Boosting collaborative ontology building with key-concept extraction

Sara Tonelli, Marco Rospocher, Emanuele Pianta and Luciano Serafini
Fondazione Bruno Kessler-irst
Via Sommarive 18 Povo
I-38123, Trento, Italy
{satonelli,rospocher,pianta,serafini}@fbk.eu

Abstract—We present a wiki-based collaborative environment for the semi-automatic incremental building of ontologies. The system relies on an existing platform, which has been extended with a component for terminology extraction from domain-specific textual corpora and with a further step aimed at matching the extracted concepts with pre-existing structured and semi-structured information. The system stands on the shoulders of a well-established user-friendly wiki architecture and it enables knowledge engineers and domain experts to collaborate in the ontology building process.

We have performed a task-oriented evaluation of the tool in two real use cases, one for incrementally constructing the missing part of an environmental ontology, and the other one for validating the quality of an existing ontology in terms of terminological covering of the domain it was supposed to represent. In both cases, the tool effectively supported the users in the specific task to be performed, thus showing its usefulness for knowledge extraction and ontology engineering.

I. INTRODUCTION

The development of ontologies is a crucial task in many fields in which knowledge needs to be shared and reused, such as knowledge management, e-commerce, database design, educational applications, etc. One of the main purposes for building ontologies is to eliminate the conceptual and terminological confusion that often occurs among the members of a community that share information of various kinds.

Building an ontology is a complex task that requires a considerable amount of human effort, and although several attempts have been proposed to automatize some steps of the process (see for example the ‘ICML Workshop on Learning and Extending Lexical Ontologies by Using Machine Learning Methods’ held in Bonn in 2005), it cannot be currently performed without human intervention.

We claim that the creation of ontologies can benefit from the automatic extraction of knowledge from documents related to the ontology domain and to its linking with available structured and semi-structured resources, such as WordNet\(^1\) and Wikipedia\(^2\). However, we believe that the user must be in charge of the final decision about which concepts should be integrated into the ontology in the light of some suggestions and additional information provided by an automatic system.

We present an online environment in which the ontology building process can be performed collaboratively in a Wiki-like fashion. To start the construction (or the extension) of an ontology, the user can exploit a domain corpus, from which the terminological component of the system automatically extracts a set of domain-specific key-concepts. These key-concepts are further linked to existing external resources to get other useful information such as the concept definition, the synonyms and the hypernyms. Finally, the user can easily select through the interface which concepts should be imported into the ontology.

The paper is structured as follows: in Section II an overview on existing systems for ontology building and extension is given, while in Section III we detail the environment we have developed by describing the whole workflow and the single components. In Section V a use case is discussed in which the system was successfully employed to semi-automatically create a specific ontology. Finally, we draw some conclusions and discuss future work directions in Section VI.

II. RELATED WORK

The approach commonly adopted for building ontologies is based on manual effort, especially when the knowledge to be encoded is particularly complex and multi-faceted. Several past studies (e.g., [2], [3], [4]) have tried to define a methodology or ‘best practice’ for building ontologies, which usually includes i) the identification of the goals, intended application, target users of the ontology, ii) the ontology building process, iii) the evaluation and iv) the ontology documentation. The second step relies on several sub-tasks, ranging from the identification of core concepts and relationships to the coding in some formal language and to the possible integration with existing ontologies. Some of these sub-tasks can be performed (semi-)automatically through a so-called Ontology Learning process, which has been the topic of a number of recent workshops (e.g., [5], [6]) and books (e.g., [7], [8]).

