JRC EUROVOC INDEXER - JEX

MANUAL

MOHAMED EBRAHIM

RALF STEINBERGER

MARCO TURCHI

Publication date (version): 15.05.2012

ABOUT JEX

The *JRC EuroVoc Indexer* JEX is a software package developed at the European Commission's *Joint Research Centre* (JRC)[1] aiming to provide end users with a tool for the automatic annotation of documents with descriptors from the EuroVoc[2] thesaurus. It can also be used by advanced users for research purposes, for using the training method developed at the JRC[3] to train the system on a different type of ontology or just to improve the semantic performance of the indexing tool by training it on more data.

We have trained JEX to index documents in the following **22 languages**: Bulgarian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish and Swedish. However, JEX can be trained in many more languages, assuming that training data exists (documents previously classified manually). As the EuroVoc thesaurus is available in at least five further languages (Basque, Catalan, Croatian, Russian and Serbian), we can assume that training documents for these languages exist, as well.

This manual is divided into three parts. Part I gives background information about the system. Part II introduces the different functionalities provided by the system and how to use them. Part III contains a reference section for all the configuration options of the system.

[1] http://ec.europa.eu/dgs/jrc/index.cfm

[2] http://eurovoc.europa.eu/

[3] http://langtech.jrc.ec.europa.eu/Eurovoc.html

TABLE OF CONTENTS

PART I. BACKGROUND INFORMATION

1. BACKGROUND INFORMATION

1.1. INTRODUCTION

EuroVoc is a multilingual thesaurus used by the European Parliament (EP), by parts of the European Commission and by many national and regional parliaments or other organisations in the European Union to categorise, search and retrieve their documents. The European Commission's Joint Research Centre (JRC) has developed software that tries to categorise documents in many different languages automatically according to the EuroVoc classes. The automatic classification is less accurate than that carried out by human professionals, but it has the advantage that it is extremely fast and perfectly consistent. The software can thus be used to automatically classify or re-classify large historic document collections with a new version of EuroVoc. It is also possible to use this software as an interactive tool to combine the virtues of the machine and those of the human professional. In this setting, the software automatically assigns EuroVoc descriptors to a text and the human indexer verifies and corrects the automatic results. Interactive indexing would thus help guarantee indexing consistency while speeding up the assignment process. The Spanish *Congress of Deputies* in Madrid has been using the JRC's Eurovoc indexing software as an interactive tool since 2006.

1.2. MANUAL ASSIGNMENT OF EUROVOC DESCRIPTORS TO TEXT

As of May 2012, EuroVoc consists of 6797 descriptors (also referred to as *classes* or *categories*) which are organised into a hierarchical structure of up to 8 levels. Human indexing professionals are typically librarians or linguists, employed or free-lance, with a developed conceptual understanding of the themes dealt with in the thesaurus.

The classification of documents with EuroVoc is done for at least two purposes: (a) by looking at the EuroVoc descriptors for any document, users can see what the document is about, i.e. the descriptors have a summarising function; (b) users can retrieve documents about a certain subject domain by searching for specific or generic EuroVoc descriptors (e.g. NUCLEAR ENERGY or PLUTONIUM). As EuroVoc is multilingual and each descriptor has exactly one translation in each of the other languages, users can also see in their own language what a foreign language document is about and they can retrieve documents in other languages by searching with descriptors in their own language.

1.3. AUTOMATIC CATEGORISATION OF DOCUMENTS ACCORDING TO EUROVOC

From a technical point of view, EuroVoc descriptor assignment is a multi-label categorisation, i.e. each document is classified into any of the EuroVoc classes (each descriptor stands for one class) and each document can belong to more than one class, i.e. more than one class label can be attached to a single document. There are roughly two approaches to automate classification, described here in a rather simplistic way: (a) people can manually write rules saying, for instance, that if some words occur in a document and others don't, then the document could be classified into a certain class. For instance, if any of the words 'nuclear material(s)', 'plutonium' or 'uranium' occur in a text, then the text can be classified as belonging to the class RADIOACTIVE MATERIALS. Writing such rules for all descriptors and for all EuroVoc languages would be rather time-consuming. The rules would additionally need to be updated with time as the useful word combinations may change over time. (b) Computer programs that are fed with many manually classified documents can learn (using statistical methods) what words are typical for documents belonging to each class. We refer to these typical words as *associates*. When the software then finds associates for a certain class in a new document, it may conclude that this class is appropriate for this document. Within these Machine Learning approaches (group b), there are many variants. Automatic EuroVoc indexing is a particularly challenging task because (a) the number of classes is extremely large (there are over 6,000 EuroVoc classes), (b) each document belongs to several classes (multi-label categorisation) and (c) EuroVoc descriptors are used rather unevenly (distribution imbalance), so that for many descriptors there are not enough training documents (documents from which the software can learn the typical words or word combinations).

The JRC's software uses the Machine Learning approach (approach b). This means that the software needs to be trained with manually classified documents. The software uses a bag-of-words approach, i.e. it takes into account which words occur how frequently in a text, but it does not consider the sequence and context of words. This is a rather common approach. Software trained automatically on certain text types (e.g. legislation) will work reasonably well when classifying texts of the same type, but they will perform badly when trying to classify a different text type (e.g. news). It is thus moderately useful to train the software on the legal EU documents and to try to categorise other types of documents.

For a more detailed description of how JRC's EuroVoc indexing technology works, why it was developed and how it can be used, please read the following publication, which can be found at http://langtech.jrc.ec.europa.eu/JRC_Publications.html.

> Steinberger Ralf, Mohamed Ebrahim & Marco Turchi (2012). **JRC EuroVoc Indexer JEX - A freely available multi-label categorisation tool**. Proceedings of the 8th international conference on Language Resources and Evaluation (LREC'2012), Istanbul, 21-27 May 2012.

At the same place, you also find further related publications.

1.4. POSSIBLE USES OF JEX

JEX can be used by parliamentary libraries as an interactive indexing tool to improve consistency and speed of the human indexing process. While the accuracy achieved by the fully automatic process may not be good enough for normal parliamentary needs, it may still be useful to index large document collections that would not otherwise be indexed at all. Finally, JEX as a fully automatic tool can also be used as a software ingredient to a number of multilingual and cross-lingual Language Technology applications. These include cross-lingual clustering and classification; the detection of document translations; cross-lingual plagiarism detection; the linking of related documents across languages; to support the lexical choice in Machine Translation; and the ranking of sentences in topic-specific summarisation.

