DG EMPL

Framework Contract Ref.: VC/2014/0246

Service Order Number: VC/2014/0511

ESCO Mapping Pilot Phase

DLV008 – Final Report Draft

|  |  |
| --- | --- |
| ***Brussels*** |  |
| ***02/09/2014*** |  |
| ***ESCO Mapping Pilot Phase*** |  |
| ***Version: 0.1*** |  |
| **Final Report Draft** |  |
|  |  |
|  |  |

Table Of Contents

[1 Introduction 4](#_Toc397419360)

[1.1 Purpose of the document 4](#_Toc397419361)

[1.2 Intended audience 4](#_Toc397419362)

[2 Introduction to the mapping pilot 5](#_Toc397419363)

[3 National classifications to SKOS 6](#_Toc397419364)

[4 Example Scenarios 7](#_Toc397419365)

[4.1 Cases created by the contractor 7](#_Toc397419366)

[4.2 Cases provided by the PES 7](#_Toc397419367)

[5 Creating the MApping 8](#_Toc397419368)

[5.1 Available tools for the mapping 8](#_Toc397419369)

[5.2 Methodology 9](#_Toc397419370)

[5.3 Preparing the data for mapping 9](#_Toc397419371)

[5.4 Alignment rules 9](#_Toc397419372)

[5.5 Manual Corrections 9](#_Toc397419373)

[5.6 Creating the final mapping 9](#_Toc397419374)

[6 Evaluating the Mapping 10](#_Toc397419375)

# Introduction

## Purpose of the document

The purpose of the Final Report Draft is to provide a status of the ESCO Mapping Phase As the project progresses. The document will be updated as the different stages in the project are completed.

## Intended audience

The present document is intended to be read by the following people:

* ESCO Secretariat
* Delegates of the Public Employment Services that participate in the ESCO Mapping Pilot.
* TenForce staff that provide support to DG EMPL

# Introduction to the mapping pilot

TODO

# National classifications to SKOS

# Example Scenarios

The mappings between the NOC and ESCO need to be tested on real world data. To that end job vacancies (JV) and curricula vitae (CV) were tagged with ESCO terms. A first exercise was done by the contractor to discover the possible problems in this area as soon as possible. Later, real life cases were requested from the PESs as well.

Currently only TenForce CV/JV have received ESCO tagging, documents from the AGRI, VET and HOSPI sectors have been tagged. The TenForce CV and JV have been translated into Dutch, Spanish and Czech. French translations are still to be done. The translated cases have not been tagged with NOC terms yet, as it would be better to wait for the actual content from the PESs and tag those with ESCO and the relevant NOC.

## Cases created by the contractor

CVs and JVs were collected from [www.indeed.com](http://www.indeed.com). This site was chosen because it is freely available, has both CVs and job vacancies available and is being used in different countries in Europe. CVs and JVs were collected for:

* Hospitality and tourism
  + picked 'tour leader' as the example occupation
  + searched for cook resume, found terrible quality and highly mixed CVs. Examples available. It is not easy to find the 'upper end' CVs with little information that we thought would exist.
* Agriculture/Forestry and Fishery
  + terribly hard to find good resumes in Flanders/the Netherlands, it is just too small.
  + good CV/JV found for fisherman in the UK/US
* Veterinary
  + found good CV/JV for veterinary nurse

The CVs often contain all the experience a person had previously. This means they are often a mix of many different occupations, and not necessarily only the occupation the person is looking for.

We expect there to be a bell-curve that relates CV quality to education level of the writer. Low education and very high education will both generate low quality CVs.

### Tagging process

Tagging of CVs and JVs is currently a manual process. It is supported by an open source tool that is being developed (https://github.com/Rahien/RDFazer) that accepts an HTML file and that allows the user to select parts of text and tag it with elements from a taxonomy. These tags are inserted in the HTML file as RDFa. In this way, the document can be understood by both human readers and machines. To human users, the concepts show up as hyperlinks to the concept on the ESCO portal page. A machine receives a set of triples that link the pieces of text to ESCO URIs.

Because a single tag can refer to multiple concepts, every tag is represented as a blank node (a ‘highlight’) that links to the URIs that are tagged.