Ontology learning is a generic term pertaining all information extraction tasks and approaches aimed at (semi) automatically extracting relevant concepts and relations from a corpus to be included in an explicit, formal specification, i.e. the ontology. The corpus can be processed in different ways to retrieve such information, for example it can be first parsed [9], or collocations can be identified [10] or also semantic graphs can be retrieved [11]. Also the choice of the

\(^1\)http://wordnet.princeton.edu
\(^2\)http://www.wikipedia.org
corpus from which concepts and relations should be retrieved represents a relevant issue. In a seminal work by [12], the authors first proposed to use the web as a corpus for ontology learning, although all collected documents must undergo a selection process in order to create a corpus that is both relevant for the learning process and specific for the domain to be defined. Some authors prefer to handle texts from well-established and widely known resources such as the BioMed corpus for the biomedical domain. In our case, given the complexity of ontology development, we devised a methodology for semi-automatic ontology building that on the one hand exploits knowledge from general resources like WordNet and Wikipedia, and on the other hand relies on domain-specific corpora. Many applications for ontology building have been recently developed using different techniques (for an overview, see [13]). Some of the tools based on a similar approach to our (semi)-automatic, data-driven methodology are the following:

**OntoLearn** [14]: The tool first extracts specific terminology from domain corpora, then filters the terms using natural language processing and statistical techniques, and then assigns them a semantic interpretation in order to find taxonomic relations. More specifically, compound terms are split into single concepts, and each of them is disambiguated by assigning a WordNet synset. All this knowledge is then merged to create semantic graph representations expressing taxonomic relations to be added to an existing ontology. Although this methodology is extremely interesting, especially as regards the detection of semantic interconnections between concepts, we believe that automatic word sense disambiguation could prune away some senses that the user may want to consider. Therefore, in our system we opt for a more inclusive approach (for example suggesting several possible interpretations of a concept).

**OntoGen** [15]: The tool supports the user in building an ontology by extracting possible concepts and relations between them from domain texts. The user can select a topic through a graphical user interface, and the system automatically suggests some potential subtopics from a set of selected documents. In a first phase, the main concepts are extracted from the documents using either centroid vectors or Support Vector Machine. Then, the selection of subtopics is performed through Latent Semantic Indexing (LSI) or k-means clustering. The system also includes a text visualization component and an ontology population module that shows to the user instances and the possible associated concepts with a confidence score. The functionalities of this tool meet some requirements that we and the possible associated concepts with a confidence score. The ontology population module that shows to the user instances and the possible associated concepts with a confidence score. The example above show on the one hand that the need for interactive tools to support the ontology building process is a relevant problem, and on the other hand that the use of domain texts is now a consolidated technique, which we integrate also in our system. Nevertheless, we introduce also some novel elements w.r.t. previous works, for example the collaborative approach to ontology building through an online interface, the possibility to exploit external dictionaries and also the support to the creation of an ontology in different languages (for the moment, English and Italian).

Many other tools have been developed for ontology-based information extraction, and integrated for example in query answering systems [17] or used for extracting information from Wikipedia infoboxes [18]. However, such applications are not directly related to our ontology building approach. For a survey, see [19].

### III. MAIN SYSTEM COMPONENTS

We develop a general workflow that, starting from a domain-specific textual corpus, first extracts a list of terms, then retrieves additional information from available lexical resources and finally enables the user to add new concepts to the ontology through a user-friendly graphical interface. The workflow has been made available in an online collaborative environment which can be potentially used for any kind of ontology domain. We have taken advantage of the existing infrastructure for conceptual modelling provided by MoKi [20] and we have enriched it with a terminology-extraction component. Besides, we have integrated the tool with external lexical resources, which include WordNet sense repository [1], WordNet domains [21], Wikipedia (in different languages) and possibly domain-specific dictionaries. The framework we have designed enables the users to take all decisions in the ontology construction process, supported by the software that helps by suggesting relevant concepts.