1.5. USING STOP WORD LISTS TO IMPROVE THE PERFORMANCE

The tool thus learns automatically lists of topically related words (*associates*) that are correlated with each class (descriptor). Most of these words are indeed even intuitively related to the class, but there are other words that should not be part of any list of these lists. For instance, the words 'were', 'regard', 'question' and 'paragraph' are not related to any specific descriptor. Such words should be ignored by the system. In many cases, the learning algorithm detects that these words are evenly distributed over all documents and that they should not be part of any descriptor's associate list. However, sometimes this is not the case. To ensure that these words do not get considered during the classification process, language-specific *stop word* lists are used. Words on this stop word list will be ignored when the system decides to assign a descriptor. We found that elaborating a good list of stop words will improve the performance of the JEX tool more than any other measure. To produce a good stop word list, it is best if users know Eurovoc and the EuroVoc-indexed documents well.

For English and a small number of other languages, the JRC has already produced relatively large stop word lists (containing hundreds or even thousands of words). Wherever these exist, they are part of the JEX distribution. However, for most of the 22 languages, such lists have not yet been elaborated, or only generic (not EuroVoc-specific) stop word lists are being used. It is thus of vital importance that you produce an extensive list of stop words for your language in order for the tool to perform well in that language. When producing the stop word list, it is useful to keep in mind that the main purpose is to find words that do not distinguish between the classes. For the EuroVoc documents, which are mostly in the legal domain, words such as 'paragraph', 'bullet', 'decision', 'European', 'chairman', etc., should be part of this stop word list. A practical way to produce these stop word lists is to manually select among the most frequent words of your text collection. Note that stop words will apply to all descriptors, i.e. stop words are not the means to remove specific words from a descriptor's associate list.

1.6. IMPROVING AUTOMATIC CATEGORISATION OVER TIME

Typically, the more training documents there are for each class (i.e. for each descriptor), the better Machine Learning systems will perform.

The JRC provides pre-trained EuroVoc indexing software for all official EU languages except Irish (there are not enough documents that have been translated into Irish). The software has been trained with between 20,000 and 40,000 documents per language. However, the major advantage of the software is that it can be re-trained with the users' own documents. The users can thus re-train the software periodically with their newest documents, which may have been classified manually or interactively, and the performance should in principle get better with time.

20,000 manually classified documents may sound like a lot, but as the usage frequency of the over 6,000 EuroVoc descriptors is rather unevenly distributed, this is not enough to train for the classification of many descriptors. The reason is that the software needs to train automatic classifiers for each descriptor so that it needs enough documents for each descriptor. 50 documents per descriptor would be a good number to produce rather good classification results (exact numbers may differ depending on various other parameters), i.e. the software would need about 50 documents to which this descriptor has been assigned manually. However, in our training set of about 40,000 documents, there are only 802 descriptors for which fifty or more manually indexed documents exist. On the other hand, reasonable results can also be achieved with smaller amounts of training documents per descriptor.

The automatic system can thus be expected to work relatively well for those descriptors for which many training documents exist and less well for the others. Should users of the system want to train the system for certain descriptors (e.g. for descriptors that have just been added in a new version of EuroVoc), they can search for documents for which this new descriptor is appropriate, manually index the documents and then re-train the system. The system should then be able to classify new documents with this newly trained descriptor.

1.7. USING JEX TO INDEX DOCUMENTS WITH THESAURI OTHER THAN EUROVOC

JEX can in principle be used to train the classifiers for descriptors that are not part of EuroVoc, or for a classification scheme that combines EuroVoc descriptors with your own categories. Note that – in that case – some JEX functionality may not work (e.g. functionality having to do with the hierarchical display of the thesaurus). JEX was optimised for multi-label categorisation using large numbers of classes. You can use it for other purposes, but you may then not fully exploit the software's strength.

1.8. A MULTILINGUAL CLASSIFICATION TOOL – DIFFERENCES BETWEEN LANGUAGES

The JRC's software operates on any text-based documents encoded in UTF-8 and can, in principle, be trained for any language that uses white space to separate words (in Chinese, for example, words are not visibly separated), independently of the writing system. The software will thus also work with Bulgarian (using Cyrillic) and Greek texts (using Greek characters).

However, there are other differences between languages that do matter: The system tries to identify, separately for each descriptor, which words are typical for the documents indexed with this descriptor (the *associates*). It may find, for instance, that the words 'religion' and 'religious' are very typical for documents indexed with the EuroVoc descriptor RELIGION. In more highly inflected languages, such as those from the Balto-Slavonic or Finno-Ugric language families, many more different morphological forms for these words may be found in the text. The software treats any of these word forms as being completely independent and unrelated and it has to learn for each of these word forms that they may be related to the descriptor RELIGION. You may thus consider using linguistic pre-processing software (e.g. lemmatisers) that changes the document representation. Our own experiments with using lemmatisers for the language Czech, English, Estonian and French did not improve the classification performance[1], but many more pre-processing steps and parameter setting could be tried out. Such work must be carried out by specialists in computational linguistics. Should you be interested in collaborating with computational linguists and you do not know any for your language, you can ask the JRC.

1.9. EXPECTED PERFORMANCE

The JRC has evaluated the software for all 22 languages automatically, by measuring the accuracy of the automatically assigned EuroVoc descriptors against the previously manually assigned descriptors. The results, visualised in **Figure 1**, show that the performance is surprisingly similar across all languages. The performance is expressed using the F1 measure, which gives equal weight to the accuracy of the assignment (precision) and to the coverage (recall). Note that the JRC's software assigns a ranked list of descriptors rather than a set so that evaluation results can be calculated for the best descriptor (the descriptor that the system found to be most likely to be appropriate), for the two best, for the three best, etc. As six was the most frequently used number of descriptors assigned to text, we evaluated the first six best assignments for all documents. However, if in a test document only three descriptors had been manually assigned, three of the six automatically assigned descriptors were thus automatically marked as wrong, which lowered the average performance for all documents and descriptors.

