Semantic Summarization of News from Heterogeneous Sources

Flora Amato, Antonio d'Acierno, Francesco Colace, Vinenzo Moscato, Antonio Penta, Antonio Picariello

Abstract Summarization techniques are becoming an essential part of everyday life, basically because summaries allow users to spend less time making effective access to the desired information. In this paper, we present a general framework for retrieving relevant information from news articles and a novel summarization algorithm based on a deep semantic analysis of texts. In particular, a set of triples (subject, predicate, object) is extracted from each document and it is then used to build a summary through an unsupervised clustering algorithm exploiting the notion of semantic similarity. Finally, we leverage the centroids of clusters to determine the most significant summary sentences using some heuristics. Several experiments are carried out using the standard DUC methodology and ROUGE software and show how the proposed method outperforms several summarizer systems in terms of recall and readability.

1 Introduction

Seeking bits of information from a large amount of data still remains a difficult and time consuming task for a wide range of people such as stu-

Flora Amato, Vincenzo Moscato and Antonio Picariello
DIETI - University of Naples, e-mail: flora.amato,vmoscato,picus@unina.it

Antonio d'Acierno
ISA - Consiglio Nazionale delle Ricerche (CNR) e-mail: antonio.dacierno@isa.cnr.it

Antonio Penta
University of Turin, Department of Computer Science, Via Pessinetto, 12, 10149, Torino (Italy) e-mail: penta@di.unito.it

Francesco Colace
Dipartimento di Ingegneria dell'Informazione, Ingegneria Elettrica e Matematica Applicata, University of Salerno, Fisciano (Italy) e-mail: fcolace@unisa.it

F. Xhafa et al. (eds.), *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, Lecture Notes on Data Engineering and Communications Technologies 1, DOI 10.1007/978-3-319-49109-7_29

dents, news reporters, and many other types of professionals. The exponential growth of the Web has made the search and track of information apparently easier and faster, but the huge information overload requires algorithms and tools for a fast and easy access to the specific *desired* information, discriminating between "useful" and "useless" information, especially in the era of *Big Data.*

We can consider as an example a typical workflow related to a news reporter that has just been informed of a plane crash in Milan and that would quickly like to gather more details about this event[1, 2, 3]. We suppose that the following textual information [1] on the accident, extracted from some web sites already reporting the news, are publicly available:

"*A Rockwell Commander 112 airplane crashed into the upper floors of the Pirelli Tower in Milan, Italy. Police and ambulances are at the scene. The president, Marcello Pera, just moments ago was informed about the incident in Milan, he said at his afternoon press briefing". "It was the second time since the Sept 11 terror attacks on New York and Washington that a plane has struck a high-rise building. Many people were on the streets as they left work for the evening at the time of the crash. Ambulances streamed into the area and pedestrians peered upward at the sky. The clock fell to the floor. The interior minister had informed the senate president, Marcello Pera, that the crash didn't appear to be a terror attack".*

Our basic idea consists in finding, from each sentence, a sequence of relevant information that are then clustered into subsets with a *similar* information content; thus we would like to discard repeated sentences, and to consider only the relevant ones for each cluster. More in details, we can obtain that by reducing the information, that is contained into a sentence, to a regular and simple form: for example, using NLP algorithms [4, 5, 6] we retrieve a list of triples formed by $\langle subject, verb, object\rangle$, where the verb is reported in infinitive form while subject and object are nouns. For the previous air crash example, the extracted triples are[2] : ⟨airplane, crash, tower⟩, ⟨police, be, scene⟩, ⟨ambulance, be, scene⟩, ⟨president, inform, incident⟩*, ⟨terror, attack, New York⟩, ⟨terror, attack, Washington⟩, ⟨plane, strike, building⟩, ⟨people, be, street⟩, ⟨ambulance, stream, area⟩, ⟨pedestrian, peer, sky⟩, ⟨clock, fall, floor⟩, ⟨minister, inform, president⟩, ⟨crash, appear, terror attack⟩*.

Successively, a clustering algorithm creates the clusters reported in Table 1. The clusters are obtained by computing the *semantic similarity* between each couple of triples[7, 8]. Finally, we assume that it is possible to obtain a useful summary by just considering the sequence of centroids and eventually re-loading the associated original sentence, as in the following: "*A Rockwell Commander 112 airplane crashed into the upper floors of the Pirelli Tower in Milan, Italy. Police and ambulances are at the scene. The interior minister had informed the senate president, Marcello Pera, that the crash didn't appear to be a terror attack.*"

[1] The analyzed texts come from news articles, thus they are generally quite simple as concerning grammar and syntax.

