Cristian Lai, Giovanni Semeraro, Alessandro Giuliani (Eds.)

Proceedings of the 7th International Workshop on Information Filtering and Retrieval

Workshop of the XIII AI*IA Conference

December 6, 2013
Turin, Italy
http://aixia2013.i-learn.unito.it/course/view.php?id=26

Preface

The series of DART workshops provides an interactive and focused platform for researchers and practitioners for presenting and discussing new and emerging ideas. Focusing on research and study on new challenges in intelligent information filtering and retrieval, DART aims to investigate novel systems and tools to web scenarios and semantic computing. Therefore, DART contributes to discuss and compare suitable novel solutions based on intelligent techniques and applied in real-world applications.

Information Retrieval attempts to address similar filtering and ranking problems for pieces of information such as links, pages, and documents. Information Retrieval systems generally focus on the development of global retrieval techniques, often neglecting individual user needs and preferences.

Information Filtering has drastically changed the way information seekers find what they are searching for. In fact, they effectively prune large information spaces and help users in selecting items that best meet their needs, interests, preferences, and tastes. These systems rely strongly on the use of various machine learning tools and algorithms for learning how to rank items and predict user evaluation.

Submitted proposals received two or three review reports from Program Committee members. Based on the recommendations of the reviewers, 7 full papers have been selected for publication and presentation at DART 2013.

When organizing a scientific conference, one always has to count on the efforts of many volunteers. We are grateful to the members of the Program Committee who devoted a considerable amount of their time in reviewing the submissions to DART 2013.

We were glad and happy to work together with highly motivated people to arrange the conference and to publish these proceedings. We appreciate the work of the Publicity Chair Fedelucio Narducci from University of Milan-Bicocca for announcing the workshop on various lists. Special thanks to Cristina Baroglio and Matteo Baldoni for the support and help in managing the workshop organization.

We hope that you find these proceedings a valuable source of information on intelligent information filtering and retrieval tools, technologies, and applications.

December 2013

Cristian Lai

Giovanni Semeraro

Alessandro Giuliani

Organization

Chairs

• Cristian Lai (CRS4, Center for Advanced Studies, Research and Development in Sardinia, Italy)

• Giovanni Semeraro (University of Bari Aldo Moro, Italy)

• Alessandro Giuliani (University of Cagliari, Italy)

Publicity Chair

• Fedelucio Narducci (University of Milan-Bicocca, Italy)

Program Committee

• Marie-Hélène Abel (Technology University of Compiègne, France)

• Gianbattista Amati (Fondazione Ugo Bordoni, Italy)

• Liliana Ardissono (University of Torino, Italy)

• Giuliano Armano (University of Cagliari, Italy)

• Pierpaolo Basile (University of Bari Aldo Moro, Italy)

• Roberto Basili (University of Rome "Tor Vergata", Italy)

• Federico Bergenti (University of Parma, Italy)

• Ludovico Boratto (University of Cagliari, Italy)

• Annalina Caputo (University of Bari Aldo Moro, Italy)

• Pierluigi Casale (Eindhoven University of Technology, Netherlands)

• José Cunha (University Nova of Lisbon, Portugal)

• Juan Manuel Fernández (Barcelona Digital Technology Center, Spain)

• Marco de Gemmis (University of Bari Aldo Moro, Italy)

• Emanuele Di Buccio (University of Padua, Italy)

• Nima Hatami (University of California at San Diego, US)

• Fumio Hattori (Ritsumeikan University , Japan)

- Leo Iaquinta (University of Milan-Bicocca, Italy)
- Jose Antonio Iglesias Martinez (University of Madrid, Spain)
- Francesca Alessandra Lisi (University of Bari Aldo Moro, Italy)
- Pasquale Lops (University of Bari Aldo Moro, Italy)
- Massimo Melucci (University of Padua, Italy)
- Maurizio Montagnuolo (RAI Centre for Research and Technological Innovation, Italy)
- Claude Moulin (Technology University of Compiègne, France)
- Gabriella Pasi (University of Milan-Bicocca, Italy)
- Vincenzo Pallotta (University of Business and International Studies at Geneva, Switzerland)
- Marcin Paprzycki (Polish Academy of Sciences, Poland)
- Agostino Poggi (University of Parma, Italy)
- Sebastian Rodriguez (Universidad Tecnologica Nacional , Argentina)
- Paolo Rosso (Polytechnic University of Valencia, Spain)
- Eloisa Vargiu (Barcelona Digital Technology Center, Spain)

Table of Contents

Ambient-Intelligence Trigger Markup Language
A new approach to Ambient Intelligence rule definition

Juan Manuel Fernández, Sergi Torrellas, Stefan Dauwalder, Marc Solà, Eloisa Vargiu and Felip Miralles

Barcelona Digital Technology Center, jmfernandez@bdigital.org, storellas@gmail.com, {sdauwalder, msola, evargiu, fmiralles}@bdigital.org

Abstract. Assistive technologies need to constantly adapt and react to user needs. To this end, ambient intelligence techniques could be adopted. One approach consists of defining suitable rules to trigger actions or suggestions to the users. In this paper, ATML (Ambient intelligence Trigger Markup Language), a novel suitable language, is presented and described. ATML is aimed at defining and describing actions and rules in the field of ambient intelligence and context-awareness. To show how useful ATML is, we briefly introduce its current implementation in BackHome, an EU project concerning physical and social autonomy of people with disabilities.

1 Introduction

Today's appliances have successfully become integrated to such an extent that we use them without consciously thinking about them. Computing devices have transitioned in this past half a century from big mainframes to small chips that can be embedded in a variety of places ranging from communication appliances (e.g. mobile phones) to simple applications (e.g. weather sensors). In this setting, various industries have silently distribute computing devices all around us, often without us even noticing, both in public spaces and in our more private surroundings with small amounts of intelligence providing them autonomy to perform small-scale decisions to modify the environment.

The advances in the miniaturization of electronics allow purchasing sensors, actuators and processing units at very affordable prices [4] favouring the inclusion of these elements in our normal day activities and houses. This novel approach can be networked with the coordination of highly intelligent software applications to understand the events and relevant context of a specific environment. This knowledge enables to take sensible decisions in real-time or a posteriori to adapt the features of applications to the real setting in which they are.

These elements are to be coordinated by intelligent systems that integrate the available resources to provide an intelligent environment. This confluence of topics has led to the introduction of the area of Ambient Intelligence (AmI) that is defined as a digital environment that proactively, but sensibly, supports people in their daily lives [3]. AmI is aligned with the concept of the disappearing computer [29][25]: "*Technologies that disappear weave themselves into everyday life to the point that they are indistinguishable*".

Networks, sensors, human-computer interfaces, pervasive computing and artificial intelligence are all relevant but none of them conceptually fully cover AmI. It is, though, AmI which brings all these together to provide flexible and intelligent services to users acting in their environments. Indeed, AmI relies in the application of artificial intelligence techniques to provide added value services to the end-users.

Ambient Assisted Living (AAL) fosters the provision of these intelligent environments for the independent or more autonomous living of people with disabilities, via the seamless integration of info-communication technologies in homes and residences [18]. Assistive Technologies (ATs) are becoming of crucial importance in this AAL scenario as they are often used to provide support at home to engage and promote independence [6].

Among other application fields, let us recall here the importance that AmI takes on personalized assistance of people with disabilities and the improvement of AT to their impairments and needs. In particular, AmI helps monitoring the context features and behaviour and facilitates the control on environmental appliances (e.g. lights, windows). Those systems combine all the above features with the understanding of situations in which to perform actions pro-actively, such as triggering emergency alarms (e.g. fire, gas leak).

In order to adapt the AT and to react properly to the different situations that a user may need, several AmI techniques should be developed. One approach can be the use of rules which should trigger actions, or suggestions to the user, in order to react properly to the situation. To fulfill the need to express these rules we present, in this paper, a suitable language called ATML (Ambient intelligence Trigger Markup Language). Based on RuleML [19], ATML is aimed at defining and describing actions and rules in the field of AmI and context-awareness. The underlying idea is to adapt the RuleML to the AmI systems and applications in the area of ATs.

As a practical demonstration of the usefulness of ATML, we present its current implementation in BackHome[1], an EU project concerning physical and social autonomy of people with disabilities, by using mainly Brain-Neural Computer Interfaces (BNCIs) and integrating other assistive technologies as well.

The paper is organized as follows: Section 2 briefly introduces AmI. Section 3 discusses the representation of rules and actions in the field of AmI. Section 4 presents the proposed language and its fundamentals. Section 5 illustrates the benefits of ATML in BackHome through an example of usage. Section 6 ends the paper with conclusions.

2 Ambient Intelligence

According to Augusto and McCullagh [4] we may see Ambient Intelligence (AmI) as the confluence of Pervasive Computing (aka, Ubiquitous Computing), Artificial Intelligence (AI), Human Computer Interaction (HCI), Sensors, and Networks. First, an AmI system pervasively senses the environment by relying on a network of sensors. The gathered information is, then, processed by AI techniques to provide suitable actions to be performed on the environment through controllers and/or specialized HCI.

[1] www.Backhome-FP7.eu

According to [10], the AmI is placed in the confluence of a multi-disciplinary and heterogeneous ecosystem. This position allows the AmI applications to get the information of the surroundings, actuate and change the environment, the different human interfaces available, as well as apply some reasoning techniques. The conjunction of all these fields is used by the AmI systems always respecting the privacy of the user. Following this description, a system incorporates AmI principles if the following characteristics are met: *Sensitive*, AmI systems have to incorporate the ability to perceive their immediate surroundings; *Responsive*, AmI systems have to be able to react in front of the context occurring; *Adaptive*, AmI systems are to be flexible enough to accommodate the responses along the time; *Transparent*, AmI systems have to be designed to be unobtrusive; *Ubiquitous*, AmI systems have to be concealed so as to minimize the impact of bulky and tedious appliances; and *Intelligent*, AmI systems have to incorporate intelligent algorithms to react in front of specific scenarios. These principles can be applied to several fields ranging from education to health or security. As we have already commented, among others, this work focusses on ATs. The capability of AmI techniques for recognizing activities [5] [20], monitoring diet and exercise [15], and detecting changes or anomalies [11] support the key idea of providing help to individuals with cognitive or physical impairments. For instance, AmI techniques can be used to provide reminders of normal tasks or the step sequences to properly realize and complete these tasks. For those with physical limitations, automation and inclusion of AI to their home and work environment may become a response for independent living at home [30].

Several artefacts and items in a house can be enriched with sensors to gather information about their use and in some cases even to act independently without human intervention. Some examples of such devices are white goods (e.g., oven and fridge), household items (e.g., taps, bed and sofa) and temperature handling devices (e.g., air conditioning and radiators). Applying AmI in this scenario may: (a) increase safety (e.g., by monitoring lifestyle patterns or the latest activities and providing assistance when a possibly harmful situation is developing) and/or (b) improve comfort (e.g., by adjusting temperature automatically).

In addition, AmI allows the home itself to take decisions regarding its state and interactions with its residents. There are several physical smart homes that have been designed with this theme in mind. For instance, the MavHome project treats an environment as an intelligent agent, which perceives it using sensors and acts on the environment using powerline controllers [13].

3 Ambient Intelligence and Triggered Actions

Being implemented in real-world environments, AmI involves problems such as incompleteness and uncertainty of the information available about the user and the environment. In fact, we generally deal with information that might be in some way correct, in somewhere incorrect, and in some part missing. Thus, an elaborated reasoning process that deals with those information drawbacks might be performed to successfully define an accurate knowledge representation. To this end, AmI relies on the context as a model of the current situation of the user and its immediate environment [14].

In order to use context effectively, we must understand what context is and how it can be used. A precise notion of context is essential in an intelligent environment, even more in assistive applications since an understanding of how context can be used helps application designers to determine the context-aware behaviours necessary to support in their applications. Nevertheless, the context is not a static but a dynamic concept composed of entities like people, devices, locations or even computing applications which, in their turn, are characterized by attributes. Once these concepts are properly integrated into the design, the system is entitled to take the necessary actions according to different combinations of these entities using different techniques such as rules or analysis of this information provided by the context.

The process of understanding the context is not explicit and thus not trivial; it depends on previous knowledge. Experience provides means to classify and highlight specific situations and to relate them with the received stimuli of the sensory system. It is supposed that same situations are promoted by the same stimulus, and those implications resultant from those situations serve as an extension on the recall of experience [23]. Sensing of location, environmental conditions and capturing explicit interactions are the general inputs for context extraction. Nevertheless, it is desirable for smart environments to interpret the available information by perceptual means similar to those of humans [24], for this reason it is necessary to describe the situations in a human readable format. The ATML fulfil this point offering the possibility to describe all the situations where the sensory system is involved and the actions that can be interesting to trigger.

Current implementations of AmI in the field of SmartHome and ATs are focused on providing modifiable assistance according to the user context, the so-called personalized assistance. The more frequent implementations are based on machine learning algorithms and intelligence systems. Nevertheless, this is not the only possible approach: there exist systems based on the use of rule based engines that determine the actions to be triggered by the system. Some examples can be find at the literature, for instance Acampora and Loia [1] present a distributed AmI system, based on agents, communication protocols (TCP/IP) and Fuzzy Logic [31] using as a language for description of knowledge and rules, the Fuzzy Markup Language [2]. In our project, it is not necessary the use of distributed logic and agents.

On the other side, the platform DOAPAmI [17] uses a Domain Specific Language (DSL) in order to define the complete platform including services, sensors, profiles of physical platforms where executing the system. This DSL also allows the specification of rules for the different situations, but focused on the proper running of the system not on the user needs and preferences. Papamarkos et al. [21] present an Even-Condition-Action centred approach based on RuleML [19] and focused on Semantic Web, far from the focus of our objective. Other example of use of XML languages in order to define rules based on the context was presented by Schmidt [22] introducing an extension of the Standard Generalized Markup Language (SGML) defined in [7] [8] is introduced. This extension allows the language to define triggers and represent the context to improve the user interface in small devices. So the context represented by this proposal is only related to the attributes and characteristics of the device where the application shows its user interface.

4 The ATML Language

As we have already commented in the previous section, some applications use rules engines to define the actions to be triggered in several situations. Although, these rules can be defined in several ways, frequently are platform dependent and not based on standards. In order to establish a platform-independent and flexible definition of these rules, and to specify conditions and actions to be triggered (hereinafter, triggers), we propose to use a human readable XML-based language called AmI Triggering Markup Language (ATML). This approach allows exporting the rules definitions to any AmI system and reaching the same status on the configuration of the intelligence.

ATML is compliant with the RuleML. In fact, most of ATML definition relies on the RuleML definitions. RuleML is an initiative part of the research community's effort to develop the Semantic Web. RuleML is, at its heart, an XML syntax for rule knowledge representation that is interoperable among major commercial rule systems. RuleML is based on a fundamental rule knowledge representation, declarative logic programs, which expressively extend ordinary logic programs with features for prioritized conflict handling and procedural attachments to perform actions and queries.

Let us consider an example of trigger definition:

```
<Trigger>
  <name>Activate Managed Ambience</name>
  <Properties>
    <occurrence>Continuous</occurrence>
    <enabled>true</enabled>
  </Properties>
  <Implies>
   . . .
  </Implies>
</Trigger>
```

As you can see, a trigger in ATML is firstly described by its *name*, in order to distinguish it from the overall set of triggers. This is very helpful for both user and designers to perform the import/export operations as rules can be easily identifiable. After assigning the name, the set of *properties* has to be defined. Properties incorporate different attributes helping to understand how to interact with the rule. As an example, let us consider an *occurrence* property that provides the time frame in which the rule needs to be evaluated. In the code of the example reported above, we can also find the enabled property that indicates to the system if that rule is active and we have to check and react, if needed.

After these two sections the rules define the implications of the action. That section includes two main sub-sections: *head* and *body*:

```
<Implies>
    <head>
      <TriggerAction>
        <op> <rel>open</rel> </op>
        <who>trigger</who>
        <device>CUR_DD_001</device>
      </TriggerAction>
```

```
    </head>
    <body>
      <And>
        <Atom>
          <op> <rel>greater than</rel> </op>
          <var>TEMP_ENV_001</var>
          <value>28</value>
        </Atom>
        <Atom>
          <op><rel>lower or equal than</rel></op>
          <var>TEMP_HVAC_001</var>
          <value>23</value>
        </Atom>
      </And>
    </body>
  </Implies>
```

Head is used to define the actions (*TriggerAction*) to be executed whenever the condition of the rule is met. Every *TriggerAction* is defined by a single action to be performed when a given condition (defined in the body section) is met. To define a *TriggerAction*, three different tags are necessary:

- *op*, which indicates the command to be performed on the target, in the example the *op* indicates that the value of the variable $TEMP\_ENV\_001$ must be greater than *28*;
- *who*, describing which system performs the operation (in the example the trigger engine defined by the identifier trigger);
- *device*, which defines the targeted appliance (in the example, the device with identifier CUR_DD_001).
- *body*, which expresses the condition of the rule by means of comparison and logic operation. The rule defined in the example contains two conditions, the first one related with the value of the variable $TEMP\_ENV\_001$ and the second one with the value of $TEMP\_HVAC\_001$.

A single comparison operation, called *atom*, includes an operand, namely *equal*, *not equal*, *greater than*, *lower than* and *contains*. To concatenate and combine different atoms, the three classical logic operands (*and*, *or*, and *not*) are allowed. It is remarkable to say that logic operations are allowed to be nested, that is, a logic operation may contain inner logic operands.

5 Real Case of ATML Use: BackHome Project

ATML is currently used in the BackHome project. The project BackHome, an European initiative funded by the FP7 program, is willing to play a role in empowering the end-users to become more autonomous and independent in their activities of daily life by means of a novel concept in ATs. BackHome is about boosting physical and social autonomy of people with disabilities taking a broad approach and is aimed at supporting the transition from institutional care to home, post rehabilitation and discharge

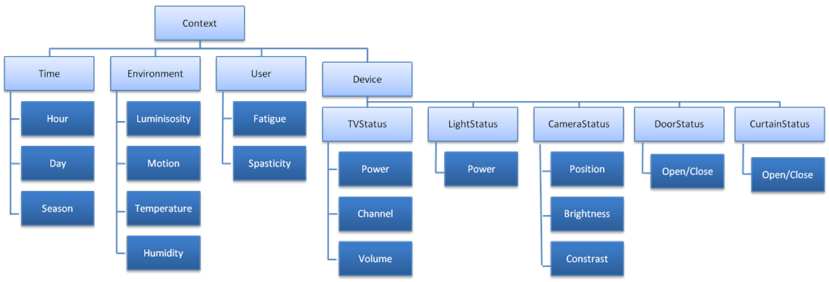

Fig. 1. Context definition in BackHome.

[12]. The project offers as an innovative AAL platform, a sensor-based Telemonotoring and Home Support System (TMHSS), devoted to help the user to be more independent by providing an AAL environment improved with the AmI principles [28] [27]. BackHome also takes care of the isolation problems often associated to disability and, therefore, incorporates eInclusion with the possibility to interact with the most popular social networks such as Facebook or Twitter, and other Internet related services, like Web browsing and e-mail. Finally, the system has added value features in the field of telemedicine: cognitive rehabilitation and quality of life automatic assessment [26]. Within the project, the achievement of these objectives strongly relies on the usage of BNCIs as principal interface but integrating other assistive technologies as well. BNCIs rely on the direct measures of brain activity complemented with other technologies. This project is a perfect environment to test the flexibility and scalability of ATML.

5.1 Ambient Intelligence in BackHome

Suitable AmI features are provided in BackHome. In particular, the sensor-based TMHSS is aimed at acquiring contextual information through data coming from sources of different nature: BNCI system that allows monitoring ElectroEncephalo-Gram (EEG), ElectroOculoGram (EOG), and ElectroMyoGram (EMG); wearable, physiological, and biometric sensors, such as ElectroCardioGram (ECG), heart-rate sensor, respiration-rate sensor, Galvanic Skin Response (GSR) sensor, EMG switches, and inertial sensors (e.g., accelerometer, gyrocompass, and magnetometer); environmental sensors (e.g., temperature and humidity sensors); SmartHome devices (e.g., wheelchairs, lights, TVs, doors, windows and shutters); devices that allow interaction activities (e.g., a desktop PC); as well as devices to perform rehabilitation tasks (e.g., a robot).

As part of the design of BackHome, it was mandatory to devise a knowledge representation of the context which needs to be captured, and stored, from the sources presented in the previous list. The outcome of the context formalization is depicted in Figure 1 in which the different values of the environment together with parameters of the user (e.g. fatigue) are presented. This definition of the context incorporates different categories taken into account when evaluating the context:

1. Time Variables: representing the current moment taking place.

2. Environmental Variables: these variables refer to those measures that give direct information of the context such as luminosity or motion.
3. User Variables: providing information about physiological measurements (e.g. fatigue and spasticity).
4. Device Variables: these are the variables referred to the status of the devices controlled by the AmI-enabled system.

BackHome takes advantage of AmI to provide advanced assistance through two main assistive services based on AmI: Personalized Adaptation and Proactive Performance. The *Personalized Adaptation* is an approach based on the user's preferences and habits. This method learns the habits from the user applying machine learning techniques in order to infer the behavioral patterns of the system. Currently, it includes the proposals of the BrainAble [16] project based on AdaBoost with C45 as a weak algorithm [9]. The *Proactive Performance* is a module that applies rules user defined which should be activated for a particular situation without an explicit request from the user. When the environment arrives to a given status, which matches with any of the pre-established rules, the system reacts consequently. Currently, the system updates the interface suggesting to the user some actions that might be interesting for him/her.

5.2 Use-Case Scenario

In the current implementation of the BackHome project, some proactive context-trigger actions have been designed and developed. Context-Trigger actions are clear examples of the proactive nature of AmI: whenever a rule condition is met the corresponding action is triggered.

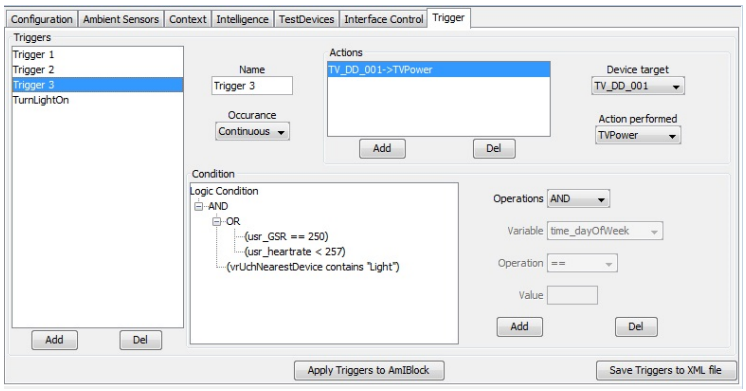

Fig. 2. Trigger Definition User Interface in BackHome.

These rules, expressed in ATML, are configurable by the end-user with a dedicated interface (see Figure 2), which facilitates the creation of the rules which are stored in a separated file. As such, the rules can be defined as portable across different AmI-enabled systems. As an example, let us consider the next piece of code that presents a rule expressed in ATML:

```
<Trigger>
  <name>Activate Managed Ambience</name>
  <Properties>
    <occurrence>Continuous</occurrence>
    <enabled>true</enabled>
  </Properties>
  <Implies>
    <head>
      <TriggerAction>
        <op> <rel>open</rel> </op>
        <who>trigger</who>
        <device>CUR_DD_001</device>
      </TriggerAction>
      <TriggerAction>
        <op> <rel>PowerOn</rel> </op>
        <who>trigger</who>
        <device>HVAC_DD_001</device>
      </TriggerAction>
    </head>
    <body>
      <And>
        <Atom>
          <op> <rel>greater than</rel> </op>
          <var>TEMP_ENV_001</var>
          <value>28</value>
        </Atom>
        <Atom>
          <op><rel>lower or equal than</rel></op>
          <var>TEMP_HVAC_001</var>
          <value>23</value>
        </Atom>
        <Neg>
          <Atom>
            <op> <rel>equal</rel> </op>
            <var>MOTION_ENV_001</var>
            <value>true</value>
          </Atom>
        </Neg>
      </And>
    </body>
  </Implies>
</Trigger>
```

As discussed in Section 4, first, the name is defined and then the required properties (e.g., *occurrence* and *enabled*). The rule itself is defined under the tag *Implies* and is divided in two sections, *head* and *body*. The *head* contains the different actions (called *TriggerAction*) to be made once the condition of the body is met. Each *TriggerAction* defines who executes the action, on which device it is performed, and under the tag *op/rel* the command to execute is specified. In this case, the trigger will execute *open* on the device *CUR_DD_001*, and *PowerOn* on *HVAC_DD_001*. The condition of the

trigger is defined in the body section, where different logic operands called *Atom* are linked by logic operations, *And*, *Or* and *Neg*. The *Atom* contains the variable, the value, and like with the actions, under the tag *op/rel* the operand to them.