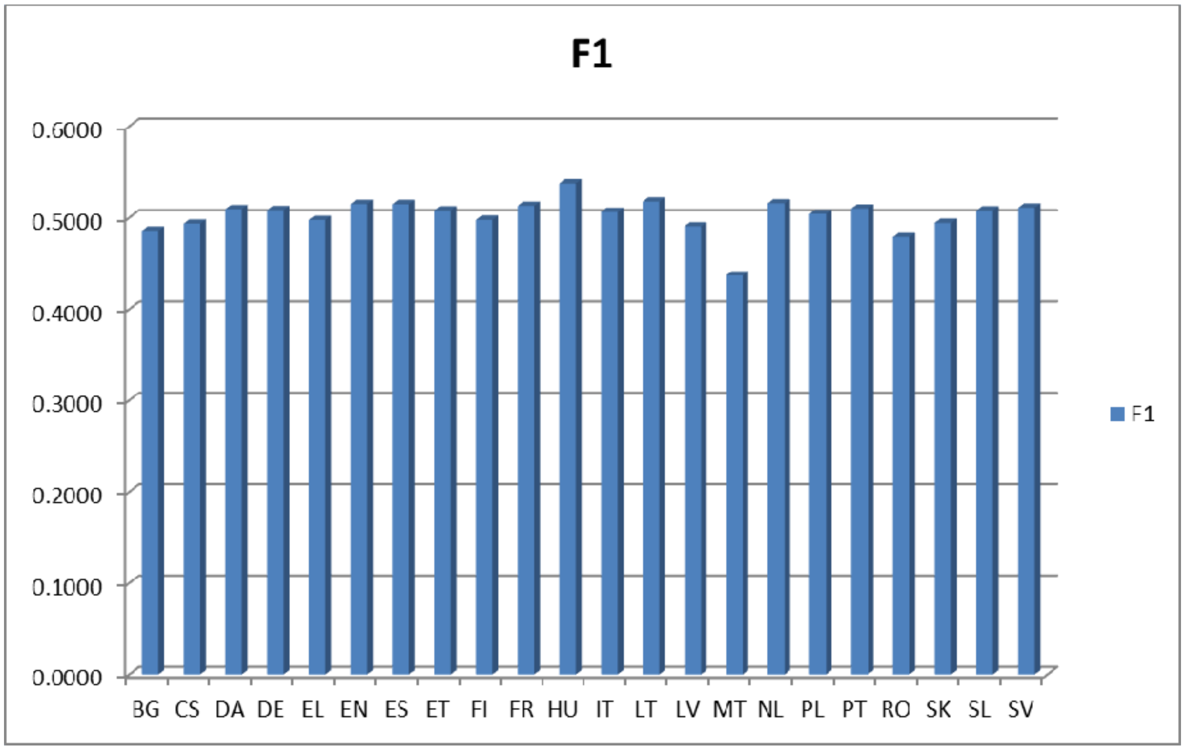


FIGURE 1. PERFORMANCE LEVELS (F1) ACHIEVED FOR 22 LANGUAGES.

For English and Spanish, a large-scale **manual evaluation** was additionally carried out. In this evaluation, professional (human) indexers evaluated the automatic assignment. The F1-measure for the top ten automatically assigned descriptors was

[1] Ebrahim Mohamed, Maud Ehrmann, Marco Turchi & Ralf Steinberger (2012). Multi-label EuroVoc classification for Eastern and Southern EU Languages. In: Cristina Vertan & Walther v. Hahn: *Multilingual processing in Eastern and Southern EU languages - Low-resourced technologies and translation*. Cambridge Scholars Publishing, Cambridge, UK.

.63 and .67. These values are slightly higher than those of the automatic evaluation.

The professional human evaluators also judged the previously manually assigned descriptors by their professional peers, without knowing which descriptors had been assigned automatically and which ones manually. They judged 78% and 87%, respectively, of the manually assigned descriptors as good. This shows that even the human inter-annotator agreement for this highly complex task is far from reaching 100%, and it presents the performance of the fully automatic tool in a better light. For details on this evaluation (and more technical details on the indexing algorithm), see:

> Pouliquen Bruno, Steinberger Ralf, Camelia Ignat (2003). **Automatic annotation of multilingual text collections with a conceptual thesaurus**. In: Proceedings of the Workshop Ontologies and Information Extraction at the Summer School *The Semantic Web and Language Technology - Its Potential and Practicalities* (EUROLAN'2003). Bucharest, Romania, 28 July - 8 August 2003.

1.10. DOWNLOAD AND INSTALLATION OF JEX

JEX is implemented in Java; to run the application java 6 or higher is required. It can be used on Windows and on un*x-like machines, as well as on Apple Macs. The software runs on any modern computer without any particular specifications. The indexing process is extremely fast so that even large document collections can be indexed in little time, but training the system on document collections of the size we used (20 to 40 thousand documents per language) takes a few hours per language.

JEX can be downloaded from http://langtech.jrc.ec.europa.eu/Eurovoc.html. It is packaged, separately for each of the 22 languages, in zip files. There are two versions of the software: (1) the basic version of the software, which can index documents and which should be sufficient for most users; and (2) a more complete version, which additionally contains the training data and which allows specialists to re-train the software. The downloaded files should be unzipped in a directory of your choice, but the directory structure of the application must be maintained. The application should be installed on the machine where it runs as it would run rather slowly if the files had to be accessed across a network. No further installation is necessary besides the unzipping.

PART II. FUNCTIONALITIES PROVIDED BY THE SYSTEM

There are two ways of using the indexing system. The easy way is to use the *graphical user interface* (GUI), which allows users to select a set of files to be indexed and to show, edit and save the result of the indexing. The *command line interface* allows users to carry out batch indexing and also to use the system for training their own data. All executable files can be found in the *bin* directory. All scripts can be configured using conventional java *properties* files. These properties files can be found in the *config* directory. The following chapters will explain the two usage modes in detail.

2. GRAPHICAL USER INTERFACE

You can start the graphical user interface (GUI) by double-clicking on the program file **start_gui.{bat|bash}** in the directory *bin*. The GUI can be used to index a set of files and to display the result of the indexing. Users are allowed to manually change the result and to save it.