[2] Particular cases in which verb is a modal verb or is in a passive or negation form - see triples marked with '*' - have to be opportunely managed.

Cluster 1	⟨ambulance, be, scene⟩ ⟨police, be, scene⟩ ⟨ambulance, stream, area⟩ ⟨people, be, street⟩ ⟨pedestrian, peer, sky⟩
Cluster 2	⟨minister, inform, president⟩ ⟨president, inform, incident⟩
Cluster 3	⟨airplane,crash,tower⟩ ⟨plane, strike, building⟩ ⟨clock, fall, floor⟩ ⟨terror, attack, Washington⟩ ⟨terror, attack, New York⟩ ⟨crash, appear, terror attack⟩

Table 1: The clustering results on a set of triples (centroids are underlined).

Summarizing, we take advantages of some NLP facilities in order to propose a novel approach for text summarization based on the extraction of semantic descriptors of documents, avoiding approaches that are time-consuming and require domain dependent settings[9, 10, 11, 12].

In particular, the semantic content of documents is captured by a set of triples and we then propose a methodology to semanitcally cluster "similar information", thus a summary may be decribed as the sequence of sentences that are associated to the most representative clusters. The idea of using triples as semantic units for representing content of web documents is well studied in the Resource Description Framework (RDF) [3] in the Semantic Web Community.

2 A model for automatic multi-document summarization

2.1 Basic elements of the model

The summarization problem may be stated as follows: given a set of source documents, let us produce an accurate and all-sided summary that is able to reflect the *main concepts* expressed by the original documents, matching some *length restrictions* and without introducing *additional* and *redundant* information.

Our idea is inspired by the text summarization models based on *Maximum Coverage Problem* ([13, 14]), but differently from them we design a methodology that combines both the syntactic and the semantic structure of a text. In particular, in the proposed model, the documents are segmented into several linguistic units (named as *summarizable sentences*) in a preprocessing

[3] http://www.w3.org/TR/WD-rdf-syntax-971002/

stage, and each linguistic unit is then characterized by a set of conceptual units (named as *semantic atoms*) containing the meaning of a sentence. Our main goal is to *cover* as many conceptual units as possible using only a small number of sentences.

Let us give some preliminary definitions about our idea of “summarizable sentence” and “semantic atoms”, and for the sake of clarity, we introduce the example sentences in Table 2 to better explain the introduced definitions.

Document	Sentence	Text of Sentence
1	1.a	People and ambulances were at the scene.
	1.b	Many cars were on the street.
	1.c	A person died in the building.
2	2.a	Marcello Pera denied the terror attack
	2.b	A man was killed in the crash.
	2.c	The president declared an emergency.

Table 2: Example Sentences

Definition 1 (Summarizable Sentence and Semantic Atoms). A *Summarizable Sentence* σ defined over a document D is a couple:

$$\sigma = \langle s, \{t_1, t_2, \ldots, t_m\}\rangle \tag{1}$$

s being a sentence belonging to D and $\{t_1, t_2, \ldots, t_m\}$ being a set of atomic or structured information that expresses in some way the semantic content related to s.

In particular, t_i is defined as a set of couples $\langle A_i, \mathcal{V}_i\rangle$, where A_i is any relevant attribute on which a generic classifier is trained and $\mathcal{V}_i$ is a string or a set of strings in the sentence that is classified as a value for A_i[15, 16]. Let us call t_i as *Semantic Atom.*

Example 1 *In Table 3 there are depicted some possible summarizable sentences that can be extracted from the sentences introduced in Table 2. We introduces attributes like “Sub” (Subject), “Verb”, “Obj” (Object) and “Person” to describe the syntactic structure or the named entities of a sentence.*

Given a set of documents, a *summary* is a set of summarizable sentences. The set has to satisfy some length restriction conditions and include the most representative content of all input data. For now, let us assume the existence of a similarity function $sim(t_i, t_j) \in [0, 1]$ able to compute the semantic similarity between two semantic atoms and another function able to *score* the semantic atoms based on their importance. We will give more details on both these aspects in the following subsections.