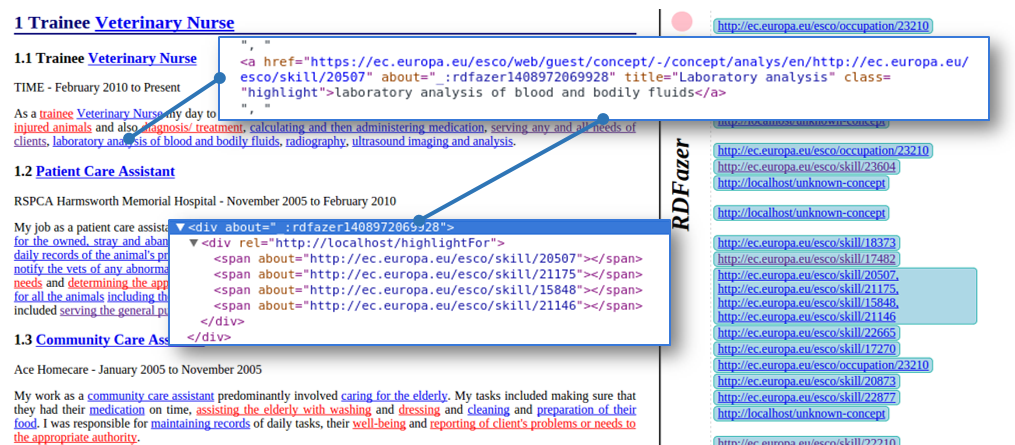


Figure 1: Tagged CV of a Veterinary Nurse

The process of finding the concepts that are related to the term is currently fully manual, the tool does not support search yet. This means that to tag the CV/JVs with ESCO terms, look-ups were done on the ESCO portal and the concepts were linked manually. However the developer of the tool is confident that he can add a connection with a SPARQL endpoint to allow searching for concepts in any endpoint.

Tagging documents with machine readable data is being used to enhance search results on search engines. An example is the google search engine. In the image below, information on IMDB regarding the cast of a movie is included as Microdata and then used in the google search results of this movie. Microdata is an alternative format for inserting machine readable content in a web page.

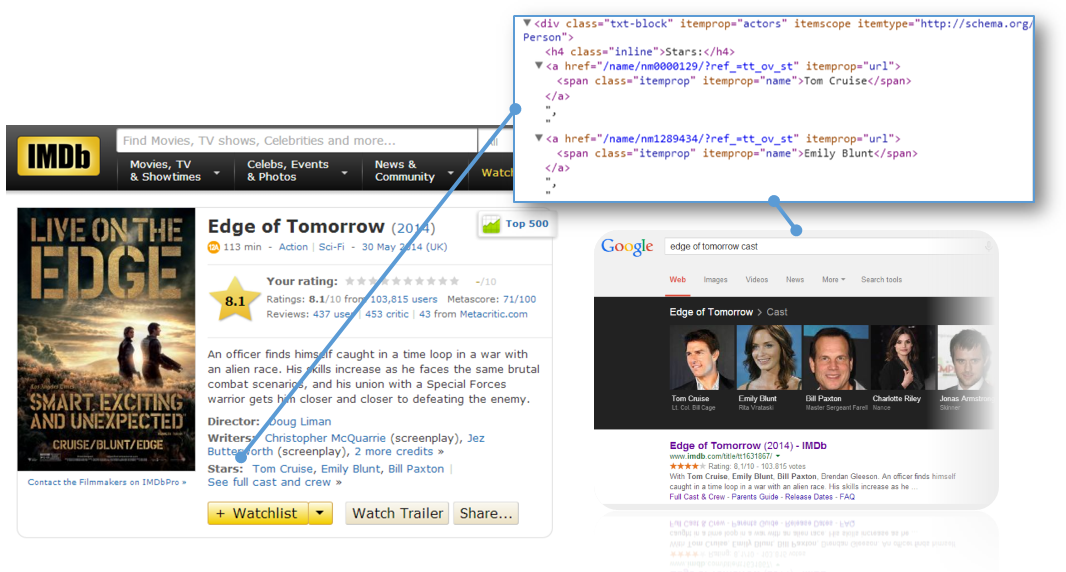


Figure 2: Use of Microdata in search engines

### Tagging with ESCOv0

During the Tagging process with ESCOv0 terminology, the following observations were made:

#### No definition in ESCOv0

ESCOv0 concepts do not have a definition or scope note. Therefore, it can be difficult to distinguish the intended meaning of terms. For instance, the skill 'Filming (video production)' could mean actually taking a camera and shooting a video, or it could mean taking the role of producer in making a movie. Only the link with the occupations can tell us that it is in fact the first.