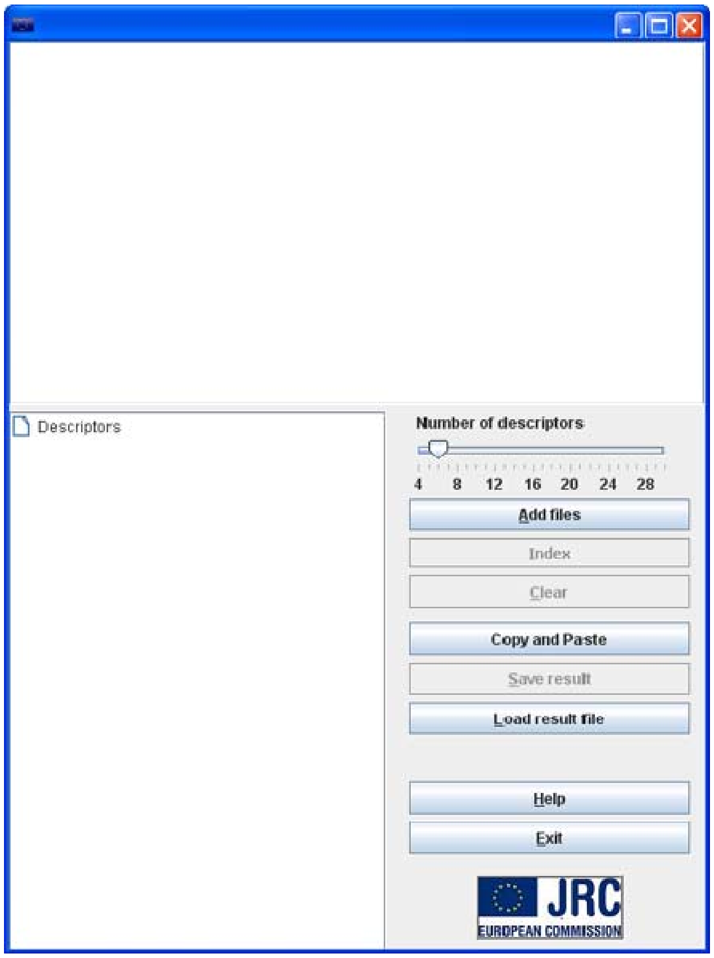


FIGURE 2. MAIN JEX WINDOW.

2.1. INPUT

Using the button *Add Files*, users can select a set of files to be indexed. The System can process any utf-8-encoded text files. The types of files which the users are allowed to feed to the system can be configured. In the default configuration, users are allowed to select only files with the ending "html", "xml" and "txt". If you have any other type of text files you wish to add, please change the parameter AcceptedFormats according to your needs.

Warning

The system can only process text files that are encoded in UTF-8. If non-text files (e.g. PDF, Microsoft-Word or image files such as jpg or tiff) or files not encoded in UTF-8 are used as input to the system, the results are not defined, meaning that the results may seem random.

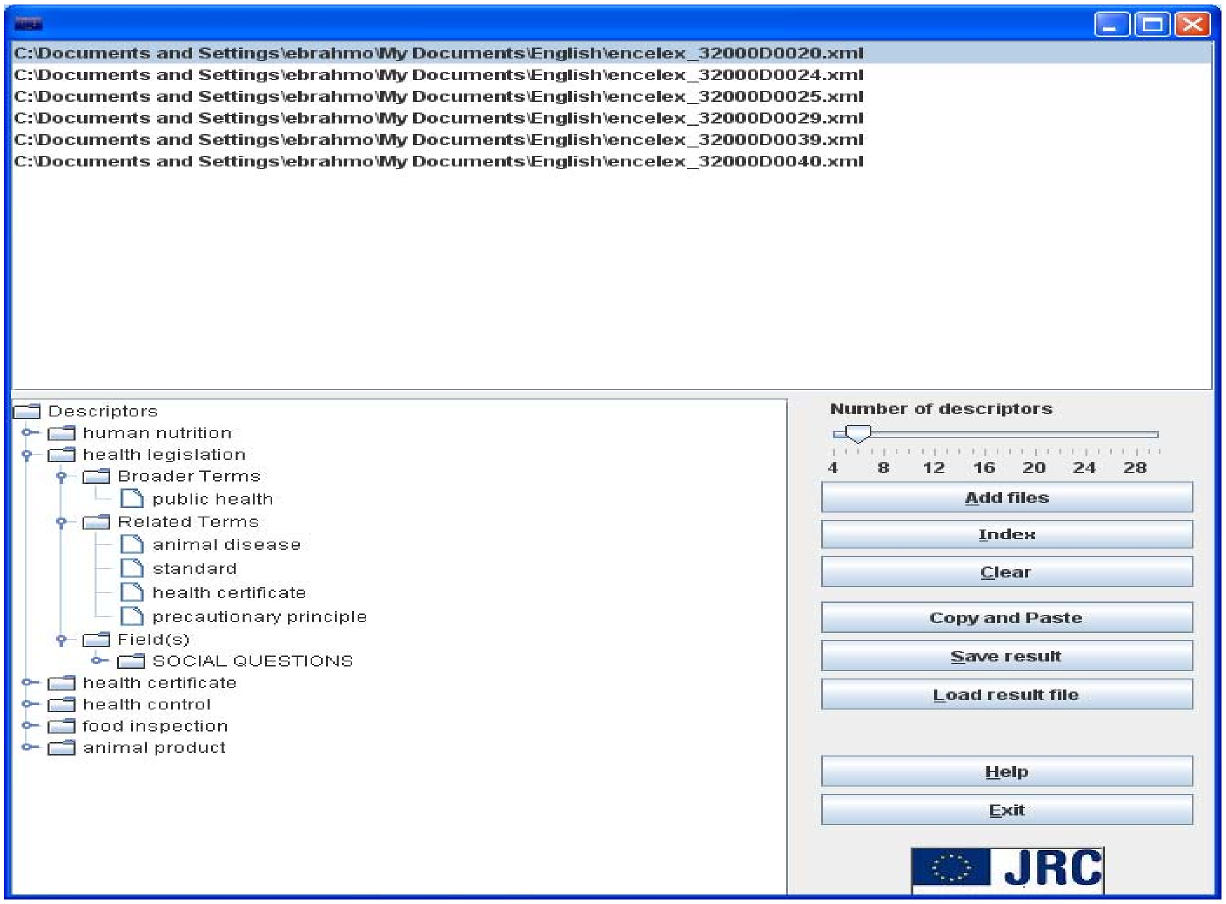
Users can also use the button *Copy and Paste* to use as an input some text copied from a different application, or to manually write the text.

Note

This is useful for testing purposes; for example to check if a document that includes certain terms will be assigned to a given descriptor.

2.2. INDEXING

After adding the files to the system, you click the *index* button in order to start the indexing process, as shown in **Figure 3**. When indexing is completed, clicking on any of these documents (in the top part of the window) will display the descriptors assigned to this document, in the bottom left of the window.



