

KNOWledge in the MAKing in European Society

Report on Ontologies and Tagging

*September 2018*

*Diana Maynard, Johann Petrak, Xingyi Song and Adam Funk (USFD)*



**DELIVERABLE INFORMATION**

|  |  |
| --- | --- |
| **Name of the deliverable** | Report on Ontologies and Tagging |
| **Number of the deliverable** | D2.4 (preliminary) |
| **Related WP number and name** | WP2 Ontologies |
| **Related task number and name** | T2.2 Methodology for ontology creation and semi-automatic tagging |
| **Deliverable dissemination level** | PU |
| **Deliverable due date** | 30 October 2018 |
|  |  |
| **Main author(s)** | Diana Maynard |
| **Contributing partners** | USFD |

**REVISIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Comments (Y/N)** | **Project partner** |
| **1** |  |  |  |

The sole responsibility for the content of this document lies with the authors. It does not necessarily reflect the opinion of the European Union. The European Commission is not responsible for any use that may be made of the information contained therein.

# Table of contents

[Table of contents 3](#_Toc527711586)

[1 Introduction 4](#_Toc527711587)

[2 Ontology structure 5](#_Toc527711588)

[3 Ontology population 6](#_Toc527711589)

[3.1 Improving the initial keyword generation 6](#_Toc527711590)

[3.2 Keyword enrichment 6](#_Toc527711591)

[3.2.1 Corpus pre-processing 7](#_Toc527711592)

[3.2.2 Training the embeddings 8](#_Toc527711593)

[3.2.3 Scoring the embeddings 8](#_Toc527711594)

[4 Classification 9](#_Toc527711595)

[4.1 Scoring process 9](#_Toc527711596)

[4.2 The classification tool 10](#_Toc527711597)

[5 Topic modelling-based classification 13](#_Toc527711598)

[6 Next steps 14](#_Toc527711599)

[7 References 15](#_Toc527711600)

# Introduction

This document is a preliminary version of D2.4, which updates D2.2, and describes the work on ontologies and annotation since the first versions of the ontology. The full version of D2.4 will be produced towards the end of the project, when the final version of the ontology is released.

In the first version of the ontology, the design of the ontology structure was described, and preliminary population of the ontology with keywords related to the topics in the ontology was carried out. This enabled preliminary tagging of a set of publications, patents and projects with topics, by means of the keywords and a weighting mechanism. The extension to this deliverable described some improvements to the initial ontology population and tagging methodologies.

In this document, we describe substantial progress in (1) refining the ontology structure (classes and subclasses, corresponding to topics); (2) improving the mechanisms for keyword generation; and (3) scoring the documents during the classification process. We also outline our plans for the remaining work to be done in the project. Improvements to the ontology have been made based on a continuous cycle of update and testing on the documents in our database. In practical terms, the resulting classification of documents is much improved. In technical and methodological terms, we have developed new NLP techniques for generation and scoring of keywords, and have demonstrated that the combination of NLP, deep learning and ontologies can enhance standard classification-based approaches typical of the STI field.

# Ontology structure

Annotation with the second version of the ontology produced much higher quality results already. However, it became clear that some ontology classes were problematic. This was largely due to the starting point of existing mapping schemes and the Nature.com ontology, which in some cases were flawed, and in other cases just did not fit our structure properly. We identified several particular issues.

First, some subclasses did not really fit the higher level class description (such as security, which is specific to public security and safety, but had subclasses that were less relevant). Second, there was some confusion between related high level classes such as Energy and Climate, and some reorganisation was necessary.

# Ontology population

Our experience with annotating the documents in our collection with version 2 of the ontology showed substantial improvement over the first version, but still some issues remained. We have pursued two strands of work: first, on improving the initial keyword generation, and second, on specifically improving the methodology for creating the enriched keywords.

## Improving the initial keyword generation

Our analysis showed problems with certain topics that had too few, too many, or badly chosen keywords. We have worked on mitigating this in the following ways.