#### Level of detail

The level of detail in ESCOv0 is heterogeneous. Sometimes, there is only a general skill available, that does not allow you to cover the finer detail expressed in a job posting (and presumably required by the job market). An example is 'Foster social networks'. Other skills are much too detailed. The best example that I could find was 'Rifle stock knurling'. This makes tagging CVs and JVs harder in the sense that it is difficult to find the ESCOv0 concept that matches best.

This is especially problematic when only the more specific term is available, for instance in the context of a 'veterinary nurse', the skill 'radiography' had to be tagged, but the only possible match in ESCO v0 was 'Radiography (dental nursery)'.

Currently, tagging with skill groups has not been done, but could be useful to reduce this problem, e.g. the words 'team work' could refer to the skill group 'work as part of a team'. In fact during the mapping pilot, we can consider mapping skills and skill groups as one, skill groups can then be seen simply as more general skills.

#### Some sectors are poorly covered in ESCO v0

Some sectors have very poor coverage in ESCO v0, for example the Veterinary sector. Many medical treatments are not split between animals and humans. This may give confusing results. E.g. 'Operating theatre assistance'. In this case, the assumption was made that if a concept does not specify the context, it can be applied in any context.

#### Qualifications are very poor in ESCO v0

Tagging qualifications and education in CVs and/or JVs is practically impossible, given the very small amount of Qualifications that are available in ESCO.

#### Concepts Impossible to tag with ESCO

The following elements could not be tagged with ESCO v0 concepts:

* Wages/Advantages
* Experience
* Many qualifications/education
* Projects that were taken part in
* Tasks, though these can mostly be transformed to skills

## Cases provided by the PES

Currently, we have received Job Seeker Scenarios (JSS) and Job Vacancy Scenarios (JVS) from the Spanish PES, the French PES and the Dutch PES. The JSS and JVS from the Spanish PES have been translated into Dutch, English, Spanish and Czech. The French translations are still pending. For the Dutch scenarios, the translation to English was started.

### Dutch Scenarios

The Dutch Scenarios describe:

* JSS1: Gastvrouw horeca (hospitality hostess), this includes the orinal resume of the client
* JSS2: Cafetaria medewerker (employee in a cafeteria)
* JSS3: Reisadviseur (travel consultant)
* JSS4: Hostess tourisme (tourism hostess)
* JVS1: Medewerker bediening (waiter/waitress)
* JVS2: Reisadviseur (travel advisor)

The JSS and the JVS are in the form of excel files that contain the output of a computer input form (e.g. a form on a webpage).

### French Scenarios

The French Scenarios describe:

* Somelier (wine waiter)
  + 2 JVS as unstructured information
  + 2 JVS as structured information from the Pole Emploi systems
  + 2 JSS both unstructured CVs
  + The ‘fiche metier’ Sommelerie from Rome
* Assistance de direction d'hôtel-restaurant (assistant hotel/restaurant manager)
  + 3 JSS scenarios in the form or emails exchanged with the client
  + 2 JVS as unstructured information
  + 2 JVS as structured information from the Pole Emploi systems

The JSS and JVS are mostly unstructured text documents. The documents

### Spanish Scenarios

The Spanish Job Seeker Scenarios describe:

* Cocineros (cook)
* Recepcionistas de hotel (Hotel receptionist)
* Cocineros and Ayudantes de cocina / Comida rápida (cook, kitchen assistant/fast food cook)
* Agencia de viajes and Recepcionistas de hotel (travel agent, hotel receptionist)
* Cocineros, Ayudantes de cocina / Comida rápida, Recepcionistas de hotel, Camareros (Cook, Kitchen assistant / fast food cook, Hotel receptionist, Waiter)
* Animación turística, Recepcionistas de hotel (tourist animator, hotel receptionist)
* Recepcionistas de hotel (hotel receptionist)
* Recepcionistas de hotel, Agencia de viajes, Camareros, Cocineros, Animación turística, Ayudantes de cocina / Comida rápida (hotel receptionist, travel agent, waiter, tourist animator, kitchen assistant/fast food cook)
* Hostelería / Turismo (hospitality/tourism)

The Spanish Job Vacancy Scenarios describe:

* Recepcionistas De Hotel (Hotel Receptionist), 2 times
* COCINEROS (cook), 3 times
* CAMAREROS (waiter), 4 times

The JSS and JVS are all the results of online input forms.