The general approach to ontology extension presupposes that the user has collected a domain corpus in one of the languages supported by the tool that can be used to extract the most relevant terms in the documents, which we assume represent domain-specific terminology. We rely on a terminology extraction step because we agree that ‘terminology is the surface appearance, in texts, of the domain knowledge.
of a community. Because of their low ambiguity and high specificity, these words are also particularly useful for conceptualization of a domain ontology [14]. The workflow for ontology building and extension is displayed in Figure 1. First, the module for terminological extraction is run on the domain corpus in order to obtain a set of domain-specific terms, that are seen as candidate concepts for the integration into the ontology. Such terms are obtained using KX, a robust system for keyphrase-extraction [22] which can be easily configured by the user. Then, additional information coming from available lexical resources is retrieved and displayed, so that the user can decide whether to integrate it into the ontology under development or to discard it. Further details about the single workflow components are given in the following subsections.

A. MoKi

MoKi [20] is a collaborative MediaWiki-based tool for modeling ontological and procedural knowledge in an integrated manner. The main idea behind MoKi is to associate a wiki page to each basic entity of the ontology, i.e., concepts, object and datatype properties, and individuals. Each basic entity is associated to a MoKi page, composed of an unstructured part and a structured part. The unstructured part contains informal text, possibly enriched by formatting information, links to other MoKi pages or to external resources, uploaded images, and so on. In the current implementation, the content of this part is stored according to the standard MediaWiki markup format. The structured part, which is delimited by specific tags to separate it from the unstructured text, contains knowledge stored according to the modelling language adopted. In the current implementation, the structured part of a page contains a RDF/XML serialisation of a set of OWL statements formalising the element described in the page.

MoKi implements a multi-mode access to the page content, to support easy usage both by domain experts (who own knowledge about a domain, sometimes implicitly, but usually lack the skill of making this knowledge explicit in a formal model) and knowledge engineers (who have the capability of encoding a piece of informal knowledge into a formal model, but usually have no or limited understanding of the domain to be modeled), thus facilitating them to play an equally central role in the modelling activities.

MoKi implements a multi-mode access to the page content, to support easy usage both by domain experts (who own knowledge about a domain, sometimes implicitly, but usually lack the skill of making this knowledge explicit in a formal model) and knowledge engineers (who have the capability of encoding a piece of informal knowledge into a formal model, but usually have no or limited understanding of the domain to be modeled), thus facilitating them to play an equally central role in the modelling activities.

Fig. 2. The lightly-structured access mode for editing the structured part of a MoKi concept page.

The fully-structured access mode allows the user to edit/view the content of the structured part of a MoKi page using the full expressivity of the modelling language adopted. That is the user is able to view/edit formal statements (OWL axioms) describing the element associated to the page. This access mode specifically address the needs and requirements of knowledge engineers. Instead, the purpose of the lightly-structured access mode (see Figure 2) is to allow users with limited knowledge engineering skills, to edit/view the content of the structured part of the MoKi page in a simplified and less formal way. In the current version of the tool, the lightly-
structured access mode is provided through a form, where the user can view and edit simple statements which can be easily converted to/from OWL statements.

MoKi also offers model overview pages, that is special pages dynamically created from the (structured) content of the pages describing model elements. This model overview pages allow to explore (either graphically or in a tabular-based form) the generalisation and part/subparts decomposition hierarchies of concepts, as well as the classification of the individuals.

B. KX

KX [22] is a system for key-concept extraction based on statistical and linguistic information, which can be run both on a single document to get its most relevant keyphrases and on a document collection to extract domain-specific terminology. In both cases, the basic extraction methodology remains the same. In our system, KX was integrated in the ontology building workflow in order to extract domain-specific terminology. The extraction process is possible only if a domain corpus is available. Starting from such corpus, the extraction steps are the following:

Step 1 – N-gram Extraction: the list of all possible n-grams (longer than one) ordered by frequency is extracted from the corpus. The maximum length of n-grams can be set by the user; default is 4. N-grams with frequency lower than a threshold (default 5) are filtered out.

