Solving a bi-criteria integer linear programming model to enhance tweet contextualization

Fernando Paredes
Escuela de Ingeniería Industrial,
Universidad Diego Portales
Av. Ejército 441, Santiago de Chile
fernando.paredes@udp.cl

Javier Pereira
Escuela de Ingeniería Informática y Telecomunicaciones,
Universidad Diego Portales
Av. Ejército 441, Santiago de Chile
javier.pereira@udp.cl

Abstract

We propose a bi-criteria integer linear programming approach to support constructive searching of non-dominated document passage assemblies in tweet contextualization, formulating this problem in terms of coverage of a given set of interesting terms in a tweet taken from Twitter, with the minimum number of passages and minimum redundancy in a feasible assembly. We show that resolution of this problem could be very efficient, without the disadvantages of problem size or sparse information data structure, as found in precedent works, where the same formulation was proposed.

KEYWORDS: bi-criteria approach, tweet contextualization, efficient algorithm

1 Introduction

In a previous work [15], we proposed a bi-criteria integer linear programming approach to support exploratory search of a software architect, aiding him/her to identify a restricted list of non-dominated interesting assemblies, i.e. sets of components. Observing that the most relevant components do not necessarily correspond to those satisfying the largest number of properties, but rather to those combining well with other components, we proposed an enumeration algorithm which provides a list of all potentially interesting assemblies. This allowed us to provide a restricted sublist containing potential candidates which may be recommended to the architect. Our problem was formulated as the identification of component-sets covering the whole set of expected architectural properties, minimizing the number of components and redundancy of non-dominated sets.

In a different version of this problem, we explored a portfolio constructive model, formulated in terms of satisfaction of a given set of technical requirements, with the minimum number of projects and minimum redundancy [16]. An algorithm issued from robust portfolio modeling was adapted to a vector model, modifying the dominance condition as convenient, in order to find the set of non-dominated portfolios, as solutions of the same kind of bi-criteria integer linear programming model proposed in the case of component assemblies mentioned above. Therefore, we proposed a heuristic process finding a first optimal solution, on a mono-criteria version of this problem, which was further used as a first feasible solution aiding to find new non-dominated solutions. Numerical examples showed that this heuristic improves computational efficiency of the original algorithm and it is highly efficient for small-size problems.

In the present article, we show that the proposed models and algorithms may be used in a completely different situation, looking for contexts of opinions expressed in Twitter, that is *tweet contextualization*. In fact, search engines usually identify lists of documents that may be relevant to a user's information need. Thus, locating the relevant information is left to the user. Nevertheless, in most domains (digital libraries, maintaining assistance, recommender systems, among others) users are not necessarily interested on single artifacts or lists of ranked items; their needs are more oriented to general objectives, searching for elements aiding to compare, interpret, aggregate, analyze, synthesize and discover knowledge [9]. Users request explanations to sense-making processes and learning in a contextualized manner, making results transparent, examinable, trustworthy or satisfactory [17]. For instance, developments in digital libraries domain raise necessity for exploratory search used in contexts of multi-faceted, open-ended, persistent, opportunistic, iterative and multi-tactical information-seeking problems [10]. It seems that exploratory search systems help users to sense making, compelling for context awareness and understanding in support of decision-making, and to information foraging, i.e. searching for information pieces in a learning processes [11]. Context awareness is the achievement of perceiving elements of the environment, comprehending their collective meaning, and projecting their status into the near future.

Tweet contextualization is a kind of task concerning the discovering of Internet resources sets (documents, multimedia material, document passages, images, etc.) that best explain what a tweet of Twitter talks about. This is a task referred as Focused Retrieval, taking different forms: Passage Retrieval from a document corpus, Element Retrieval from an XML document, Page Retrieval from books, as well as Question Answering (see https://inex.mmci.uni-saarland.de/about.html). Here, we propose a model of context-awareness retrieval of resources in a restricted formulation, where only document passages are retrieved. Therefore, the problem is formulated as follows: "given a new tweet, the system must provide some context about the subject of the tweet, in order to help the reader to understand it; this context should take the form of a readable summary, not exceeding 500 words, composed of passages from a provided Wikipedia corpus". This problem is formulated as a challenge in the INEX 2012 Tweet Contextualization Track (see https://inex.mmci.uni-saarland.de/tracks/qa/).