C:\Documents and Settings\ebrahmo\My Documents\English\encelex_32000D0020.xml
C:\Documents and Settings\ebrahmo\My Documents\English\encelex_32000D0024.xml
C:\Documents and Settings\ebrahmo\My Documents\English\encelex_32000D0025.xml
C:\Documents and Settings\ebrahmo\My Documents\English\encelex_32000D0029.xml
C:\Documents and Settings\ebrahmo\My Documents\English\encelex_32000D0039.xml
C:\Documents and Settings\ebrahmo\My Documents\English\encelex_32000D0040.xml

```
Descriptors
  human nutrition
  health legislation
    Broader Terms
      public health
    Related Terms
      animal disease
      standard
      health certificate
      precautionary principle
    Field(s)
      SOCIAL QUESTIONS
  health certificate
  health control
  food inspection
  animal product
```

FIGURE 3. MAIN JEX WINDOW WITH DOCUMENTS LOADED AND THE INDEXING RESULT SHOWN.

Note

The slider *Number of descriptors* can be used to configure how many descriptors should be displayed for a given document. The default value is six. After changing the value on the slider, you need to press the *Index* button again.

2.3. DESCRIPTOR DISPLAY

Descriptors are displayed in the bottom left window. The display language by default is the same as the language of the text. A descriptor is displayed together with information on its position in the hierarchical EuroVoc structure. If you click on any of the descriptors you can see the following information:

Broader Terms

A list of descriptors with a scope wider than the shown descriptor.

Related Terms

A list of descriptors related to the shown descriptor.

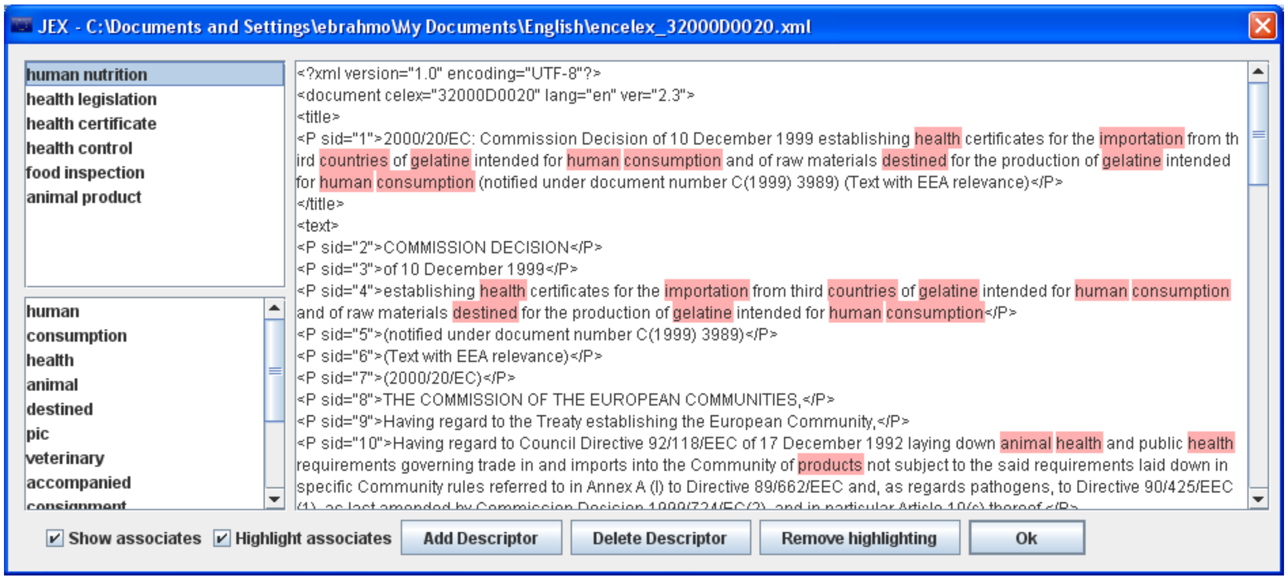
Field

The field(s) to which this descriptor belongs.

Micro-thesauri

The micro-thesaurus, or the micro-thesauri, to which a descriptor belongs are shown under the node *Field*.

If you click on any document in the JEX main window twice, a new text display window will open to show the full text of the document and the descriptors assigned to the document. As shown in **Figure 4**, when you select any of the descriptors, all words in the document which led to the assignment of this descriptor will be highlighted (right part of the window). Also, all the words associated with that descriptor – whether found in the document or not – will be shown (bottom left part). The display of both kinds of information can be toggled by the check boxes *Show associates* and *Highlight associates*.



human nutrition
health legislation
health certificate
health control
food inspection
animal product

human
consumption
health
animal
destined
pic
veterinary
accompanied
consignment

```
<?xml version="1.0" encoding="UTF-8"?>
<document celex="32000D0020" lang="en" ver="2.3">
<title>
<P sid="1">2000/20/EC: Commission Decision of 10 December 1999 establishing health certificates for the importation from th
ird countries of gelatine intended for human consumption and of raw materials destined for the production of gelatine intended
for human consumption (notified under document number C(1999) 3989) (Text with EEA relevance)</P>
</title>
<text>
<P sid="2">COMMISSION DECISION</P>
<P sid="3">of 10 December 1999</P>
<P sid="4">establishing health certificates for the importation from third countries of gelatine intended for human consumption
and of raw materials destined for the production of gelatine intended for human consumption</P>
<P sid="5">(notified under document number C(1999) 3989)</P>
<P sid="6">(Text with EEA relevance)</P>
<P sid="7">(2000/20/EC)</P>
<P sid="8">THE COMMISSION OF THE EUROPEAN COMMUNITIES,</P>
<P sid="9">Having regard to the Treaty establishing the European Community,</P>
<P sid="10">Having regard to Council Directive 92/118/EEC of 17 December 1992 laying down animal health and public health
requirements governing trade in and imports into the Community of products not subject to the said requirements laid down in
specific Community rules referred to in Annex A (I) to Directive 89/662/EEC and, as regards pathogens, to Directive 90/425/EEC
(1), as last amended by Commission Decision 1999/724/EC(2), and in particular Article 10(c) thereof,</P>
```

FIGURE 4. TEXT DISPLAY WINDOW.

2.4. EDITING THE INDEXING RESULTS

You can use the text display window to manually edit the results provided by JEX. To delete any descriptor, select that descriptor from the list of descriptors in the left part of the window and click on the button *Delete Descriptor*. In order to add new descriptors, click on the button *Add Descriptor* to call the EuroVoc thesaurus display window shown in **Figure 5**.