Step 2 – Multi-word Extraction: the n-gram list is filtered with the aim of retaining only Multi-Word Expressions (MWEs), i.e. combinations of words expressing a unitary concept, for example ‘light beam’ or ‘access control’. Filtering is carried out through two mechanisms. First we exclude all n-grams containing at least one stop-word (taken from a list configurable by the user), e.g. ‘because’, ‘is’, etc. Note that prepositions are not included in the stop-word list, because we want to be able to extract multi-words such as ‘powder of pollen’. Then we retain only n-grams matching one of a predefined list of lexical patterns (e.g. Adjective+Noun, Noun+Preposition+Noun). What we get from this filtering is an ordered list of probable MWEs. Items on top of the list are more frequent in the corpus, and have a higher probability of being actual MWEs.

Step 3 – Multi-word Recognition: with the list produced in Step 2, we go back to the corpus and recognize the actual occurrence of MWEs in each text, possibly solving local ambiguities between nested or partially overlapping MWEs. After multi-word recognition, the units of our text are not tokens anymore, but lexical units, where a lexical unit can include one or more tokens.

Step 4 – Ranking of key-concept by frequency: we now group lexical units into key-concepts by recognizing some morphological variants and synonyms (to a limited extent). For each text we can now build a list of key-concepts ordered by frequency. A black-list can be used by the user to avoid that certain specific concepts are included in the key-concept list of any text.

Step 5 – Re-ranking of key-concepts by relevance: the initial list of key-concepts ordered by frequency is re-ranked by taking into consideration various criteria such as first occurrence of key-concepts in the text (early occurrence points to higher relevance), and degree of specificity (longer concepts are assumed to be more specific, and higher specificity implies higher relevance). Re-ranking can also be based on key-concept inverse document frequency (K-IDF).

Step 6 – Terminology extraction: a list of the most relevant terms in the corpus is extracted by merging in a unique list the percentage of the top-most key-concepts from each document.

All these steps are performed in a pipeline by the terminology extraction component after that the user has uploaded a corpus through the interface (the most popular document format is supported by the system: Adobe PDFs, Microsoft Office documents, OpenOffice documents, plain text files) and has started the automatic extraction process. However, some parameters can be manually set to modify the output of some of the steps described above. A screenshot of the parameters to define through the MoKi interface is displayed in Figure 3.

The languages currently available are English and Italian. This means that the uploaded domain corpus must be in one of these two languages, and also that the ontology under construction will include concepts either in English or in Italian. The available languages are constrained on the one hand by the underlying text processing component that is needed for the pattern-based filtering of multiword expressions (see Step 2). On the other hand, they influence also the kind of information provided through additional resources (see following subsection). However, the system could be extended to other languages by integrating a new morphological analyzer in KX and accordingly defining PoS patterns (see Step 2) to recognize MWEs. This shows that the effort required to support further languages is rather limited.

As for the domain option, environment, information technologies and news are available for English, while architecture, medicine and news can be chosen for Italian. This means that some pre-processing of pre-existing domain corpora has been done, which outputted relevant statistical information used for the final key-concept ranking. In particular, the inverse document frequency of each key-concept in a domain corpus (K-IDF) has been computed, which is used in the final ranking to give more relevance to key-concepts with a higher K-IDF (i.e. not occurring in many documents of the domain corpus). If no domain is set, this information is not used.

Another important parameter is the percentage of relevant concepts to return (see Step 6), which is 15% by default. Note that this value should be adapted to the size of the document set used for terminology extraction: if the document set is small, 15% may correspond to few key-concepts, while on a big corpus this may return a very long list.

As for the parameters used to set the frequency threshold for MWEs, they are used in Step 3 in order to select as MWEs only the n-grams occurring with a certain frequency in the single document or in the corpus. If a domain has been chosen, the frequency at corpus level takes into account both
the current corpus from which terminology should be extracted and the pre-processed domain corpus.

As for the preference for specific key-phrases, five different degrees of preference are available, which have an impact on Step 5. More specifically, a strong preference means that the final top-ranked key-concepts will tend to be more specific, i.e. longer than those with lower relevance. This parameter activates a combination of different mechanisms, for example the final relevance score can be multiplied by the length of the key-concept in tokens, or shorter key-concepts that are stringwise included in longer ones can be put at the bottom of the rank and their relevance can be transferred to the others.