First, analysis of the distribution of documents to topics showed that some classes had very low numbers of documents assigned to them. Investigation showed that this was primarily due to either just low numbers of keywords, or to poor choice of keywords. This was particularly the case with topics such as “social inequality”, where the keywords were mostly too general to be useful. To resolve this, we manually reviewed the seed keywords for these classes, and added some new ones which were more targeted. This meant that in the enrichment process, a better set of additional keywords was then produced.

Second, the distribution analysis also showed that some classes had very high numbers of documents assigned to them, e.g. “public engagement”. Investigation showed that some very general keywords were produced by the enrichment process. To resolve this, we reviewed the results and added a blacklist of terms that should not be generated by the enrichment (or indeed, at any other keyword generation stage).

## Keyword enrichment

We have also improved the generation of the enriched keywords. The basic workflow for this is as follows:

1. Generate a set of seed keywords associated with each ontology class
2. Extend these keywords by finding semantically similar terms in a large corpus, using word embeddings trained on that corpus (extract a set of terms, then find the ones most similar to seeds)
3. Score the keywords according to how representative they are of that class
4. Generate prior probabilities using PMI for term combinations, based on frequency of co-occurrence in the training data (this is used later in the classification tool – see Section 4).

Due to the low quality of some of the keywords generated in the first stage, we revisited this process. As before, we now automatically generate keywords from class names, descriptions, and related information (e.g. DBpedia, skos, etc.) using term recognition tools, but these are now separated into:

* *preferred* (e.g. coming directly from class names or other “good” sources)
* *generated* (e.g. coming from descriptions – might not be so high quality).

Only preferred terms are used for the enrichment process, and they also get a higher weighting at document classification time (see Section 4). Figure 1 shows an example of a class description for the topic “smart cities and communities”, where relevant terms have been extracted automatically (highlighted in yellow) by NLP tools. These would be classified as “generated”, while terms derived from the class name itself (e.g. “smart cities”, “smart cities and communities”), would be classified as “preferred”.

The basic idea is thus to:

Figure 1: Keywords extracted automatically from a class description

**Sustainable development of urban areas** is a challenge of key importance. It requires new, efficient, and **user-friendly technologies** and services, in particular in the areas of **energy**, **transport** and **ICT**. However, these solutions need integrated approaches, both in terms of research and development of advanced technological solutions, as well as deployment. The focus on **smart cities technologies** will result in commercial-scale solutions with a high market potential.

* create a set of additional keywords for each class in the ontology using an automatic unsupervised approach;
* create a large corpus of patent, publication and project abstracts as well as relevant policy documents;
* extract new candidate terms from this corpus;
* train domain-specific word embeddings for these terms, in such as way that we can have vectors also for multi-word terms;
* use the embeddings to find the similarity between the seed terms and the new terms;
* use the similarity to decide which new terms to keep, and which concept to map them to.

The following steps are taken for the enrichment process:

### Corpus pre-processing

Our corpus consists of 2.6 million documents in total, comprising project, patent and publication abstracts, and a set of policy documents. Pre-processing consists of the following steps:

1. Run a GATE application for linguistic pre-processing, which consists of POS tagging, lemmatisation, entity finding, etc. This is used to find (1) all occurrences of original ontology keywords in corpus (both of which are lemmatised), and (2) single and multi-word term candidates in the corpus (filtering out any Named Entities (e.g. names of people, places etc.)
2. Merge the ontology matches and the term candidate, and create (potentially overlapping) keyword candidates
3. Calculate the canonical lemmatized string for these candidates
4. Calculate term statistics for all term candidates (using tf, df, idf)

This results in a set of 1.2 million keyword candidates in 180 million locations in the corpus.