# Creating the Mapping

This section describes how the mapping between ESCO and the PES was created. The mapping was done automatically, with a manual review step. In this document, the tools that were researched for the mapping are described, then a general methodology for mapping taxonomies to ESCO is proposed, and finally the steps in the methodology are examined one by one.

## Available tools for the mapping

In the context of the ESCO Mapping Pilot the following taxonomy alignment tools were examined:

* Amalgame
* ITM-Match
* Silk Alignment Framework

The research into alignment tools follows a depth first approach. This means that while all tools were examined on a basic level first, the focus was then put more on Silk to run into all possible problems as soon as possible and thus get early results.

### Amalgame

Amalgame (AMsterdam ALignment GenerAtion MEtatool) is a plugin for the Cliopatria environment and runs in an SWI-Prolog environment. They showcase an interface that allows the easy and comprehensible combination of elements of alignment rules. The user interface of Amalgame is easily the most configurable of the examined tools. However, the SWI-prolog environment may present a problem for deployment at EC premises. Furthermore, Amalgame requires that the taxonomies to be aligned are stored in a local store in the Cliopatria environment, it does not allow communication with remote triple stores.

Amalgame is open source and is available from <https://github.com/jrvosse/amalgame.git>. Its license is currently still unclear.

The manual correction interface of Amalgame has not yet been investigated.

Currently, this tool was not yet tested on actual data to be aligned.

### ITM-Match

ITM-Match is a module of the ITM system that is already being used to manage the ESCO taxonomy and its supporting taxonomies. While the user interface does not allow much configuration of the linkage rules, the fact that it is a module for an existing ESCO component makes the installation on the EC infrastructure very easy. The requirement for linking is that the taxonomies to be aligned are loaded in the ITM system.

ITM-Match is a closed source proprietary tool developed by Mondeca.

The manual correction interface of ITM-Match is very good. It shows for every concept being matched which alternatives exist for matching in the other taxonomy and can show additional information on the concepts to make an informed decision on whether a match is correct.

Currently, this tool was not yet tested on actual data to be aligned.

### Silk Alignment Framework

The SILK Alignment framework allows the creation of alignment rules through a graphical user interface. It can connect to any triple store. The interface for creating alignment rules is not as advanced as the one from Amalgame, but still allows very detailed configuration. Silk is written in Scala and therefore runs on the Java virtual machine. It can be deployed using the play framework or the Scala simple build tool (sbt).

The manual correction interface of Silk is good. It shows the score of a possible alignment and how this score was calculated. However, it only shows information of the concepts that was used in a mapping rule, no additional properties can be examined to make a decision on the quality of the match. The user also cannot (easily) see how many other elements match an element.

Silk is available as open source (<https://github.com/silk-framework/silk>) and is licensed under the Apache v2 license.

### Comparison table

The following table summarizes the information about the tools for easy comparison

|  |  |  |  |
| --- | --- | --- | --- |
|  | Amalgame | ITM-Match | Silk |
| Configuration | Very good | Poor | Good |
| Manual correction | ? | Very good | Good |
| Results | ? | ? | Good |
| Open Source | Yes | No | Yes |
| License | ? | Proprietary | Apache v2 |
| Environment | SWI-prolog | Java (war file) | Scala (Java VM) |

## Methodology

The methodology for creating a mapping is similar for all tools that were used. There are different stages in the methodology:

* Selecting the data to be mapped
* Preparing the data
* Constructing a mapping rule
* Manually correcting the data
* Creating the final mapping

The following sections will explain these steps in detail.

### Selecting the data to be mapped

Mapping two taxonomies in its core means that two lists of concepts are being examined and the concepts that mean the same thing are extracted. It is worthwhile to reduce the size of these lists for two reasons:

* Because the search space is smaller, the manual validation step will have to work on a smaller set of data (i.e. the data of a single SREF)
* Because the search space can be limited to a single SREF, the mapping will be more precise, e.g. occupations will never be automatically matched to occupations from a completely different sector.

For these reasons every SREF in the Mapping Pilot was matched separately. This section details how to select the data that is relevant to a certain SREF in ESCOv1 terminology.