Configure and Run

<table>
<thead>
<tr>
<th>Language: english</th>
<th>Domain: environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of relevant concepts to return: 15</td>
<td></td>
</tr>
<tr>
<td>Take multword expressions that occur at least:</td>
<td></td>
</tr>
<tr>
<td>± either 2 times in a document</td>
<td></td>
</tr>
<tr>
<td>or 8 times in the corpus</td>
<td></td>
</tr>
<tr>
<td>Maximum length of multword expressions: 4</td>
<td></td>
</tr>
<tr>
<td>Prefer key-concepts occurring early in the text:</td>
<td></td>
</tr>
<tr>
<td>Prefer specific key-phrases: Medium Preference</td>
<td></td>
</tr>
<tr>
<td>Extract relevant concepts</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. Parameters of the terminology extraction component

C. Additional resources

As shown in Fig. 1, after domain-specific terminology has been extracted, the key-concepts are possibly linked with external existing resources, which are described below:

WordNet and WordNet domains: WordNet is a large lexical database of English where content words (nouns, adverbs and adjectives) are grouped into sets of synonyms called synsets, each corresponding to a concept. Synsets are connected to each other through semantic and lexical relations and build a concept taxonomy freely available in electronic format. Due to the high quality of the encoded information and to its wide coverage, WordNet has become one of the most widely used electronic resources in natural language processing applications, and has been often integrated in the ontology building process (see for example [14]). An advantage of WordNet is that a lot of information is encoded for each lemma, for example the list of synonyms sharing the same synset, as well as a textual definition (the gloss) and a set of semantic relations, for example meronymy and hyperonymy links between the synsets. All this information is extremely useful while building an ontology. The gloss, for instance, can be used to provide a textual definition of a concept, while is_a and part_of relations can be mapped onto the ontology under development. For all these reasons, after a list of relevant key-concepts has been obtained from a domain corpus, the MoKi environment allows the user to retrieve the information encoded in WordNet for each concept, if any, and to import it in the ontology. Since WordNet lists all possible senses for a given word, a disambiguation step would be required if we want to identify only the synsets related to a specific domain. Nevertheless, we do not want to integrate into our system a word sense disambiguation component that discards some senses that the user may consider relevant, or that he may want to look up anyway. So, we introduce a domain-based ranking of the synsets using WordNet Domains (hence WND), so that the synsets are displayed following a domain-specific ordering of the senses. In order to do this, we exploit the domain labels assigned to the synsets with the semi-automatic methodology described in [21]. More specifically, for each key-concept extracted with KX, we look up in WordNet if it occurs at least in one WordNet synset. Then, we check for each of the retrieved synsets the WND they belong to. For each domain configurable through the interface (e.g. environment, medicine, etc.) a list of non compatible WND has been manually defined as well as a list of the most relevant ones. If a retrieved synset belongs to a WND that has been defined as not compatible, then it is not displayed in the system output. Instead, if it occurs in the list of the most relevant WND, it is displayed at the top of the synset list. If no ontology domain has been set through the web interface, all synsets are displayed in random order. The output of this linking process is the list of synsets containing a given key-concept, ordered by relevance for the given ontology domain. Other information encoded in WordNet is also displayed, for example the gloss, the synonyms sharing the same synset of the key-concept, the synsets related through is_a relations to the current one and the label from the Suggested Upper Merged Ontology (SUMO).