As a result, the rule of the example opens the curtains and turns on the air conditioning when the condition of evaluating the environment is met. The specific condition evaluated for this rule is based on the measure of the environmental temperature, the target temperature of the HVAC and the presence of someone in the surroundings.

6 Conclusion

The latest progress in three domains, i.e., microelectronics, communication and networking technologies, as well as intelligent agents and user interfaces has given rise to the idea of ambient intelligence. It provides added value services by combining the features of the different appliances creating smart environments. This paper presents a novel markup language called Ambient intelligence Triggering MarkUp Language (ATML). This new language, based on RuleML, is aimed at describing in a flexible and scalable way all the possible situations where an AmI system can react to fit the needs and preferences of the users in different situations. In order to validate the design of the ATML and the robustness of the concept it was incorporated to the BackHome project, an European-funded FP7 project, aimed at creating assisting environments for people with severe impairments. It is an ongoing project where the services and devices to be included are growing and changing with the evolution of the project and it will be a real test of ATML. Also, the users tests (technicians or not) will be a valuable input that will help to improve the definition and will be presented in the next steps of the project.

Acknowledgement

The research leading to these results has received funding from the European Communitys, Seventh Framework Programme FP7/2007-2013, BackHome project grant agreement n. 288566.

References

1. Acampora, G., Loia, V.: Using fml and fuzzy technology in adaptive ambient intelligence environments. International Journal of Computational Intelligence Research **1**(1), 171–182 (2005)
2. Acampora, G., Loia, V.: Using fuzzy technology in ambient intelligence environments. In: Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on, pp. 465–470. IEEE (2005)
3. Augusto, J.: Ambient intelligence: Basic concepts and applications. In: J. Filipe, B. Shishkov, M. Helfert (eds.) Software and Data Technologies, *Communications in Computer and Information Science*, vol. 10, pp. 16–26. Springer Berlin Heidelberg (2008)
4. Augusto, J.C., Nakashima, H., Aghajan, H.: Ambient intelligence and smart environments: A state of the art. In: in Handbook of Ambient Intelligence and Smart Environments, pp. 3–31. Springer (2010)

5. Barger, T., Brown, D., Alwan, M.: Health status monitoring through analysis of behavioral patterns. In: 8th congress of the Italian Association for Artificial Intelligence (AI*IA) on Ambient Intelligence, pp. 22–27. Springer-Verlag (2003)
6. Bechtold, U., Sotoudeh, M.: Assistive technologies: Their development from a technology assessment perspective. Gerontechnology **11**(4), 521–533 (2013)
7. Brown, P., Bovey, J., Chen, X.: Context-aware applications: from the laboratory to the marketplace. Personal Communications, IEEE **4**(5), 58–64 (1997)
8. Brown, P.J.: The Stick-e Document: a Framework for Creating Context-aware Applications. In: Proceedings of EP'96, Palo Alto, pp. 259–272 (1996)
9. Casale, P., Fernández, J.M., Rafael, X., Torrellas, S., Ratsgoo, M., Miralles, F.: Enhanching user experience with brain neural computer interfaces in smart home environments. In: 8th IEEE International Conference of Intelligent Environments 2012, INTENV12 (2012)
10. Cook, D.J., Augusto, J.C., Jakkula, V.R.: Ambient intelligence: Technologies, applications, and opportunities (2007)
11. Cook, D.J., Youngblood, G.M., Jain, G.: Algorithms for smart spaces. Technology for Aging, Disability and Independence: Computer and Engineering for Design and Applications, Wiley (2008)
12. Daly, J., Armstrong, E., Miralles, F., Vargiu, E., Müller-Putz, G., Hintermller, C., Guger, C., Kuebler, A., Martin, S.: Backhome: Brain-neural-computer interfaces on track to home. In: RAatE 2012 - Recent Advances in Assistive Technology & Engineering (2012)
13. Das, S.K., Cook, D.J.: Health monitoring in an agent-based smart home. In: In Proceedings of the International Conference on Smart Homes and Health Telematics (ICOST, pp. 3–14. IOS Press (2004)
14. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. **5**(1), 4–7 (2001)
15. Farringdon, J., Nashold, S.: Continuous body monitoring. In: Y. Cai (ed.) Ambient Intelligence for Scientific Discovery, *Lecture Notes in Computer Science*, vol. 3345, pp. 202–223. Springer Berlin Heidelberg (2005)
16. Fernández, J.M., Dauwalder, S., Torrellas, S., Faller, J., Scherer, R., Omedas, P., Verschure, P., Espinosa, A., Guger, C., Carmichael, C., Costa, U., Opisso, E., Tormos, J., Miralles, F.: Connecting the disabled to their physical and social world: The BrainAble experience. In: TOBI Workshop IV Practical Brain-Computer Interfaces for End-Users: Progress and Challenges (2013)
17. Fuentes, L., Jimenez, D., Pinto, M.: An ambient intelligent language for dynamic adaptation. In: Proceedings of Object Technology for Ambient Intelligence workshop (OT4AmI), Glasgow, Uk (2005)
18. Gentry, T.: Smart home technologies for people with cognitive impairment: An affordable, rehabilitative approach. In: Handbook of Ambient Assisted Living, pp. 535–548 (2012)
19. Grosof, B.N.: Representing e-commerce rules via situated courteous logic programs in RuleML (2003)
20. Nambu, M., Nakajima, K., Noshiro, M., Tamura, T.: An algorithm for the automatic detection of health conditions. Engineering in Medicine and Biology Magazine, IEEE **24**(4), 38–42 (2005)
21. Papamarkos G., P.A., Wood, P.T.: Event-Condition-Action Rule Languages for the Semantic Web. In: Proceedings of Workshop on Semantic Web and Databases, Palo Alto, pp. 309–327 (2003)
22. Schmidt, A.: Implicit human computer interaction through context. Personal Technologies **4**(2-3), 191–199 (2000)
23. Schmidt, A.: Ubiquitous computing-computing in context. Ph.D. thesis, Lancaster University (2003)
24. Schmidt, A.: Interactive context-aware systems interacting with ambient intelligence. Ambient intelligence (Part 3), 159–178 (2005)

25. Streitz, N.: From human computer interaction to human?environment interaction: Ambient intelligence and the disappearing computer. In: C. Stephanidis, M. Pieper (eds.) Universal Access in Ambient Intelligence Environments, *Lecture Notes in Computer Science*, vol. 4397, pp. 3–13. Springer Berlin Heidelberg (2007)
26. Vargiu, E., Fernández, J.M., Miralles, F.: Context-aware based quality of life telemonitoring. In: Distributed Systems and Applications of Information Filtering and Retrieval. DART 2012: Revised and Invited Papers. C. Lai, A. Giuliani and G. Semeraro (eds.) (inpress)
27. Vargiu, E., Fernández, J.M., Torrellas, S., Dauwalder, S., Solà, M., Miralles, F.: A sensor-based telemonitoring and home support system to improve quality of life through bnci. In: 12th European AAATE Conference (2013)
28. Vargiu, E., Miralles, F., Martin, S., Markey, D.: BackHome: Assisting and telemonitoring people with disabilities. In: RAatE 2012 - Recent Advances in Assistive Technology & Engineering (2012)
29. Weiser, M.: Some computer science issues in ubiquitous computing. Commun. ACM **36**(7), 75–84 (1993)
30. Youngblood, G.M., Cook, D.J., Holder, L.B.: A learning architecture for automating the intelligent environment. In: Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3, IAAI'05, pp. 1576–1581. AAAI Press (2005)
31. Zadeh, L.A.: Fuzzy sets. Information and control **8**(3), 338–353 (1965)

Using Bloom filters in data leak protection applications

Sergey Butakov

Information Security and Assurance Department,
Concordia University College of Alberta, Edmonton, AB, Canada
sergey.butakov@concordia.ab.ca

Abstract. Data leak prevention systems become a must-have component of enterprise information security. To minimize the communication delay, these systems require fast mechanisms for massive document comparison. Bloom filters have been proven to be a fast tool for membership checkup with some allowed level of false positive errors. Taking into account specific needs of fast text comparison this paper proposes modifications to the Matrix Bloom filters. Approach proposed in this paper allows to improve density in Matrix Bloom filters with the help of special index to track documents uploaded into the system. Density is improved by combining a few documents in one line of the matrix to reduce the filter size and to address the problem of document removal. The experiment provided in the paper outlines advantages and applicability of the proposed approach.

Keywords: Data Leak Protection, Bloom Filters, Text Search

1 Introduction

Data leak protection (DLP) systems become a must-have part of enterprise information security. One of the main functions for these systems is content filtering. Content filtering in DLP is different from the one in antivirus or spam protection applications. In DLP it relies on the internal sources for comparison rather than on signature database maintained by let's say antivirus supplier. Simple content filtering in DLP could be based on keyword screening. More advanced mechanisms allow comparing text in question with a set of documents that are listed for internal purposes only. It is assumed that comparison must be done on the level of paragraphs or sentences and thus lead us to the task of similar text detection. This task of fast detection of near-duplicate documents is not new. It has been studied for years, starting with applications in simple text search in the 1970s [1]. Additional momentum has been added later by massive projects in genome sets comparison and various research fields in internet study. The latter includes near –duplicate web page detection [2], filtering of web addresses [3], internet plagiarism detection [4], and many others. Although there are many methods to for fast text comparison many of them are not suitable for large scale tasks. One of the approaches called Bloom Filters offer speed of comparison while generating some false positive results as a trade-in for the speed.

Bloom Filters (BFs) were introduced in 1970 as a tool to reduce the space requirements for the task of fast membership checkup [5]. As Bloom mentioned in his paper, this method allows significant reduction in the memory requirements at the price of small probability of false positive results [5]. He also indicated that some applications are tolerable to false positives as they require two-stage comparison: the first step is to quickly identify if an object is a member of a predefined set and the second step is to verify the membership. Text comparison can be considered as one of such tasks when on the first phase the system must reduce the number of documents to be compared from hundreds of thousands to thousands. This step reduces the workload for the second phase of comparison where the system needs to find and mark similar segments in the documents.

1.1 Motivating applications

Insider misuse of information is considered to be the top reason for data leaks [6]. Regardless of the fact that such misuses can be either intentional or unintentional, it is always good to have additional defense mechanisms embedded into the information flow to limit the impact of penetration [7]. Commercial Data Leak Prevention (DLP) systems are developed by major players on information security market such as Symantec or McAfee. One of the tasks for such systems is to make sure that documents belonging to the restricted group do not leave the network perimeter of an organization without explicit permission from an authorized person [8]. DLP systems can work on different layers. On application layer they can analyze keywords, tags or the entire message [9]. The decision to allow the document to be transmitted outside should be made quickly and should be prone to false negative errors. False positive could be acceptable up to the certain level if there are embedded controls in the system for manual verification and approval. An additional desired feature for DLP systems is the ability to detect obfuscated texts where some parts of the text are not confidential but other parts are taken from the restricted documents. BFs could play a preselector role that would facilitate fast search by picking from the large archive of protected documents some limited number of candidate sources that could be similar to the document in question. A detailed comparison can be done using other methods that could be slower but more granular.

1.2 Contribution

BFs have been studied in details in the last four decades. In this paper we reviewed the problems related to their practical implementation. In its original form, the BF can identify that the object is similar to one or more objects in a predefined set of objects but it cannot identify which one it is similar to. In other words BFs do not contain localization information. Matrix Bloom Filter (MBF) [10, 11] allows such identification but, as shown in the next section, may impose significant requirements on memory to maintain the set of documents. Dynamic BFs [12] allow better memory utilization and deletion of elements, but still lack localization property. Approach proposed in this paper allows density improvement for an MBF by constructing spe-

cial index to track documents uploaded to MBF. In addition to index, the document removal algorithm with localized workload has been proposed.

Experiment in the last section shows that compacting documents in the MBF can significantly reduce memory consumption by the filter at the cost of maintaining smaller index of the documents that have been uploaded into the filter.

2 PREVIOUS WORKS

The first part of this section discusses original BF and some major modifications that occurred to improve its performance and usability; the second part highlights the areas where BFs can be improved in applications related to near-similar document detection.

2.1 Bloom Filters with Variations

BF was introduced as memory-efficient method for the task of fast membership checkup [5]. The main idea of BF is to use randomized hash functions to translate an object from set S= $\{x_0, x_1, \ldots, x_{(n-1)}\}$ into a binary vector of fixed size m. BF employs k independent hash functions $h_0, h_1, \ldots, h_{k-1}$ to map each element to a random number over the range $\{0, \ldots, m-1\}$. To insert element x into BF, the corresponding bits of BF – $\{h_i(x), 0 \le i \le k-1\}$ – have to be set to 1. To check if element x' is already in BF the corresponding values of $\{h_i(x'), 0 \le i \le k-1\}$ must be calculated and compared with the values in the filter. If all corresponding bits are set in 1 than x' already exists in the filter. Due to randomized nature of h_i BF may produce a false positive [5]. The probability of false positive can be estimated as follows [13]:

$$p' = \left(1 - \frac{1}{m}\right)^{kn} \approx e^{-kn/m} \quad (1)$$

As it can be seen from (1) the false positive probability can be reduced by increasing the filter size m. The optimal size of BF can be defined using n - number of elements in S and acceptable risk of false positive - p'. F has distinctive features:

- BF has constant insert and checkup time due to the fixed size of the filter – constant m. This feature makes it an attractive tool to check membership in large collections because checkup time does not grow along with the collection size.
- Removal of any element from BF requires the entire filter to be recreated. The fact that a single bit in the filter can be set to *1* by many elements leads to the necessity to recreate the entire filter if one element has to be removed.
- If the number of elements in set $\{S\}$ grows above n then the new value for m has to be calculated and filter must be recreated.

The last four decades have brought many variations to BFs, aiming to overcome problems of member elements removal, filter fixed size limitations, and some others. Counter BFs [14] use bit counters instead of a single bit value in the filter. This approach allows to store the number of times the particular index has been initialized.

The element addition/removal to/from the filter can be done by increasing/decreasing corresponding counter. Although such an approach keeps the membership checkup operation simple and computationally effective, it increases memory requirements. For example, one byte counter that can store up to 255 initializations of the particular index would increase the BF memory consumption by the factor of 8.

Dynamic or matrix Boom filters (MBF) [12] handle the growing size of $\{S\}$ by adding additional zero-initialized vectors of size m to the filter when number of elements in S reaches n. Each new vector is considered a brand new zero-initialized BF. Such an approach allows indefinite growth of S at the cost of additional checkups because the membership has to be checked in each row in the matrix. The number of checkups required can be considered as linear to the size of S and it is of $\theta(|\{S\}|/n)$ order. Removal of any text would cause the corresponding row to be removed from the matrix.

For the applied problems of document comparison, the speed of comparison can be considered as a priority metric, while the speed of update operations such as addition and deletion of elements can be less of the priority. This metric selection is based on the assumption that a set of documents for the comparison is relatively steady if compared to the information flow that has to go through the filter.

2.2 Related Problems

Near to duplicate document detection usually works on the level of a few consecutive words or a string sequence of the same length without selecting word from the test. Such granularity allows to locate meaningful similarities in the texts while staying prone to minor text alternations. The sequence of words or characters can be called a shingle or a grammar [15, 16]. A full set of these sequences defines text $T = \{t_0, t_1, \ldots, t_{z-1}\}$. If sequencing is done on the word level, then z is close to the number of words in the document. If sequencing is done on the character level it could lead to excessive representations. Many algorithms with proven performance and robustness have been developed to address this issue. For example, Winnowing algorithm provides $d=2/(w+1)$ density of the selection where w is number of characters in the sequence (w-gram) [16].

If we are to compare two documents, T_i and T_j of length z_i and z_j respectively, and the comparison has to be done using BF, then m must satisfy the following inequality: $m \geq max\{z_i, z_j\}$. Obviously, if BF is to be used to compare one document T' with many documents $\{T\}$ then BF has to be large enough to fit all of these many documents. Straightforward sequencing of all the documents followed by placing shingles into the BF will not work because classical BF is not local. In other words, if two documents contain exactly the same shingle (sequences of words or characters) and both documents were placed in BF, the same shingle flips to *1* same bits and therefore there will be no way to identify the source document. It is obvious that similar problem appears when there is a need to remove a document from the comparison set $\{T\}$. As Geravand & Ahmadi suggested, the localization problem can be addressed by matrix BF.

They proposed to use a set of BFs where each BF represents shingles from one document [11]. The document that is scheduled for the checkup generates its own BF which will be compared with every row in matrix BF on the next step. Although a set of independent BFs does not look like a typical BF, it has one important property: the number of *XOR* operations to check the membership in the entire matrix BF is linear to the matrix size. This important performance property is achieved by using the same fixed *m* for all rows of matrix BF.

This approach has the disadvantage of inefficient memory consumption by each row in the matrix because of two factors: the requirement to have one document per row and the requirement to have constant *m*. The last one provides the possibility to calculate BF for *T'* only once and therefore dramatically improve the comparison speed. The first requirement can be relaxed with the additional indexing added to the matrix. The following section proposes such an index and outlines related algorithms.

3 PSEUDO-MATRIX BLOOM FILTER

Placing all the shingles for the entire document collection in one large BF is not an option in DLP. These systems require not only the confirmation that a particular shingle from the document in question belongs to the collection but also require linking the same shingles to the potential source document. The proposed data structure resembles the one proposed by Geravand & Ahmadi for the task of plagiarism detection. Pseudo-Matrix BF (PMBF) will consist of an unlimited number of BFs of the same size [11]. The major alternation is to use one BF to store more than one document where it is possible. From the equation (1) we can find *m* that would allow to limit *p* for a maximum of *n* elements [13]:

$$m \geq -\frac{n\ ln(p)}{(ln(2))^2} \qquad (2)$$

Keeping in mind that *m* is fixed for the entire PMBF to achieve the fast comparison speed, we can state that if there is a restriction to place one and only one document in single BF we can meet the following cases:

- Number of shingles in a document exceeds *n*, thus the entire document cannot be placed in a single row;
- Number of shingles in a document is much less than *n*, thus the BF is underpopulated which makes it memory inefficient.

Suboptimal memory allocation would require *p* to be slightly less than *n*. To provide the such memory allocation by PMBF, two additional indexes have to be maintained: one to keep track of the remaining capacity of each row and another one to store indexes of documents placed in each row of PMBF. Figure 1 provides an example of a PMBF. The example assumes that each row in BLOOM FILTERS matrix is populated with two documents *(T)*. The example can be extended to any number of documents in a single row. INDEX part of the structure links filter rows with uploaded texts $\{T_i\}$. For each row the corresponding INDEX list identifies the documents

that are stored in this row. For example, Figure 1 assumes that each document is stored in two rows. As the last part of the paper shows in real applications there will be no such even distribution and many documents could be stored in one row. If *BF'* compiled from *T'* has a certain level of similarity with BF_i we can say that *T'* is probably similar to one of the documents *(T)* located in BF_i. The detailed comparison must be done to confirm this. It can be done with any appropriate method such as Levenstein's distance; this stage of comparison is out of scope of this paper.

The proposed data structure allows relatively easy document removal with no increase in memory consumption. If T_i to be removed, then only the row(s) that contain contains T_i has to be recreated. For example, in Figure 1 if document T_{1000} are to be removed from the collection then only BF_{h-1} and BF_h have to be recreated making it relatively minor work from the computational perspective. Thus the additional work is not very computationally expensive if comparing with matrix BF presented in [11] but there is higher shingle density in the matrix. The density of each individual filter in the matrix can be evaluated using the index. Under-populated filters can be filled up with additional text segments.

BLOOM FILTERS		**INDEX**
BF_0	----	
. . .	----	
BF_{h-2}: T_{998}, T_{999}	----	I_{998}; I_{999};
BF_{h-1}: T_{999}, T_{1000}	----	I_{999}; I_{1000};
BF_h: T_{1000}, T_{1001}	----	I_{1000}; I_{1001};
BF_{h+1}: T_{1001}, T_{1002}	----	I_{1001}; I_{1002};
. . .	----	
BF_{d-1}	----	

Fig. 1. Example of Pseudo-Matrix Bloom Filter with two documents in a row

The proposed architecture has the following important features:

- Compared to an MBF proposed earlier, it increases density for each BF_i, therefore increasing its memory efficiency.
- Increased density leads to better computational efficiency of the data structure. Although the comparison computations are still of $\theta(n*d)$ order but in this structure d is the number of rows required to allocate the document corpus instead of total number of documents. As the experiment below shows it could be few times less than the total number of documents.
- It keeps number of binary checkups linear to the size of the collection *{T}*.
- It allows relatively easy document removal.
- Matrix size is pseudo linear to the total number of shingles in the collection.

Example calculations of the size and density provided below show the memory efficiency of the proposed approach. Second part of the experiment shows applicability and limitations of the proposed approach.

4 EXPERIMENT

Two experiments have been conducted to test the proposed approach of PMBF utilization. First experiment has been aimed to evaluate the compression ratio of the PMBF and second experiment has been conducted to study applicability of PMBF on real data.

4.1 Evaluation of memory consumption by PMBF

The goal of the first experiment was to show on the real data that PMBF has distinctive size advantage over matrix BF. In this case the experimental calculations have been performed on the simulated plagiarism corpus used for the PAN'2009 plagiarism detection competition [17]. The corpus contained 14,428 source documents simulating plagiarism. The average number of words in a document is about 38,000; 95% of all documents have less than 123,000 words. The calculations provided in the Table 1 were done with the following assumptions:

- Text shingling is done by sliding window on the word basis. Keeping in mind that shingles (consecutive phrases of *y* words) are relatively short, we can safely assume number of shingles to be equals to the number of words in the text. The number of words in a shingle may affect the granularity of the search in PMBF but this question is out of scope of this paper.
- The order in which the documents are placed in the filter does not affect its memory capacity. In practice, this will not be absolutely true because if random documents are constantly being added and deleted to/from PMBF, it could lead to a fragmented index thus decreasing the performance of the index search. The issue can be addressed by setting up some threshold to protect under-populated rows from accepting short sequences of shingles. This option will be explored in the future research.

As it can be seen from Table 1, for PAN'09 corpus PMBF has compressed the information in 1:3.1 ratio comparing to MBF because the straightforward implementation of one text per row approach would require 14,428 rows in matrix instead of 4489. Moreover, in MBF about 5% of the documents would not fit a single row and therefore would be out of search scope. The option to increase n up to the size of the largest document in the corpus (~428,000 words) would increase an MBF size by the factor of 4.

The compression ratio of 1:3.2 also indicates that 95% of the documents will be collated in one or two rows of the matrix. The latter case covers those documents that start in one row and end in the next one; therefore, in 95% of cases only two rows will

have to be recreated if such a document is scheduled for removal from the filter. On the related issue of space reuse, we can suggest that algorithms similar to garbage collection can be implemented to take care of the released space after a document has been removed from PMBF. The garbage collection can be implemented in the system's off-peak time when search results can be delayed.

Table 1. Filter Size and Operations for the Corpus.

Value	Explanation
p=0.05	5% false positive probability
n=123,000	Filter of this size will accommodate 95% of texts. If text has more than 123,000 words it will be placed in more than one filter. If text has less than 123,000 words empty space will be occupied by another text.
m=766,933 bit.	See equation (2). 766,933 bite ~= 94 kilobytes
d=4489	Total number of rows in PMBF = Total number of words in corpus / Number of words to fit in one row
M ~=411 Megabytes	Total size of PMBF to accommodate PMBF for the entire corpus of 14,428 documents.

Obviously, improvement in space allocation comes with a cost of more comparisons to be done to confirm the similarity of the document, if an MBF similarity of *BF'* and BF_i means that respective texts *T'* and T_i must be compared more closely. In the case of PMBF, the similarity of BF' and BF_i means that *T'* must be compared with more than one document. According to the example above, in 95% of cases on average *T'* will have to be compared with a minimum of 4 documents and a maximum of 8 documents because 95% of the documents are located in one or two rows. Therefore, on one hand increasing *n* helps to improve the compression, but it also increases the workload for the system component that performs detailed comparison.