This article is organized as follows. In Section 2, the problem stated is modeled as a bi-criteria integer linear program. Next, an algorithm grounded on a constructive approach is presented in Section 3. In Section 4 our approach is tested and results are compared to those obtained in a previous work. Finally, Section 5 is dedicated to conclusions.

2 Model

Let us consider a Twitter user searching for different interesting combinations of document passages that best explain the speech context of selected tweets. Let us also assume that a repository of document passages is available such that each tweet considered by this active user acts as a *query* on such collection. Therefore, the problem consists of automatically identifying the set of passages explaining the inherent semantics of a given tweet, knowing that a systematic manual exploration may be a very hard task, even not affordable.

In general, we know that a tweet is a 140 characters maximum string containing different kinds of words, expressions (individual names, slang words, branding phrases, product trade marks, geographical or temporal anchors, etc.) and special symbols (hashtags, user nicknames, webpage links, etc.). Indeed, we are going to assume that a Natural Language Process Engine (NPLE) is available such that a tweet is pre-processed, extracting from it only the interesting words and symbols. In fact, we could assume that a target vocabulary is defined *a priori*, which is used by the NPLE in order to identify words contained in a specific tweet, searching for literal meaning [4]. In another approach, we could assume that the NPLE is in charge of determining the latent semantic in a tweet. In this case, the focus of the vocabulary is the *context of mention* [13], that is the semantics entirely derived from content. Any entity mentioned in text can be surrounded by other words that provide context. For instance, the word "apple" might be mentioned in the context of either cooking or computing terms. Therefore, we use context also to disambiguate away polysemy, the existence of multiple word senses and thus the source of ambiguity. In any case, we assume that a subset of terms contained in a tweet is extracted in order to build a "representative" version of it. This subset will consist of a vector of terms $T = \{t_1, t_2, \ldots, t_m\}$.

Actually, the tweet contextualization problem will be formulated as the focused retrieval process where the interesting document passage assemblies are proposed to the active user. The main restriction of this process is that a feasible assembly must contain the whole set of terms in a tweet vectorial representation. Therefore, two criteria should be satisfied. First, in order to limit the size of an interesting assembly, it must contain the minimal number of passages as possible. Second, low-redundancy assemblies should be proposed, that is, those where any two passages on them minimize the common tweet terms. The problem formulated as the satisfaction of these two criteria, the tweet contextualization means finding the non-dominated assemblies among the whole set of possible sets of document passages built-up from the collection.

We have already formulated this problem as a bi-criteria integer linear program [15, 16]. In fact, let us consider:

T, a tweet represented by terms $t_1, t_2, \ldots, t_m$, $m = 1, 2, \ldots$;

$P = \{p_1, p_2, \ldots, p_n\}$, a the collection of passages p;

$\mathscr{P} = 2^P$ the set of possible assemblies, $a, b \in \mathscr{P}$, $f_i : \mathscr{P} \to \Re$;

$A = (a_{ji})_{n \times m}$, the matrix of information, that is the coverage of tweet terms by passages;

$a_{ji} \in \{0, 1\}$, the passage p_j coverage of term t_i;

$x_j \in \{0, 1\}$, such that $x_j = 1$ if p_j included in a feasible solution, and $x_j = 0$ if not;

y_i, number of times the i-th term is covered.

The following program is a model of our problem, where the whole set of non-dominated assemblies simultaneously minimize the number of passages in a composite and the assembly redundancy, i.e. the number of times the terms are covered by more than one passage.

$$\begin{aligned} &\min \sum_i y_i, \quad \min \sum_j x_j \qquad (1)\\ &st.\\ &\sum_j a_{ji} x_j = y_i, \forall i\\ &y_i \geq 1, \forall i\\ &x_j \in \{0, 1\}, \forall j. \end{aligned}$$

We have shown that this problem implies finding the Pareto front, which cannot be solved by a traditional approach [15], using an algorithm based on a constructive procedure, adapted from a preference programming approach [5]. This algorithm, as originally presented, allows the progressive generation of non-dominated assemblies, in a two phases procedure: generation of candidates and pruning.