### Training the embeddings

This step generates embeddings (vector representations from our keyword candidates and corpus. The following steps are undertaken:

1. calculate a set of 330 stopwords (also used for scoring later on) and a set of unique multi-word terms from the original ontology keywords;
2. train the embeddings using Python GenSim, removing stopwords and single-letter words from the corpus
3. Use sentences as training examples, generated in the following way:
   * match each sentence against the list of multi-word terms;
   * for each multi-word term, create one sentence where each lexical unit is a separate multi-word term;
   * create one sentence also where all multi-word terms are single lexical units.

This process results in the generation of 591,526 embeddings.

### Scoring the embeddings

We have investigated various ways of calculating embeddings to represent ontology topics and measuring similarity between the keyword and class. Best results have been achieved so far with a method we term *centrboth*. For each class, we calculate the average embedding for the set of preferred terms, and another average embedding for the set of non-preferred terms related to the class. The final embedding is the weighted average of both.

We then use a method we term *simonly*. This is the 0/1 normalised cosine similarity between the embeddings representing the ontology class (*centrboth*) calculated in the previous step, and the embedding representing the candidate term. In both cases for *simonly*, we take the unweighted average, since using the weighted (tf, idf) average did not work well in early experiments.

# Classification

The annotation tool classifies documents according to the best matching topics from the ontology. Each topic is matched based on a number of keywords, but a complex process defines how to rate the “quality” of these keywords (how well they indicate a particular topic) and how to combine the various keyword scores for each topic. This has been significantly enhanced since D2.3. For example, a single keyword might be relevant for more than one topic, but it might be more relevant for one topic than another, so it would get a higher score. At the classification stage, all relevant keywords and topics are scored, resulting in a list of topics and scores, together with the keywords they comprise (and their scores). At a later stage, cut-off thresholds are established for each document type, so that only the highest and most relevant topics will be added to the database for that document.

## Scoring process

The original scoring process was based on the number of matching keywords in a document and topic, normalised by document length, with some additional weighting for longer terms (which are thought to be more specific). We have enhanced the scoring process in a number of ways.

First, we deal with the problem of ambiguity. Some keywords are good indicators of a topic only when they appear in the same document as another keyword. For example, “packaging” could relate to many topics, but if it appears in the same document as the term “microelectronics” (not just as a multi-word term, i.e. “microelectronics packaging”) it can be considered to be a good indicator of the topic of MNE (micro- and nano-electronics). We therefore want to weight more strongly terms that appear together with some specific other terms. The question is then how to find which are these “specific other terms”.

PMI (Pointwise Mutual Information) is an indicator of lexical cohesion which considers terms to be more closely related the more often they occur together in a large training corpus. We therefore use this to “boost” the score of certain keyword pairs that occur together in a document – terms with high PMI will be more strongly boosted. To do this, we pre-calculate on our training corpus pairwise collocation statistics for all term candidates. We then select only those where the normalised PMI value >0 and the minimum occurrence frequency is 20 (based on heuristic experimentation). This gives us 309,932 pairs from an initial 2.8 million pairs.

From our enrichment process, each keyword already has a keyword score (kw). In the document classification stage, we generate a base score from this keyword score, as follows:

* kw score is multiplied by 2 if it fulfils certain criteria (e.g. a patent classification keyword is matched in a patent document, or a project classification keyword is matched in a project document)
* kw score is multiplied by 1.1 if it is a preferred term

Next, the base score is boosted by 0.5 \* the score of the highest scoring direct superclass (if any). This accounts for the fact that a superclass is less specific a term, but is still relevant, so we want to boost the more specific terms over the more general ones so that we annotate the document at the most specific level possible.

Next, we integrate the PMI boosting. All keywords for a document are looked up in the matching pairs PMI table generated previously. We use the highest value of any matching pair, and boost the score by 1+PMI value.

We finally generate two scores for each topic:

* unboosted: 100\*base score / doc length
* boosted: 100\*base score + PMIboost / doc length

Both scores are generated because we want to compare which version is better. Current experiments have shown that the boosted score tends to be better.

## The classification tool

Separate documentation for users of the classification tool is attached to this deliverable. This explains the technical details of how to run the web service. In summary, the software provides a REST service on the USFD servers which accepts documents, classifies them according to the topics in the ontology, and returns classification and keyword information in JSON. This information is fed back into the KNOWMAK database.