Document 1	
Summarizable Sentences	Semantic Atoms
$\sigma_{1.1}{=}\langle 1.a, t_1^{1.1}, t_2^{1.1}\rangle$	$t_1^{1.1}{=}\{\langle Sub, people\rangle, \langle Verb, were\rangle, \langle Obj, scene\rangle\}$, $t_2^{1.1}{=}\{\langle Sub, ambulances\rangle, \langle Verb, were\rangle, \langle Obj, scene\rangle\}$,
$\sigma_{1.2}{=}\langle 1.b, t_1^{1.2}\rangle$	$t_1^{1.2}{=}\{\langle Sub, cars\rangle, \langle Verb, were\rangle, \langle Obj, street\rangle\}$.
$\sigma_{1.3}{=}\langle 1.c, t_1^{1.3}\rangle$	$t_1^{1.3}{=}\{\langle Sub, person\rangle, \langle Verb, died\rangle, \langle Obj, building\rangle\}$.
Document 2	
Summarizable Sentences	Semantic Atoms
$\sigma_{2.1}{=}\langle 2.a, t_1^{2.1}, t_2^{2.1}, t_3^{2.1}\rangle$	$t_1^{2.1}{=}\{\langle Sub, Marcello\ Pera\rangle, \langle Verb, denied\rangle, \langle Obj, attack\rangle\}$ $t_2^{2.1}{=}\{\langle Person, Marcello\ Pera\rangle\}$,
$\sigma_{2.2}{=}\langle 2.b, t_1^{2.2}\rangle$	$t_1^{2.2}{=}\{\langle Sub, man\rangle, \langle Verb, was\ killed\rangle, \langle Obj, crash\rangle\}$.
$\sigma_{2.3}{=}\langle 2.c, t_1^{2.3}\rangle$	$t_1^{2.3}{=}\{\langle Sub, president\rangle, \langle Verb, declared\rangle, \langle Obj, emergency\rangle\}$

Table 3: Example of Summarizable Sentences extracted from the documents in Table 2.

Now we are in position to introduce the concept of "Summarization Algorithm" as follows.

Definition 2 (Summarization Algorithm). Let $\mathcal{D}$ be a set of documents, a *Summarization Algorithm* is formed by a sequence of two functions ϕ and χ. The semantic partitioning function (ϕ) partitions $\mathcal{D}$ in K sets $\mathcal{P}_1, \ldots, \mathcal{P}_K$ of summarizable sentences having similar semantics in terms of semantic atoms and returns for each set the related *information score* by opportunely combining the score of each semantic atom:

$$\phi : \mathcal{D} \rightarrow \mathcal{S}^* = \{\langle \mathcal{P}_1, \hat{w}_1\rangle, \ldots, \langle \mathcal{P}_K, \hat{w}_K\rangle\} \tag{2}$$

s. t. $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset, \forall i \neq j$.

The *Sequential Sentence Selection* function (χ):

$$\chi : \mathcal{S}^* \rightarrow \mathcal{S} \tag{3}$$

selects a set of the sentences $\mathcal{S}$ from original documents containing the semantics of most important clustered information sets in such a way that:

1. $|\mathcal{S}| \leq L$,
2. $\forall \mathcal{P}_k, \hat{w}_k \geq \iota$, $\nexists t_j, t_j \in_\sigma \mathcal{S}^* : sim(t_i, t_j) \geq \gamma, t_i \in_\sigma \mathcal{S}$.

ι and γ being two apposite thresholds. With abuse of notation, we use the symbol $\in_\sigma$ to indicate that a semantic atom comes from a sentence belonging to the set of summarizable sentences $\mathcal{S}$.

Note that we can select any unsupervised clustering algorithm that is able to partition the documents space into several clusters using the semantic similarity among semantic atoms. Once obtained a partition of the space in terms of clusters of semantic atoms, we select a number of sentences, trying to: (i) maximize the semantic coverage - the most representative sentences of each cluster should be considered starting from the most important clusters, i.e. those having the highest average information score; (ii) minimize the redundancy by selecting one sentence for each cluster that is most representative in terms of semantic content and not considering similar sentences.

Now, we are going to explain the following points of our model are: (i) how to represent and extract semantic atoms of a document, (ii) how to evaluate the similarity between two semantic atoms, (iii) how to calculate a score for each semantic atom, and finally, (iv) how to define suitable semantic partitioning and sentence selection functions.