Wikipedia: Since Wikipedia has become one of the largest online repositories of encyclopedic knowledge, with millions of articles available for a large number of languages (>3,600,000 for English), it would be very useful to access its information in order to integrate it into the ontology building process. In particular, its basic entries, the articles (or pages), can be seen as concepts identified by a unique reference, the url, and the fact that they are linked to each other and associated with a category provides a sort of structure, even if it is not as accurate as the WordNet hierarchy. One of the main issues related to the integration of Wikipedia-based information in such tools, however, is the need of a word sense disambiguation (WSD) step, because many concepts or senses are usually associated with a word. For example, for the term ‘Frost’ the Wikipedia disambiguation page lists two weather phenomena, one sense corresponding to the ‘Frost’ surname, four senses related to places, thirteen senses in the entertainment field and three other senses classified as ‘Other’. In our ontology building environment, we decided not to introduce a WSD step, but rather to provide the user with some information extracted from Wikipedia in order to enable him to select the most suitable information. More specifically,
we apply a WordNet-based detour to Wikipedia through BabelNet [23]. This resource is a large semantic network where lexicographic and encyclopedic knowledge from WordNet and Wikipedia have been automatically integrated. The authors report a mapping precision of 81.9%. Through this resource, we are able to display for each synset linked to a key-concept also the corresponding Wikipedia page, if it was listed in BabelNet. For Italian, we have extracted also the pages in the Italian Wikipedia corresponding to the English ones, so that both links can be displayed. A specific configuration was introduced for the PESCaDO European project\(^8\), dealing with environmental information in English, Swedish and Finnish: when the English and environment parameters are selected, the link to the Finnish and Swedish equivalent of the English Wikipage is also shown. This should enable a domain-expert that is not a native English speaker to look up information in his/her language before adding a concept into the ontology under construction. This feature can be potentially extended to any language available in Wikipedia by exploiting the outgoing links from a Wikipedia page to its correspondence in other languages. The list with the interlingual links has been extracted while developing the system, so that it is not necessary to directly access and crawl Wikipedia during the ontology building process, which would substantially slow down the whole workflow.

**External resources:** The system is able to exploit and integrate knowledge coming also from additional external resources, like dictionaries. Again, key-concepts can be looked up in an electronic dictionary and, if a match is found, the corresponding information (e.g. definition, possible translations, etc.) is displayed. For the moment, this feature is active only by selecting the English and environment parameters, and allows the user to see additional information about environment-related terms and the corresponding Swedish and Finnish translation coming from the Finnish Meteorological Dictionary\(^9\). Nevertheless, we plan to make other dictionaries available in the future also for other domains, since they only need to be encoded in a tabular format to be read and matched by the system.

Each time a key-concept extraction process is executed, the system performs an additional operation: each concept extracted from the document corpus is matched against the concepts currently defined in the ontology, in order to understand if the concept (or a synonym of it) is already defined in the ontology under development. If a matching is recognised, the key-concept is marked with a label “already defined”, thus reducing the chance that unnecessary duplicated content is added to the ontology. In the current version of the system, the matching is performed by applying normalized matching techniques on concepts (and synonyms) labels.

IV. Application scenarios

The system that we have developed has several possible applications for ontology building. More specifically, we foresee at least four possible tasks that could be accomplished during the ontology modelling phase with the help of the system.

a) **Ontology construction boosting:** This is the typical situation occurring when users want to build a domain ontology, and no structured resources are already available. In this case (see Section V-A), the system can boost the building step by reducing the human effort needed to look up domain-specific information in reference corpora and to search for additional knowledge coming from external resources.

b) **Ontology extension:** In this second application scenario, the system can be used to iteratively extend an existing ontology. In this case, the ontology and the corpus can increase in parallel. The user can add new documents to the reference corpus and iteratively use them to validate and extend the ontology. That is, the terminology extraction component is employed to monitor the ontology extension process. The matching functionality presented in Section III that automatically aligns the key-concepts extracted from a corpus against an ontology, turns out to be particularly useful in this scenario.

c) **Ontology validation:** The matching functionality proves to be very useful also to validate an already existing ontology (see Section V-B). This typically happens when an ontology is available on the web and the user needs to check if this is relevant and adequate for a given domain, in order to decide whether to re-use it. The matching functionality helps determining if the ontology uploaded in the system provides a good representation of the domain considered, at least for what concerns the terminological aspects.