Since the previous version of the classification tool, some improvements have been made – mainly in terms of providing additional output. Instead of producing a single score for a topic, multiple scores have been produced so that the different mechanisms can be evaluated. These match the different scores described in the previous section: standard score and boosted score, each with or without PMI boosting. For the boosted score, the URI of the topic which boosted it is also given. This also means that if necessary, different scoring mechanisms can be used for different kinds of documents.

An example of the output (in JSON format) is shown in Figure 2. This can be interpreted as follows:

* **Classification** shows the topic URL (in this case *antibiotics*).
* **Boosted by** shows the topic that boosted the score (in this case, *antimicrobials*). This means that keywords were found for both these two topics, but since *antimicrobials* is a superclass (more general) than *antibiotics*, the latter gets a score boost from the former.
* **Keywords** shows the relevant keywords found in the document that match this topic (in this case, *antibiotics* and *bacteria*).
* For each keyword, there are features **kind** and **score**
  + **Kind** shows the provenance of the keyword (was it automatically **generated** by the NLP tools, was it a **preferred** term (generated directly from a topic name, or manually added, and thus thought to be highly correct), or an **enriched** term (generated via the keyword enrichment techniques). This is useful to understand how a keyword was generated in case of a bad match (or indeed, a good one).
  + **Score** shows the score for that keyword, based on the scoring procedure described previously.
* **Score** (for the topic) shows the aggregated score for all keywords matching that topic, including the boosting process.
* **topicID** shows the number of the topic in the ontology, which is later added to the database (rather than the topic name) along with the document, once cutoff thresholds have been applied.
* **Unboosted score** (for the topic) shows the aggregated score for all keywords matching that topic, without the boosting process.

## Topic assignment

The annotation tool assigns to each document as many topics as it finds matches for, without making any decision about which ones are valid. For example, some topics might have a very low score, and are clearly not relevant, but are not excluded at that stage. Instead, a further topic assignment stage is defined after the annotation. The reason for this is that this strategy might vary for different kinds of document. Furthermore, the strategies are based on an analysis of the entire set of annotated documents, and thus cannot be done at annotation time where each document is processed individually. We describe in this section the approaches for deciding on topic assignment. The strategies were decided by manually testing and verifying a variety of strategies. We plan to harmonise these as much as possible in the next stage of the project.

### Projects

To decide on a strategy for accepting topic assignments to projects, a first analysis was made of a selection of projects, to get a feel for trouble spots and possible thresholds. The main conclusions were:

1. Contributions from both parent-class boosting and corpus-based PMI were positive, so that was the score used.
2. Boosting could produce a high-scoring class whose score was predominately from the parent class. These classes did not seem to be useful, so it was decided to eliminate all classes where the boosted score was more than double the unboosted score. For this, the scores without the PMI contribution were used for comparing the size of the boost.
3. Both an absolute threshold and some form of project-specific relative threshold were deemed necessary. The absolute threshold was decided at 4.5, and all class assignments with lower scores were eliminated. The relative threshold needs to be fairly high; thus this was decided as 0.8 times the maximum score assigned to the project, with all lower-scoring classes eliminated.

 The relative threshold is thus chosen for the final assignment, with no top-N limit (i.e. no maximum absolute number of topics).

### Publications

The current strategy for accepting topic assignment to publications is as follows:

1. All articles and reviews in WoS between 2010-2016 were annotated with the classification service.

2. The boosted score including PMI was taken for each class of a publication (since this was deemed to be the best)  
3. The median score per class was calculated, and used as a threshold  
4. For each publication, only scores above the median were selected  
5. Out of these, only the highest scoring class for each publication was selected

### Patents

Since the first version of the ontology and annotation tool, the strategy for patents has changed somewhat.