3 Algorithm

The algorithm presented in this section allows the identification of the whole set of non-dominated passage sets. It applies an exhaustive approach [15] that bases on the candidates generation and a pruning process implementing a generation procedure of passage assemblies. At the current level of the procedure, any dominated or potentially dominated assembly is pruned. The algorithm is defined as follows,

```
$N_0 = \emptyset,\ L_1 = \{\emptyset\},\ \eta = MM;$
$For(k = 1, n, k++)do$
    $C_k = candidates(L_k)$
    $L_{k+1} = \{a \in C_k \mid \quad |Cov(a)| < |T|\}$
    $N_k = \{a \mid \nexists\, a' \in N_{k-1} \cup (C_k \setminus L_{k+1}) \quad such\, that \quad a' \succ a\}$
    $\eta = \min_{a \in N_k} \eta(a)$
$FunctionCandidates(C_k)$
    $L = \emptyset$
    $For \quad a \in C_k \quad do$
        $a = a \cup \{p_k\}$
        $If\ (\eta(a) \leq \eta)\, Then\, L = L \cup \{a\}$
    $Return\, L$
```

The function *Candidates* generates the candidate assemblies by progressively joining new passages to non-pruned assemblies coming from a previous round of the algorithm. An initial redundancy value is set at MM, a number big enough that will be modified the first time a feasible solution is found (i.e., an assembly covering T). Actually, the algorithm progressively generates non-feasible candidates until a feasible solution is found, which sets the first values for the minimal redundancy (η) and the assembly size.

We have shown that depending on the coverage structure of passages over terms, the initial values of minimal redundancy and assembly size could be identified after a very expensive searching process. Indeed, we have proposed different procedures to improve the efficiency of the resolution procedure [15, 16]. However, in the precise case of tweet contextualization the problem size may be very low. In fact, let us assume that a rough query is launched against the passage collection such that only the passages containing at least one term of the tweet are retrieved. Thus, the set P may be restricted enough as a few or relatively low number of components. Next, if a latent semantic approach is applied to build a vectorial representation of a tweet, then very few terms will be extracted, at the case, three or four. Under these conditions, it should be common to have instances of this problem very easy to solve using the proposed algorithms.

4 Some experiments

In order to test our approach, a group of instances have been defined. An instance corresponds to a set of n passages, m tweet terms and a coverage matrix $A^T_{m\times n} = (a_{ji})$, where $a_{ji} = 1$ if the passage p_j covers the i-th term and $a_{ji} = 0$, if not. In a previous work we have verified that algorithms are very sensitive to density of A [16], defined as the ratio of zero values to $(m \times n)$ cells. In fact, given an instance, thirty different matrices have been defined, filled in a random way, according to the expected density value. As a consequence, each instance has been simulated thirty times and the average time to solution, measured in milliseconds, has been calculated.

Average time to find the Pareto front for the algorithm with two data structures is presented in Table 1. Cases where no entry is shown for an instance mean that the respective algorithm does not respond in less than one minute (see below), that is the respective model is not capable to solve the problem in such time. The WUB columns refer to cases where the algorithm is applied without regard to the way that coverage is distributed in the information matrix. At the contrary, OWUB applies the algorithm over a pre-processed information matrix, resulting from convenient permutations that will allow finding of feasible solutions as soon as possible in the iteration procedure.

Table 1: Time in [ms], as a function of 0's density (δ)

Passages	**Terms**	$\delta = 0.5$		$\delta = 0.75$		$\delta = 0.8$	
		WUB	OWUB	WUB	OWUB	WUB	OWUB
25	4	1	0	1	0	6	0
50	4	11	0	6	0	5	0
100	4	90	2	61252	0		
200	4	764	11				

It is clear that the OWUB is the most efficient version. However, density of zeroes in the information matrix is a critical issue. Fortunately, a finest viewing on the way that passages are selected from the passage collection guarantees that high-density information matrices will not be built. In consequence, cases where $\delta \leq 0.5$ will be found. These are very promising results for the proposed problem.

Notice that, contrarily to the precedent works, the problem of tweet contextualization benefits from a convenient information data structure. It means that good behavior of algorithms is enhanced in this situation. However, some aspects remain out of our formulation: coherence among passages, trash remotion, syntax problems, etc. That will be treated in our future research.

5 Conclusions

In previous works, we proposed a bi-criteria integer linear programming approach to support constructive searching of non-dominated assemblies, formulated in terms of satisfaction of a given set of requirements, with the minimum number of components and minimum redundancy in a feasible assembly . This formulation was applied in the context of component-based software architecture, but also proved to be useful in project portfolios generation. In this article, we have proposed that the same approach could be very practical for tweet contextualization, that is, the task concerning the discovering of Internet resources sets (documents, multimedia material, document passages, images, etc.) that best explain what a tweet of Twitter talks about, from a given collection of document passages. Interestingly, the same problem structure of previous works may be used in this case, without the disadvantages of problem size or density, as found in our previous experiments.