4.2 Applicability of PMBF for DLP tasks

Goal of the second experiment was to test the applicability of PMBF for text similarities detection. As it was mentioned earlier, DLP systems have to be prone to false negatives and keep false positives in reasonable range. In our case PMBF is used on the first stage of comparison which is to indicate the potential similarity among documents thus reducing the range of the detailed comparison from thousands of potential matches to more manageable number which can be processed on the second page by more precise but slower methods.

Two data sets were used for the evaluation. One data set included 500 documents with text inserts from other 50 source documents. Each randomly placed insert had been taken from only one of 50 sources. The length of the inserts varied from 5% to 50% of the total document length. Each insert was placed as one piece in the random

place of the document. All the documents consisted of about 4000 words. Due to the fact that the documents in this corpus were about the same length the compression was not implemented in this part of the experiment and therefore each document had been placed in the separate row of PMBF. Each of 50 source documents has been compared with all 500 documents from this corpus. The level of similarity was evaluated as number of ones in the same cells of the one row [11]. It was naturally expected to see some similarity among totally different documents as BFs are tailored to have some level of false positives. The question was to see if the actual inserts produce higher level of similarity which is distinct from the background similarity levels. The experiment indicated that for all 50 sources it was true. For each source document the documents that had some parts from that source produced distinctive pikes on the similarity level graph. An example of such graph for one of the sources is presented on Figure 2. It shows 10 distinct similarity levels for 10 sources that included from 5% (leftmost pike) to 50% (rightmost pike) of the text from that particular source. Based on this part of the experiment we can state that PMBF are suitable for preliminary comparison of text documents in the cases where large portions – 5% or above – were copied without variations from one of confidential documents.

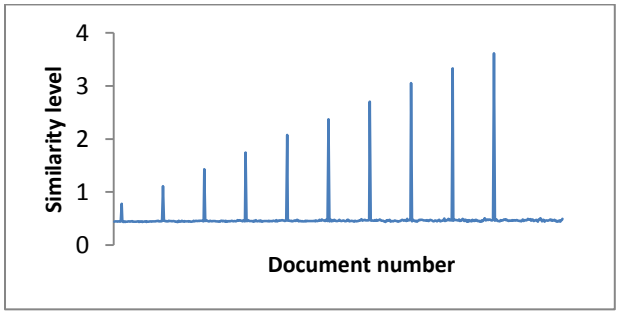

Fig. 2. Example of source document comparison for the first corpus of 500 documents.

Second phase of experiment was conducted to evaluate level of false positives and false negatives on the larger corpus with obfuscated text. Training corpus from PAN'09 plagiarism detection competition [17] was used for PMBF performance evaluation. Although classic BF is prone to false negatives this may not be the case for PMBF. When maximum size of a document – *n* in equations (1) and (2) - is being used to allocate bit string, it represents maximum number of shingles in the document. But in fact the proper estimation of the number of all potential elements in one row should take into account all possible shingles. For example, if triplet of words is being used as a shingle than proper estimation of n would be all possible triplets from the dictionary: dictionary of 10^4 words would generate about 10^{12} combinations. Therefore estimating *n* as maximum number of shingles in a document increases false positives. Such increase eventually will lead to false negatives as less strong comparison criteria will be used to reduce number of false positives.

PAN'09 corpus consists of 14,428 documents [17]. Number of pair-wise comparisons for all the documents in the corpus would be about $(14*10^3*14*10^3)/2 \sim 10^8$.

PMBF will be used on the first phase of DLP checkup process to reduce this number to at least thousands or less. The comparison process was done using two approaches to populate PMBF. In first approach one document was placed in one row only but each row may contain many documents. In the second approach longer documents were distributed among many lines. As it can be seen from Table 2 second approach leaded to much less false positives but as a trade in for the better compression number of false negatives doubled. Additionally in the second approach each row in the PMBF were populated only up to 50% of its potential capacity to reduce false positives. Both approaches produced some false negative results that are not desirable for DLP systems. Detailed study of false negatives indicated that all of them were caused by the documents that contained less than 1% of the text from the source documents. Moreover 26 documents out of 117 false negatives text inserts from the source documents were highly obfuscated. Since the comparison was done on the word level without any additional tools to tackle text obfuscation we can state that these false positives were expected. Two methods produced different amount of false positives. In the second case when density of BF was reduced by 50% the number of false positives decreased significantly. This feature of PMBF gives DLP developers additional choice – if they are ready to use twice as much memory for PMBF then the second stage of comparison will be much less loaded because of much lower level of false positives.

These two experiments indicated that PMBF can be used in DLP if it is acceptable that only larger portions of the restricted documents will be filtered out. This leaves the attackers with the potential to split the document into the smaller chunks to avoid filtering. This would be very suitable when DLP is intended to protect users from errors such as typos in the email address or other mistakes where users do not have intentions to violate the document distribution rules.

Table 2. False positive and false negative results of the experiment for two approaches to populate PMBF

	Document in one line	Distributed documents
False positive results	301977 (~1%)	9322 (~0.01%)
False negative results	51 (~5%)	117 (~11%)

5 CONCLUSION

The paper proposes to use improved data structure based on Bloom filter to address the issue of fast document comparison. The proposed data structure and algorithms allow better memory allocation in Bloom filters aimed on document comparison. Additional index allows BF to locate the potentially similar documents even if few documents have been placed in a single row of the filter. This index also allows computationally effective document removal operations.

One limitation of the proposed approach is related to the fact that using PMBF makes sense only if the entire filter could be allocated in the computer RAM where fast bitwise comparison is possible. Placing parts of the filter on the disk will fade its speed advantage. Based on the experiment above we can state that even minimal server configuration with few Gigabytes of RAM can handle hundreds of thousands of documents which seems to be suitable for DLP systems for a medium enterprise. As first experiment shows PMBF provides noticeable size advantage over matrix BF. Second experiment indicated that PMBFs are applicable for filtering in DLP if document in question includes larger portion (above 5%) of the restricted document. This limitation may not be a problem depending on the purposes of the specific DLP.

ACKNOWLEDGEMENTS

Author would like to acknowledge productive discussions on Bloom Filters with Dr. A. Tskhay, and Mr. V. Shcherbinin as well as help with experiments from Mr. L. Shi and Mr. V. Dyagilev.

REFERENCES

1. Knuth, D., Morris, Jr., J., and Pratt, V. Fast Pattern Matching in Strings. SIAM Journal on Computing 1977 6:2, 323-350 (1977)
2. Brin S., Davis J., and García-Molina H. Copy detection mechanisms for digital documents. SIGMOD Rec. 24, 2 (May 1995), 398-409. DOI=10.1145/568271.223855
3. Cormack G. V. (2008). Email Spam Filtering: A Systematic Review. Found. Trends Inf. Retr. 1, 4 (April 2008), 335-455. DOI=10.1561/1500000006 (1995)
4. Butakov S. and Scherbinin V. The toolbox for local and global plagiarism detection. Comput. Educ. 52, 4 (May 2009), 781-788. DOI=10.1016/j.compedu.2008.12.001 http://dx.doi.org/10.1016/j.compedu.2008.12.001.(2009).
5. Bloom B. H. Space/time trade-offs in hash coding with allowable errors. Commun. ACM 13, 7 (July 1970), 422-426. DOI=10.1145/362686.362692. (1970).
6. Liu, S.; Kuhn, R. Data Loss Prevention. IT Professional , vol.12, no.2, pp.10-13, March-April 2010 doi: 10.1109/MITP.2010.52. (2010).
7. Blackwell C. A security architecture to protect against the insider threat from damage, fraud and theft. In Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW '09), Article 45 , 4 pages. DOI=10.1145/1558607.1558659 http://doi.acm.org/10.1145/1558607.1558659. (2009).
8. Lawton G. New Technology Prevents Data Leakage. Computer 41, 9 (September 2008), 14-17. DOI=10.1109/MC.2008.394 http://dx.doi.org/10.1109/MC.2008.394. (2008).
9. Potter B. Document Protection: Document protection. Netw. Secur. 2008, 9 (September 2008), 13-14. DOI=10.1016/S1353-4858(08)70108-9 http://dx.doi.org/10.1016/S1353-4858(08)70108-9. (2008).
10. Wang J.; Xiao M.; Dai Y. "MBF: a Real Matrix Bloom Filter Representation Method on Dynamic Set," Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on , vol., no., pp.733-736, 18-21 Sept. 2007 doi: 10.1109/NPC.2007.107 (2007).

11. Geravand, S.; Ahmadi, M. A Novel Adjustable Matrix Bloom Filter-Based Copy Detection System for Digital Libraries, Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on , vol., no., pp.518-525, Aug. 31 2011-Sept. 2 2011 doi: 10.1109/CIT.2011.61 (2011)
12. Guo, D.; Wu, J.; Chen, H.; Luo, X.; Theory and Network Applications of Dynamic Bloom Filters, INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings , pp.1-12, April 2006 doi: 10.1109/INFOCOM.2006.325 (2006)
13. Broder A.Z. and Mitzenmacher M. Network Applications of Bloom Filters: A Survey. Internet Mathematics, Vol 1, No 4, 2005. P485-509 (2005).
14. Fan L., Cao P., Almeida J., Broder A. 2000. Summary cache: a scalable widearea web cache sharing protocol. IEEE/ACM Trans. on Networking, vol.8, no.3, pp.281-293, (2000).
15. Karp R. M. and Rabin M. O. Pattern-matching algorithms. IBM Journal of Research and Development, 31(2):249–260, 1987. (1987).
16. Schleimer S., Wilkerson D., and Aiken A. Winnowing: Local Algorithms for Document Fingerprinting. Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 76-85, June 2003. (2003).
17. Potthast M., Eiselt A., Stein B., Barrón-Cedeño A., and Rosso P. PAN Plagiarism Corpus PAN-PC-09. http://www.uni-weimar.de/medien/webis/research/corpora, (2009)

Dense Semantic Graph and its Application in Single Document Summarisation

Monika Joshi[1], Hui Wang[1] and Sally McClean[2]

[1] University of Ulster, Co. Antrim, BT37 0QB, UK
joshi-m@email.ulster.ac.uk , H.Wang@ulster.ac.uk
[2] University of Ulster, Co. Londonderry, BT52 1SA, UK
sally@infc.ulst.ac.uk

Abstract Semantic graph representation of text is an important part of natural language processing applications such as text summarisation. We have studied two ways of constructing the semantic graph of a document from dependency parsing of its sentences. The first graph is derived from the subject-object-verb representation of sentence, and the second graph is derived from considering more dependency relations in the sentence by a shortest distance dependency path calculation, resulting in a dense semantic graph. We have shown through experiments that dense semantic graphs gives better performance in semantic graph based unsupervised extractive text summarisation.

1 Introduction

Information can be categorized into many forms -- numerical, visual, text, and audio. Text is abundantly present in online resources. Online blogs, Wikipedia knowledge base, patent documents and customer reviews are potential information sources for different user requirements. One of these requirements is to present a short summary of the originally larger document. The summary is expected to include important information from the original text documents. This is usually achieved by keeping the informative parts of the document and reducing repetitive information.

There are two types of text summarization: multiple document summarisation and single document summarization. The former is aimed at removing repetitive content in a collection of documents. The latter is aimed at shortening a single document whilst keeping the important information. Single document summarisation is particularly useful because large documents are common especially in the digital age, and shortening them without losing important information is certain to save time for the users/readers. The focus of our research is on single document summarisation. In order to process a text document, it should be broken down into parts and then represented in a suitable form to facilitate analysis. Various text representation schemes have been studied, including n-gram, bag of words, and graphs. In our research we use graphs to represent a text document. The graph is constructed by utilising semantic relations such as dependency relations between words within the sentence.

We propose a novel graph generation approach, which is an extension of an existing semantic graph generation approach [4] by including more dependencies from dependency parsing of the sentence. This results in dense semantic graph. We evaluated both graphs in a text summarisation task through experiments. Results show that our dense semantic graph outperformed the original semantic graph for unsupervised extractive text summarization.

The next section gives a short literature review of the earlier graph based approaches to text summarisation. In section 3, a detailed description is provided concerning the construction of two different semantic graphs that were used in our study. Section 4 discusses extractive summarisation based on these semantic graphs and section 5 describes the experiments and results. After that conclusion of the analysis follows.

2 Previous Work on Graph based Text Summarisation

Earlier researchers have used graph representation of documents and properties of graphs to extract important sentences from documents to create a short summary. Graph based text summarisation methods such as LexRank [1], TextRank [2] and Opinosis [3] have shown good performance. There are two types of graph that are constructed and used to represent text. Lexical graph uses the lexical properties of text to construct a graph. LexRank and Text Rank are lexical graph based approaches. They construct graphs by connecting two sentences/smaller text units as nodes in the graph based on the degree of content overlap between them.

On the other hand, semantic graph is based on semantic properties of text. Semantic properties are: Ontological relationship between two words such as synonymy, hyponymy; relationship among set of words representing the syntactic structure of sentence such as dependency tree and syntactic trees. A set of words along with the way they are arranged provides meaning. The same set of words connected in different ways gives different meaning.

According to the semantic properties utilised for graph construction, various representations have been reported in literature for semantic graphs [4, 5]. Some of the approaches utilize the lexical database WordNet to generate ontological relations based semantic graph. In this sentences are broken into terms, mapped to WordNet synsets and connected over WordNet relations [6]. In one of the approaches called semantic Rank [7], sentences are connected as nodes and the weight of the edges between them is the similarity score calculated by WordNet and Wikipedia based similarity measures. Other approaches to generate semantic graphs try to utilize the dependency relations of words in a sentence along with the ontological relations between words. Utilizing this particular order of connection also forms the basis of research work done on semantic graphs in our study. In this area of semantic graph generation most of the work has been concentrated on identifying logical triples (subject-object-predicate) from a document and then connecting these triples based on various semantic similarity measures [4]. Predicate (or verb) is the central part of any sentence, which signifies the main event happening within the sentence. Thus it was

mostly agreed to consider the verb and its main arguments (subject and object) as the main information presented in the sentence, and use this as a basic semantic unit of the semantic graph. Various researches have been done on this graph in the field of supervised text summarisation.

We have evaluated two semantic graphs which are based on the dependency structure of words in a sentence. The first graph is triple(subject-object-verb) based semantic graph proposed by Leskovec et al [4]. The second graph is a novel approach of semantic graph generation proposed in this paper, based on the dependency path length between nodes. Our hypothesis is that moving to a dense semantic graph, as we have defined it, is worthwhile. The principle idea behind this new graph has been used in earlier research in kernel based relation identification [8]. However it has not been used for construction of a semantic graph for the complete document. The next section describes more details about this graph.

3 Semantic Graphs

In the research carried out in this paper, we have analysed the difference between performances when more dependency relations than just subject-object-verb are considered to construct a semantic graph of the document. In this direction, we have developed a methodology to select the dependencies and nodes within a shortest distance path of dependency tree to construct the semantic graph. First we will describe the previous use of graphs and then we will introduce the graph generated by our methodology.

3.1 Semantic graph derived from a triplet (Subject-Object-verb)

Leskovec et al. [4] has described this graph generation approach for their supervised text summarization, where they train a classifier to learn the important relations between the semantic graph of a summary and the semantic graph of an original text document. In this graph the basic text unit is a triple extracted from sentence: subject-verb-object. This is called triple as there are three connected nodes. Information such as adjectives of subject/object nodes and prepositional information (time, location) are kept as extra information within the nodes. After extracting triples from every sentence of the text document two further steps are taken: i. co-reference and anaphora resolution: all references to named entities (Person, Location etc.) and pronoun refeences are resolved. ii. Triples are connected if their subject or object nodes are synoymous or referring to the same named entity. Thus a connected semantic graph is generated.

3.2 Dense Semantic graphs generated from shortest dependency paths between Nouns/Adjectives

We have observed that various named entities such as location/time which are important information, are not covered in the subject-predicate-object relations. As this

information is often added through prepositional dependency relations, it gets added to nodes as extra information in the semantic graph generated by previous approaches. However these named entities hold significant information to influence ranking of the sentences for summary generation and to connect nodes in the semantic graph. This has formed the basis of our research into new way of semantic graph generation. First we elaborate the gaps observed in previous approach of semantic graph generation and then give the details of the new semantic graph.

Gaps identified in triple (subject-object-verb) based semantic graph.

The kind of information loss observed in the previous semantic graphs has been described below:

- Loss of links between words in sentence
  Some connections between named entities are not considered because they do not come into the subject/object category. This information is associated with subject/object, but does not get connected in the semantic graph, as they are not directly linked through a predicate. For example consider the sentence below:

  President Obama's arrival in London created a joyful atmosphere.
  The triple extracted from this sentence is:
  Arrival->create->atmosphere

  Here the information *London*, *Obama* is added as extra information to node *Arrival*, and *Joyful* is added to node *Atmosphere*. However a direct link between *London* and *atmosphere* is missing, whereas a reader can clearly see this is atmosphere of London. This connection can be identified in our shortest dependency path graph as shown below:

  *London-**prep-in**->Arrival-**nsubj**->created-**dobj**->atmosphere*

- Loss of inter-sentence links between words

  Some named entities which are not subject/object in one sentence are subject/object of another sentence. When creating a semantic graph of complete document, these entities are the connecting words between these sentences. In the previous graph these connections are lost as shown below by two sentences.

  He went to church in Long valley.
  One of the explosions happened in Long Valley.

  The triple extracted from these sentences is:

  He->went>church
  Explosion->happened->long valley

In the semantic graph derived from triples of the above 2 sentences, we do not have both these sentences connected, because the common link *Long Valley* is hidden as extra information in one semantic graph.

- Identification of subject is not clear

  In a few cases, identification of a subject for the predicate is not very accurate with current dependency parsers. This case occurs in the clausal complement of verb phrase or adjectival phrases called dependency relation "xcomp". Here the determination of subject for clausal complement is not very accurate, as the subject is external.

Construction of shortest distance dependency path based semantic graph

To overcome these gaps, we construct the semantic graph by connecting all noun and adjectives which are connected within a shortest path distance in the dependency tree of that sentence. From the literature review it has been identified that nouns are the most important entities to be considered for ranking sentences. So we have decided to include nouns as nodes in the semantic graph. We also considered adjectives, as they modify nouns and may present significant information. The length of the shortest path is varied from 2-5 to analyse its effect on the efficiency of the PageRank score calculation. The following steps are followed to construct the semantic graph

- Co-reference resolution of named entities
  The text document is preprocessed to resolve all co-references of named entities. We replace the references with the main named Entity for Person, Location, and organization.

- Pronominal resolution
  After co-reference resolution, text is preprocessed for pronominal resolution. All reference (he, she, it, who) are resolved to referring named entities and replaced them in text.

- Identifying nodes and edges of the semantic graph

  The shortest path distance based Semantic graph is defined as G= (V, E), Where

$$V = \left\{ \bigcup_{word_i \in document} Word_i : \text{pos}(Word_i) \in \{\text{JJ}*, \text{NN}*\} \right\} \quad (1)$$

In (1) $\text{pos}(Word_i)$ provides part of the speech tag of $Word_i$. According to Penn tag set for part of speech tags, "JJ" signifies Adjectives and "NN" signifies Noun.

$$\text{Edge set } E = \{\bigcup_{u,v \in V}(u, v) : SD(u, v) \leq limit\} \quad (2)$$

In (2) $SD(u, v)$ is the shortest distance from u to v in the dependency tree of that sentence and *limit* is the maximum allowed shortest path distance, which is varied from 2-5 in our experiments.

We have used Stanford CoreNLP package for co-reference resolution, identification of named entities and dependency parse tree generation[9] [10]. To develop the graphs and calculate the page rank scores of nodes we use the JUNG software package[1]. First we extract dependency relations for each sentence. Then we generate a temporary graph for the dependency tree of that sentence in JUNG .Then Dijkstra's shortest path algorithm is applied to find the shortest distance between nodes. From this temporary graph we find vertices and edges based on equations (1) and (2) to construct the semantic graph.

Fig. 1 and 2 show two graphs, triple based semantic graph and shortest distance dependency path based semantic graphs for the given excerpt of 2 sentences below, taken from the *Long Valley* document of DUC2002 data.

A text excerpt taken from DUC 2002 data.

The resort town's 4,700 permanent residents live in Long Valley, a 19-mile-long, 9-mile-wide volcanic crater known as a caldera. Eruptions somewhat smaller than Mount St. Helens' happened 550 years ago at the Inyo craters, which span Long Valley's north rim, and 650 years ago at the Mono craters, several miles north of the caldera.

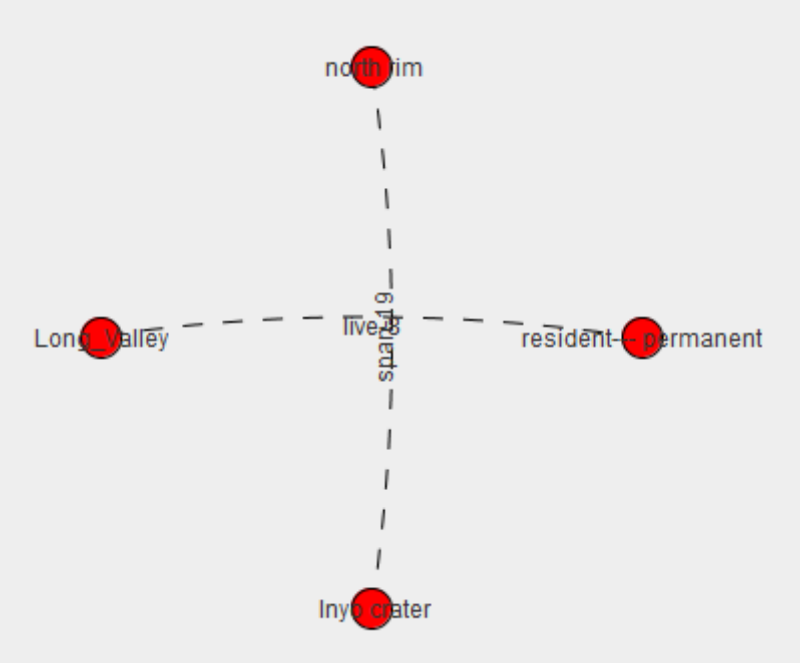

Fig. 1. The triple based semantic graph for the text excerpt taken from DUC 2002 data

[1] http://jung.sourceforge.net/

The next section describes the methodology to rank sentences based on the semantic graph described in this section.

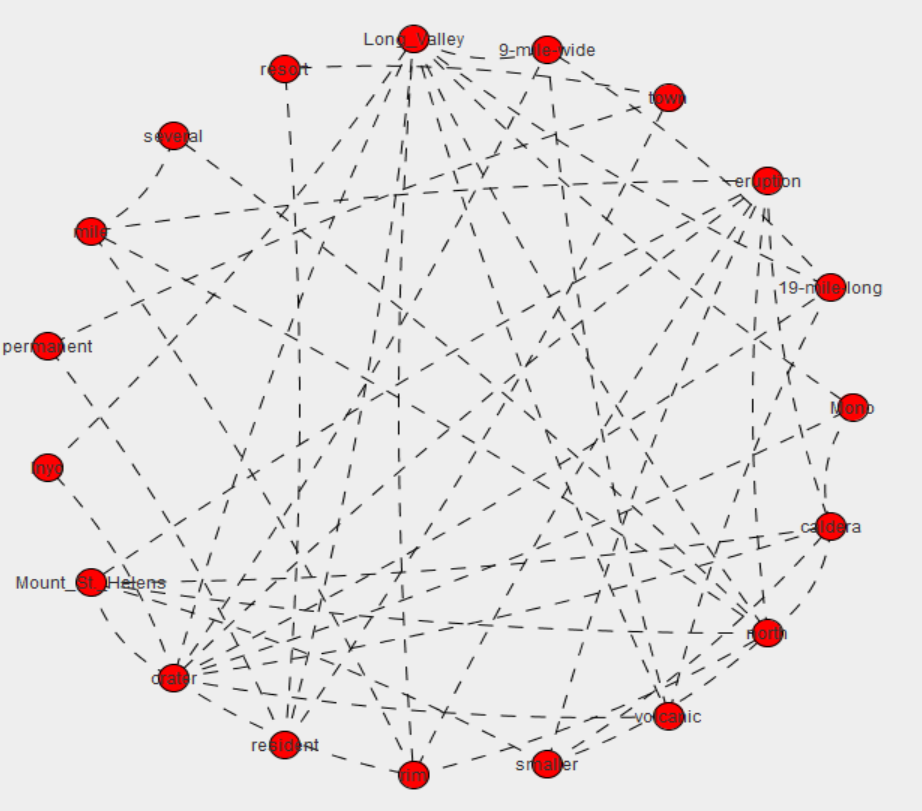

Fig. 2. Sematic graph based on the shortest dependency path between nouns/adjectives (shortest distance=2) for the text excerpt taken from the DUC 2002 data

4 Extraction of Sentences for Summary Generation

In this paper we want to analyse the impact of dense semantic graphs on text summarisation and provide a comparison with the summarisation results of earlier triple based semantic graphs. To achieve this, first we rank the semantic graph by one of the graph ranking algorithm. We have used PageRank method to rank the semantic graph nodes.

