). Web Text Categorization for Large-scale Corpus. In Proceedings of the 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), vol. 8, pp. 188-191.

[14] Yildirim S., Yildiz T. (2012). Corpus-Driven Hyponym Acquisition for Turkish Language. In Proceeding of the 13th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2012), LNCS 7181, Springer, pp. 29-41.

[15] Mitchell M. (1996). An Introduction to Genetic Algorithms. MIT Press, Cambridge MA.

[16] Srinivasan S., Ramakrishnan S. (2012). Cultural Algorithm Toolkit for Multi-objective Rule Mining. International Journal on Computational Sciences & Applications (IJCSA) Vo2, No.4, pp. 9-23.

[17] Jin X. and Reynolds R. G. (2002). Data Mining Using Cultural Algorithms and Regional Schemata. In Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2002), IEEE Computer Society, Washington, DC, USA, p. 13.

[18] Bird S., Klein E., and Loper E. (2009). Natural Language Processing with Python. Published by O'Reilly Media, Inc., CA, USA, 482 pages, ISBN: 978-0-596-51649-9.

[19] Stefanides, A. (2010). You speak Greek, you just don't know it. Ianos publications, ISBN 978-960-6882-24-1.

[20] Reynolds R.G (1994). An introduction to cultural algorithms. In Proc. of the 3rd Annual Conference on Evolutionary Programming, World Scientific Publishing, pp 131–139.

[21] http://python.org/