1. Class-patent allocations are only retained if they are attributed to patents with a boosted score including PMIwhich is higher than the average of this score for all annotated patents for this class. (note that previously, median was used instead of average).
2. Intra-class analysis of the keywords for each patent was performed, in order to select only classes which have more than one keyword. The idea of this step is to select classes only if they are built on **co-occurrences** (it does not matter where these co-occurrences are: titles, abstracts, or IPC descriptions). The idea behind this is that a combination of two keywords is a semantic unit, which should make the classes more relevant for a given patent. This is essentially similar to what the PMI boost does, but provides additional weeding out of documents which have a single high-scoring keyword (which could be erroneous). It is possible that this step will not be needed if the initial scoring mechanism and/or keyword assignment is improved sufficiently.

### Harmonisation

The strategies for each document type have so far been determined independently. While it may be that different strategies are required for different document types, it would nevertheless be ideal to harmonise these as much as possible. For example, the additional keyword restriction employed for patents might be useful for the other document types also. There is some discrepancy between the use of only the highest scoring class for publications, with the strategy of having no absolute limit on number of classes for projects and patents. This is worth investigating further. However, the planned improvements to both the ontology structure and the keyword generation may require further changes to the thresholding strategies of all document types, so work on this will only take place after the next round of annotation in 2019.

Figure 2: Example of JSON output from the classifier

{

"classification": {

"http://www.gate.ac.uk/ns/ontologies/knowmak/antibiotics": {

"boostedBy": "http://www.gate.ac.uk/ns/ontologies/knowmak/antimicrobials",

"keywords": {

"antibiotics": {

"kinds": [ "generated", "preferred" ],

"score": 1.1527377521613833

},

"bacteria": {

"kinds": ["generated"],

"score": 0.5763688760806917

... },

"score": [ 4.322766570605188, 4.4159785333 ],

"topicID": "38",

"unboostedScore": [ 2.5936599423631126, 3.75354899915 ],

},

"http://www.gate.ac.uk/ns/ontologies/knowmak/antimicrobial\_resistance": {

"boostedBy": "http://www.gate.ac.uk/ns/ontologies/knowmak/antimicrobials",

"keywords": {

...

} },

"score": [ 8.069164265129682, 9.12545454545 ],

"topicID": "42",

"unboostedScore": [ 6.340057636887607, 7.35454545454 ]

},

"http://www.gate.ac.uk/ns/ontologies/knowmak/antimicrobials": {

"keywords": {

...

} },

"score": [ 3.4582132564841506, 4.54545452388 ],

"topicID": "43",

"unboostedScore": [ 3.4582132564841506, 4.54545452388 ],

}, },

"doc\_type": "publication",

"doc\_type\_applied": "publication",

"error": "\_none\_",

"identifier": "12348874",

"internalID": "4dec1ce0-cead-4a94-b16c-6fada1a26f49"

}

# Topic modelling-based classification

In addition to the keyword-based classification approach, we are also experimenting with an alternative approach based on topic modelling in order to rank the topics according to similarity with each document. We will then compare the approaches to see which is the best.

Latent Dirichlet allocation (LDA) is the most widely used topic modelling algorithm in natural language processing. LDA assumes that the document is a consistent mixture of *k* topics. Therefore, we can use LDA to classify the input documents by comparing the similarity between the probability distribution of the input document topics to the distribution of the ontology class topics.

The current approach includes the following steps:

At training time, we:

1. train an LDA model (*n* topics) based on relevant training documents;
2. find the most relevant Wikipedia page as ontology class documents;
3. calculate the topic distributions for each class, based on the class document and trained model.

To train the model, we use a collection of abstracts from projects, patents, and publications, as well as policy documents (the same collection as used for the embeddings training). The training documents are preprocessed by removing stop words and stemming.  To find the most relevant Wikipedia page, we first use the class label as a search query, and score the Wikipedia page with gestalt pattern matching between class label and the Wikipedia page. If the matching score is less than 0.7, then we reset the search queries as keywords, and use the most relevant page. Once we have the trained topic model and class documents, we can then calculate the distribution on each topic of the document, and this will return *n* element vectors for each class.