The PageRank score of $node_i$ is calculated as:

$$PageRank(node_i) = (1-d) + d * \sum_{node_j \in In(node_i)} \frac{PageRank(node_j)}{|Out(node_j)|} \quad (3)$$

Where *d* is the probability of jumping from $node_i$ to any random node in the graph, typically set between 0.1-0.2. $In(node_i)$ is the set of incoming edges to $node_i$ and $Out(node_j)$ is the set of outgoing edges of $node_j$. Initially PageRank of all nodes is intialised with arbitrary values, as it does not affect the final values after convergence. In this paper semantic graphs are undirected graphs so incoming edges of a node are equal to outgoing edges.

After calculating PageRank score of the nodes in the semantic graph, the score of sentence S_i in the text document is calculated by following equation:

$$Score_{S_i} = \sum_{node_j \in graph \cap S_i} PageRank(node_j) \quad (4)$$

where $node_j$ is the stemmed word/phrase in the graph representation. Scores are normalised after dividing by the maximum score of sentences. After calculating normalized scores of all sentences in the text document, sentences are ordered according to their scores. As per the summary length, higher scoring sentences are taken as summary sentences.

In addition to this summary generation method, we have also tried to analyze impact of including additional features together with PageRank scores on semantic graph based text summarisation. This was done in a separate experimental run where we have included sentence position as an additional feature for scoring of sentences. Since the data we have experimented with is news data, a higher score is given to early sentences of the document. So the score of a sentence S_i after including sentence position, *i* as a feature is given by:

$$newScore_{S_i} = 0.1 \times (Count_{sentences} - i) \Big/ Count_{sentences} + 0.9 \times Score_{S_i} \quad (5)$$

After calculating the new score of the sentences, higher scoring sentences are extracted as the summary as in previous summarisation method. The next section describes the experimental setup.

5 Experiments

We have experimented on two single document summarisation corpuses from Document Understanding Conference (DUC), DUC-01 and DUC-02.

DUC-01 contains 308 text documents and DUC-02 contains 567 text documents. Both sets have 2 human written summaries per document for evaluation purposes. We have used the ROUGE toolkit to evaluate system generated summaries with reference summaries, that are 2 human generated summaries per document [11]. The ROUGE toolkit has been used for DUC evaluations since the year 2004. It is a recall oriented evaluation metric which matches n-grams between a system generated summary and reference summaries.

$$\text{Rouge} - \text{N} = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (6)$$

Rouge-1 is 1-gram metric. Rouge-2 is 2-gram metric. Rouge-W is the longest weighted sequence metric, which gives weight to consecutive longest sequence matches.

ROUGE scores were calculated for different summarisation runs on triple based semantic graphs and shortest dependency distance path based semantic graphs. On triple based graphs two summarisation tasks were run for DUC01 and DUC-02 data. The first considered PageRank only and the second used PageRank, sentence position (Triple based, Triple + position). On the Shortest distance dependency path based semantic graph, 6 summarisation tasks were run for both datasets. The first 4 runs are based on PageRank scores alone by varying shortest distance from 2-5: shortest distance 2 (SD-2), shortest distance 3 (SD-3), shortest distance 4 (SD-4) and shortest distance 5 (SD-5). The fifth and sixth run include sentence position as feature with SD-4 and SD-5(SD-4 + position, SD-5+ position). We have also compared our results with the results of the text summarisation software *Open Text Summarizer(OTS)* [12], which is freely available and has been reported to perform best between other available open source summarizers.

6 Results and Analysis

Figure 3 shows the ROUGE-1, ROUGE-2 and ROUGE-W scores for DUC-01 data achieved by different experimental runs described in section 5. The Rouge evaluation setting was a 100 words summary, 95% confidence, stemmed words and no stop words included during summary comparison.

In figure 3, we have observed that the lowest Rouge scores are reported with the triple based experiment. By including position, results for triple based experiment are improved. Rouge-1 scores for SD-2, SD-3, SD-4, and SD-5 improves systematically and are better than triple based and triple based + position. This shows that as the shortest length of dependency path was increased from 2 to 5, the Rouge score has improved due to better ranking of the nodes in the semantic graph. This better ranking can be attributed to more connections found after increasing the path distance to find links in the dependency tree. A similar trend of increase in ROUGE-2 and ROUGE-W scores are observed for experiments on DUC-02 data in SD-2, SD-3, SD-4, SD-5, SD-4+position, and SD-5+position.

Although benchmark OTS results are always higher than best results achieved by our approach, it is useful to observe that our results are comparable to the benchmark results, as the main purpose of our research is to analyse the impact of dense semantic graphs on text summarisation compared to previous semantic graph. Table I gives results in a numerical form for the DUC-01 experiments. Figure 4 and Table II shows the scores for the DUC 02 dataset. For both corpuses the ROUGE scores improves on shortest dependency based graph, until distance 5. During results analysis we have observed that the ROUGE score decreases or becomes approximately constant if we increase distance after 5.

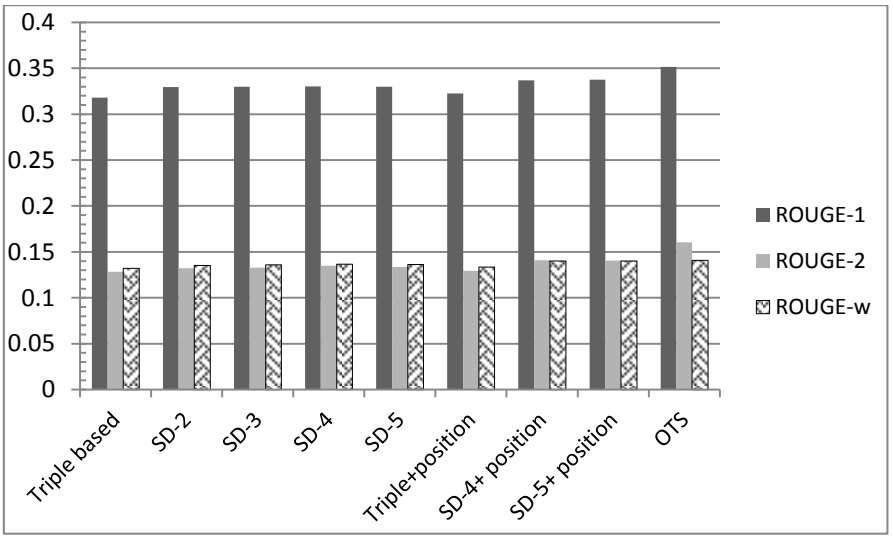

Fig. 3. ROUGE scores obtained for a summarisation test on DUC-01 data

Table 1. ROUGE scores for a summarisation test on DUC-01 data

System	Rouge-1	Rouge-2	Rouge-W
Triplet based	0.31793	0.12829	0.13214
SD-2	0.32964	0.13229	0.1354
SD-3	0.3298	0.13301	0.1359
SD-4	0.33037	0.1351	0.13671
SD-5	0.32974	0.13365	0.13621
Triple + position	0.3224	0.12923	0.13355
SD-4+ position	0.33676	0.14106	0.14017
SD-5+ position	**0.33753**	**0.14049**	**0.14023**
OTS	**0.35134**	**0.16039**	**0.14093**

Including sentence position as a feature, improves the summarisation results on both triple based graph and shortest distance dependency path based semantic graph. Also in this case, ROUGE scores for summarisation run on shortest distance dependency path based semantic graph are higher than for triple based semantic graphs. This also indicates that we can include more features to improve the results further. Overall results indicate that shortest distance based semantic graphs performs better in ranking the sentences and are comparable to benchmark system OTS.

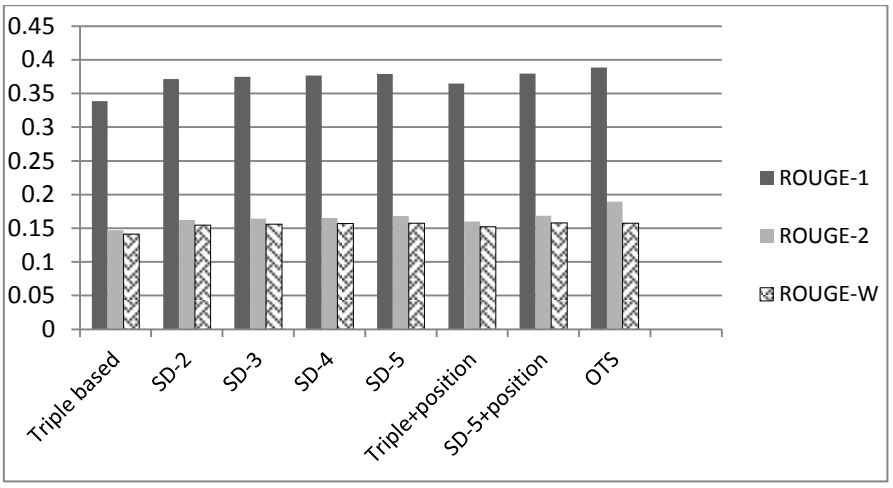

Fig. 4. ROUGE scores obtained for summarisation test on DUC-02 data

Table 2. ROUGE scores for summarisation test on DUC-02 data

System	Rouge-1	Rouge-2	Rouge-W
Triplet based	0.33864	0.14714	0.14143
SD-2	0.37154	0.16221	0.15465
SD-3	0.37494	0.16409	0.1563
SD-4	0.37666	0.16498	0.15694
SD-5	0.37919	0.168	0.15778
Triple + position	0.36465	0.16016	0.15231
SD-4+ position	0.37666	0.16498	0.15694
SD-5+position	**0.37937**	**0.16846**	**0.15793**
OTS	**0. 0.38864**	**0.18966**	0.15766

7 Conclusion

PageRank based summarisation is a novel approach for both our approaches. Earlier for triple based semantic graph, PageRank node score was considered as a feature for supervised text summarisation. In this paper we have looked at unsupervised single document summarisation. In the evaluation, we have seen that only PageRank based summarisation results do not exceed the benchmark results, but are comparable. Benchmark OTS system utilises a language specific lexicon for identifying synonymous words and cue terms. In future work, we can include a similar lexicon to identify more relation between words to improve the performance. In this paper we have hypothesised that if more dependency relations are considered for semantic graph generation it gives better PageRank scores and thus improves the ranking accuracy for

extraction of summary sentences. Although triple based graphs are more visually understandable they can be enhanced by adding more dependencies. When sentence position was included as an extra feature, it improved the Rouge scores. Also it is noticeable that summarisation results for shortest distance dependency path based semantic graph are similar to results after including the additional feature *sentence position*. This makes this graph equally useful in domains where sentence position does not have an effect on importance.

In future work we will apply semantic similarity and word sense disambiguation to improve the connectivity of the graph and identify more relations between nodes.

References

1. G. Erkan and D. R. Radev, "LexRank: graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, no. 1, pp. 457–479, Jul. 2004.
2. R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," in *Proceedings of Empirical Methods in Natural Language Processing*, 2004.
3. K. Ganesan, C. Zhai, and J. Han, "Opinosis : A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010, no. August, pp. 340–348.
4. J. Leskovec, N. Milic-Frayling, and M. Grobelnik, "Extracting Summary Sentences Based on the Document Semantic Graph," *Microsoft Technical Report TR-2005-07*, 2005.
5. D. Rusu, B. Fortuna, M. Grobelnik, and D. Mladenić, "Semantic Graphs Derived From Triplets With Application in Document Summarization," *Informatica Journal*, 2009.
6. L. Plaza and A. Díaz, "Using Semantic Graphs and Word Sense Disambiguation Techniques to Improve Text Summarization," *Procesamiento del Lenguaje Natural Revista*, vol. 47, pp. 97–105, 2011.
7. G. Tsatsaronis, I. Varlamis, and K. Nørvåg, "SemanticRank: ranking keywords and sentences using semantic graphs," *COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics*, no. August, pp. 1074–1082, 2010.
8. R. C. Bunescu and R. J. Mooney, "A Shortest Path Dependency Kernel for Relation Extraction," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, no. October, pp. 724–731.
9. K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03*, 2003, vol. 1, pp. 173–180.
10. H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky, "Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task,", Jun. 2011.
11. C. Lin and M. Rey, "R OUGE : A Package for Automatic Evaluation of Summaries," in *Proceedings of the ACL-04 Workshop: Text Summarization Branches Out*, 2004, pp. 74–81.
12. V. a. Yatsko and T. N. Vishnyakov, "A method for evaluating modern systems of automatic text summarization," *Automatic Documentation and Mathematical Linguistics*, vol. 41, no. 3, pp. 93–103, Jun. 2007.

Automatic extraction of cause-effect relations in Natural Language Text

Antonio Sorgente, Giuseppe Vettigli, and Francesco Mele

Institute of Cybernetics "Eduardo Caianiello" of the National Research Council
Via Campi Flegrei 34, 80078 Pozzuoli (Naples) Italy
{a.sorgente, g.vettigli, f.mele}@cib.na.cnr.it

Abstract. The discovery of causal relations from text has been studied adopting various approaches based on rules or Machine Learning (ML) techniques. The approach proposed joins both rules and ML methods to combine the advantage of each one. In particular, our approach first identifies a set of plausible cause-effect pairs through a set of logical rules based on dependencies between words then it uses Bayesian inference to reduce the number of pairs produced by ambiguous patterns. The SemEval-2010 task 8 dataset challenge has been used to evaluate our model. The results demonstrate the ability of the rules for the relation extraction and the improvements made by the filtering process.

Keywords: Natural Language Processing, Information Extraction, Relations extraction, Causal relations.

1 Introduction

The extraction of causal relations from English sentences is an important step for the improvement of many Natural Language Processing applications such as question answering [1, 2], document summarization and, in particular, it enables the possibility to reason about the detected events [3, 4]. Besides, many websites[1] specialized in web intelligence provide services for the analysis of huge amounts of texts and in this scenario the extraction of causal information can be used for the creation of new insights and for the support of the predictive analysis.

The automatic extraction of causal relations is also a very difficult task because the English presents some hard problems for the detection of causal relation. Indeed, there are few explicit lexico-syntactic patterns that are in exact correspondence with a causal relation while there is a huge number of cases that can evoke a causal relation not in a uniquely way. For example, the following sentence contains a causal relation where *from* is the pattern which evokes such relation:

*"Pollution **from** cars is causing serious health problems for Americans."*

[1] One of the most prominent examples is http://www.recordedfuture.com/

In this case, the words (*pollution* and *cars*) connected by the cue pattern (*from*) are in a causal relation while in the following sentence the *from* pattern doesn't evoke the same type of relation:

"A man ***from*** *Oxford with leprosy was cured by the water."*

Although most of the existing approaches for discovering causal relations are centered on the extraction of a pair of words or noun phrases that are in a causal relation, they do not discriminate causes and effects.

In this paper we propose an approach based on a set of rules that uses the dependency relations between the words. It is able to extract the set of potential pairs cause-effect from the sentence, then we use a Bayesian approach to discard the incorrect pairs. In particular, we identify words that are in a causal relation within a single sentence where the relation is marked by a specific linguistic unit and the causation is explicitly represented (both arguments of the relations are present in the sentence [5]). In particular, we detect nominal words denoting an occurrence (an event, a state or an activity), or nouns denoting an entity, either as one of its readings (like *breakfast*, which can denote an entity, or like *devastation*, which can denote an event) or metonymies (like *the mall*, which can stand in for shopping).

The rest of this paper is organized as follows. In Section 2 we present a brief review of the previous works about causal relations extraction from text. Section 3 describes the proposed method. Results are presented in Section 4. At the end we offer some discussion and conclusions.

2 Related works

In this section we will briefly introduce some approaches proposed by other authors concerning the automatic extraction of causal knowledge.

In [5] a method, based on Decision Trees, for the detection of marked and explicit causations has been proposed. The authors showed that their method is able to recognize sentences that contain causal relations with a precision of 98% and a recall of 84%. However, this method is not able to detect the causes and the effects.

The task 4 of SemEval-2007 [6] and the task 8 of SemEval-2010 [7] concerned about the classification of pairs of words. In each sentence a specific pair of words is already annotated and the target of the tasks consists in classifying the pairs according to the relation evoked in the sentence. The tasks take in account seven types of relations, one of which is the causal relation. In SemEval-2010, Rink et al. [8] had the best results. They obtained an overall precision of 89% and an overall recall of 89% using a SVM classifier, for the specific class of the causal relations they obtained a precision of 89% and a recall of 89%.

An approach to identify cause and effect in sentence was proposed in [2]. In this work, a semi-automatic method to discover causal relations having the particular pattern *<NP verb NP>* was defined. They reported a precision of 65% on a corpus containing a set of documents related to terrorism.

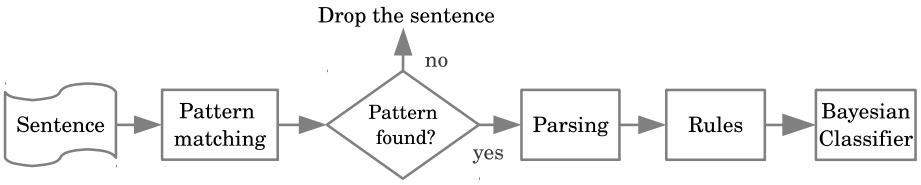

Fig. 1. Steps for the detection of causes and effects.

A system for mining causal relations from Wikipedia is proposed in [9]. The authors used a semi-supervised model in order to select lexico-syntactic patterns represented by the dependency relations between the words able to extract pair of nominals in causal relation. They reported a precision of 76% and a recall of 85%. The patterns discovered by their algorithm are not able to discriminate the causes from the effects.

In order to predict future events from news, in [10] the authors implemented a method for the extraction of causes and effects. In this case, the domain of interest was restricted to the headlines of newspaper articles and a set of hand-crafted rules was used for this task (with a precision of 78%). In [11] regarding a medical abstracts domain, separated measures of precision and recall for causes and effects are reported: a precision of 46% and a recall of 56% for the causes and a precision of 54% and a recall of 64% for the effects. In the last two works mentioned, the approaches proposed are able to discriminate between causes and effects, but they are limited to particular domains.

3 Our approach

In this work, the goal is to extract from a sentence S a set of pairs *cause-effect* $\{(C_1, E_1), (C_2, E_2), \ldots, (C_n, E_n)\}$ where (C_i,E_i) represents the ith cause-effect pair in S. To this end, we propose a method showed in Fig. 1. First, we check if the sentence contains a causal pattern. Then, if a pattern is found the sentence is parsed and a set of rules is applied. A Bayesian classifier is applied to filter out the pairs produced by the rules derived from ambiguous patterns.

3.1 Lexico-syntactic patterns

For the extraction of the pairs, we have defined a set of lexico-syntactic patterns that represent the structure of the causal relations in the sentence. In order to identify the lexico-syntactic patterns, we have inspected the structure of the sentences that contain causal relations in the train dataset provided for the task 8 of SemEval-2010.

The patterns identified are:

– *Simple causative verbs* are single verbs having the meaning of "causal action" (e.g. *generate, trigger, make* and so on).

– *Phrasal verbs* are phrases consisting of a verb followed by a particle (e.g. *result in*).
– *Nouns + preposition* are expressions composed by a noun followed by a preposition (e.g. *cause of*).
– *Passive causative verbs* are verbs in passive voice followed by the preposition *by* (e.g. *caused by*, *triggered by*, and so on).
– *Single prepositions* are prepositions that can be used to link "cause" and "effect" (e.g. *from*, *after*, and so on).

Pattern	Regular expression
Simple causative verbs	(.*) <cause|generate|triggers|...> (.*)
Phrasal verbs / Noun + preposition	(.*) <result|cause|lead|...> <in|of|to> (.*)
Passive causative verbs	(.*) <caused|generated|triggered|...> by (.*)
Single prepositions	(.*) <from|after|...> (.*)

Table 1. List of lexico-syntactic patterns and related regular expression used to detect causal sentence.

For each lexico-syntactic pattern a regular expression (see Table 1) is defined to recognize the sentences that contain such pattern, and a set of rules is defined in order to detect causes and effects.

3.2 Rules

The rules for the detection of causes and effects are based on the relations in the dependency tree of the sentence, in particular on the Stanford dependencies representation [12]. The rules are made analyzing the most frequent relations that involve the words labeled as cause or effect in the dependency tree. For example, in the case of *phrasal verb* we have observed that the cause is linked to the verb while the effect is linked to the preposition or in the case of single preposition both cause and effect are linked to the preposition. The defined rules are introduced as Horn-Clauses. The main rule that allows to detect the cause-effect relation is:

$$cause(S, P, C) \wedge effect(S, P, E) \rightarrow cRel(S, C, E). \quad (1)$$

where $causeS(S, P, C)$ means that the C is a cause in S in accordance to the pattern P while $effectS(S, P, E)$ means that E is the effect in C with respect to P.

Rules for Simple causative verbs For this pattern, generally the cause and effect are respectively the subject (rule 2) and the object (rule 3) of the verb.

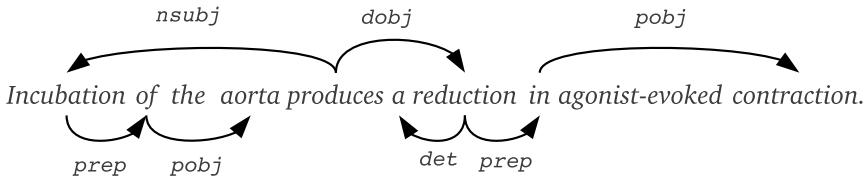

Fig. 2. Dependencies among the words of the sentence *"Incubation of the aorta produces a specific reduction in agonist-evoked contraction"*.

Examples of verbs which evoke causal relation are *cause, create, make, generate, trigger, produce, emit* and so on. We indicate with $verb(S, P, V)$ that the verb V of the sentence S belongs to pattern P of simple causative verb (row 1 in Table 1), while the relation $nsubj(S, V, C)$ is true if C is the subject of V and $dobj(S, V, E)$ is true if E is the direct object of V. The rules defined are:

$$verb(S, P, V) \wedge nsubj(S, V, C) \rightarrow cause(S, P, C), \tag{2}$$

$$verb(S, P, V) \wedge dobj(S, V, E) \rightarrow effect(S, P, E). \tag{3}$$

If we consider, for example, the dependency tree of the sentence *"Incubation of the aorta produces a specific reduction in agonist-evoked contraction"* (showed in Fig. 2), applying the rules 2 and 3, we have that *Incubation* is the cause and *reduction* is the effect.

Rules for Phrasal verbs / Noun + preposition For this pattern, the cause is linked to the verb (or noun) while the effect is linked to the preposition. We indicate with $prep\_verb(S, P, V)$ that the verb or the noun V of the sentence S belongs to the pattern P of phrasal verbs (row 2 in Table 1). While, $prep(S, E, Pr)$ is true when Pr is a propositional modifier of V. The rule defined for the detection of the causes is:

$$prep\_verb(S, P, V) \wedge nsubj(S, V, C) \rightarrow cause(S, C) \tag{4}$$

while, the detection of the effect depends on the preposition. Then, we have defined the following rule:

$$\begin{aligned} &prep\_verb(S, P, V) \wedge preposition\_of(S, V, Pr) \wedge \\ &\wedge prep(S, E, Pr) \rightarrow effect(S, P, E). \end{aligned} \tag{5}$$

A particular case is the causal relation introduced by the expression **"due to"**. For this pattern, with respect to previous rules, the relations are inverted. So, the cause is linked to the preposition and the effect to the verb. The rules are:

$$\begin{aligned} &prep\_verb(S, P, due) \wedge prep(S, due, to) \wedge pobj(S, to, C) \rightarrow cause(S, P, C), \\ &prep\_verb(S, P, due) \wedge nsubj(S, due, E) \rightarrow effect(S, P, E). \end{aligned} \tag{6}$$

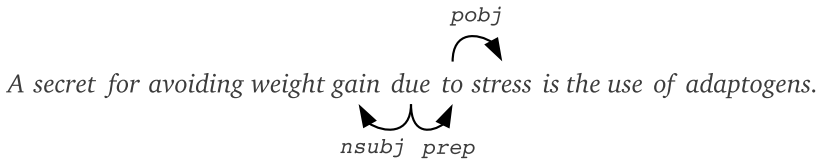

Fig. 3. Dependencies among the words of the sentence *"A secret for avoiding weight gain due to stress is the use of adaptogens"*.