2.2 Extracting semantic atoms from a text

We adopted the principles behind the RDF framework used in the Semantic Web community to semantically describe web resources. The idea is based on representing data in terms of a triple $\langle subject, verb, object\rangle$. In the Semantic Web community subjects and objects are web resources while verbs are predicates/relations defined in schemata or ontologies, in our case instead we attach to the elements of triples the tokens extracted by processing documents using NLP techniques. Thus, the semantic content of a document can be modeled by a set of structured information $\mathcal{T}=\{t_1,\ldots,t_n\}$, where each element t_i is a semantic atom described by the following couples $t_i=\{\langle sub, val\rangle$, $\langle verb, val\rangle$, $\langle obj, val\rangle\rangle\}$, and we can call t_i as *summarization triple*. Since now, we use in the rest of the paper the name *triple* and *summarization triple* with the same meaning.

The triples are extracted from each sentence in the documents by applying a set of rules on the *parse tree* structure computed on each sentence. In our rules, subjects and objects are nouns or chunks while the verbs are reported in the infinitive form.[4] It is worth to be noted that in the case of long sentences more triples may be associated to a sentence. The rules are obtained by defining a set of patterns for subject, verb and object which includes not only part of speech features but also parse tree structures. In particular, we start from the patterns described in [17] in order to include not only relations

[4] We do not consider any other grammatical units for a sentence such as adjective, preposition and so on. This because we are not interested in detecting the sentiment or opinion in the text or to provide just triples themselves as final summarization results to the user, but we exploit them like“ pointers” to original sentences that are the real components of our summaries. Thus, we want to ensure that the quality of these pointers is enough to get together similar sentences.

but also subjects and objects and we add to the pattern expressions features related to the sentence linguistic structure (parse tree).

2.3 Semantic similarity function

In our model, we decided to compare two semantic atoms based on the similarity measure obtained by the comparison of the elements hosted by a summarization triple, namely subject, predicate, and object. In particular, let us consider two sentences and assume to extract from them two triples t_1 and t_2; we define as similarity between two t_1 and t_2 the function:

$$sim(t_1, t_2) = F_{agr}(F^{sim}(sub_1, sub_2), F^{sim}(pred_1, pred_2), F^{sim}(obj_1, obj_2)) \tag{4}$$

The function F^{sim} is used to obtain the similarity among values of the semantic atoms, while F_{agr} is an aggregation function. The following functions are some examples of the easiest combination strategies:

$$F_{agr}(t_1, t_2) = \sum_{i=1}^{3} \alpha_i \cdot F^{sim}(t_1[i], t_2[i]); \tag{5}$$

$$F_{agr}(t_1, t_2) = \max_{i \in \{1,2,3\}} \left[\alpha_i \cdot F^{sim}(t_1[i], t_2[i])\right] \tag{6}$$

where the constraint $\alpha_1 + \alpha_2 + \alpha_3 = 1$ is used to obtain a convex combination of the results and we consider the same function for each element of a triple.

If the F^{sim} takes into account the information stored in a knowledge base, the function is computed based on the "*semantic*" aspects of its input, otherwise we can apply any similarity among words like the one based on the well-known edit distance.

In particular, we use the *Wu & Palmer* similarity ([18]) for computing the similarity among elements of our triples. This similarity is based on the *Wordnet Knowledge Base*, that lets us compare triples based on their semantic content.

3 The proposed framework

In Figure 1, we show at a glance the summarization process that consists of the following steps: (i) *Web Search* - this activity has the task of retrieving a set of HTML documents that satisfy some search criteria using a *Search Engine* external component; (ii) *Text Extraction* - the sentences are extracted from the several web sources by parsing the related HTML pages and analyzing the HTML tags; (iii) *NLP and Triples Extraction* - NLP processing techniques are performed on the input sentences, in particular *Named Entities Recognition* (NER), *Part Of Speech* (POS) tagging, Parse Tree Generation, Anaphora and Co-Reference Resolutions, successively in this stage, semantic triples are detected for each sentence, analyzing the related Parse Tree and using appropriate heuristics; (iv) *Similarity Matrix Builder* - a matrix containing the similarity values for each couple of triples is computed; (v) *Clustering* - a proper clustering algorithm is applied on the input matrix; (vi) *Sentence Selection* - this activity performs a sentence selection to generate the summary using our proposed algorithm; (vii) *Summary Building* - this activity performs a sentence ordering to generate the final summary[19].