However, it must be noticed that the problem of tweet contextualization benefits from downstream pre-processing tasks, looking for convenient problem structures, which is not the case for our precedent works. Further research needs to be done in order to include other criteria in this specific situation, for instance, coherence among passages, trash remotion, syntax problems, etc.

References

[1] Bouyssou, D., Marchant, T., Pirlot, M., Perny, P., Tsoukias, A. and Vincke, P., 2006. Evaluation and Decision Models. International Series in Operations Research & Management Science, Vol. 86, Springer-Verlag.

[2] Doerner, K., Gutjahr, W., Hartl, R., Strauss, C. and Stummer, C., 2004. Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection. Annals of Operations Research 131, 79-99.

[3] Ehrgott, M., Gandibleux, X., 2000. A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektrum 22, 425-460.

[4] Iwanska, L.M., 2000. Natural Language Is a Powerful Knowledge Representation System: The UNO Model. In: L.M. Iwanska and S.C. Shapiro (eds.), Natural Language Processing and Knowledge Representation, AAAI Press, Menlo Park, USA, 7-64.

[5] Liesiö, J, Mild, P. and Salo, A., 2007. Preference programming for robust portfolio modeling and project selection. *European Journal of Operational Research*, 181, 1488-1505.

[6] Liesiö, J, Mild, P. and Salo, A., 2008. Robust portfolio modeling with incomplete cost information and project interdependencies. European Journal of Operational Research 190, 679-695.

[7] López, C., Astudillo, H. and Pereira, J., 2007. Towards Robustness Analysis of Component-based Systems built on Imperfect Information 1st International Workshop on Living with Uncertainties (IWLU01), co-located with the 22nd International Conference on Automated Software Engineering (ASE), Atlanta, Georgia, November 6th 2007.

[8] Lourenço, J., Bana e Costa, C., 2009. PROBE - A multicriteria decision support system for portfolio robustness evaluation. Working paper LSEOR 09.108, ISSN 2041-4668 (Online).

[9] Maes, P., 1994. Agents that reduce work and information overload, Communications of the ACM, 37(7), 30-40.

[10] Mäkelä, E., Suominen, O. and Hyvnen, E., 2007. Automatic Exhibition Generation Based on Semantic Cultural Content. Proceedings of the Cultural Heritage on the Semantic Web Workshop at the 6th International Semantic Web Conference (ISWC 2007), Busan, Korea, November 12.

[11] Marchionini, G., 2006. Exploratory search: from finding to understanding. Commun. ACM 49(4), 41-46.

[12] Markowitz, H., 1952. Portfolio Selection. The Journal of Finance, March, 1952, 77 -91.

[13] Mehra, P., 2012. Context-aware Computing. IEEE Internet Computing 16(2), 12-16.

[14] Ogryczak, W., 2000. Multiple criteria linear programming model for portfolio selection. Annals of Operations Research 97,143-162.

[15] Paredes, F. and Pereira, J., 2010. A bi-criteria integer programming and algorithmic approach for software architecture alternatives modeling. ALIO-INFORMS Joint International Meeting, Buenos Aires, Argentina, June 6-9, 2010.

[16] Paredes, F., Pereira, J. and Candia, A., 2011. Towards a requirements-based bi-criteria approach to identify interesting project portfolios. The 21st International Conference on Multiple Criteria Decision Making, Jyväskylä, Finland, June 13-17, 2011.

[17] Pirolli, P., Card, S., 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. Proceedings of International Conference on Intelligence Analysis, 2-4.

[18] Roy, B., 1996. Multicriteria methodology for decision aiding. Kluwer, Dordrecht.

[19] Simon, H., 1955. A Behavioral Model of Rational Choice, Quarterly Journal of Economics, 69(1), 99-118.

[20] Steuer, R., Qi, Y. and Hirschberger, M., 2008. Portfolio Selection in the Presence of Multiple Criteria. In HANDBOOK OF FINANCIAL ENGINEERING, Springer Optimization and Its Applications, Volume 18, Part I, 3-24.

[21] Zheng, J., Cailloux, O. and Mousseau, V., 2011. Constrained Multicriteria Sorting Method Applied to Portfolio Selection. In 2nd International Conference on Algorithmic Decision Theory. (Oct. 26-28, 2011, DIMACS, Rutgers University, New Jersey.