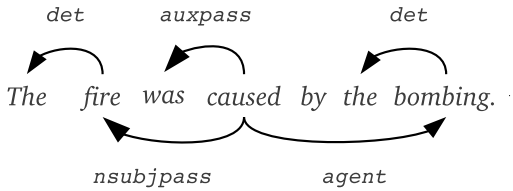

Fig. 4. Dependencies among the words of the sentence *"The fire was caused by the bombing"*.

where $pobj(S, Pr, C)$ is true when C is the object of a preposition Pr.

In this case, applying the rules on the dependency tree of the sentence *"A secret for avoiding weight gain due to stress is the use of adaptogens"* (showed in Fig. 3), we are able to correctly detect *stress* as cause and *gain* as effect.

Rules for Passive causative verbs In this pattern the cause is the word that has an *agent* relation with the verb. In fact, as reported in [12], the *agent* is the complement of a passive verb which is introduced by the preposition *by* and does the action, while the effect is the passive subject of the verb. We indicate with $passive(S, P, V)$ that the verb V of the sentence S belongs to the pattern P of passive causative verbs (row 3 in Table 1).

The rules defined are:

$$\begin{aligned} &passive(S, P, V) \wedge agent(S, V, C) \rightarrow cause(S, P, C), \\ &passive(S, P, V) \wedge nsubjpass(S, V, E) \rightarrow effect(S, P, E) \end{aligned} \tag{7}$$

where $agent(S, V, C)$ is true when C is the complement of a passive verb V and $nsubjpass(S, V, E)$ is true when E is the subject of V.

In this case, applying the rules on the dependency tree of the sentence *"The fire was caused by the bombing."* (showed in Fig. 4), we are able to correctly detect *bombing* as cause and *fire* as effect.

Rules for Single prepositions For this pattern, the cause and effect are linked, in the dependence tree, to the preposition that evokes the causal relation.

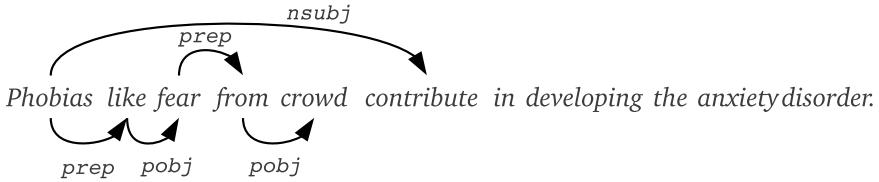

Fig. 5. Dependencies among the words of the sentence *"Phobias like fear from crowd contribute in developing the anxiety disorder"*.

We use $preposition(S, P, Pr)$ to indicate that the preposition Pr of the sentence S belongs to the pattern P of single preposition (row 4 in Table 1). In this case, the rules defined are:

$$\begin{aligned} &preposition(S, P, Pr) \wedge pobj(S, Pr, C) \rightarrow cause(S, P, C), \\ &preposition(S, P, Pr) \wedge prep(S, E, Pr) \rightarrow effect(S, P, E). \end{aligned} \tag{8}$$

In many cases the effects have a direct link with the verb that precedes *from* or *after*. In order to handle those situations we defined the following rule:

$$preposition(S, P, Pr) \wedge prep(S, V, Pr) \wedge nsubj(S, V, C) \rightarrow effect(S, P, C). \tag{9}$$

If we consider, for example, the dependency tree of the sentence *"Phobias like fear from crowd contribute in developing the anxiety disorder"* (showed in Fig. 5), applying the rules 8 and 9, we have that *crow* is the cause and *fear* is the effect.

3.3 Rules for multiple causes and effects

The rules presented above allow to detect a cause and an effect for each pattern that match in a sentence. If there are two or more causes for an effect, we want to detect them all. For example, in the sentence

"*Heat, wind and smoke cause flight delays.*"

the *and* relation indicates that the *delays* (effect) is caused by *Heat*, *wind* and *smoke*, so we have three causes. To deal with these situations we have defined rules that propagate the causal relation along the conjunct dependencies:

$$cause(S, P, C1) \wedge conj(S, C1, C2) \rightarrow cause(S, P, C2). \tag{10}$$

The same rule is defined to propagate through conjunctions the effect:

$$effect(S, P, E1) \wedge conj(S, E1, E2) \rightarrow effect(S, P, E2) \tag{11}$$

where $conj(S, C1, C2)$ is true when $C1$ and $C2$ are connected by a coordinating conjunction (*and*, *or*).

The Fig. 6 shows the dependency tree of sentence *"Heat, wind and smoke cause flight delays"* and we can see that that applying the and rule we detect all causes.

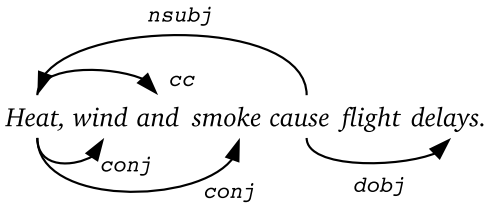

Fig. 6. Dependencies among the words of the sentence *"Heat, wind and smoke cause flight delays"*.

3.4 Pairs filtering

The patterns and the rules defined above, due to their empirical nature, are not able to produce exact results. Hence, there are some pairs (C, E) that are not in causal relation. In order to remove the erroneous pairs detected we used a binary classifier to discriminate *causal* and *non-causal* pairs. This problem is a subtask of the task 8 of the SemEval-2010 where only the causal relation have been considered. To implement the classifier we have chosen to use the Bayesian classification method. Considering the (hypothetical causal) pair $r \equiv cRel(C, E)$, the Bayes' rule becomes:

$$P(c_i|r) = \frac{P(r|c_i)P(c_i)}{P(r)}, \tag{12}$$

with $i = 1, 2$ where c_1 is *causal* and c_2 is *non-causal*. The following features have been associated to the relation r:

- *Lexical features*. The words between C and E.
- *Semantic features*. All the hyponyms and the synonyms of each sense of C and E reported in WordNet [13].
- *Dependency features*. The direct dependencies of C and E in the dependency parse tree.

For each pair r we have extracted a set of features F and for each feature $f \in F$ we have estimated $P(f|c_i)$ by counting the number of causal relations having the feature f, then dividing by the total number of times that f appears. We have used Laplace smoothing applying an additive constant α to allow the assignment of non-zero probabilities for features which do not occur in the train data:

$$P(f|c_i) = \frac{\#(c_i, f) + \alpha}{\#f + \alpha|F|}. \tag{13}$$

Assuming that the features are independent from each other we computed

$$P(r|c_i) = \prod_{f \in F} P(f|c_i). \tag{14}$$

Class	Precision	Recall	F-score
causal	91%	94%	92%
non-causal	98%	97%	98%

Table 2. Results of the classifier for the discrimination of causal pairs on the train set of the SemEval-2010 task 8 using 10-fold cross validation ($\alpha = 1$).

According to the Bayesian classification rule, the relation is classified as *causal* if

$$P(c_1|r) \geq P(c_2|r) \tag{15}$$

and as *non-causal* otherwise.

In order to test the classification framework we used 10-fold cross validation on the train set of the SemEval-2010 dataset. The results are summarized in Table using precision, recall and f-score, they are slightly better to the best ones obtained at the SemEval-2010 challenge, the improvement can be explained by the fact that we consider only the causal relation.

4 Evaluation

We have evaluated our method on a test corpus made extending the annotations of the SemEval-2010 (Task 8) test set. In the original dataset in each sentence only one causal pair has been annotated. We have extended the annotation with the causal pairs not considered by the SemEval annotators. In the cases where an effect is caused by a combination of events or a cause produces a combination of events, pair cause-effect is annotated separately. Our corpus is composed by 600 sentences, 300 of them contain at least a causal relation and the other 300 without causal relations.

The dependency trees have been computed using the Stanford Statistical Parser [14] and the rules for the detection of cause-effect pairs have been implemented in XSB Prolog [15].

The performances have been measured globally and per sentence. The metrics used are *precision*, *recall* and *F-score* in both contexts. Let us define *precision* and *recall* in the global context as

$$P_{global} = \frac{\#\text{correct retrieved pairs}}{\#\text{retrieved pairs}}, \tag{16}$$

$$R_{global} = \frac{\#\text{correct retrieved pairs}}{\#\text{total pairs in } D}, \tag{17}$$

where D is the set of all the sentences in the dataset. The *precision* and *recall* to measure the performances *per sentence* are defined as

$$P_{sentence} = \frac{1}{|M|} \sum_{s \in M} \frac{\#\text{correct retrieved pairs in } s}{\#\text{retrieved pairs in } s}, \tag{18}$$

	Precision	Recall	F-score	α
Global	49%	66%	56%	*no filter*
Per sentence	56%	67%	61%	
Global	55%	65%	59%	0
Per sentence	59%	66%	62%	
Global	**71%**	**58%**	**63%**	0.2
Per sentence	**72%**	**57%**	**64%**	
Global	70%	54%	61%	0.5
Per sentence	70%	53%	60%	
Global	70%	56%	63%	0.7
Per sentence	71%	56%	62%	
Global	71%	54%	62%	1
Per sentence	72%	54%	61%	

Table 3. Results obtained during the tests.

$$R_{sentence} = \frac{1}{|D|} \sum_{s \in D} \frac{\#\text{correct retrieved pairs in } s}{\#\text{total pairs in } s}, \tag{19}$$

where M is the set of the sentences where the rules found at least a causal pair. In both cases the F-score is defined as

$$F = 2\frac{P \cdot R}{P + R}.$$

The *per sentence* metrics measure the ability of the system to extract all the causal pairs contained in a given sentence while the *global* metrics measure the ability of the system to extract all the causal pairs contained in the entire corpus.

The results of the evaluation are summarized in Table 3. We can see that the precision of the rules stand-alone is around 50% and the recall is around 60%. While, the application of the filter, in the best case, increases the precision of 20% but with a slight lowering of the recall. We can also observe that the best performances of the filter are obtained with Laplace smoothing setting $\alpha = 0.2$. For highest values of α we obtained the same precision, but a significant lowering of the recall.

5 Conclusion & Future Work

In this work we have presented a method for the detection and the extraction of cause-effect pairs in English sentences that contain explicit causal relations. In particular, we have used an hybrid approach which combines rules, for the extraction of all possible causes and effects, and a Bayesian classifier to filter the erroneous solutions.

The presented method have been evaluated on an extended version of the dataset used for the task of the SemEval-2010 challenge. The results achieved by our approach are encouraging, especially if we consider that the dataset contains data extracted from various domains.

In future work we will refine the rules presented and experiment other filtering techniques. Also, we will extend this system in order to handle also implicit causal relations.

References

1. Atzeni, P., Basili, R., Hansen, D.H., Missier, P., Paggio, P., Pazienza, M.T., Zanzotto, F.M.: Ontology-based question answering in a federation of university sites: The moses case study. In: NLDB. (2004) 413–420
2. Girju, R., Moldovan, D.: Mining answers for causation questions. In: In AAAI symposium on. (2002)
3. Mele, F., Sorgente, A.: Ontotimefl a formalism for temporal annotation and reasoning for natural language text. In Lai, C., Semeraro, G., Vargiu, E., eds.: New Challenges in Distributed Information Filtering and Retrieval. Volume 439 of Studies in Computational Intelligence. Springer Berlin Heidelberg (2013) 151–170
4. Mele, F., Sorgente, A., Vettigli, G.: Designing and building multimedia cultural stories using concepts of film theories and logic programming. In: AAAI Fall Symposium: Cognitive and Metacognitive Educational Systems. (2010)
5. Blanco, E., Castell, N., Moldovan, D.I.: Causal relation extraction. In: LREC. (2008)
6. Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., Yuret, D.: Classification of semantic relations between nominals. Language Resources and Evaluation **43**(2) (2009) 105–121
7. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Saghdha, D.., Romano, L., Szpakowicz, S.: Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals
8. Rink, B., Harabagiu, S.: Utd: Classifying semantic relations by combining lexical and semantic resources. In: Proceedings of the 5th International Workshop on Semantic Evaluation. SemEval '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 256–259
9. Ittoo, A., Bouma, G.: Extracting explicit and implicit causal relations from sparse, domain-specific texts. In: Proceedings of the 16th international conference on Natural language processing and information systems. NLDB'11, Berlin, Heidelberg, Springer-Verlag (2011) 52–63
10. Radinsky, K., Davidovich, S.: Learning to predict from textual data. J. Artif. Int. Res. **45**(1) (September 2012) 641–684
11. Khoo, C.S.G., Chan, S., Niu, Y.: Extracting causal knowledge from a medical database using graphical patterns. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics. ACL '00, Stroudsburg, PA, USA, Association for Computational Linguistics (2000) 336–343
12. Marneffe, M.C.D., Manning, C.D.: Stanford typed dependencies manual (2008)
13. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
14. catherine De Marneffe, M., Maccartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: In LREC 2006. (2006)

15. Sagonas, K., Swift, T., Warren, D.S.: Xsb as an efficient deductive database engine. In: In Proceedings of the ACM SIGMOD International Conference on the Management of Data, ACM Press (1994) 442–453

A Keyphrase Generation Technique Based upon Keyphrase Extraction and Reasoning on Loosely Structured Ontologies

Dario De Nart, Carlo Tasso

Artificial Intelligence Lab
Department of Mathematics and Computer Science
University of Udine, Italy
{dario.denart,carlo.tasso}@uniud.it

Abstract. Associating meaningful keyphrases to documents and web pages is an activity that can greatly increase the accuracy of Information Retrieval and Personalization systems, but the growing amount of text data available is too large for an extensive manual annotation. On the other hand, automatic keyphrase generation, a complex task involving Natural Language Processing and Knowledge Engineering, can significantly support this activity. Several different strategies have been proposed over the years, but most of them require extensive training data, which are not always available, suffer high ambiguity and differences in writing style, are highly domain-specific, and often rely on a well-structured knowledge that is very hard to acquire and encode. In order to overcome these limitations, we propose in this paper an innovative unsupervised and domain-independent approach that combines keyphrase extraction and keyphrase inference based on loosely structured, collaborative knowledge such as Wikipedia, Wordnik, and Urban Dictionary. Such choice introduces a higher level of abstraction in the generated KPs that allows us to determine if two texts deal with similar topics even if they do not share a word.

1 Introduction

Due to the constant growth of the amount of text data available on the Web and in digital libraries, the demand for automatic summarization and real-time information filtering has rapidly increased. However, such systems need metadata that can precisely and compactly represent the content of the document. As broadly discussed in literature and proven by web usage analysis [16], is particularly convenient for such metadata to come in the form of *KeyPhrases*(KP), since they can be very expressive (much more than single Keywords), pretty much straightforward in their meaning, and have a high cognitive plausibility, because humans tend to think in terms of KPs rather than single Keywords. In the rest of this paper we will refer to *KP generation* as the process of associating a meaningful set of KPs to a given text, regardless to their origin, while we will call *KP extraction* the act of selecting a set of KP from the text and *KP*

inference the act of associating to the text a set of KP that may not be found inside it. KP generation is a trivial and intuitive task for humans, since anyone can tell at least the main topics of a given text, or decide whether it belongs to a certain domain (news item, scientific literature, narrative, etc., ...) or not, but it can be extremely hard for a machine since most of the documents available lack any kind of semantic hint.

Over the years several authors addressed this issue proposing different approaches towards both KP extraction and inference, but, in our opinion, each one of them has severe practical limitations that prevent massive employment of automatic KP generation in *Information Retrieval*, *Social Tagging*, and *Adaptive Personalization*. Such limitations are the need of training data, the impossibility of associating to a given text keyphrases which are not already included in that text, an high domain specificity, and the need of structured, detailed, and expansive domain knowledge coded in the form of a thesaurus or an ontology.

In this paper we propose an unsupervised KP generation method that combines KP Extraction and KP inference based on Ontology Reasoning upon knowledge sources that though not being formal ontologies can be seen as loosely structured ones, in order to associate to any given text a meaningful and detailed set of keyphrases.

The rest of the paper is organized as follows: in Section 2 we briefly introduce some related works; in Section 3 we present our keyphrase extraction technique; in Section 4 we illustrate our keyphrase inference technique; in Section 5 we discuss some experimental results and, finally, in Section 6 we conclude the paper.

2 Related Work

Many works over the past few years have discussed different solutions for the problem of automatically tagging documents and Web pages as well as the possible applications of such technologies in the fields of Personalization and Information Retrieval in order to significantly reduce information overload and increase accuracy. Both keyphrase extraction and inference have been widely discussed in literature. Several different keyphrase extraction techniques have been proposed, which usually are structured into two phases:

- a *candidate phrase identification* phase, in which all the possible phrases are detected in the text;
- a *selection* phase in which only the most significant of the above phrases are chosen as keyphrases.

The wide span of proposed methods can be roughly divided into two distinct categories:

- *Supervised approaches*: the underlying idea of these methods is that KP Extraction can be seen as a *classification* problem and therefore solved with a sufficient amount of training data (manually annotated) and machine learning algorithms [19]. Several authors addressed the problem in this direction

[18] and many systems that implement supervised approaches are available, such as KEA [20], Extractor[2], and LAKE [4]. All the above systems can be extremely effective and, as far as reliable data sets are available, can be flawlessly applied to any given domain [10]. However, requiring training data in order to work properly, implies two major drawbacks: (i) the quality of the extraction process relies on the quality of training data and (ii) a model trained on a specific domain just won't fit another application domain unless is trained again.

– *Unsupervised approaches*: this second class of methods eliminates the need for training data by selecting candidate KP according to some ranking strategy. Most of the proposed systems rely on the identification of *noun phrases* (i.e. phrases made of just nouns) and then proceed with a further selection based on heuristics such as frequency of the phrase [1] or upon phrase clustering [2]. A third approach proposed by [12] and [9], exploits a graph-based ranking model algorithm, bearing much similarity to the notorious Page Rank algorithm, in order to select significant KPs and identify related terms that can be summarized by a single phrase. All the above techniques share the same advantage over the supervised strategies, that is being truly domain independent, since they rely on general principles and heuristics and therefore there is no need for training data. However, such generalist approaches may not always lead to excellent results, especially when dealing with peculiar documents whose structure does not satisfy the assumptions that drive the KP extraction process.

Hybrid approaches have been proposed as well, incorporating semi-supervised domain knowledge in an otherwise unsupervised extraction strategy [15], but still remain highly domain-specific. Keyphrase extraction, however, is severely limited by the fact it can ultimately return only words contained in the input document, which are highly prone to ambiguity and subject to the nuances of different writing styles (e.g: an author can write "mining frequent patterns" where another one would write "frequent pattern mining"). Keyphrase inference can overcome these limitations and has been widely explored in literature as well, spanning from systems that simply combine words appearing in the text in order to construct rather than extract phrases [3] to systems that assign Keyphrases that may built with terms that never appear in the document. In the latter case, KPs come from a controlled dictionary, possibly an ontology; in such case, a classifier is trained in order to find which entries of the exploited dictionary may fit the text [6]. If the dictionary of possible KPs is an ontology, its structure can be exploited in order to provide additional evidence for inference [13] and, by means of ontological reasoning, evaluate relatedness between terms [11]. In [14] a KP inference technique is discussed, which is based on a very specific domain OWL ontology and which combines both KP Extraction and inference, in the context of a vast framework for personalized document annotation. KP inference based on dictionaries, however, is strongly limited by the size, the domain coverage, and the specificity level of the considered dictionary.

3 System Overview

In order to test our approach and to support our claims we developed a new version of the system presented in [14] which introduces an original innovation, i.e. the exploitation of a number of generalist online External Knowledge Sources, rather than a formal ontology, in order to improve extraction quality and infer meaningful KPs not included in the input text but preserving domain independence.

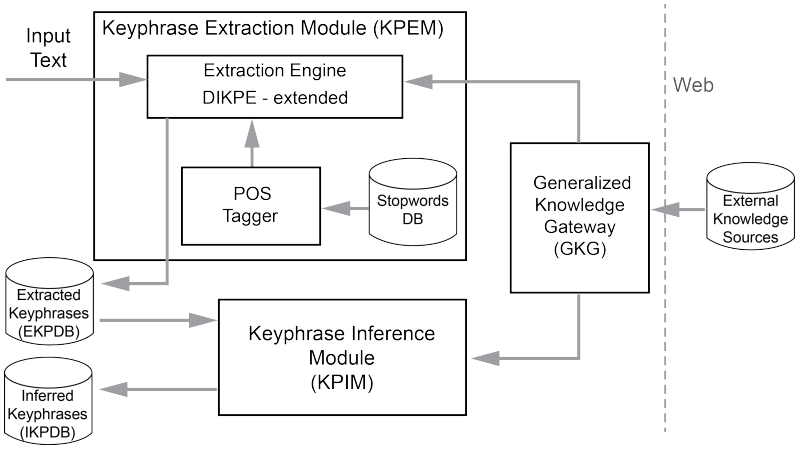

Figure 1. Architecture of the System

In Figure 1 the overall organization of the proposed system is presented. It is constituted by the following main components:

- A *KP Extraction Module* (*KPEM*), devoted to analyse the text end extract from it meaningful KPs. It is supported by some linguistic resources, such as a *POS tagger* (for the English Language) and a *Stopwords Database* and it accesses online some *External Knowledge Sources* (*EKS*) mainly exploited in order to provide support to the candidate KPs identified in the text (as explained in the following section). The KPEM receives in input an unstructured text and it produces in output a ranked list of KPs, which is stored in an *Extracted Keyphrases Data Base*(*EKPDB*).
- A *KP Inference Module* (*KPIM*), which works on the KP list produced by the KPEM and it is devoted to infer new KPs, (possibly) not already included in the input text. It relies on some ontological reasoning based on the access to the External Knowledge Sources, exploited in order to identify concepts which are related to the concepts referred to by the KPs previously extracted by the KPEM. Inferred KPs are stored in the *Inferred KP Data Base* (*IKPDB*).

The access to the online External Knowledge Sources is provided by a Generalized Knowledge Gateway (GKG). Both the EKPDB and the IKPDB can be accessed through Web Services by external applications, providing in such a way and advanced KP Generation service to interested Web users, which can exploit such capability in other target applications.

4 Phrase Extraction

KPEM is an enhanced version of *DIKPE*, the unsupervised, domain independent KP extraction approach described in [14] and [8]. In a nutshell, DIKPE generates a large set of candidate KPs; the exploited approach then merges different types of knowledge in order to identify meaningful concepts in a text, also trying to model a human-like KP assignment process. In particular we use:*Linguistic Knowledge* (POS tagging, sentence structure, punctuation); *Statistical Knowledge* (frequency, tf/idf,...); knowledge about the *structure* of a document (position of the candidate KP in the text, title, subtitles, ...); *Meta-knowledge* provided by the author (html tags,...); knowledge coming from *online external knowledge sources*, useful for validating candidate keyphrases which have been socially recognized, for example, in collaborative wikis (e.g. Wikipedia, Wordnik, and other online resources).
By means of the above knowledge sources, each candidate phrase, is characterized by a set of features, such as, for example:

- *Frequency*: the frequency of the phrase in the text;
- *Phrase Depth*: at which point of the text the phrase occurs for the first time, the sooner it appears, the higher the value;
- *Phrase Last Occurrence*: at which point of the text the phrase occurs for the last time, the later it appears, the higher the value;
- *Life Span*: the fraction of text between the first and the last occurrence of the phrase;
- *POS value*: a parameter taking into account the grammatical composition of the phrase, excluding some patterns and assigning higher priority to other patterns (typically, for example but not exclusively, it can be relevant to consider the number of nouns in the phrase over the number of words in the phrase).
- *WikiFlag*: a parameter taking into account the fact that the phrase is or is not an entry of collaborative external knowledge sources (EKS).