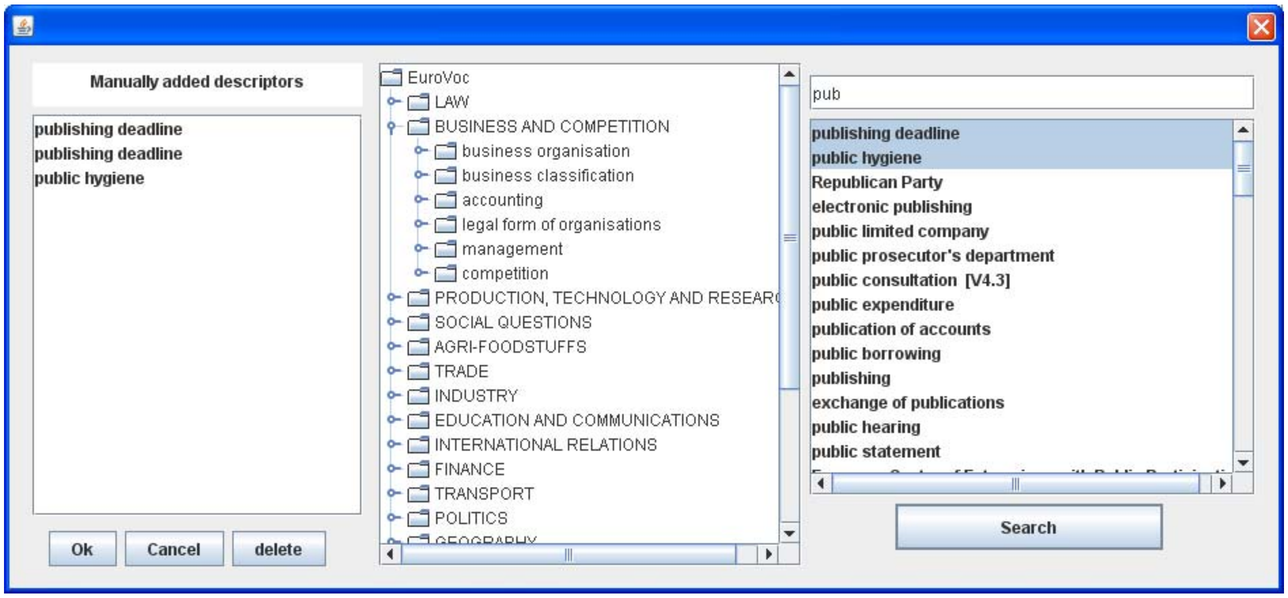


FIGURE 5. EUROVOC DISPLAY WINDOW.

The window is divided into three main parts. In the right part you can search for descriptors in the EuroVoc hierarchy, by typing into the search box at the top. We provide a basic search functionality which allows you to search for a descriptor by any of its sub-strings. For example to search for the descriptor COMMERCIAL TRANSACTION, you can use "comm", "cial" or "trans". You can also use lower and upper case interchangeably. The result of a search is a list of descriptors, which you can add to the left part of the window, either by double-clicking on the descriptor or by dragging the relevant descriptors from the right side and dropping it in the section on the left. The middle part of the window represents EuroVoc as a tree. Each top level node of the tree represents a EuroVoc field and beneath each field you will find the associated micro-thesauri. In turn, each micro-thesaurus includes its associated descriptors. You can browse this tree and add elements from the tree to the left part, using drag and drop.

At any time, you can delete erroneously added descriptors again from your list by selecting them and then clicking the *Delete* button. When you click the button *OK*, the descriptors in the section *manually added descriptors* will be added to your document.

2.5. SAVE AND LOAD RESULT

Clicking on the button *Save result* in the main JEX window allows you to save the result of indexing in xml format. To view and possibly amend a previously saved file, simply click on the *Load result file* button.

3. COMMAND LINE INTERFACE

The command line interface to the EuroVoc indexing tool allows users to index many documents in batch mode. It also allows advanced users to train the system on their own data and to automatically evaluate the results of training the system. The input to the system should be in a format which we call the *compact format* (file extension = cf). Please consult section 5. Compact format definition for a description of this format. We provide with the system a pre-processing module which will create this format for indexing purposes. Also there is a tool to help create the format for training purposes. Each command provided by the interface can be configured using a *properties* file (file extension *properties*), which is located in the directory *config*. Most of the parameters do not need to be changed. However, advanced users wanting to experiment with the system can try out different settings that might be more suitable for their needs. A reference to the different parameters is given in the reference section. All executable programs mentioned here can be found in the `bin` directory of the installation. On a Windows operating system, please use the versions with the file extension "bat"; for un*x-like systems and Apple Mac, please use the versions with the suffix "sh".

3.1. INDEXING

The indexing task is divided into three steps, which will be explained in the following sub-sections:

3.1.1. PRE-PROCESSING

In this step, the input documents will be converted to a standardised format, the so-called *compact format* and all input files will be concatenated into one big compact format file with the file extension cf. You can use the command **bin/PreProcess.{bat|sh}** to create this cf file, or you can write your own routines to produce it. The **bin/PreProcess.{bat|sh}** command will take as its input a directory with a set of files and create one single cf file out of it. The only options that users need to change are the path to the input text documents and the path and name of the output document (the cf file). For a full reference to the configuration of this module, please refer to the reference section.

3.1.2. INDEXING

This is the module that carries out the indexing task. It takes as its input the compact format file produced in the pre-processing step. The command **Index.{bat|sh}** assigns to each document a list of descriptors. The output of this process is an XML file which contains, for each file id, a list of the automatically assigned descriptors. The only options that users need to change are the path and the name of the cf input file, as well as the path and name of the output file. For a full reference to the configuration of this module, please look at the reference section

3.1.3. POST-PROCESSING

This is an optional step which takes as input the file produced by the indexing process and adds more information to the annotation gained in the indexing process. For each assigned descriptor a list of related terms, broader terms and micro thesauri will be produced by the command **PostProcess.{bat|sh}**. By default, the result will be written to an XML file. However, if the original documents were XML documents and the id used in the compact format file is the local path to a file, the user may choose to write the information into this input file under a tag called <*EuroVoc*>. For a full reference to the configuration of this module, please have a look at the reference section.

3.1.4. EXECUTING ALL STEPS AT ONCE

After configuring the system according to your preferences. You can use the command **AllInOneStep.{bash|bat}** to execute all of the above steps.