d) **Ranking of ontology concepts:** Finally, another possible task to be performed by the system is the evaluation of the relevance of the concepts in an ontology. Given that an ontology can include thousands of concepts, it is not always easy to understand if some of them are more relevant than others, thus representing a sort of core knowledge of the domain ontology (an information which usually is not explicitly encoded in ontologies). Indeed, users may find this information very important in order (i) to have a better understanding of the ontology, e.g. helping them in selecting the terms from which to start inspecting more in details the ontology, or (ii) to decide, in the case of building the ontology from scratch, which part of the ontology should be first enriched with axioms and additional information. For this reason, the user can take advantage of the relevance score associated with every key-concept extracted from the domain corpus. The rank of a key-concept that has been matched with a concept in the ontology can represent the importance of such ontology concept with respect to the domain.

V. **Use Case Analysis**

A. **Use Case 1: Pollen Ontology construction boosting**

We have employed the system within the context of the PESCaDO project for building an ontology describing the classification of pollens. This ontology is one of the modules of a larger Knowledge Base used by a service-based system to provide user-oriented and user-tailored environmental information and decision support. Since to the best of our knowledge

---

\(^8\)http://www.pescado-project.eu/
\(^9\)http://mot.kielikone.fi/mot/endic/netmot.exe?UI=ened
no OWL ontologies describing pollens were available at the time of building the PESCaDO Knowledge Base, we opted to build this ontology module from scratch, applying our approach for the extraction of concepts from a domain corpus.

A crucial step in our approach is determining the domain corpus upon which to run the key-concepts extraction. Due to the requirements of the ontology to be built - a lightweight ontology describing a classification of pollens - we assembled the domain corpus with the texts extracted from the daily pollen reports available for each European country at http://polleninfo.org. We believe that daily pollen reports constitute an appropriate domain corpus for the target ontology to be built, as all pollens usually flowering in Europe are sooner or later mentioned in the bulletins.

With the help of a tool for automatically downloading and cleaning web pages [24], we collected a plain text corpus composed of 390 pollen bulletins (541,000 tokens). The key-concept extraction was performed with the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>english</td>
</tr>
<tr>
<td>Domain</td>
<td>environment</td>
</tr>
<tr>
<td>Percentage of relevant concepts to return</td>
<td>20</td>
</tr>
<tr>
<td>Take multiset expressions that occur</td>
<td></td>
</tr>
<tr>
<td>- (at least) times in a document</td>
<td>2</td>
</tr>
<tr>
<td>- or (at least) times in a corpus</td>
<td>8</td>
</tr>
<tr>
<td>Maximum length of multiset expressions</td>
<td>4</td>
</tr>
<tr>
<td>Prefer key-concepts occurring early in text</td>
<td>no</td>
</tr>
<tr>
<td>Prefer specific key-phrases</td>
<td>max. preference</td>
</tr>
</tbody>
</table>

The document threshold is set to 2 because the texts are short, consisting mainly of two-three sentences. For the same reason, it would not be useful to prefer key-concepts occurring early in text.

The system outputted 91 key-concepts. 26 of them are pollen names which we further validated by manually comparing them with the online Pollen Atlas[10]. Note that they include key-concepts up to 4 tokens, such as *oil seed rape pollen*, which are usually very difficult to extract in an automatic way. Most of the other key-concepts in the list refer to geographical names denoting the locations described in the bulletins (for example *central areas*, *switzerland*, *europe*) and to terms from the environmental domain such as *flowering*, *birch pollen loads* and *mould spore*. 38 key-concepts in the list are provided with additional information. More specifically, for 23 of them a synonym has been displayed, 15 have been associated with at least one synset and one Wikipedia page, and 3 of them were found also in the Environmental Dictionary, thus a Finnish and Swedish translation were also displayed.