A weighted mean of the above features, called *Keyphraseness* is then computed and the KPs are sorted in descending keyphraseness order. The weight of each feature can be tuned in order to fit particular kinds of text, but, usually, a generalist preset can be used with good results. The topmost n KPs are finally suggested.
In this work, we extended the DIKPE system with the GKG to access EKS, allowing access to multiple knowledge sources at the same time. We also added

a more general version of the WikiFlag feature.This feature is computed as follows: if the phrase matches an entry in at least one of the considered knowledge sources, the its value is set to 1, otherwise the phrase is split into single terms and the WikiFlag value is the percentage corresponding to the number of terms that have a match in at least one of the considered knowledge sources. By doing so, a KP that does not match as phrase, but is constituted by terms that match as single words, still gets a high score, but lower than a KP that features a perfect match. The WikiFlag feature is processed as all the other features, concurring to the computation of the keyphraseness and, therefore, influencing the ranking of the extracted KPs. The rationale of this choice is that a KP is important insofar it represents a meaningful concept or entity, rather than a random combination of words, and matching a whole phrase against collaborative human-made knowledge sources (as the EKS are) guarantees that it makes better sense, providing a strong form of human/social validation. This also reduces the tendency of the system to return typos, document parsing errors, and other meaningless strings as false positives.

Another improvement over the original DIKPE approach is represented by the fact that, instead of suggesting the top n KPs extracted, the new system evaluates the decreasing trend of Keyphraseness among ordered KPs, it detects the first significant downfall in the keyphraseness value, and it suggests all the KPs occurring before that (dynamic) threshold. By doing so, the system suggests a variable number of high-scored KPs, while the previous version suggests a fixed number of KPs, that could have been either too small or too large for the given text.

5 Phrase inference

The KP Inference Module (KPIM), as well as the knowledge-based WikiFlag feature described in the previous section, rely on a set of external knowledge sources that are accessed via web. We assume that (i) there is a way to match extracted KPs with entities described in EKSs (e.g.: querying the exploited service using the KP as search key) and (ii) each one of the EKSs considered is organized according to some kind of hierarchy, as shown in (Figure 2), even if very weak and loosely structured, in which is possible to associate to any entity a set of parent entities and another set made of related entities. Such sets may be void, since we do not assume each entity being linked to at least another one, nor the existence of a root entity that is ancestor to all the other entities in the ontology.

Even if such structure is loose, assuming its existence is not trivial at all, but an increasing number of collaborative resources allow users to classify and link together knowledge items, generating a pseudo-ontology. Clear examples of this tendency are Wikipedia, where almost any article contains links to other articles and many articles are grouped into *categories*, and Wordnik, an online collaborative dictionary where any word has sets of hypernyms, synonyms, hyponyms

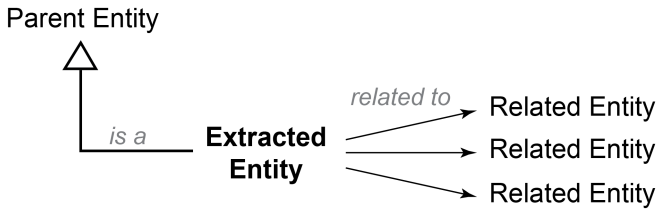

Figure 2. Example of the assumed Knowledge Source structure.

and related words associated. Recently also several entertainment sites, like Urban Dictionary, have begun to provide these possibilities, making them eligible knowledge sources for our approach. Knowledge sources may be either generalist (like Wikipedia), or specific (like the many domain-specific wikis hosted on *wikia.com*) and several different EKS can be exploited at the same time in order to provide better results.

In the case of Wikipedia, parent entities are given by the *categories*, that are thematic groups of articles (i.e.: "Software Engineering" belongs to the "Engineering Disciplines" category). An entry may belong to several categories, for example the entry on "The Who" belongs to the "musical quartets" category as well as to the "English hard rock musical groups" one and the "Musical groups established in 1964" one. Related entities, instead, can be deduced by links contained in the entry associated to the given entity: such links can be very numerous and heterogeneous, but the most closely related ones are often grouped into one or more *templates*, that are the thematic collections of internal Wikipedia links usually displayed on the bottom of the page, as shown in Figure 3. For instance, in a page dedicated to a film director, it is very likely to find a template containing links to the all movies he directed or the actors he worked with.

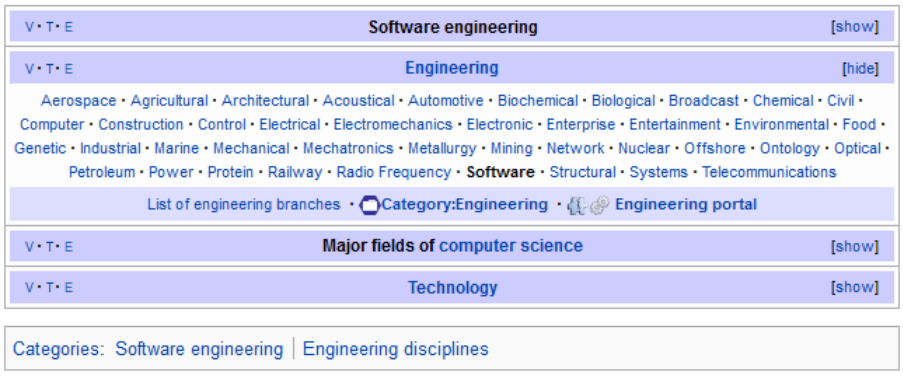

Figure 3. The lowest section of a Wikipedia page, containing templates (the "Engineering" template has been expanded) and categories (bottom line).

Wordnik, instead, provides hierarchical information explicitly by associating to any entity lists of hypernyms (parent entities) and synonyms (related entities).

The inference algorithm considers the topmost half of the extracted KPs, that typically is still a significantly larger set than the one suggested, and, for each KP that can be associated to an entity, retrieves from each EKS a set of parent entities and a set of related entities. If a KP corresponds to more than one entity on one or more EKSs, all of the retrieved entities are taken into account. The sets associated to single KPs are then merged into a table of related entities and a table of parent entities for the whole text. Each retrieved entity is scored accordingly to the sum of the Keyphraseness value of the KPs from which it has been derived and then it is sorted by descending score. The top entries of such tables are suggested as meaningful KPs for the input document.

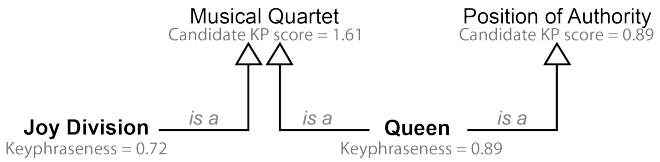

Figure 4. inference and scoring of parent entities.

By doing so, we select only entities which are related or parent to a significant number of hi-scored KPs, addressing the problem of polysemy among the extracted KP. For instance, suppose we extracted "Queen" and "Joy Division" from the same text (Figure 4): they both are polysemic phrases since the first may refer to the English band as well as to a regent and the latter to the English band or to Nazi concentration camps. However, since they appear together, and they are both part of the "musical quartets" category in Wikipedia, we it can be deduced that the text is about music rather than politics or World War II.

6 Evaluation

Formative tests were performed in order to test the accuracy of the inferred KPs and their ability to add meaningful information to the set of extracted KPs, regardless of the domain covered by the input text. Three data sets, dealing with different topics, were processed, article by article, with the same feature weights and exploiting Wikipedia and Wordnik as External Knowledge Source. For each article a list of extracted KPs and one of inferred KPs were generated, then the occurrences of each KP were counted, in order to evaluate which portion of the data set is covered by each KP. We call *set coverage* the fraction of the data set labelled with a single KP. Since the topics covered in the texts included in each data set are known a-priori, we expect the system to generate KPs that associate the majority of the texts in the data set to their specific domain topic.

Extracted Keyphrase	Set coverage	Inferred Keyphrase	Set Coverage
program	0,13	Mathematics	0,47
use	0,12	Programming language	0,26
function	0,12	move	0,25
type	0,10	Computer science	0,22
programming language	0,10	Set (mathematics)	0,17
programming	0,08	Data types	0,15
functions	0,07	Aristotle	0,16
class	0,07	Function (mathematics)	0,14
code	0,06	C (programming language)	0,14
COBOL	0,06	Botanical nomenclature	0,12
chapter	0,05	C++	0,11
variables	0,05	Information	0,08
number	0,05	Java (programming language)	0,08

Table 1. The most frequently extracted and inferred KPs from the "programming tutorials" data set.

The first data set contained 113 programming tutorials, spanning from brief introductions published on blogs and forums to extensive articles taken from books and journals, covering both practical and theoretical aspects of programming. A total of 776 KPs were extracted and 297 were inferred. In Table 1 are reported the most frequently extracted and inferred KPs. As expected, extracted KPs are highly specific and tend to characterize a few documents in the set (the most frequent KP covers just the 13% of the data set), while inferred ones provide an higher level of abstraction, resulting in an higher coverage over the considered data set. However some Inferred KPs are not accurate, such as " Botanical nomenclature " that clearly derive from the presence of terms such as "tree", "branch", "leaf", and "forest" that are frequently used in Computer Science, and "Aristotele" which comes from the frequent references to Logic, which Wikipedia frequently associates with the Greek philosopher.
The second data set contained 159 car reviews taken from American and British magazines written by professional journalists. Unlike the previous data set, in which all the texts share a very specific language and provide technical information, in this set different writing stiles and different kinds of target audiences are present. Some of the reviews are very specific, focusing on technical details, while others are more aimed at entertaining rather than informing. Most of the considered texts, however, stand at some point between these two ends, providing a good deal of technical information together with an accessible and entertaining style.

In Table 2 the most frequently extracted and inferred KPs are reported. While extracted KPs clearly identify the automotive domain, inferred ones don't, with only the 44% of the considered texts being covered by the "Automobile" KP and the 64% being labelled with "English-language films". However this is mostly due to the fact that several reviews tend to stress a car's presence in popular

Extracted Keyphrase	Set coverage	Inferred Keyphrase	Set Coverage
car	0,16	United States	0,66
sports car	0,08	English-language films	0,64
SUV	0,06	Automobile	0,44
fuel economy	0,05	United Kingdom	0,33
ride	0,05	American films	0,16
looks	0,04	Internet Movie Database	0,16
Lotus	0,04	Japan	0,14
GT	0,04	2000s automobiles	0,11
top speed	0,04	Physical quantities	0,09
gas mileage	0,04	2010s automobiles	0,09
look	0,04	Germany	0,09
hot hatch	0,03	Sports cars	0,08

Table 2. The most frequently extracted and inferred KPs from the "car reviews" data set.

movies (eg: Aston Martin in the 007 franchise or any given Japanese car in the Fast and Furious franchise) and only 18 out of 327 (5.5%) different inferred KPs deal with cinema and television. KP such as "Unites States" and "United Kingdom" are also frequently inferred due to the fact that the reviewed cars are mostly designed for USA and UK markets, have been tested in such countries, and several manufacturers are based in those countries. As a side note, 98% of the considered text are correctly associated with the manufacturer of the reviewed car. The third data set contained reviews of 211 heavy metal albums published in 2013. Reviews were written by various authors, both professionals and non-professionals, and combine a wide spectrum of writing styles, from utterly specific, almost scientific, to highly sarcastic, with many puns and popular culture references.

Extracted Keyphrase	Set coverage	Inferred Keyphrase	Set Coverage
metal	0,23	Music genre	1
album	0,21	Record label	0,97
death metal	0,17	Record producer	0,54
black metal	0,17	United States	0,48
band	0,16	Studio album	0,16
bands	0,08	United Kingdom	0,11
death	0,08	Bass guitar	0,09
old school	0,07	Single (music)	0,08
sound	0,06	Internet Movie Database	0,07
albums	0,05	Heavy metal music	0,07
power metal	0,05	Allmusic	0,06

Table 3. The most frequently extracted and inferred KPs from the "album reviews" data set.

In Table 3 are reported the most frequently extracted and inferred KPs. All the documents in the set were associated with the Inferred KP "Music Genre" and the 97% of them with "Record Label", which clearly associates the texts with the music domain. Evaluation and development, however, are still ongoing and new knowledge sources, such as domain-specific wikis and Urban Dictionary, are being considered.

7 Conclusions

In this paper we proposed a truly domain independent approach to both KP extraction and inference, able to generate significant semantic metadata with different layers of abstraction for any given text without need for training. The KP extraction part of the system provides a very fine granularity, producing KPs that may not be found in a controlled dictionary (such as Wikipedia), but characterize the text. Such KPs are extremely valuable for the purpose of summarization and provide great accuracy when used as search keys. However, they are not widely shared, meaning, from an information retrieval point of view, a very low recall. On the other hand, the KP inference part generates only KPs taken from a controlled dictionary (the union of the considered EKS) that are more likely to be general and, therefore, shared among a significant number of texts.

As shown in the previous section, our approach can annotate a set of documents with good precision, however, a few unrelated KPs may be inferred, mostly due to ambiguities of the text and to the generalist nature of the exploited Knowledge Sources. This unrelated terms, fortunately, tend to appear in a limited number of cases and to be clearly unrelated not only to the majority of the generated KPs, but to also each other. In fact, our next step in this research will be precisely to identify such false positives by means of an estimate of the *Semantic Relatedness*[17], [7] between terms in order to identify, for each generated KP, a list of related concepts and detect concept clusters in the document.

The proposed KP generation technique can be applied both in the Information Retrieval domain and in the Adaptive Personalization one. The previous version of the DIKPE system has already been integrated with good results in RES [5], a personalized content-based recommender system for scientific papers that suggests papers accordingly to their similarity with one or more documents marked as interesting by the user, and in the PIRATES framework [14] for tag recommendation and automatic document annotation. We expect this extended version of the system to provide an even more accurate and complete KP generation and, therefore, to improve the performance of these existing systems, in this way supporting the creation of new Semantic Web Intelligence tools.

References

1. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases. In: Advances in Artificial Intelligence, pp. 40–52. Springer (2000)

2. Bracewell, D.B., Ren, F., Kuriowa, S.: Multilingual single document keyword extraction for information retrieval. In: Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on. pp. 517–522. IEEE (2005)
3. Danilevsky, M., Wang, C., Desai, N., Guo, J., Han, J.: Kert: Automatic extraction and ranking of topical keyphrases from content-representative document titles. arXiv preprint arXiv:1306.0271 (2013)
4. DAvanzo, E., Magnini, B., Vallin, A.: Keyphrase extraction for summarization purposes: The lake system at duc-2004. In: Proceedings of the 2004 document understanding conference (2004)
5. De Nart, D., Ferrara, F., Tasso, C.: Personalized access to scientific publications: from recommendation to explanation. In: User Modeling, Adaptation, and Personalization, pp. 296–301. Springer (2013)
6. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: Proceedings of the seventh international conference on Information and knowledge management. pp. 148–155. ACM (1998)
7. Ferrara, F., Tasso, C.: Integrating semantic relatedness in a collaborative filtering system. In: Mensch & Computer Workshopband. pp. 75–82 (2012)
8. Ferrara, F., Tasso, C.: Extracting keyphrases from web pages. In: Digital Libraries and Archives, pp. 93–104. Springer (2013)
9. Litvak, M., Last, M.: Graph-based keyword extraction for single-document summarization. In: Proceedings of the workshop on multi-source multilingual information extraction and summarization. pp. 17–24. Association for Computational Linguistics (2008)
10. Marujo, L., Gershman, A., Carbonell, J., Frederking, R., Neto, J.P.: Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. arXiv preprint arXiv:1306.4886 (2013)
11. Medelyan, O., Witten, I.H.: Thesaurus based automatic keyphrase indexing. In: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries. pp. 296–297. ACM (2006)
12. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Proceedings of EMNLP. vol. 4. Barcelona, Spain (2004)
13. Pouliquen, B., Steinberger, R., Ignat, C.: Automatic annotation of multilingual text collections with a conceptual thesaurus. arXiv preprint cs/0609059 (2006)
14. Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., Tasso, C.: Automatic keyphrase extraction and ontology mining for content-based tag recommendation. International Journal of Intelligent Systems 25(12), 1158–1186 (2010)
15. Sarkar, K.: A hybrid approach to extract keyphrases from medical documents. arXiv preprint arXiv:1303.1441 (2013)
16. Silverstein, C., Marais, H., Henzinger, M., Moricz, M.: Analysis of a very large web search engine query log. In: ACm SIGIR Forum. vol. 33, pp. 6–12. ACM (1999)
17. Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. In: AAAI. vol. 6, pp. 1419–1424 (2006)
18. Turney, P.D.: Learning to extract keyphrases from text. national research council. Institute for Information Technology, Technical Report ERB-1057 (1999)
19. Turney, P.D.: Learning algorithms for keyphrase extraction. Information Retrieval 2(4), 303–336 (2000)
20. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: Kea: Practical automatic keyphrase extraction. In: Proceedings of the fourth ACM conference on Digital libraries. pp. 254–255. ACM (1999)

Enabling Advanced Business Intelligence in Divino

Danilo Croce, Francesco Garzoli, Marco Montesi,
Diego De Cao, and Roberto Basili

Department of Enterprise Engineering
University of Roma, Tor Vergata
00133 Roma, Italy
{croce,garzoli,montesi,decao,basili}@info.uniroma2.it

Abstract. This paper presents the system targeted in the Divino project, funded under the Industria 2015 framework of the Italian Ministry of Industry. The resulting platform embodies an innovative portal technology where Social Web functionalities, User Profiling and Aspect-based Opinion Mining are integrated through Liferay, a well known Enterprise Portal Technology. The proposed approach allows analysts to bootstrap an opinion-mining system by interacting with data-driven functions based on effective Online Machine Learning paradigms. The evaluation of the proposed methods is carried out in the targeted domain, i.e. the marketing of national wine products, one of the major focus area of the Made in Italy track of Industria 2015.

1 Introduction

In Business Intelligence, analysts have nowadays access to a variety of public forums where opinions and sentiments about companies, products and strategies are expressed in unstructured form. Opinion Mining (OM) [11] tackles different problems that arise in this scenario, such as determining if a segment of text (sentence, paragraph or section) is *opinionated*, identifying the *opinion-holder* (the person or organization who expresses the opinion) or determining the *polarity* (i.e. how positive or negative each opinion is). For business intelligence, it is also useful to classify each opinion according to the *aspect* of the analyzed product, such the flavor or taste of a wine.

This paper discusses the system targeted in the Divino project, funded under the Industria 2015 framework of the Italian Ministry of Industry. The resulting platform embodies an innovative portal technology where Social Web functionalities, User profiling and Aspect-based Opinion Mining (OM) are integrated. On the one hand, users can visit a portal enjoying a community interested in the eno-gastronomic domain of wine. When logged, the so-called Divino User has a deeper interaction with the portal, leaving message in the forum, designing a personalized blog or buying items in a specialized e-shop; every registered user becomes part of a Social Network, determining friendship-based links with

other users. On the other hand, an Opinion Mining workflow has been implemented to capture people opinions and preferences expressed within the portal. These are enriched by crawling and processing specialized sites and blogs from the Web. Opinions are stored in a semi-structured form and meaningfully summarized to be consumed by Market Analysts. Based on the Enterprise Portal Technology known as Liferay, the system results in a Web Portal where different users can enjoy and interact, always providing valuable information for Business Intelligence processes.

The proposed OM workflow is quite general and it can be used to bootstrap and adapt an OM system to a target domain. This can be achieved by applying online Learning Algorithms [3], training classifiers that recognize topics, aspects and opinions in texts, comments and blogs. The online learning paradigm is appealing as it allows an interaction between the system and a Market Analyst, who can incrementally refine the domain by validating classifiers predictions. The applicability of the proposed approach is then evaluated in the targeted domain of the national and international marketing of wine products, one of the major focus area of the Made in Italy track of Industria 2015. In the rest of the paper, Section 2 discusses the OM process in Divino. Section 3 provides a description of the resulting portal. Section 4 provides the experimental evaluation and Section 5 derives the conclusions.

2 Modeling Opinion Mining in Divino

If we are interested in detecting opinions about wines, all textual units containing information related to the target products must be carefully retrieved. Let us consider the following excerpt related to the wine domain:

> *La gamma aziendale prevede un vino rosso basato su uve ciliegiolo in purezza, il Ciliegiolo Golfo del Tigullio doc, vinificato in acciaio, che dona al vino netti ma delicati sentori di ciliegia, violetta e una sottile vena speziata (pepe) senza mancare di una buona aciditá e tannicitá.*[1]

It contains information about a wine, the "*Ciliegiolo Golfo del Tugullio doc*", i.e. the entity to which the author refers. As we are interested in opinions related to specific aspects of wine, such as *flavor* and *taste*, textual units containing objective expressions can be neglected. Words like "*sentori netti ma delicati*" and "*buona aciditá e tannicitá*" here give a positive connotation to the *Aroma* and *Taste* aspects, respectively. Moreover, even if not made explicit, the underlying domain must be properly addressed as it allows to reject texts related to other products, e.g. cars or mobile phones.

Many approaches have been defined to determine and recognize opinions in texts, as discussed in [8, 11, 14], ranging from different text genres, from newswire

[1] Translation: *The product range contains a red wine derived from Ciliegiolo grapes, that is the Ciliegiolo Golfo del Tigullio doc, vinified in stainless steel, which gives strong but delicate hints of cherry, violet and a slightly spicy note (pepper) without missing a good acidity and tannin levels.*

[17] to social media, such as Twitter [10]. These studies let to the development of several corpora with detailed opinion and sentiment annotations, e.g., the MPQA corpus [16] of newswire text. These corpora have proved very valuable as resources for learning about the language of sentiment in general. As discussed in the following section, in Divino we applied empirical methods in order to automatically train classifiers able to associate sentences to specific classes useful to characterize the writer opinion. More formally, our ultimate aim is therefore to extrapolate structured information such as the n-tuple $\langle u, t, h, r, a, b\rangle$ where:

- u is the **Textual Unit**, e.g. a sentence or paragraph expressing an opinion;
- t is the **Topic** related to u, e.g. the `WineryProduct`, that represents the opinion domain;
- h is the **Opinion Holder**, the person or organization expressing the opinion (here the blog author);
- r is the **Opinion Target**, that is the entity subjectively valued (e.g. *Ciliegiolo Golfo del Tugullio doc.*);
- a is the **Aspect** for r in the domain t (e.g. *flavor* or *taste*);
- b is the **Polarity**, associated with a target r and its specific aspect a, e.g. Positive, Negative or Neutral.

In the next section, data-driven learning algorithms to associate each u to the proper n-tuple will be discussed.

2.1 The Opinion Mining Workflow

Behind the Divino portal, an OM workflow has been developed to structure opinions, as discussed above. We defined a specific ontology providing a meta-model from which domain-specific OM workflows are derived, not shown here for space reasons. In the Divino project, the workflow shown in Figure 1 has been implemented.

In the **Data Gathering** phase, a dedicated *Web Crawler* downloads documents from wine specialized sites, blogs and forums. *Chaos* [1], the *Natural Language Processing* (*NLP*) processor made available at the University of Tor Vergata, analyzes such documents to extract morpho-syntactic and semantic information required by the workflow.

In the **Information Extraction** phase the *Target Extractor* allows to identify sentences mentioning one or more target products. In the domain addressed by Divino, examples of target can be wines, such as *Barolo* or *Taurasi*, or Varietal, such as *Syrah* or *Merlot*. This module is based on the Name Entity Recognizer and Classifier (NERC) made available by Chaos. The *Target Propagator* finds sentences referring to targets, even if they are not explicitly mentioned.

The core **Sentiment Analysis** functionalities determine opinions and are realized as a sequence of classification steps. Among all existing Machine Learning paradigms, we investigated the class of Online Learning Algorithms. The goal, as in a traditional fashion, is to predict classes for instances. In addition, soon after the prediction is made, it can then be used to refine the prediction hypothesis used by the algorithm. In a traditional setting, the training phase would

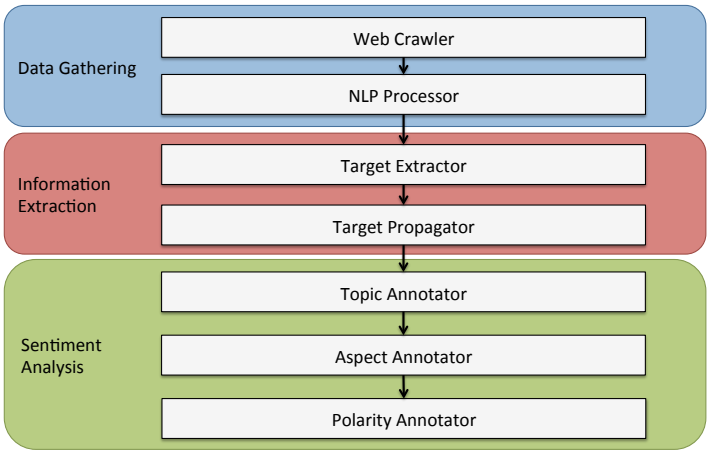

Fig. 1: The OM workflow

have started *ex-novo*, re-considering all training examples. Such online schemas allow implementing mechanisms for relevance feedback: it incrementally refines the domain classifiers and adapts the resulting analysis to the target domain.