3.2. TRAINING

The software will be distributed readily trained on those documents that are accessible to the JRC at the moment, but users can re-train the system on their own data, or on a combination of their own data with the JRC's data. Training data must always consist of texts and (manually) assigned descriptors. The result of the training is a collection of *classifiers*, one for each descriptor. Training the system on a set of documents is divided into two tasks. First the compact format needs to be created from the training documents; In a second step, the classifiers will be produced. The commands for training are in the directory `workspace/bin` in the top level directory of the installation.

3.2.1. PRE-PROCESSING FOR TRAINING

It is not possible to provide a module that would allow creating a compact file for all types of input documents JEX users may have. However, we provide a module which can ease the effort of creating the final compact format file. First one or more intermediate compact format files should be created. Such an intermediate cf file is like the one described in the 5. Compact format definition with the exception that the text of a document is written as is with numbers, stop words etc. The number of removed stop words is thus also not written. Our tool will take one or more of those files, do the necessary pre-processing of the text, calculate the number of stop words, write it in the right place and combine all the input cf files into one big cf file. For a complete reference on how to configure this step, please look at the reference section

3.2.2. TRAINING

The training step will take as its input a cf file and the output will be a list of classifiers to be used for indexing and a *dict*[1] file which maps between the tokens in the cf and integers.

Note

When using the classifiers for indexing, please do not forget to use the associated dict file with them, as otherwise the results will be undefined.

Please use the command **Train.{bat|sh}** to start the training. For a detailed reference to the configuration of this module please have a look at the reference section.

3.3. EVALUATION

Evaluation is an indispensable part of training, as it gives you a clear estimation of how the system will perform in terms of recall and precision. We provide two commands to evaluate the performance of the system.

3.3.1. N-FOLD CROSS VALIDATION

The command **RunExperiment.{bat|sh}** allows you to randomly divide a set of documents into a training set and a test set, then to run the training module on the training set and to check the performance by comparing the automatically assigned categories against the previously manually assigned descriptors. The experiment will be repeated N times with different permutations of training and test documents so as to get an accurate idea of the performance of the system. You can configure how many times the experiment should be run and the percentage of the test set, among other options. For details, please have

[1]The purpose of this file is to decrease the memory footprint of the system and to increase the performance.

a look at the reference section.

3.3.2. EVALUATE THE PERFORMANCE ON A DEFINED DOCUMENT SET

The command **evaluate.{bat|bash}** allows you to test the performance of the system on a defined set of documents. This is useful to more exactly compare the performance between different runs. In order to use this command, you first have to train the system and to compile the test set manually. For a detailed reference on the configuration of this command please have a look at the reference section.

PART III. SYSTEM CONFIGURATION

All components of the system can be configured by the users using the *properties* files in the installation's config directory. In all parameters that need a path as their value, the default value will be a relative path to the directory of the installation.

Some of the parameters listed below are described in more detail in the publication:

> Pouliquen Bruno, Ralf Steinberger, Camelia Ignat (2003). **Automatic Annotation of Multilingual Text Collections with a Conceptual Thesaurus**. In: Proceedings of the Workshop *Ontologies and Information Extraction* at the Summer School '*The Semantic Web and Language Technology - Its Potential and Practicalities*' (EUROLAN'2003). Bucharest, Romania, 28 July - 8 August 2003.

4. REFERENCE TO THE CONFIGURATION FILES

4.1. THE GRAPHICAL USER INTERFACE

AcceptedFormats

This parameter defines which files can be processed by the application. Here you can define a list of file endings separated by a comma; only files ending with the one of the endings defined here will be shown to the users in the add files dialogue. Default: "*html, htm, xml, txt*"

classifiersDir

The path to a directory which includes the classifiers to be used for the indexing. You should change the value of this parameter only if you carry out your own training. Default value is *resources/classifiers*.

dict

The path to the dict file. This file contains a mapping between words and the numerical representation in our system. The numerical representation of words is used to improve the performance of the system. Please change the value of this parameter only if you re-train the system yourself. Default is *resources/dict*.

ThesaurusInfo

The path to a directory which includes the XML version of EuroVoc. Default is *resources/ThesaurusStructure*. Usually there is no need to change the value of this parameter.

DisplayLanguage

The language in which the labels of a given descriptor will be shown. The EuroVoc descriptors can thus be shown in a different language from the language of the indexed document.

minNumCommonTokens

This value represents the minimal number of common terms between a descriptor and a document in order for the former to be considered as a potential descriptor for the latter. Default is *1*.

rank

The number of descriptors to be displayed to the user. Default is *6*.

4.2. PRE-PROCESSING

inputDir

All files in this directory will be written to the compact format file. Default is the *documents* directory in the installation's top level directory.

MultiWordsFile

The path to the multiWordsFile. This file includes a list of word n-gram sequences that should be removed from the text. This parameter is optional.

stopwords

The path to a file containing a list of stop words to be removed from the text.

output

The compact format file created will be written to this file.

Warning

If the file already exists, its content will be over-written.

forTraining

Toggles whether the produced compact format file will be used for training or indexing.

AcceptedFormats

Lists file endings (Separated by a comma) which the system should process; only a file with an ending listed here will be added to the compact format, other files within the input directory will be ignored. Default is *xml, html, txt, htm*.

4.3. INDEXING

input

The path to a compact format file to be used in the indexing process.

output

Path to a file where the output should be written.

Warning

If the file already exists, its content will be over-written.

blacklist

The path to a file which contains a list of descriptors that should never be assigned. Here you can add descriptor identifiers, that should never be assigned (e.g. deprecated EuroVoc descriptors). Default value is *resources/blacklist.txt*.

classifiersDir

The path to a directory which includes the classifiers to be used for the indexing. You should change the value of this parameter only if you carry out your own training. Default value is *resources/classifiers*.

dict

The path to the dict file. This file contains a mapping between words and the numerical representation in our system. The numerical representation of words, is used to improve the performance of the system. Please change the value of this parameter only if you re-train the system yourself. Default is *resources/dict*.

minNumCommonTokens

This value represents the minimal number of common terms between a descriptor and a document in order for the former to be considered as a potential descriptor for the latter. Default is *1*.

rank

The number of descriptors to be displayed to the user. Default is *6*.

4.4. POST-PROCESSING

ThesaurusInfo

The path to the directory that includes the XML version of EuroVoc. Default value is *resources/ThesaurusStructure*. Usually there is no need to change the value of this parameter.