The NOC all have a link with ISCO08. For example in the case of the Dutch NOC, the (only) property used to link to ISCO08 is skos:exactMatch. This ISCO08 link can be used to extract the occupations of any ESCO SREF, as long as we can translate the sector (which is NACE based) to a set of ISCO08 codes. In the rest of this section, the HOSPI sector will be used as an example.

Different paths to extract the occupations and skills relevant to an ESCO SREF were examined.

#### Path one: select per PES

Every PES has its own hierarchy of occupations. It is possible to select the occupation sectors or groups that map to the NACE codes of the target SREFs. However, this would be a custom operation for every PES. For instance, the Czech NOC has its own NACE classification, where hospitality occupations fall under the sector with URI 'http://ec.europa.eu/esco/ConceptScheme/NACE-CZ/c.18', but this URI cannot be reused in e.g. the French NOC.

#### Path two: use CV/JV content

CVs and JVs have been tagged with ESCO terms (see earlier). These ESCO terms have ISCO08 codes, either directly for occupations or indirectly for skills, through their link with occupations. So, one can select all ISCO08 codes that are relevant for the tagging and filter the Dutch classification that way. The ISCO08 codes for skills can be found by taking the skills that are tagged in the document and taking the set of all occupations that refer to these skills. By taking the union of all ISCO08 codes of these occupations, we get the ISCO08 codes of the skills in the document. Only the occupations for the sector specific ESCO skills are taken into account. Cross-sectoral skills would lead to too many ISCO08 codes, as a very wide range of occupations from different sectors use these skills.

Unfortunately, when gathering the ISCO08 codes from the skills, the query returns 129 ISCO08 codes (and eventually 1276 occupation that belong to these codes) for VET. This almost does not limit the dataset any more.

This result can be explained. A CV contains a much broader set of skills than the domain a person wants to apply for. It details all previous experience the person has.

As a backup solution, only the occupation ISCO08 codes could be selected. This has as a drawback that it selects too few ISCO08 codes. Occupations that are very much related to the sector are not included, e.g. for veterinary nurse, veterinarian (and especially the related skills!) should be included as a nurse often assists in the tasks of a veterinarian.

#### Path three: use ESCOv01

The final approach is to use a taxonomy that has already tagged concepts with both ISCO08 and NACE codes can be used to create a mapping. ESCO v01 has already tagged its occupations with both NACE and ISCO08. From this tagging, a mapping from NACE to ISCO08 can be created by looking at the occupations that are tagged with a NACE sector and listing all their ISCO08 codes as codes relevant to that sector.

The following SPARQL query creates this mapping:

CONSTRUCT {

?nace <http://localhost/esco-mapping/usesISCOGroup> ?isco .

} WHERE {

GRAPH<http://localhost/esco-v01> {

?occ <http://ec.europa.eu/esco/model#memberOfISCOGroup> ?isco.

{ ?occ <http://www.w3.org/2004/02/skos/core#broader>\* ?ancestor.

?ancestor <http://purl.org/dc/terms/subject> ?nace. }

UNION

{ ?occ <http://purl.org/dc/terms/subject> ?nace. }

}

} ORDER BY ?nace

The NACE sectors that the ESCO SREFs represent are known, these NACE codes can be used as look-up values to get the ISCO08 codes for that sector. From there, the relevant occupations in both ESCOv0 and all the NOC classifications can be selected.

The skills belonging to a sector are those used by any occupation in that sector.

This approach is the final, applied approach and the contents of the NACE to ISCO08 mappings is stored in the helper graph with URI http://localhost/esco-mapping/nace-to-isco.

### Preparing the data for mapping

While the data from both ESCO and the NOC was already made available in a linked data format, modifications are still needed to work efficiently with that data in the tools.

Amalgame limitations have not yet been researched.

ITM-Match can only compare items based on their text based values, like their labels and definitions. Numeric values and references are not examined. This would mean that references to e.g. ISCO and NACE cannot be used for comparing concepts in the mapping. To remedy this, the data can be transformed so that properties with numeric or reference values are stored as text values instead of numbers or URIs. For the ISCO tags, for instance, a text value can be created with the language code “isco”. Because the algorithm only compares labels of the same language, these new labels will not interfere with the existing labels.