In particular, the Passive Aggressive (PA) learning algorithm [3] is one of the most popular online approaches and it is generally referred as a state-of-art online method. Its core idea is quite simple: when an example is misclassified, the algorithm updates the model with the hypothesis that is more similar to the current one. Formally, let $(\mathbf{x}_t, y_t)$ be the t-th example where $\mathbf{x}_t \in \mathbb{R}^d$ is a feature vector that represents a document or sentence in a d-dimensional space, while $y_t \in \{+1, -1\}$ is the corresponding label, e.g. a sentence does/does not belong to a topic or polarity class. Let $\mathbf{w}_t \in \mathbb{R}^d$ be the current classification hypothesis. The PA classification function is $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. After receiving $\mathbf{x}_t$, the new classification function $\mathbf{w}_{t+1}$ becomes the one that minimizes the objective function $Q(\mathbf{w}) = \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C \cdot l(\mathbf{w}; (\mathbf{x}_t, y_t))$. The first term $\|\mathbf{w} - \mathbf{w}_t\|$ is a measure of how much the new hypothesis differs from the old one while the second term $l(\mathbf{w}, (\mathbf{x}_t, y_t))$ is a proper loss function assigning a penalty cost to an incorrect classification. C is the aggressiveness parameter that balances the two competing terms[2]. Minimizing $Q(\mathbf{w})$ corresponds to solving a constrained optimization problem, whose solution let to update the classifier according to the following schema: $\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha_t \mathbf{x}_t, \;\; \alpha_t = y_t \cdot \min\left\{C, \frac{H(\mathbf{w}_t; (\mathbf{x}_t, y_t))}{\|\mathbf{x}_t\|^2}\right\}$.

If $\mathbf{x}_t$ is correctly classified, the model does not change, while, after a wrong prediction, the new classification function $\mathbf{w}_{t+1}$ becomes a linear combination between the old one $\mathbf{w}_t$ and the feature vector $\mathbf{x}_t$. A kernelized version of the PA algorithm is easy to obtain and gives the possibility to exploit rich data representations, as discussed in [3, 15].

[2] In this work we will consider the hinge loss $H(\mathbf{w}; (\mathbf{x}_t, y_t)) = max(0, 1 - y_t \mathbf{w}^T \mathbf{x}_t)$

Fig. 2: Forum and news portlet for non registered users.

In the resulting workflow, given a new document, the *Topic Annotator* retrieves paragraphs related to all topics t that are compatible with the domain, e.g. `WineryProducts` or `Varietals`. Each paragraph is associated by a PA classifier to each target topic t. In order to model an open-world scenario, where not all topics are already known, the `OtherTopic` class is introduced: each paragraph classified as `OtherTopic` is not considered in the remaining processing chain by the other annotators. The *Aspect Annotator* classifies all sentences from the remaining paragraphs with respect to the active aspects a of a given topic t. Even at this level, the open-world assumption is valid, so the `OtherAspect` class is introduced. Finally, for each sentence associated to a valid aspect, the corresponding polarity is provided by another PA-based classifier with respect to the POSITIVE, NEGATIVE or NOPOLARITY classes[3]. More details about the modeling of single textual units u are provided in Section 4. At the moment of writing the Opinion Holder h is assumed to be the content creator, e.g. the author of a blog page or comment in a forum.

3 The Divino portal

The Divino portal is designed as a set of interacting services whose overall logic is integrated within the Liferay portal. Liferay[4] is a free and open source enterprise portal written in Java and distributed under the GNU Lesser General Public License and proprietary licenses. It allows to efficiently create a portal for Internet or Intranet use and it is fundamentally constructed of functional

[3] When a sentence is classified as POSITIVE and NEGATIVE at the same time, it is considered as NEUTRAL.

[4] `http://www.liferay.com/`

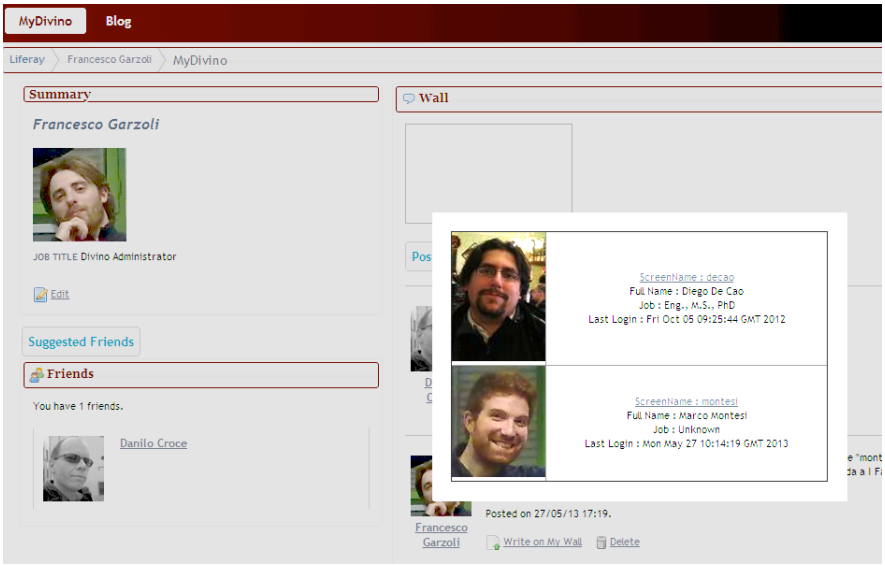

Fig. 3: The MyDivino page: in evidence the Friend Suggestion

units called *portlets*, that represent portal functionalities and produce fragments of markup code that are aggregated into a portal.

Liferay enables the creation of different users and different roles, so that every role associates a user group sharing the same permissions. Permissions are linked to Portal, Portlet and other Liferay entities. In addition to the role of Administrator, the Divino Portal handles four roles, i.e. Guest, Divino User, Annotator and Market Analyst, each enabled to access to the following functionalities.

Enjoying the Divino Portal as a Registered User. A user can visit the Divino Portal without being registered. As a *Guest*, he can view a limited set of pages providing not tailored information as well as the *Divino Forum* and e-commerce services, i.e. the *Divino Shop*, as shown in Fig. 2. A log-in step is required in order to post any message or buy items. Moreover, a *Divino Search* portlet allows to retrieve all web pages downloaded during the Data Gathering phase, described in Section 2. When logged-in at the Divino Portal, the user assumes the role of *Divino User*. He can now participate to the social activities made available in the portal within the forum and e-commerce portlets. As shown in the background of Figure 3, each user is associated to a personal *MyDivino* page where a blog can be easily populated with comments. In line with popular Social Networks, a friendship schema is applied to allow a restricted number of friends to read the personal blog. Each user can retrieve other users and ask their friendship. Every Divino User owns a profile that keep all the information about his search queries, preferences and purchased items. Such interactions with the system, as well as other information provided through a questionnaire suggested in the registration phase, are crucial for many portal functionalities. They enable

Fig. 4: Annotation interface

the design of different User Recommending and Information Filtering schemas, as discussed in [12]. At the moment of writing, a first recommending schema is used to suggest friends. All information gathered during registration provide a set of preferences P_{u_i} describing each Divino User d_i. For example, one can prefer red wines instead of white wines or wines from specific regions. A first recommending function has been implemented by estimating the similarity among user pairs d_i and d_j in terms of the Jaccard Similarity score between the sets of related preferences: $J(d_i, d_j) = \frac{|P_{d_i} \cap P_{d_j}|}{|P_{d_i} \cup P_{d_j}|}$. The score is 1 for user pairs with exactly the same interests, while it drops to 0 for "different" users. Figure 3 shows the User Suggestion, i.e. two users nominated to be friends.

Providing labeled material as Divino Annotator. The machine learning methods proposed in Section 2.1 require labeled data in order to acquire a proper model of target phenomena. The role of *Divino Annotator* allows user to access the annotation functionalities. When logged, users can retrieve, add, remove and modify documents downloaded during the Data Gathering phase. Given a document, the user annotates all paragraphs with the corresponding information, such as Topic, Aspects and Polarity. In Figure 4 the interface shows a brief part of a document related to a specific wine, the *Chianti Classico*: in particular, two sentences expressing positive comments about the *taste* aspect are shown. The contribution of the Online Learning schema is emphasized in the annotation phase. In fact, the annotator can ask the system to automatically annotate the examples and validate the proposed information. When these are validated and submitted, the model can be corrected and improved through the novel annotations, so conforming to the Annotator notion of the target domain. In a real scenario, the system is expected to produce wrong annotations during its

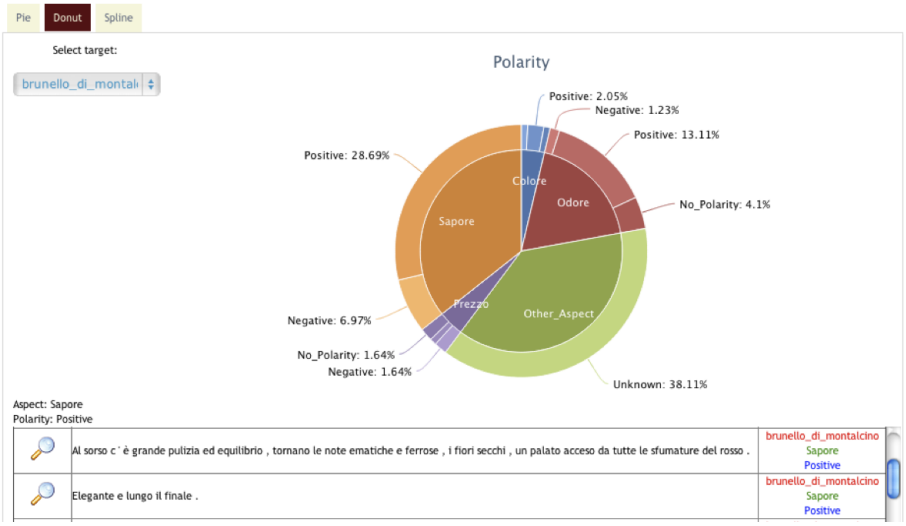

Fig. 5: Market Analyst interface

first life-cycle and to improve the annotation quality after a reasonable number of interactions with the annotators.

Advanced Business Intelligence in Divino. The automatic detection of users preferences and opinion from the portal and the corresponding translation in a semi-structured form, represents a valuable source of information for *Market Analysts* to feed Business Intelligence processes. Some of these information are automatically captured from user interactions, while other come from external sources, retrieved in the Data Gathering phase. For example, the Market Analyst can browse statistics about purchased items or the query logs from the Search Portlet. Advanced Business Intelligence techniques can also be applied in order to capitalize the knowledge extracted within the Opinion Mining process, as discussed in [11]. As an example, the Opinion Browsing portlet is shown in Figure 5: a multi-level pie chart, the so-called Donut, provides a synthetic view of opinions expressed by people within the forum or the targeted web pages. It is represented as the percentage of textual units expressing opinions about different aspects within a specified domain, such as `WineryProduct`. A fine-grained analysis can be enabled focusing on a specific target, e.g. a *Brunello di Montalcino*. For example, in Figure 5 the percentage of textual units giving positive comments about the *taste* of the product is 29,69%, while the percentage of negative comments about the *price* is 1.64%. The analyst can have a deep look on these statistics by clicking on every percentage, so visualizing the list of textual units and, if needed, can browse the source document. It is also possible to access to other reports and charts, enabling complex activities such as the monitoring of temporal trends, by visualizing the opinion depending on specific temporal based selections.

4 Experimental Evaluation

In this section, the Opinion Mining process is evaluated, as it represents the core functionality enabling Advanced Business Intelligence processes within the entire Divino Portal. In particular, the quality of classifiers powering different annotators described in section 2.1 is considered. The classification task is tackled through a Multiple Kernel approach, as discussed in [15]. Kernel methods are beneficial because the combination of kernel functions can be integrated into state-of-the-art classifiers, such as Support Vector Machines [15] or Passive Aggressive algorithm [3], as they are still kernels.

4.1 Textual Unit representation

A multiple kernel approach allows to combine the contribution of complex kernel functions to implicitly integrate different linguistic and semantic information of annotated examples. In this work, two kernels have been employed in our modeling. The Bag of Word Kernel (BOWK) reflects the lexical overlap between textual units t, represented as a vector whose dimensions correspond to different words. Each dimension represents a boolean indicator of the presence or not of a word in the text. The kernel function is the cosine similarity between vectors.

Another kernel is added, as lexical information of BOWK is highly affected by data sparseness, and words as found in test cases may often result rare or unseen in the training set. Our aim is to increase robustness to the resulting system by extending lexical information through *Distributional Analysis*. The core idea is that the meaning of a word can be described by the set of textual contexts in which it appears (*Distributional Hypothesis* as described in [6]). Words can be geometrically represented as vectors whose components reflect the corresponding contexts: two words close in the space (i.e. they have similar contexts) are likely to be related by some type of generic semantic relation, either paradigmatic (e.g. synonymy, hyperonymy, antonymy) or syntagmatic (e.g. meronymy, conceptual and phrasal association), as observed in [13]. A word-by-context matrix M is obtained through a large scale corpus analysis. Then the *Latent Semantic Analysis* [9] technique is applied to capture the statistical information of M by a lower k-dimensional space. Given two words w_1 and w_2, their similarity function σ is estimated as the cosine similarity between the corresponding projections $\boldsymbol{w_1}, \boldsymbol{w_2}$ in the space, i.e $\sigma(w_1, w_2) = \frac{\boldsymbol{w_1} \cdot \boldsymbol{w_2}}{\|\boldsymbol{w_1}\| \|\boldsymbol{w_2}\|}$. The result is that every word can projected in the reduced Word Space. The representation of a t consists of a linear combination of vectors representing words. Finally, the resulting Lexical Semantic Kernel (LSK) function is the cosine similarity between vector pairs, in line with [4], allowing to generalize the lexical information. The Word Space is acquired through the distributional analysis of a corpus made of about 2.5 million tokens; it is composed by web pages downloaded during the Data Gathering phase and pages from Wikipedia related to the `Wine` category, in order to have a space tied to the target domain. All words occurring more than 30 times are represented through vectors. The original space dimensions are generated from the set of the 20,000 most frequent words in the corpus. One dimension describes

Fig. 6: Examples of tag cloud: words referring positively to the *taste* of wine and negatively to the *price* of wine are shown on the left and right, correspondingly.

the Pointwise Mutual Information score between one feature, as it occurs on a left or right window of 5 tokens around a target. Left contexts of targets are treated differently from the right ones, in order to capture asymmetric syntactic behaviors (e.g., useful for verbs): 40,000 dimensional vectors are thus derived for each target, later reduced to $k = 250$.

As a side effect of the LSK, sentences are projected in the same representation space of words as in [9]. Given a textual unit u referring to an aspect a with a polarity p, the set of m words more semantically related to us can be obtained, namely W_t^k. By collecting all $W_{a_p}^k$ from sentences referring to a specific aspect a with a polarity p, a Tag Cloud can be obtained, as discussed in [5]. Figure 6 shows tag clouds related to the *taste* and *price* aspects. They are generated by selecting the $k = 20$ words more similar to examples used in this experimental evaluation. Notice that the word size depends on the number of times a tag is suggested by a single u.

4.2 Opinion Mining Results

In our approach, the kernel combination αBOWK + βLSK estimates the similarity between textual units, linearly combining lexical properties captured by BOWK and the lexical generalization of the LSK[5]. A set of 60 web pages has been annotated according to the schema proposed in Section 2.1. Annotations are derived from 7 specialized sites and blogs[6] from the enogastronomic domain targeted in the Divino Project. The Topic annotator is powered with a classifier associating paragraphs with respect to 4 classes, i.e. WineryProducts, Varietals, WineryBrands and OtherTopics. The analysis has been then specialized for the WineryProducts and each sentence within this topic has been classified with respect to different aspects, i.e. Taste, Aroma, Color, Price and OtherAspects. Each sentence related to a valid aspect is then

[5] Here, parameters α and β weight the combination of the three kernels. In our experiments, α and β are set to 1.

[6] We annotated pages from `www.intravino.com`, `www.enofaber.com`, `percorsidivino.blogspot.it`, `ilvinoeoltre.blogspot.it`, `grappolidivini.blogspot.it`, `simodivino.blogspot.it` and `grappolorosso.blogspot.it`.

Table 1: Number of examples for the Topic, Aspect and Polarity classifiers.

Topic	#par.
WineryBrands	71
WineryProducts	293
Varietals	25
OtherTopics	42
Total	431

Aspect	#sent.
Aroma	222
Color	60
Price	24
Taste	323
OtherAspects	239
Total	868

Polarity	#sent.
Positive	411
Negative	77
NoPolarity	141
Total	629

classified with respect to the Positive, Negative and NoPolarity classes. Table 1 shows the number of paragraphs annotated with Topic classes and the number of sentences annotated with Aspect and Polarity classes.

Table 2: Accuracy of the SVM and PA classifiers in the 10-fold cross validation schema; in parenthesis the standard deviation is reported.

	SVM	PA
Topic	85.45% (5.14%)	83,53% (7,44%)
Aspect	85.25% (3.60%)	84,67% (7,02%)
Polarity	79.17% (5.63%)	75,18% (15,89%)

In order to evaluate the robustness of the employed Passive Aggressive (PA) classifiers, we compared performances with a Support Vector Machine based classifier, which represents the state-of-the-art of kernel-based (non online) machines. In particular, the $SVM^{multiclass}$ schema described in [7] is applied[7]. A One-VS-All schema is used for the PA to realize the multi-classification: a binary classifier is used for each class and the one providing the highest classification function is selected. As the PA model depends on the order of example provided in the training phase, a 10 fold cross validation schema is applied. On the contrary, $SVM^{multiclass}$ adopts the implicit multi-class formulation described in [2]. Results are measured in terms of accuracy, i.e. the percentage of examples obtaining the correct labeling. Table 2 shows the mean results of both classifiers within the 10 folds. As expected, the SVM generally achieves slightly higher and more stable scores. It is not surprising as SVM, as a batch learning algorithm, finds the optimal solution of the classification problem, while the PA does not, according to its online nature [3]. However, high results achieved by different PA classifiers, i.e. about the 80% accuracy, confirms the applicability of online schema in the OM workflow within the Divino Portal. The slightly lower accuracy of the polarity classifiers emphasizes the complexity of capturing opinions in the domain of wine.

[7] http://svmlight.joachims.org/svm_multiclass.html

5 Conclusion

This paper shows a comprehensive web portal where Social Web functionalities, User Profiling and automatic Aspect-based Opinion Mining are integrated. The resulting portal allows people to express their preferences while enabling Market Analysts to bootstrap an opinion-mining system from scratch. The effectiveness of the proposed Online Machine Learning schema has been evaluated in a real use case in the national marketing of wine products. Future work will focus on improving the system bootstrapping capability with fewer annotated data, as well as a deeper study to combine modern Business Intelligence to semi-structured information extracted through Opinion Mining techniques.

References

1. Basili, R., Zanzotto, F.M.: Parsing engineering and empirical robustness. Nat. Lang. Eng. 8(3), 97–120 (Jun 2002)
2. Crammer, K., Singer, Y.: On the algorithmic implementation of multi-class svms. Journal of Machine Learning Research 2, 265–292 (2001)
3. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research 7, 551–585 (2006)
4. Cristianini, N., Shawe-Taylor, J., Lodhi, H.: Latent semantic kernels. J. Intell. Inf. Syst. 18(2-3), 127–152 (2002)
5. Halvey, M.J., Keane, M.T.: An assessment of tag presentation techniques. In: Proceedings of WWW 2007. pp. 1313–1314. ACM, New York, NY, USA (2007)
6. Harris, Z.: Distributional structure. In: Katz, J.J., Fodor, J.A. (eds.) The Philosophy of Linguistics. Oxford University Press (1964)
7. Joachims, T., Finley, T., Yu, C.N.: Cutting-plane training of structural SVMs. Machine Learning 77(1), 27–59 (2009)
8. Kim, S.M., Hovy, E.: Determining the sentiment of opinions. In: Proceedings of COLING. Association for Computational Linguistics (2004)
9. Landauer, T., Dumais, S.: A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review 104 (1997)
10. Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., Wilson, T.: Semeval-2013 task 2: Sentiment analysis in twitter. In: In SemEval 2013. pp. 312–320. Atlanta, Georgia, USA (June 2013)
11. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (Jan 2008)
12. Rajaraman, A., Ullman, J.D.: Recommendation Systems, chap. 9. Cambridge University Press (2011)
13. Sahlgren, M.: The Word-Space Model. Ph.D. thesis, Stockholm University (2006)
14. Seerat, B., and, F.A.: Article: Opinion mining: Issues and challenges (a survey). International Journal of Computer Applications 49(9), 42–51 (July 2012), published by Foundation of Computer Science, New York, USA
15. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York, NY, USA (2004)
16. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation 1(2), 0 (2005)
17. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of EMNLP. Stroudsburg, PA, USA (2005)

A Web Portal for Reliability Diagnosis of Bus Regularity

Benedetto Barabino[1], Carlino Casari[2], Roberto Demontis[2], Cristian Lai[2], Sara Mozzoni[1], Antonio Pintus[2], and Proto Tilocca[3]

[1] Technomobility s.r.l. - Cagliari - Italy
bbarabino@gmail.com , sara.mozzoni@technomobility.it
[2] CRS4, Center for Advanced Studies, Research and Development in Sardinia - Pula (CA) - Italy
casari@crs4.it, demontis@crs4.it, cristian.lai@crs4.it, pintux@crs4.it
[3] CTM S.p.A. - Cagliari - Italy
proto.tilocca@ctmcagliari.it

Abstract. In high frequency transit services, bus regularity - i.e. the headway adherence between buses at bus stops - can be used as an indication of service quality, in terms of reliability, by both users and transit agencies. The Web portal is the entry point of a Decision Support System (DSS), contains an environment designed for experts in transport domain. The environment is composed of tools developed to automatically handle Automatic Vehicle Location (AVL) raw data for measuring the Level of Service (LoS) of bus regularity at each bus stop and time interval of a transit bus route. The results are represented within easy-to-read control dashboards consisting of tables, charts, and maps.

1 Introduction

Nowadays there is a growing interest in the measurement of public transport service quality, which is a key factor for both users and transit agencies [1]. A relevant element of quality of service is reliability, viewed as the capability of transit operators to meet the expectations raised by the service offer in terms of multidimensional aspects such as time, passenger loads, vehicle quality, and so on [2]. In high frequency bus services, where scheduled headways between buses are 10/12 minutes (e.g. [3], [4], [5], [6]), one of the main aspects of reliability is regularity, which is faced in this paper. High quality evaluation of regularity means working on huge amounts of data, which must be collected and normalized before processing to avoid misleading information. Moreover, for efficient monitoring, it is necessary to be able to process the huge amount of data, present the results in a user-friendly way and guarantee a fast and pervasive access to them.

In this paper we propose: i) the implementation of a methodology to evaluate regularity starting from data collected by Automatic Vehicle Location (AVL); ii) a Web based system specifically designed to support experts in transport engineering domain for evaluating regularity issues.

Currently, AVL technology can collect the raw data for detailed analysis, but its use requires addressing challenges such as missing data points and possible bus overtakings. Moreover, while regular methods of bus operators are thought to operate in data-poor environments, new methods must be developed to exploit the rich-data environments provided by AVL. It is therefore important to develop new methods suitable to handle these data.

Ability to process data and quickly present results is a crucial factor for the efficient management of the service at hand (i.e. decisions based on data). Usually, in transit agencies the executives use spreadsheets to face this challenge. Raw AVL data are first downloaded in a standard PC, separated according to routes. Next, the value of regularity, per route and time period is calculated using formulas. Finally, results of the processed data (route direction, time periods and the value of regularity)are presented in a table. However, these activities are very time- and energy-consuming, because usually performed manually. Besides, results are available only locally. Thus, the need for fast procedures to effectively process AVL raw data, quickly present results and guarantee their access from everywhere. Hence, in order to shed additional light into the diagnosis of service regularity, the authors, making reference to their previous works ([7], [8]), implement in a Web application a method to derive accurate measure of bus regularity. Such method is expected to improve the quality and regularity of transit operators measurements which are too often made at a limited number of check points, on selected routes, and at limited time intervals. One more point, based on these measures, managers will be able to prioritize actions and/or give recommendations to improve the service. Last, but not least, thanks to the possibility to perform fast AVL data processing and thanks to their easy accessibility (as long as a Web connection is available), the workload of transit agencies will be reduced.