DisplayLanguage

The language in which the labels of the different thesaurus structures will be shown.

classifiersDir

The path to a directory which includes the classifiers to be used for the classification. You should change the value of this parameter only if you re-train the system yourself.

documents

The path to the compact format file which has been used for the assignment.

input

The path to the file which includes the result of the assignment.

appendResult

If this property is set to true, the result of the post-processing will be appended to the original XML file. Otherwise, the result will be written to the directory defined by the parameter *resultDir*. Default is *false*.

showDescriptorAssociates

Toggles showing the assigned descriptor associates(words) which have been found in the document. This should be enabled only while debugging the system, if there is an interest in knowing the reason for the indexing of a certain document. For daily usage, it should be disabled; otherwise a lot of memory resources will be wasted. Default is *false*.

showBroaderTerms

Toggles the display of broader terms. Default is *true*.

showRelatedTerms

Toggles the display of related terms. Default is *true*.

showMicroThesaurus

Toggles the display of micro-thesauri and fields. Default is *true*.

resultDir

The result of the post-processing will be written in this directory: each indexed document will have a file with the same name in this directory, including the information required by the user.

4.5. TRAINING

input

The compact format file to be used for training.

classifiersDir

The path to a directory in which the produced classifiers should be written.

dict

The path to a file where the dictionary should be written. The dictionary is an internal file used by JEX to enhance the performance of the system by associating each term known to the system with an integer, in order to replace the costly string equality operations with integer equality operations.

dumpReadable

Toggles writing a human readable version of the descriptors to the desk. For evaluation and test purposes there is no need to set this parameter to true; which will save the time of dumping the files to desk. However, when training with the aim of using the produced classifiers for indexing and displaying the results to an end user it must be set to true. Default is *true*.

RefCorpusWordMinFreq

Words which have a corpus frequency less than this value will be considered irrelevant. Default is *4*.

LlhThreshold

Words with a log-likelihood less than this value will not be considered in the training. Default is *5*.

minDocLength

If a document has less words than t will not his value, it will not be considered in the training. Default is *100*.

minNumTrainingDocsPerDescriptor

A descriptor has at least to appear in this number of documents in order to be trained; otherwise it will be ignored. Default is *4*.

minNumAssociatesPerDescriptor

A descriptor should have at least this number of associates. Default is *1*.

minWeightOfAssociates

If for a given descriptor a term has weight less than this value it will be ignored. Default is *2*.

maxNumAssociatesPerDescriptor

Only this number of associates of a descriptor will be used. If more associates exist, they will be ignored.

DeprecatedFile

If this parameter is defined, a check for deprecated EuroVoc descriptors will be executed during training. The value of this parameter should be a reference to a file where to write the ids of the found deprecated descriptors. The content of this file can be used to delete those descriptors or to add them to the *blacklist* file.

ThesaurusInfo

The path to a directory which includes the XML version Of EuroVoc. Default is *resources/ThesaurusStructure*. Usually there is no need to change the value of this parameter.

Beta

Beta is a parameter that allows tuning the weight (relative importance) of descriptor associates during training, by punishing words that are associates to many different descriptors. The idea is to punish words that are part of too many descriptor associate lists (descriptor profiles). Assuming that the word that can be found in most descriptor associate lists of all is found in X descriptor lists, we punish words that occur in more than X/beta (X divided by beta) descriptor lists. If beta is set to 10, words occurring in more descriptor associate lists than 10% of X are thus punished. The higher you set beta, the more you favour words that are only found in documents indexed with few descriptors. The lower you set beta, the more weight you give to words that can be found in descriptor associate lists of many different descriptors. The default value for beta is set to *10*.

4.6. RUN EXPERIMENT CONFIGURATION

input

The path to the cf file which will be used for training and testing.

blacklist

The path to a file which contains a list of descriptors that should never be assigned. Here you can add descriptor identifiers that should never be assigned (e.g. deprecated EuroVoc descriptors). Default value is *resources/blacklist.txt*.

output

The path to a file to which the result of the evaluation should be written.

testPercentage

The percentage of documents which should be used for testing. The rest of the documents will be used for training. Default is *10* (ten-fold cross-validation).

runTimes

The number of times how often the experiment should be run. Default value is *5*.

RefCorpusWordMinFreq

Words which have a corpus frequency less than this value will be considered to have the frequency of 1. This option is introduced to reduce the size of information held in memory while performing the training. Default is 4.

4.7. EVALUATE

input

The path to the cf file that includes the test documents.

output

The path to the file to which the result of the evaluation should be written.

dict

The path to the dict file. This file contains a mapping between words and the numerical representation in our system. The numerical representation of words is used to improve the performance of the system. Please change the value of this parameter, only if you re-train the system yourself. Default is *resources/dict*.

blacklist

The path to the file which contains a list of descriptors that should never be assigned. Here you can add descriptor identifiers that should never be assigned (e.g. deprecated EuroVoc descriptors). The default value is *resources/blacklist.txt*.

classifiersDir

The path to the directory which includes the classifiers to be used for the indexing. You should change the value of this parameter only if you carry out your own training. Default value is *resources/classifiers*.

minNumCommonTokens

This value represents the minimal number of common terms between a descriptor and a document in order for the former to be considered as a potential descriptor for the latter. Default is *1*.

rank

The number of descriptors to be displayed to the user. Default is *6*.

displayClassifierInfo

If this parameter is set to true, the output of the evaluation will be more verbose. For each descriptor, the system will show how many times the descriptor has been assigned manually, how many times it was automatically assigned and how many times it was assigned both automatically and manually. If the parameter is set to *false*, only recall, precision and f-measure will be shown.

5. COMPACT FORMAT DEFINITION

A compact format file is a representation of a set of text files with some meta-information. Each file in the file set is represented by two lines. The second line includes the words of a document after removing numbers, punctuation marks and stop words. The first line contains the following information separated by the hash symbol:

1. A list of category Ids[1] separated by a space (if the file will be used for training). The empty String (if the file will be used for indexing).

2. A unique identifier for the file.

3. The number of stop words that have been removed from the text.

Let us assume that we have two files on a local file system with the paths "*/usr/docs/file_1.xml*" and "*/usr/docs/file_2.xml*" with the following content:

file_1.xml

Law nr. 34 about fisheries in europe, asia and africa.

file_2.xml

Discussions about 10% income tax reduction for farmers.

If we use the path of a file as its id, this set of documents will be represented in the following way in the compact format for indexing:

```
# /usr/docs/file_1.xm # 4
Law fisheries europa asia africa

#/usr/docs/file_2.xml # 2
Discussions income tax reduction farmers
```

[1]Category ids have to be unique within the whole file.