Silk only compares direct properties of objects. Because ESCO instantiates labels of concepts to add e.g. gender information, a transformation is needed here as well. In this case, all indirect properties of an object that could be relevant for the mapping can be transformed to direct properties. This does mean that some information is lost though. Because the label of an ESCO concept must be transformed to a simple string, only its value and its language information can be retained. The gender information is lost. This will not have a large impact on the mapping though.

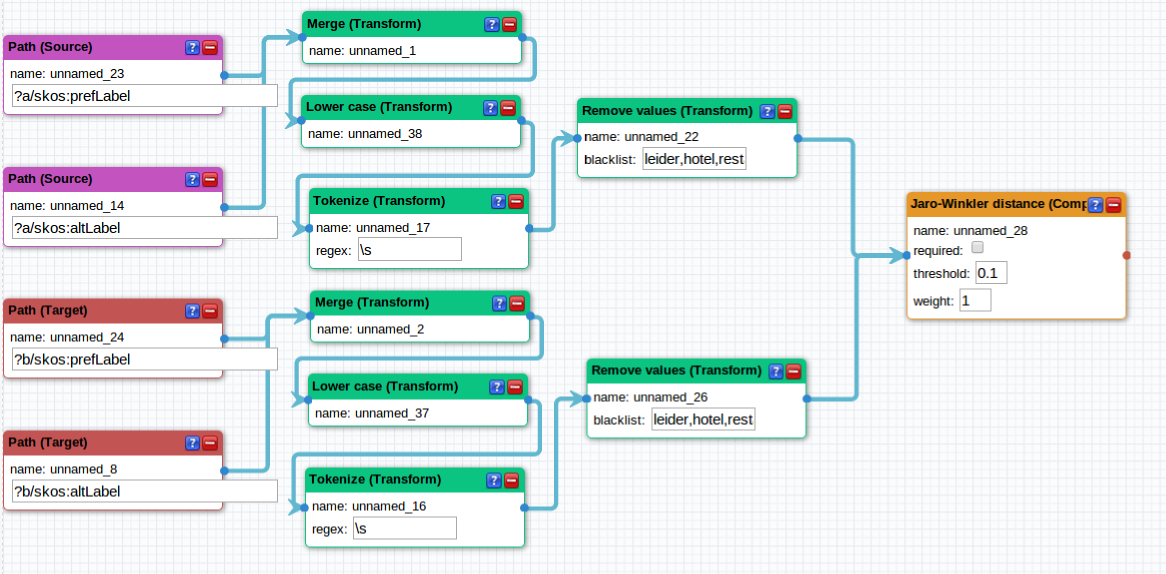
### Alignment rules

#### Building the mapping rule

In Silk, a mapping rule is constructed as follows:

* take the skos:prefLabel (preferred term) and skos:altLabel (non-preferred terms) of the concepts to be matched, and add them to a single list of things to be used as input.
* lowercase all the labels in the list so they are matched in a case-insensitive way
* tokenize the labels, splitting the labels into separate words that can be compared. Note that this is not a linguistic tokenizer, it simply splits the labels into words where it encounters white-space characters.
* remove stopwords so common words are not considered when comparing things (e.g. articles). Words that do not add meaning, like the verb 'work', are removed as well.
* compare the labels using a string comparator (Jaro-Winkler string distance) to compare the words in the labels. The Jaro-Winkler metric examines both strings and creates a score based on the number of matching characters and the number of transformations needed to get from one string to the other.

The resulting linking rule is shown in the image below. The purple (ESCO) and red (Dutch) boxes on the left show the properties taken from the taxonomies to be aligned. The green boxes in the middle show the transformations that are being done on these properties. The yellow box on the right shows the string distance comparator.



#### Reduction of search space

This mapping rule reduces the search space dramatically already. When the Dutch skills have to be matched to ESCOv0 in the context of hospi (for selection of concepts, see the 'data-transformations' section), 617 links have to be manually examined. This is quite an improvement from the about 500x500 possible skill combinations. For the occupations, 151 links have to be manually examined versus the about 150x150 possible links.

However, using some custom extensions to Silk it would be possible to improve the recall and the precision of these linkage rules considerably. This will be described below.

#### Stemming

Silk is lacking an appropriate stemmer. Stemming accepts a word and reduces it to its grammatical basis, e.g. 'running' is reduced to 'run'. The only stemming algorithm available in Silk is the Porter stemmer, which was developed for the English language. Because stemming algorithms are language dependent, this is not sufficient. For Dutch, the stemming does not work for instance. Because words are not stemmed, automatically matching the word 'running' to 'run' for instance becomes hard, as there are already 4 characters, more than half the longest string, that differ.

