A brief description of available services in the X5gon models API

This document is a quick overview of the models provided by the X5gon models API, a technical detailed documentation with specific input and output for each service can be found at http://wp3.x5gon.org/lamapidoc. Even more information can be found in the ***WP 3.2 deliverable*** . Some ipython notebooks in which basic examples on what it is possible to do with the API can be found at (https://colab.research.google.com/drive/1I5PQqN3lWPPYQIaVeKD_ZDfSd9R_Grj0)

Preprocess

This domain allows the user to recover a preprocessed textual content of a resource. All the preprocessing done is optimized on the x5gon resources.

text2phrase joins words in transcriptions that appear frequently together, and infrequently in with other context words. This preprocess can be very useful before more complex text manipulation such as doc2vec. The model used was trained on all the x5gon resources in order to detect particularly well the phrase specific to educational context.
See the model of https://radimrehurek.com/gensim/models/phrases.html for a detailled explanation.

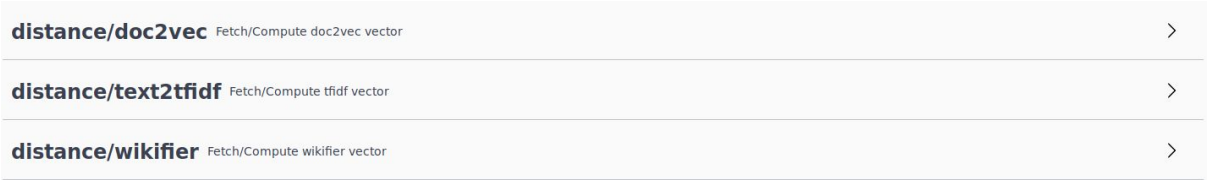
text2processedtext allows to apply some basics text preprocessing:

- lemmatization
- stop words removing
- part of speech removing to keep only nouns, verbs, adjectives, propositions.

Different endpoints are provided, the exact preprocess can be found directly in the service description in the API documentation.

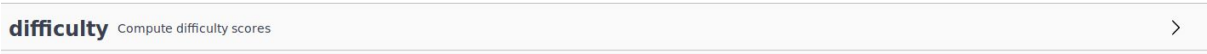
tr2text simply allows to transform dfxp file into plain text, by removing all xml tags.

Distance



In this domain are grouped all the vectorial representations of the resources. All these vectors was computed using only the whole english transcriptions of the resources.
For each representation (doc2vec, tfidf, wikifier) an endpoint (fetch) is dedicated to simply recover the vector for a given input.
And a second one (knn) is dedicated to recover neighbors of the input resource according to the corresponding representation. The distance used is the *cosine distance* between the vectors. The knn service also provides some other extra features in addition to the ranked neighbors and distances such as: proximity matrix of the neighborhood and 2d projection of this matrix using LLE (https://scikit-learn.org/stable/modules/manifold.html#locally-linear-embedding).

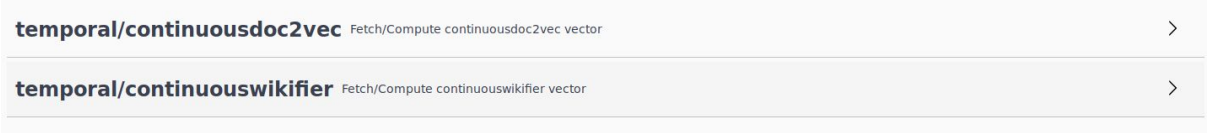
Difficulty



In the difficulty domain three metrics of difficulty are provided:

- (conpersec) The number of wikipedia concepts per second in the transcription for a user with an average reading speed.
- (charpersec) The number of characters per second in the transcription for a user with an average reading speed.
- (tfidf2technicity) A difficulty metric based on the tfidf distribution of the transcription.

Temporal domain



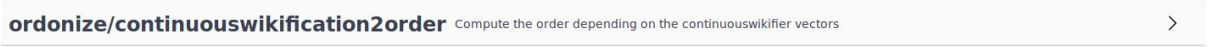
The idea is to no longer representing the resource as a big indivisible block. But on the contrary as an object whose content and therefore the concepts related to it evolve as the resource is consumed. This approach reduces the bias due to the comparison of resources of very different sizes. And also allows in particular for long resources to better capture the meaning of the content. For example, a 200-page book on computer science does not look at all like the same resources in its first chapter, where it deals with the history of computer

science, and in its last chapter, where it deals with the challenges of tomorrow's computer science.
Practically, we simply cut the whole transitions of the resource in constant sized chunk of 5000 words without overlapping between chunks. For each chunk, we simply compute the corresponding (wikifier, doc2vec) and wrap all these results in an output list.

Missing resource

This endpoint gives the most probable resource from a list of candidates, for given previous and after resources from the database. The prediction is currently done based on wikifier of each resource. The best intermediate resource is the one which maximizes the number of concepts shared with previous or after but not both.

ContinuousWikification2order



This endpoint returns the logical order for a list of candidates comparing to the principal resource based on their continuousWikifiers: given a resource and a list of candidate resources.
The model is based on the following assumption: using ContinuousWikifier we can follow the evolution of concepts through the resources; intuitively the first resource should define concepts which can be reused in the following one. More precisely, the common concepts of the two resources should appear earlier in the first resource more than in the following one (at least on average), due to the fact that the learner is familiar with them since they have already been mentioned in.