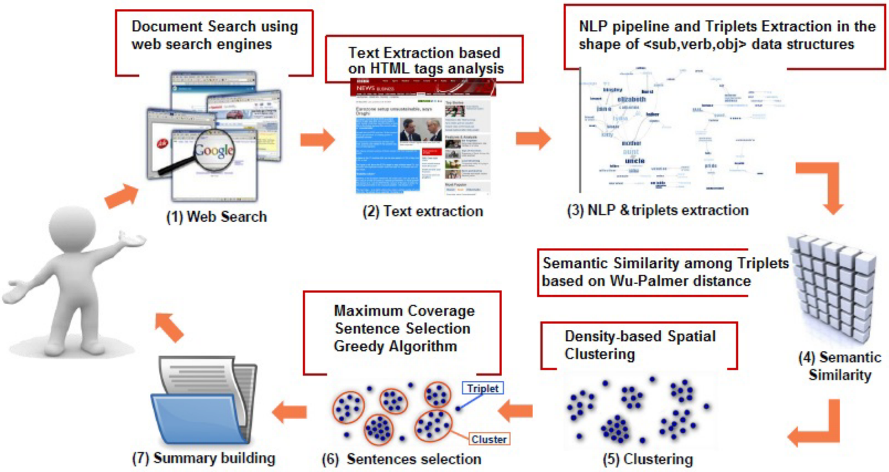
Document Search using web search engines
(1) Web Search
Text Extraction based on HTML tags analysis
(2) Text extraction
NLP pipeline and Triplets Extraction in the shape of <sub,verb,obj> data structures
(3) NLP & triplets extraction
Semantic Similarity among Triplets based on Wu-Palmer distance
(4) Semantic Similarity
Density-based Spatial Clustering
(5) Clustering
Maximum Coverage Sentence Selection Greedy Algorithm
Triplet
Cluster
(6) Sentences selection
(7) Summary building

Fig. 1: The summarization process

The system is named *iWIN* (information on the Web In a Nutshell) and provides a graphical user interface that allows the user to configure some system settings, thus allowing to tune the system and evaluate its performances under several conditions and situations.

4 Conclusions

In this paper, we described a novel approach for summarizing web documents based on semantic extraction and description of documents.

We first proposed a model of summarization based on the semantic content of a document, captured and modelled by a set of triples, i.e. a subject, verb, object.

We then proposed a novel methodology based on cluster analysis of triples, thus obtaining a summary as the sequence of sentences that are associated to the most representative clusters' triples: in particular, the centroids of the clusters are used to detect the main representative topics, then they are properly combined for producing non-repetitive and brief summaries. To the best of our knowledge, this is the first work that uses a semantic-driven approach between triples for text summarization applications[20, 21, 22, 23].

Based on this approach, we implemented a system called iWin, that provides all the functionalities of a multi-document summarization tool. We tested iWin using some well-known data sets, showing good and sometimes excellent performances with respect to classical evaluation measures in the summarization literature. In addition, we made a performances comparison with open sources and commercial summarizer systems, obtaining promising results with respect to other approaches both for query-based and for generic summaries.

Future work will be devoted to improve the current research into main directions: i) extend the proposed methodology to the query-based approach; ii) consider multimodal summaries able to combine different data coming from unstructured (image, video, audio data) or structured (linked data) repositories; iii) improve efficiency of our approach (by caching similarity values between terms that have to be computed more times, using concurrent computation threads for subjects, verbs and objects and defining a proper indexing to access to the distances matrix); iv) implement the algorithms in a parallel computing environment; v) test several and more advanced density-based clustering approaches.