Figure 4 reports the 15 top-ranked key-concepts with the corresponding relevance between parenthesis. The entries marked as “already defined” are those that have been added in the final ontology, since they are also listed in the Pollen Atlas. The concepts preceded by + were automatically provided with additional information from external resources.

B. Use Case 2: BPMN Ontology validation

With this second use case, we want to show the usefulness of our system in validating an existing ontology against some domain textual resources. We considered the BPMN Ontology[25], an ontology formalising version 1.1 of the Business Process Modelling Notation (BPMN), a reference language to formally describe business processes. The ontology, which contains 117 concepts, has been built from scratch by manually encoding the information described in the BPMN v1.1 Specification [26], the document which defines all the elements of the language and their properties.

The system outputted 91 key-concepts. 26 of them are pollen names which we further validated by manually comparing them with the online Pollen Atlas[10]. Note that they include key-concepts up to 4 tokens, such as *oil seed rape pollen*, which are usually very difficult to extract in an automatic way. Most of the other key-concepts in the list refer to geographical names denoting the locations described in the bulletins (for example *central areas*, *switzerland*, *europe*) and to terms from the environmental domain such as *flowering*, *birch pollen loads* and *mould spore*. 38 key-concepts in the list are provided with additional information. More specifically, for 23 of them a synonym has been displayed, 15 have been associated with at least one synset and one Wikipedia page, and 3 of them were found also in the Environmental Dictionary, thus a Finnish and Swedish translation were also displayed.

Figure 4 reports the 15 top-ranked key-concepts with the corresponding relevance between parenthesis. The entries marked as “already defined” are those that have been added in the final ontology, since they are also listed in the Pollen Atlas. The concepts preceded by + were automatically provided with additional information from external resources.

In the experiment, we uploaded in MoKi the BPMN ontology, and we performed the key-concepts extraction on

---

the BPMN v1.1 Specification Document. Figure 5 shows a screenshot of the top 20 key-concepts returned by the system.

With the help of the matching functionality of the system, we compared the number of key-concept that were already defined in the ontology, with respect to the top 10, 20, 50 and 100 key-concepts returned by the system. The results are reported in Table I.

<table>
<thead>
<tr>
<th>number of top key-concepts considered</th>
<th>concepts already in the ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
</tr>
<tr>
<td>50</td>
<td>26</td>
</tr>
<tr>
<td>100</td>
<td>43</td>
</tr>
</tbody>
</table>

**Table I**

**NUMBER OF EXTRACTED KEY-CONCEPTS ALREADY DEFINED IN THE ONTOLOGY, WITH RESPECT TO THE TOP 10, 20, 50 AND 100 KEY-CONCEPTS RETURNED BY THE SYSTEM.**

VI. CONCLUSIONS

We presented a novel, Wiki-based system for semi-automatic ontology building and extension. The system can be easily accessed via web browser and different users have the possibility to work at the same ontology building process. This means that experts from the ontology engineering field can easily cooperate with experts in specific domains to tailor the ontology to complex requirements. The platform includes a terminology extraction component that, starting from a domain corpus, provides the user with a ranked list of candidate concepts. The user has the possibility to easily integrate them in an ontology after a manual validation, in which additional information automatically retrieved from external resources can be consulted. Such information is derived from structured and semi-structured resources, and can be partially displayed in different languages. Furthermore, the interface supports the development of English and Italian-based ontologies.

The system has been successfully tested in the extension of an environmental ontology, providing a valuable support during the integration of pollen information. It was also employed to effectively validate the BPMN Ontology by matching it against the key-concepts extracted from the BPMN Specification Document.

In the future, we plan to extend the platform by providing more available domains through the interface. Also, we will encourage different user groups to test the system in order to collect their feedback and improve on the system usability.

ACKNOWLEDGMENT

The work described in this paper has been partially funded by the European Commission under the contract number FP7-248594, PESCaDO project.

REFERENCES