The lack of a stemmer algorithm means that many correct matches will be discarded, so the search space may have been reduced too far.

Stemmers exist for many European languages (see for instance http://snowball.tartarus.org/, BSD license) and can be imported in Silk (https://github.com/silk-framework/silk/, Apache license v2). Stemmers for all languages of the PES in the mapping pilot are available.

#### Recall over precision

The automatic matching process is always just a first step. A human user must remove the incorrect matches from the list of automatic matches. As he will only see the matches that the algorithm considers possible, it is more important that all correct matches are returned (high recall) than that there are some incorrect matches as well. In other words, the precision (percentage of returned matches that is correct) of the alignment can be lower if it increases recall (percentage of correct matches returned by the algorithm).

However, the precision should not be allowed to drop too much. If the precision is lower, it means many incorrect matches are returned by the algorithm together with the correct matches. All of these incorrect matches need to be examined by the user. In the worst case, all possible matches are returned and we nothing is gained by running the alignment algorithm when compared to a completely manual approach.

#### Improving the precision

Current algorithm works token based, that is, it looks at all the words in the labels of the concepts to be matched separately. This does indeed improve the recall of the algorithm, as detailed in the previous section, but it also includes many results that are somewhat related but should not actually match. For instance, 'Maitre de cuisine' is matched to 'Commis de cuisine'. This match is considered to be 100% the same because one word matches in the label. Words like 'hotel', 'restaurant' and 'manager' are very problematic in this context.

This could be improved by writing a custom comparator for silk that takes two full labels, tokenizes them and awards a bonus if multiple words in the labels match.

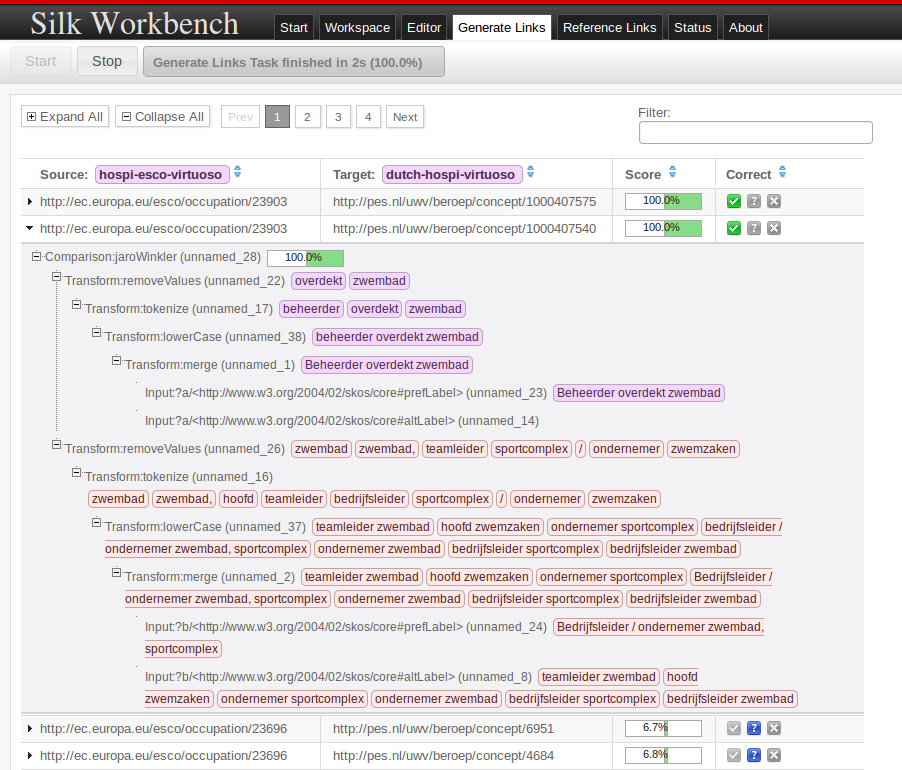
### Manual Corrections

Once the automatic mapping table between a NOC and ESCO is generated, every match in this table needs to be examined to remove the incorrect matches. This section describes this procedure and the problems that may occur during this process.