At application time, we:

1. calculate the input document topic distributions using the trained model;
2. calculate the cosine similarity between ontology class and document topic distributions;
3. return *k* most similar ontology class ids.

 The development is still underway, but in future work, we plan to:

1. apply the ontology class keywords as a guide to train the LDA topic model;
2. improve the ontology class documents by:
   1. improving the string match algorithm;
   2. searching on a different database besides Wikipedia;
   3. combine the topic modelling algorithm with the existing classification algorithm.

# Next steps

Although first results are very reasonable, there are a number of issues still to be dealt with in the project.

First, we are continuing to refine the ontology structure according to the needs of the users. The ontology will be reduced to only two levels, in order to have a simpler and more communicable structure. Note that the KNOWMAK tool already only shows indicators at two levels – numbers of documents assigned to topics at deeper levels are aggregated at the higher level.

The new ontology structure will be tested with users in RISIS2 in order to get feedback on the structure and a mechanism could be set up whereby users can suggest potential new keywords. Ongoing until June, milestone in January (new version of ontology) and June.

Second, we are currently experimenting with an alternative way of annotating documents with ontology topics, as described in Section 5. This is ongoing experimental work which we have not yet integrated into the classification tool. Experimentation to understand if this approach is feasible, and how it can be integrated with the existing keyword-based classification approach, is still to be done.

Third, we are working on ways to improve the keyword enrichment process. Alongside minor tweaks to the methodologies, we are also working on incorporating a larger training set which could make significant improvements to both this and to the topic-based classification approach. This set will consist of a larger representative sample of publications, projects, and patents related to Europe and covering the period 2000-2014.

Fourth, we are currently testing also the annotation of the social innovation documents against the ontology. These differ substantially from the existing document types, and so there may be new issues that arise after this is assessed. A first round of annotation is currently taking place, following which further improvements will be made to the ontology and potentially also to the annotation tool and scoring process.

Last, but not least, one of the most important things currently lacking is a proper evaluation of the classification process. This is scheduled for 2019, after the next version of the ontology is finished and re-annotation has been performed.

UFSD will summarize the information on the assignment process and a sykpe will be organized to harmonize and make more transparent the assignment.

By mid-November: annotation of social innovation as an experiment in order to fine-tune classes and keywords for social innovation.

New structure of the ontology delivered by end of 2018, re-annotation of the whole corpus will be done in January. Including social innovation as well. Then data will be sent again to the central database.

UFSD will work in the meantime on the evaluation process for the ontology and circulate preliminary documents.

Workshop end of January/early February to discuss the evaluation procedure.

Final annotation planned in June 2019 in order to finalize the tool by September.

A timeline is shown below for the next steps.

Evaluation: February 2019 (2 months)

Refinement of the Ontology: October-December

New version of the ontology and annotation: January

Preliminary annotation of Social Innovation: November

Ontology improvements based on feedback from social innovation: December

Beginning of November: larger training dataset to be created, experiments with retraining enrichment

June 2019: Final annotation round

# References

1. Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T., 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
2. Gorrell, G., Petrak, J. and Bontcheva, K. "Using @Twitter Conventions to Improve #LOD-based Named Entity Disambiguation." In The Semantic Web. Latest Advances and New Domains, pp. 171-186. Springer International Publishing, 2015
3. Li, G. and Wang, H., 2014. Improved Automatic Keyword Extraction Based on TextRank Using Domain Knowledge. In *Natural Language Processing and Chinese Computing* (pp. 403-413). Springer, Berlin, Heidelberg.
4. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).
5. Pennington, J., Socher, R. and Manning, C., 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*(pp. 1532-1543).
6. Zhang, Z., Petrak, J., Maynard, D. (2018). Adapted TextRank for Term Extraction: Improving Performance of Any Automatic Term Extraction Algorithms. Submitted to the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Melbourne, Australia, July 2018.