

Espacenet claims tree

1. `espacenet_maintenance_Data_TextAnalysis` (Claims Tree)

1.1. Source code in SVN

1.1.1. Introduction

The library **TextAnalysis** is responsible for identifying claims (from flat list of concatenated claims with removed mark-up) and detecting references among them. The project depends only on junit and antlr libraries, and does not depend on any parent projects. The API is independent from other espacenet/OPS code and can be easily re-used.

Under the hood the project uses Antlr framework for parsing the claims text and detecting claims in it. The project is built using Maven.

1.1.2. Description

The entry point is class **Claims** with the key method: *public static List<ClaimReferences> getClaimReferenceGuess(String text)*, which takes as input claims as a string of text (without mark-up) and creates a list of individual claims references from it. Each member in the list (an instance of class ClaimReference) contains a claim (instance of Claim class, containing claim text and a corresponding claim number) plus list of claims, which are referenced from this claim.

For example, if original hypothetical input was:

1. text of first claim
2. text of the second claims, referencing claim 1
3. text of the third claim, referencing claim 1 and 2

than the output of this method will be the following list:

claim_1, references: none

claim_2, references: claim_1

claim_3, references: claim_1, claim_2

This structure can be used to build a tree of claims.

Another method (somewhat simpler) *public static List<Claim> getClaimGuess(String text)* is able to provide just list of claims (text plus number), without references.

There is also one convenience method in **ClaimReference** class: *public static SortedMap<Claim,SortedSet<Claim>> getDependencies(Collection<ClaimReferences> allClaimReferences)*, which from list of claims with references makes a map, where key is a claim, and value is a list of other claims referring to this claim ("reversed" relationships).

1.1.3. Additional Espacenet processing

The following gives a highlevel view of how the response is handled in Espacenet:

class PublicationClaimsTileHandler in "espacenet_levelx_WebApplication" contains code which uses the library above and generates all necessary information (matrix) to be inserted into generated HTML for JavaScript to render the claims tree. It can be used as an example of how library is intended to be used;

using JavaScript "Raphael" framework and information (matrix) generated by the item above, the tree is rendered in the browser (in modern browsers using SVG technology) by a piece of javascript code. Path to the code:

/espacenet_levelx_WebApplication/components/amf/eptos/resources/htdoc/scripts/claimstree.js

2. Claims Tree parsing rules

Examples EN taken from:

<https://worldwide.espacenet.com/publicationDetails/claims?CC=EP&NR=1000000A1&KC=A1&FT=D&ND=3&date=20000517&DB=&locale=en> EP

<https://worldwide.espacenet.com/publicationDetails/claims?CC=EP&NR=3363751A2&KC=A2&FT=D&ND=3&date=20180822&DB=EPODOC&locale=en> EP

Examples DE taken from:

<https://worldwide.espacenet.com/publicationDetails/claims?CC=EP&NR=3363751A2&KC=A2&FT=D&ND=3&date=20180822&DB=EPODOC&locale=en> EP

Rule	All	English	German	French	Example EN	Example DE
SINGLE_CLAIM		('C' 'c') 'laim' ('C' 'c') 'lam' ('C' 'c') 'laimed' ('C' 'c') 'lamed'	'Anspruch'	('R' 'r') 'evendication'	EP3360431B1 Apparatus according to claim 1 EP1000000B1 Apparatus as claimed in claim 1	EP3363751B1 Verfahren nach Anspruch 1 EP1000000B1 Vorrichtung gemäß Anspruch 1
MULTIPLE_CLAIMS		('C' 'c') 'laims' ('C' 'c') 'lams'	'Anspr' . 'che' 'n'?	('R' 'r') 'evendications'	Apparatus as claimed in any of the foregoing claims	Vorrichtung gemäß einem der vorangehenden Ansprüche Verfahren nach einem der Ansprüche 1, 2, 4 oder 5
REFER_BACKWARDS		('P' 'p') 'revious' 'ly'? ('P' 'p') 'receding' ('F' 'f') 'oregoing'	'vorhergehende' ('n' 'r') 'vorangehende' ('n' 'r') 'vorangegangene' ('n' 'r') 'vorherige' ('n' 'r')	('P' 'p') 'r' . 'c' . 'dente' 's'?	Apparatus as claimed in any of the foregoing claims	Vorrichtung gemäß einem der vorangehenden Ansprüche
ANY	'any'				Apparatus as claimed in any of the foregoing claims	
SEQUENCE_SEPARATOR	'-' '\u2013'	'to' 'until'	'bis'	'\u00E0'	The elastic electrode structure according to any one of claims 1 to 9	
CONTINUATION	',' '&'	'or' 'and'	'oder' 'und'	'ou' 'et'	Apparatus as claimed in claim 1 or 2	Verfahren nach Anspruch 1 oder 2