#### Manual correction interface

The manual correction interface displays all the possible matches that were discovered the application. In the image below, the Silk interface is shown. It shows a table with four columns, the URI of the concept in the source taxonomy (ESCOv0), the URI of the concept in the target taxonomy (Dutch NOC), the score according to the linking rule and controls to approve, or remove a link. Each of these columns can be sorted on.

Every row in the table can be clicked open to see the inputs of the mapping rule, how they were transformed and compared to obtain the score of the match.



#### Reduced result set

Automatic mapping is only useful when the set of possible matches is reduced. Theoretically every concept in a NOC can be the same as any other concept in ESCO. There can be even multiple concepts that have to be mapped to one single other concept. If the search space is not reduced, this means n x m possible matches have to be examined per NOC, where n is the number of concepts in ESCO and m is the number of concepts in the NOC. Looking only at the skills related to Hospitality in ESCO and the Dutch NOC, this gives 500 x 500 = 250.000 possible matches to be examined, which is unfeasible.

If a mapping rule does not reduce the search space enough it is not useful and it must be refined further. However, the rule should not be too restrictive as it should not remove (too many) good matches from the set of returned matches.

#### When to consider a link correct

An actual case that came up when matching occupations is the following. There is a 'Chinese cook' in the Dutch occupation list. This is automatically matched to the 'Cook' occupation in ESCO, while there is also a 'Chinese Cook'. Are both links correct? Should only the most specific link be taken?

If only the most specific link should be taken, it will be hard to do automatically. This is a fairly common problem in the mapping, other examples are waiters that only work on a ship and general waiters.

#### Missing links in concept labels

The automatic mapping tools in the pilot consider two concepts the same if they have similar labels. However, the NOCs may use very different terminology than ESCO to describe some concepts. In that case, there will be no similar labels and the matches will not be returned in the set of automatic matches.

Currently the algorithms used have no way of detecting that such concepts are in fact still similar. The problem could be mitigated by fetching synonyms of the labels of concepts, but this may increase the time needed for mapping too much.

In any case, there is no guarantee that an automatic alignment tool will find every correct match between two taxonomies. As part of the manual corrections, one could select the concepts in either of the two taxonomies that have no mappings at all and manually see whether a match can be found manually.

## Creating the final mapping

Once the single matches for the concepts in ESCO and the NOCs have been found a mapping table can be constructed. This mapping table holds for every concept of one taxonomy, the matching concepts from the other taxonomy and the kind of relationship the two concepts have.

For the type of relationships, the SKOS[[1]](#footnote-1) ontology can be used.The SKOS primer[[2]](#footnote-2) defines the relationships that can be used for alignment as follows:

* **skos:closeMatch:** indicates that two concepts are sufficiently similar that they can be used interchangeably in applications that consider the two concept schemes they belong to. However, skos:closeMatch is not defined as transitive, which prevents such similarity assessments to propagate beyond these two schemes
* **skos:exactMatch:** also indicates semantic similarity—it is a sub-property of skos:closeMatch. However, it denotes an even higher degree of closeness: the two concepts have equivalent meaning, and the link can be exploited across a wider range of applications and schemes. skos:exactMatch is indeed transitive
* **skos:broadMatch** and **skos:narrowMatch**: are parallel to the skos:broader and skos:narrower relations, which enable the representation of hierarchical links, such as the relationship between one genre and its more specific species, or, depending on interpretations, the relationship between one whole and its parts;
* **skos:relatedMatch:** is parallel to the skos:related relation, which nables the representation of associative (non-hierarchical) links, such as the relationship between one type of event and a category of entities which typically participate in it. Another use for skos:related is between two categories where neither is more general or more specific. Note that skos:related enables the representation of associative (non-hierarchical) links, which can also be used to represent part-whole links that are not meant as hierarchical relationships.

The decision about which property should be used by each match may be assisted by the the alignment algorithm but should at least be evaluated by a human user. E.g. ITM-Match allows the automatic creation of broader or narrower matches.

Currently, the mapping tables that have been created were generated automatically by Silk and have not had a manual correction step. The predicate that is being used to connect the two taxonomies is the skos:relatedMatch property.

# Evaluating the Mapping

Insert evaluation grids here

*End of Document*

1. Simple knowledge organization system, <http://www.w3.org/2004/02/skos/> [↑](#footnote-ref-1)
2. Skos primer: <http://www.w3.org/TR/skos-primer/> [↑](#footnote-ref-2)