References

1. Leonard Barolli and Fatos Xhafa. Jxta-overlay: A p2p platform for distributed, collaborative, and ubiquitous computing. *Industrial Electronics, IEEE Transactions on*, 58(6):2163–2172, 2011.
2. Fatos Xhafa, Raul Fernandez, Thanasis Daradoumis, Leonard Barolli, and Santi Caballé. Improvement of jxta protocols for supporting reliable distributed applications in p2p systems. In *Network-Based Information Systems*, pages 345–354. Springer, 2007.
3. Leonard Barolli, Fatos Xhafa, Arjan Durresi, and Giuseppe De Marco. M3ps: a jxta-based multi-platform p2p system and its web application tools. *International Journal of Web Information Systems*, 2(3/4):187–196, 2007.
4. Mario Sicuranza, Angelo Esposito, and Mario Ciampi. An access control model to minimize the data exchange in the information retrieval. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–12, 2015.
5. Aniello Minutolo, Massimo Esposito, and Giuseppe De Pietro. A fuzzy framework for encoding uncertainty in clinical decision-making. *Knowledge-Based Systems*, 98:95–116, 2016.
6. Aniello Minutolo, Massimo Esposito, and Giuseppe De Pietro. Design and validation of a light-weight reasoning system to support remote health monitoring applications. *Engineering Applications of Artificial Intelligence*, 41:232–248, 2015.

7. F. Amato, A.R. Fasolino, A. Mazzeo, V. Moscato, A. Picariello, S. Romano, and P. Tramontana. Ensuring semantic interoperability for e-health applications. pages 315–320, 2011.
8. F. Amato, A. Mazzeo, V. Moscato, and A. Picariello. A framework for semantic interoperability over the cloud. pages 1259–1264, 2013.
9. Tim French, Nik Bessis, Fatos Xhafa, and Carsten Maple. Towards a corporate governance trust agent scoring model for collaborative virtual organisations. *International Journal of Grid and Utility Computing*, 2(2):98–108, 2011.
10. Valentin Cristea, F. Pop, C. Stratan, A. Costan, C. Leordeanu, and E. Tirsa. A dependability layer for large-scale distributed systems. *International Journal of Grid and Utility Computing*, 2(2):109–118, 2011.
11. Soichi Sawamura, Admir Barolli, Ailixier Aikebaier, Makoto Takizawa, and Tomoya Enokido. Design and evaluation of algorithms for obtaining objective trustworthiness on acquaintances in p2p overlay networks. *International Journal of Grid and Utility Computing*, 2(3):196–203, 2011.
12. Evjola Spaho, Gjergji Mino, Leonard Barolli, and Fatos Xhafa. Goodput and pdr analysis of aodv, olsr and dymo protocols for vehicular networks using cavenet. *International Journal of Grid and Utility Computing*, 2(2):130–138, 2011.
13. H. Takamura and M. Okumura. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th Conference of the European Chapter of the AC*, pages 781–789, 2009.
14. Dan Gillick and Benoit Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, ILP '09, pages 10–18. Association for Computational Linguistics, 2009.
15. Salvatore Cuomo, Pasquale De Michele, Ardelio Galletti, and Giovanni Ponti. *Intelligent Interactive Multimedia Systems and Services 2016*, volume 55 of *Smart Innovation, Systems and Technologies*, chapter Influence of Some Parameters on Visiting Style Classification in a Cultural Heritage Case Study, pages 567–576. Springer International Publishing, 2016.
16. Salvatore Cuomo, Pasquale De Michele, Ardelio Galletti, and Giovanni Ponti. *Data Management Technologies and Applications: 4th International Conference, DATA 2015, Colmar, France, July 20-22, 2015, Revised Selected Papers*, volume 584 of *Communications in Computer and Information Science*, chapter Classify Visitor Behaviours in a Cultural Heritage Exhibition, pages 17–28. Springer International Publishing, 2016.
17. Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.
18. Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, 1994.
19. A. D'Acierno, V. Moscato, F. Persia, A. Picariello, and A. Penta. iwin: A summarizer system based on a semantic analysis of web documents. pages 162–169, 2012.
20. G. Sannino, I. De Falco, and G. De Pietro. An automatic rules extraction approach to support osa events detection in an mhealth system. *IEEE Journal of Biomedical and Health Informatics*, 18(5):1518–1524, 2014.
21. Angelo Chianese, Fiammetta Marulli, Francesco Piccialli, Paolo Benedusi, and Jai E Jung. An associative engines based approach supporting collaborative analytics in the internet of cultural things. *Future Generation Computer Systems*, 2016.
22. A. Chianese, F. Piccialli, and I. Valente. Smart environments and cultural heritage: a novel approach to create intelligent cultural spaces. *Journal of Location Based Services*, 9:209–234, 2015.
23. Giuseppe Caggianese, Luigi Gallo, and Giuseppe De Pietro. Design and preliminary evaluation of a touchless interface for manipulating virtual heritage artefacts. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on*, pages 493–500. IEEE, 2014.