This paper is organized as follows. In Section 2, we motivate the choice of the regularity indicator, describe the challenges derived from AVL technologies, and mention a number of Web existing tools. In Section 3, we propose a methodology to evaluate regularity. In Section 4 the Web application and its control dashboards are presented. In Section 5, we present conclusions and research perspectives.

2 State of the Art

In high frequency services, regularity is a major aspect of service, and a classical topic for the transportation community. The major existing studies in the field, including details on the measure of regularity, AVL technology and existing Web tools, are presented in the following three subsections.

2.1 Measure of Regularity

Bus regularity can be measured by several indicators, which present pros and cons and denote the significant lack of a universal metric [5],[9]. The discussion about the several indicators used is not required in this paper, because

already presented in [7] and [8]. However, to summarize, we look for an indicator which should satisfy these properties: ease communication (understandable and easy-to-read), objectivity (i.e. without subjective thresholds), customer oriented (penalizing longer waiting times), independence from data distributions and ranking well-established regularity levels. As discussed in [7] and [8], the Headway Adherence (HA) measured by the Coefficient of Variation of Headways (C_{vh}) proposed by [9] is a good indicator fitting the following requirements:

- although the C_{vh} is not of immediate understanding and communication, its values represent LoSs ranked in a well-established scale of regularity from A (the best) to F(the worst);
- it is objective; LoS thresholds are related to the probability that a given transit vehicle's actual headway is off headway by more than one-half of the scheduled headway;
- it is quite customer oriented; every trip is considered in the computation of the C_{vh}, to penalize long waiting times at bus stops. The output indicated the probability of encountering an irregular service, even if it is not a measure of severity of the irregularity;
- it does not require particular applicability conditions. Since bus operators sometimes schedule high frequency services irregularly, it is important to consider different headways in different time intervals;
- it can evaluate different regularity conditions and detect bunching phenomena.

2.2 AVL Technology and Regularity

Due to economic constraints and lack of technology, early experiences in the determination of regularity measures were performed at a few random or selected check points of route (e.g. [4],[6]). Typically, collected data were aggregated manually in time periods representing slack and peak hours in the morning and in the evening. This way of working generates restricted analysis and leads to limited conclusions. When data are aggregated from checkpoints to route level, one typically loses a considerable amount of information on the regularity between consecutive checkpoints. This procedure is rarely user-oriented, because passengers are mostly concerned with adherence to the headways at their particular bus stop (e.g. [10]). Hence, in order to provide the best possible service to passengers, measures should be performed at every stop of the bus route and for every investigated time period. In this way, performing regularity measures at all bus stops and time periods removes shortcomings deriving from choosing checkpoints and aggregating data in large time periods. Nowadays, relevant support is provided by AVL technology, because it can collect huge amounts of disaggregated data on different bus stops and time periods. Most important, if properly handled and processed, AVL data have the capability to show when and where the service was not provided as planned. However, there are two main criticalities which must be faced before being able to perform accurate regularity calculation. Indeed in case of not addressed criticalities, the calculation of

regularity does not sufficiently reflect the service that customers experience and it can provide misleading information. These criticalities are:

1. Bus Overtaking (BO) which arises when the succeeding scheduled bus overtakes its predecessor in the route;
2. Missing data point, which consist of Technical Failures (TF) depending on AVL being temporarily out of work, and Incorrect Operations in the Service (IOS), such as missed trips and unexpected breakdowns.

Due to possible BO, buses might not arrive in the right order. For passengers whose aim is to board on the first useful arriving bus at bus stop, BO is irrelevant because the headway is the time elapsed between two consecutive buses, in which the last one may or may not be the scheduled bus. Hence, instead of tracking the bus (e.g. [11]), regularity measures should focus on transits (i.e. arrivals or departures) of the first bus arriving at the considered bus stop. TF and IOS result in missing data points, which are not recorded by AVL. Moreover, they result in temporal gaps. Hence, a crucial challenge is to recognize the type of missing data and handle the temporal gaps, because they have a different impact on users. The temporal gaps due to TF lead to an incorrect calculation of headways, because buses actually arrived at bus stop, but they were not recorded by AVL. Considering the temporal gaps due IOS is favorable because they are perceived by users as real. McLeod [12] provided insights, in order to determine temporal gaps due to TF and showed that less than 20% of missing data due to TF leads to good quality headway measures. In [7] and [8], in order to recognize and address BO, TF and IOS, a method has been proposed in the case of regularity analysis at the single route and at the whole bus transportation network, respectively. However, in this case two software applications are used to implement the method. Therefore, additional work must be done to implement the method by a single application in order to make AVL data a mainstream source of information when regularity calculation are performed.

2.3 Web Regularity Tools

A key factor for the effective analysis of data is building intelligible performance reports. To date, there are few state of the art of modern Web platforms, specifically designed to providing a Decision Support System (DSS) focused on reliability diagnosis of bus regularity. There are a few research works focusing on Web-based, AVL data visualization, as in [13] or data analysis algorithms and techniques, including a very basic visualization of route paths and speeds using Google maps as in [14]. On the other hand, the current state of the art of information systems technologies includes mature and reliable tools. Mainstream commercial products, or Web frameworks released under Open Source licenses, are designed and documented to integrate with other systems in order to build complex and large-scale Web applications, usually thanks to the use of Application Programming Interfaces (APIs).

Some noteworthy product and framework categories are: business intelligence (BI) tools, reporting and OLAP systems, as Jaspersoft [4] or Pentaho [5]; Web portals development platforms and Content Management Systems (CMS), as the open-source ones like Entando [6] or Joomla [7]; database management systems (DBMS), like the well-known and broadly adopted MySQL [8] or PostgresSQL [9].

3 Methodology

In this section we summarize the method implemented in the Web Portal described in section 4. The method is taken from previous author's works ([7] and [8]) where further details can be found. The method addresses three main phases such as: to validate AVL data, to address criticalities in AVL raw data and to determine the value of C_{vh} in order to illustrate the LoS of regularity over space at every bus stops and route direction - and time - at every time period - in a bus transit network.

3.1 The Validation of AVL Raw Data

Specific attention must be paid to bus stops, because bus operators measure regularities at these points, where passengers board and get off. In this methodology, the relevant elements recorded by AVL at each bus stop for each high frequency route are: day, route, direction, actual and scheduled transit times.

When comparing the numbers of actual and scheduled transits, the lack of data might be observed due to IOS and TF. In this paper, we contemplate the situation where the transport service is good, according to historical data. As a result, few IOS are expected to occur. Therefore, missing data point are fundamentally TF which must be detected and processed in order to determine correct measures of headways. For this reason, we consider the following three main steps to accept or reject data related to days and months and validate a counting section.

STEP 1. Read daily AVL data at a bus stop of a specific route and check whether the number of recorded transits is larger than or equal to a certain percentage of scheduled transits. This percentage can be set equal to 80% of scheduled transits, because McLeod [12] showed that the estimation of headway variance is still good when 20% of data are missing. If a bus stop meets this criterion in that day, it is used for the next step.

STEP 2. Perform a chi-square test on the set of bus stops selected by STEP 1 to evaluate the approximation of the actual number of transits to scheduled

[4] http://www.jaspersoft.com
[5] http://www.pentaho.com
[6] http://www.entando.com
[7] http://www.joomla.org
[8] http://www.mysql.com
[9] http://www.postgresql.org

transits. A suitable significance value for this test is $\alpha = 0.05$. If a day satisfies this criterion, it is used for the next step as well as the bus stops of that day.

STEP 3. Collect all bus stops satisfying STEP 2 in a monthly list and compute the ratio between the number of bus stops in the monthly list and the total number of bus stops. If this ratio is larger than a threshold, all monthly data are supposed to be valid. Based on our experience in preliminary tests, we recommend the use of percentages larger than 60%, which is a good threshold value, in order to cover a significant number of bus stops in a route.

A detailed example of steps 1, 2 and 3 is reported in [7].

3.2 The Handling of Criticalities

Data validation is followed by the detection of criticalities in order to correctly calculate headways between buses. This phase is applicable both in case of few and of many unexpected IOS. As illustrated in section 2.2, three types of criticalities might occur: BO, TF and IOS. Since TF and IOS lead to missing data and temporal gaps, they can be addressed almost together. Gaps must be found and processed by comparing scheduled and actual transit times. Sophisticated AVL databases can be used to match scheduled transit time data (with no gaps) with actual transit time data (with possible gaps). As a result, to address all criticalities, the following steps are carried out:

STEP 4. Address BO by ordering the sequence of actual transit times at bus stops, because BO is irrelevant for the regularity perceived by users, who are not interested in the right schedule of buses.

STEP 5. Fill up tables reporting unpredicted missed trips and unexpected breakdowns. Columns include the day, bus stops, route, direction, scheduled transit times and incorrect operation code, indicating whether it is neither a missed trip or a breakdown.

STEP 6. Consider the table of scheduled service on that day with the attributes of the table in STEP 5, whereas the incorrect operation code wil be neglected, because it is an unexpected event.

STEP 7. Match these tables, in order to generate a new table of scheduled transits with a new attribute indicating the incorrect operation code.

STEP 8. Detect TF and IOS. Match the table built in STEP 7 with data at the end of STEP 4 and detect possible gaps, when a transit between two recorded transits is missing.

STEP 9. Correct TF and IOS. Disregard gaps generated by TF, because no real headways can be derived as the difference between two consecutive transits. Keep the gaps generated by IOS, because these gaps are really perceived by users.

A detailed example of steps 4,8 and 9 is reported in [7] and [8].

3.3 The Calculation of Regularity LoS

Given a generic bus stop j at time period t along the direction d of a route r, once data have been validated and criticality have been addressed, we calculate the C_{vh} as follows:

$$C_{vh}^{j,t,d,r} = \frac{\sigma^{j,t,d,r}}{h^{j,t,d,r}} \tag{1}$$

where:

- $\sigma^{j,t,d,r}$ is the standard deviation of the differences between actual and scheduled headway at bus stop j, time interval t, direction d and route r; the values used for the evaluation span over a monthly planning horizon.
- $h^{j,t,d,r}$ is the average scheduled headway at bus stop j, time interval t, direction d and route r.

In many transit agencies, the standard time interval is one hour. Since transit agencies may add or remove some bus trips to better serve the changing demand ([14]), in this paper $h^{j,t,d,r}$ is computed as the average of headways of scheduled transits times, to account for these additional trips and possible gaps. As illustrated previously, Eqn. (1) provides results in a monthly planning horizon whose set is denoted by S. The elementary observation is denoted by $x_i (i = 1, n)$ and represents the precise headway deviation at the end of STEP 9. However, in order to provide monthly aggregated statistics for week and type of day, Eqn. (1) is also calculated for them, considering the sub-sets S_1 and S_2. The related elementary observations are denoted by $x_{1j} (j = 1, , m)$ and $x_{2k} (k = 1, , p)$ and represents the precise headway deviation at the end of STEP 9, when they are related to the week and type of day, respectively. Therefore, to summarize, the results provided by Eqn. (1) refer to the considered sets defined as:

S - the set representing the headway deviations (x_i) within the month.
S_1 - the set representing the headway deviations (x_{1i}) within the considered week in the month.
S_2 - the set representing the headway deviations (x_{2i}) within the considered type of day in the month.

The calculated values of C_{vh} can be converted into the LoS according to [9]. LoSs can be represented by dashboards as illustrated in the next section.

4 Web Portal

The web portal is the entry point of the DSS (hereafter the "system") composed of an environment designed for transit industry experts. The environment is designed to primarily handle AVL raw data for measuring the LoS of bus regularity at each bus stop and time period of a transit bus route. The web portal is powered by Entando, a Java Open Source portal-like platform for building information, cloud, mobile and social enterprise-class solutions. It natively combines portal, web CMS and framework capabilities. The portal provides dashboards

that support experts with significant and useful data. A main feature of the dashboards is to identify where and when regularity problems occur. Dashboards are intended to show summaries and consist of tables, charts and maps. The items located on the dashboards include:

- regularity table - a table showing the LoS; the executive may select a route direction and a time period; the table shows for each bus stop of the route the LoS of regularity in different colors;
- regularity line chart - a multi-line chart showing one chart for each time period; the charts are distinguished by color and may be immediately compared;
- mapper - a Google Maps technology based interface; the map shows the path of a selected route direction. The bus stops associated with the route are represented with different colors depending on the LoS of regularity; the executive may interact with the map changing the time slot within the time period.

4.1 Components of the System

The web portal is part of the whole system. The Figure 1 shows the components of the system.

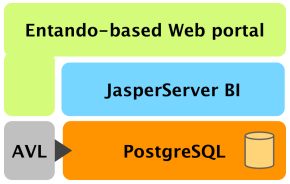

Fig. 1. Components of the system.

At the bottom, the AVL is the technology which collects raw data during the transport service. As illustrated in section 3, collected data deals with real measures including actual and scheduled bus transit time, bus information, route information, bus stops information. A pre-filtering process is necessary before storing in the database in order to harmonize data in case of non-homogeneous values. The database is a PostgreSQL instance extended with spatial and geographic features, PostGIS[10]. 451Research[11] estimates that around 30% of tech companies use PostgreSQL for core applications as of 2012. PostgreSQL is an Open Source solution that strongly competes with proprietary database engines and is supported by a consistent community of users. The use of the database in the system is twofold. It is first necessary to store the data collected by the

[10] http://www.postgis.net
[11] https://451research.com/

AVL. An entity-relationship diagram defines how data is scattered among the tables. The database is also necessary for the administrative activities required to manage the portal. An instance of JasperServer is responsible for dashboards creation. JasperServer is a Business Intelligence (BI) tool and a reporting and analytics platform. With JasperServer it is possible to create single reports or dashboards faster. The dashboards are pre-defined through the JRXML markup language and are available for integration in the system. Entando Web Portal provides the user interface.

4.2 Implementation of the method

The implementation of the method consists in four modules, that manage the four stages of the data flow process: data import; data processing; data pre-aggregation and data management.

All the modules use PostgreSQL functions in PL/pgSQL language to perform database's tasks and Entando modules in Java language to start and supervise the execution of each task as the Web application. The data are collected in a single database. A database schema is created for each month in which data is elaborated.

The main entity is represented as a table named "Bus_Stop". A Bus Stop entity contains date attributes, which are used to generate dynamically the path in use in a particular year and month. Moreover, a geometric attribute contains the polyline, which starts at the previous bus stop and ends at the considered bus stop. In this way, using spatial aggregate functions, one is able to merge polyline of consecutive bus stops per path code, in order to derive spatial characteristics (geometry) of the Path entity.

The first module contains the functions that implement the first phase of the methodology illustrated in Section 3: the raw data of a month are imported and validated. The system loads these primary data in two tables: the 'AVL' table where each row contains real transit at bus stops and the 'Scheduler' table which contains the scheduled transit. Then the system validates the 'AVL' by applying the three steps described in Section 3.1. The parameters of transits percentage (80%), the chi-square test value ($\alpha = 0,05$) and the threshold of the percentage ratio between the number of bus stops that pass the chi-square test and the total number of bus stops (60%) can be changed by the analyst.

The second module contains the tools that implement the second phase of the methodology. The module generates the "Differences" facts table. This table contains the difference between actual and scheduled headways between two consecutive buses as measure and two multi level dimensions: year, month, day and time slot are the temporal dimension; route, path and bus stop are the logical dimension.

The third module implements the third phase of the methodology. It generates the "Regularity" facts table that contains the regularity measures evaluated over three distinct type of day aggregations: by week (4 measures), by day of week (7 measures) and by the entire month. The pre-aggregation uses the eqn.(1) to calculate the C_{vh} measure over each set of samples defined in Section 3.3. The

fourth module contains a set of functions to manage data and reports. The System can also manage the JasperServer configuration for its connection to the database and regularity reports definition via the Entando/JasperServer Connector. Currently, a Mondrian Olap Cube for "regularity" or "differences" facts tables is not yet configured.

4.3 Reliability Diagnostic Tools

AVL technology is installed in-busses. It records several data, such as actual arrival times at every bus stop in minutes and seconds. Such information belongs to the class of time-at-location data collected by vehicles and can be used for off-line analysis. As a vehicle finishes its service, it moves back to the depot, where data recorded during the daily shift are downloaded. Daily data are stored in a central database. The diagnostic tools are realized using some JasperServer functionalities. First, the database containing the facts tables is connected to the JasperServer business intelligence engine. Then, as shown in figure 2, three type of reports are created through the JasperServer tools: the regularity table, the multi-line charts and the mapper .

Each report shows the regularity measures of all bus stops of a particular route direction. For the sake of clarity in representation, the value of measures is represented by colours depending on LoS. The red colour represents LoS **F** ($C_{vh} > 0,75$, i.e. most vehicles bunched), the orange colour indicates LoS **E** ($0.53 < C_{vh} < 0.74$, i.e. frequent bunching), the yellow colour shows LoS **D** ($0.40 < C_{vh} < 0.52$, i.e. irregular headways, with some bunching). Other colour gradations mean LoS from **A** to **C** ($C_{vh} < 0.40$, i.e. satisfactory regularity). When LoS are not available, they are denoted by **null**.

The executive selects the route direction and the day aggregation type. Moreover, when a report is showed, the executive can select the time slot.

The regularity classes (see figure 2) of a bus stop in a selected time slot are represented as a table in the regularity table report, as a coloured poly-lines and icons in the mapper report or as a coloured lines in the multi line charts report. It is important to highlight that figure 2 is the result of different screens, than there is no exact correspondence among the colouring. In order to permit a map representation of bus stop and path, the geometries in the "Bus_Stops" table are transformed in the WGS84 projection and GeoJSON format using Postgis functionality and linked to the reports table.

5 Conclusion

In bus transit operators the measure of regularity is a major requirement for high frequency public transport services. Besides, it is necessary to properly account for the efficient monitoring of quality of service and for the perspectives of both bus operators and users. In this paper we have implemented a methodology to evaluate regularity starting from data collected by AVL, and proposed the integration of technologies in a web portal as an environment designed to

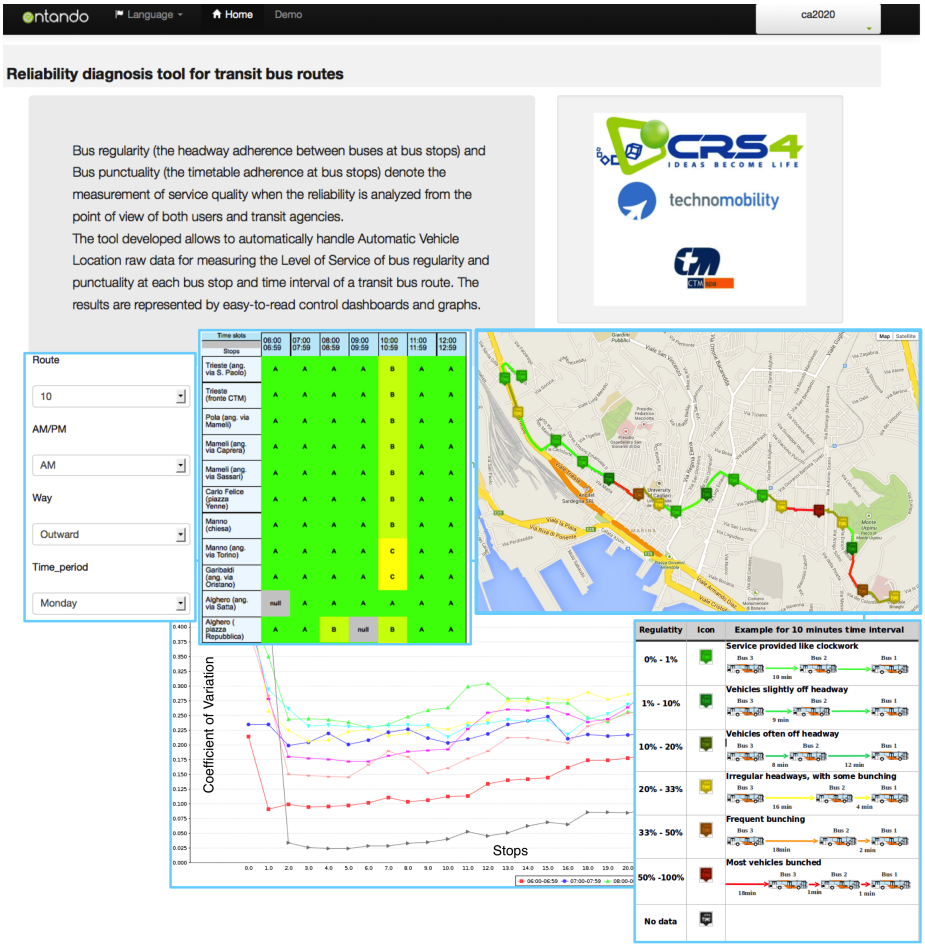

Fig. 2. Diagnostic tools.

support bus transit operators experts in evaluating regularity issues. This paper shows that it is possible to handle huge AVL data sets for measuring bus route regularity and understand whether a missing data point is a technical failure or an incorrect operation in the service, providing a detailed characterization of bus route regularity at all bus stops and time periods by AVL technologies. The web portal ensures tool access from everywhere and anywhere. This procedure results in significant time and energy savings in the investigation of large data sets.

The next step will be to extend the web portal for both operators and users, then at a later stage for transit agencies and passengers. Illustrating the practical effectiveness of this procedure will be important to implement a real case study. User-friendly control dashboards help to perform an empirical diagnosis of

performance of bus route regularity. Transit managers can use easily-understood representation and control rooms operators following buses in real time, to focus on where and when low regularities are expected to occur. Moreover, possible cause of low level of service will be investigated, in order to put the bus operator in the position of selecting the most appropriate strategies to improve regularity. In addition, the method and the integration of technologies will be adapted for the measurement of punctuality in low-level frequency services.

References

1. European Committee for Standardization, B.: Transportation logistics and services. european standard en 13816: Public passenger transport service quality definition, targeting and measurement. (2002)
2. Ceder, A.: Public Transit Planning and Operation: Theory, Modelling, and Practice. Butterworth-Heinemann, Oxford, England (2007)
3. Marguier, P., Ceder, A.: Passenger waiting strategies for overlapping bus routes. Transportation Science **18(3)** (1984) 207–230
4. Nakanishi, Y.: Bus performance indicators. Transportation Research Record **1571** (1997) 3–13
5. Board, T.R.: A Guidebook for Developing a Transit Performance-measurement System. Report (Transit Cooperative Research Program). National Academy Press (2003)
6. Trompet, M., Liu, X., Graham, D.: Development of key performance indicator to compare regularity of service between urban bus operators. Transportation Research Record: Journal of the Transportation Research Board **2216**(1) (12 2011) 33–41
7. Barabino, B., Di Francesco, M., Mozzoni, S.: Regularity diagnosis by automatic vehicle location raw data. Public transport **4(3)** (2013) 187–208
8. Barabino, B., Di Francesco, M., Mozzoni, S.: Regularity analysis on bus networks and route directions by automatic vehicle location raw data. Forthcoming in IET Intelligent Transport Systems (2013)
9. Board, T.R.: Transit capacity and quality of service manual, 2nd ed. Transit Cooperative Research Program Report 100. Transportation Research Board (2003)
10. Kimpel, T.: Time Point-level Analysis of Transit Service Reliability and Passenger Demand. Portland State University (2001)
11. Mandelzys, M., Hellinga, B.: Identifying causes of performance issues in bus schedule adherence with automatic vehicle location and passenger count data. Transportation Research Record: Journal of the Transportation Research Board **2143**(1) (12 2010) 9–15
12. McLeod, F.: Estimating bus passenger waiting times from incomplete bus arrivals data. J. Oper. Res. Soc. **58(11)** (2007) 1518–1525
13. Oluwatobi, A.: A gps based automatic vehicle location system for bus transit . Academia.edu (2013)
14. Cortés, C.E., Gibson, J., Gschwender, A., Munizaga, M., Zuniga, M.: Commercial bus speed diagnosis based on gps-monitored data. Transportation Research Part C: Emerging Technologies **19**(4) (2011) 695 – 707