4
Publishing Tool

In this chapter, we present a tool that implements the publishing technique described in Chapter 3. The tool supports the four stages of the publishing process:

- Extraction of a transcription from a multimedia asset
- Translation of the transcription to several languages
- Generation of a time-aligned Web document using the transcription and content-independent metadata
- Enriching the Web Document with machine-readable data.

4.1
Overview of the Publishing Tool

In this section, we outline how the publication tool works, in an exact parallel with the process described in Chapter 3.

Figure 9 shows a flowchart that summarizes how the tool publishes an audio/video (A/V) asset, according to the technique presented in Chapter 3. We use number from 100 to 106 to indicate the publication stage.

Assume, for the purposes of this example, that a user (100) provides desires to publish an A/V asset. To facilitate the publication, the tool provides a graphical user interface (GUI), as illustrated in Figure 20. The GUI may prompt the user to enter some configuration parameters such as confidence, support, video id, SRT file. The tool may use the associated closed caption (101), or an ASR (102) to generate the time-aligned transcription.

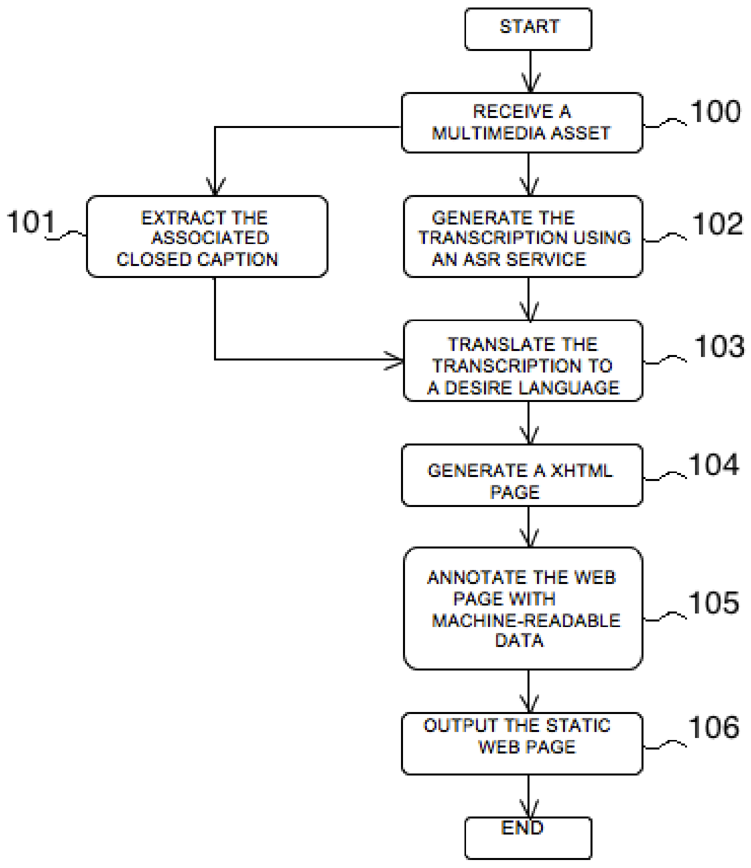

START
RECEIVE A MULTIMEDIA ASSET
100
EXTRACT THE ASSOCIATED CLOSED CAPTION
101
GENERATE THE TRANSCRIPTION USING AN ASR SERVICE
102
TRANSLATE THE TRANSCRIPTION TO A DESIRE LANGUAGE
103
GENERATE A XHTML PAGE
104
ANNOTATE THE WEB PAGE WITH MACHINE-READABLE DATA
105
OUTPUT THE STATIC WEB PAGE
106
END

Figure 9 Flowchart of processing for performing publication in accordance with the implementation.

The tool translates (103) the time-aligned transcription into the desired languages, supported by the Google Translation API.

The tool generates a XHTML page (104), which contains content-independent metadata (title, description, etc.) and descriptive metadata (transcription). Each time-aligned excerpt of the transcription contains a hyperlink that points to the exact segment of the spoken content where the speech occurs.

The tool performs an annotation process (105) to enrich the Web content, This process annotates the content-independent metadata using Dublin Core properties. Also, the tool detects entities in the Web document and incorporates references to DBPedia, Freebase, Schema.org or any other Tripleset available through a SPARQL endpoint resource.

Finally, the tool outputs (106) a static Web page that becomes visible to traditional search engines, and that can be easily indexed by traditional as well as semantic search engines.

4.2 Details of the Publication Tool

In this section, we provide additional details about how the publication tool works with the help of an example.

Assume that a user provides an audio/video (A/V) asset transcription to the tool. The transcription is sent asynchronously via JavaScript without reloading the page. The file format must follow the SubRip[11] text file format (*.srt) that is one of the most popular text-based subtitle formats. This is a simple format, where each caption is divided into three sections:

- Section 1 is a sequential count of captions, starting with 1.
- Section 2 is the start timecode, followed by the string " --> ", followed by the end timecode. Timecodes are in the format:
  - HH – Hours (00, 02 etc.)
  - MM – Minutes (00-59)
  - SS – Seconds (00-59)
  - MIL – Miliseconds (00-1000)

  The end timecode can optionally be followed by display coordinates (example " X1:100 X2:600 Y1:050 Y2:100"). Without the display coordinates, each line of the subtitle will be centered and the block will appear at the bottom of the screen.
- Section 3 is the caption text, composes of one or more lines of text, which will be displayed on the screen. The only formatting tags accepted are the following:
  - <b>text</b>: put text in boldface.
  - <i>text</i>: put text in italics.

[11] http://zuggy.wz.cz/

- <u>text</u>: underline text.
- <font color="#00ff00">text</font>: apply green color formatting to the text (you can use the font tag only to change color).

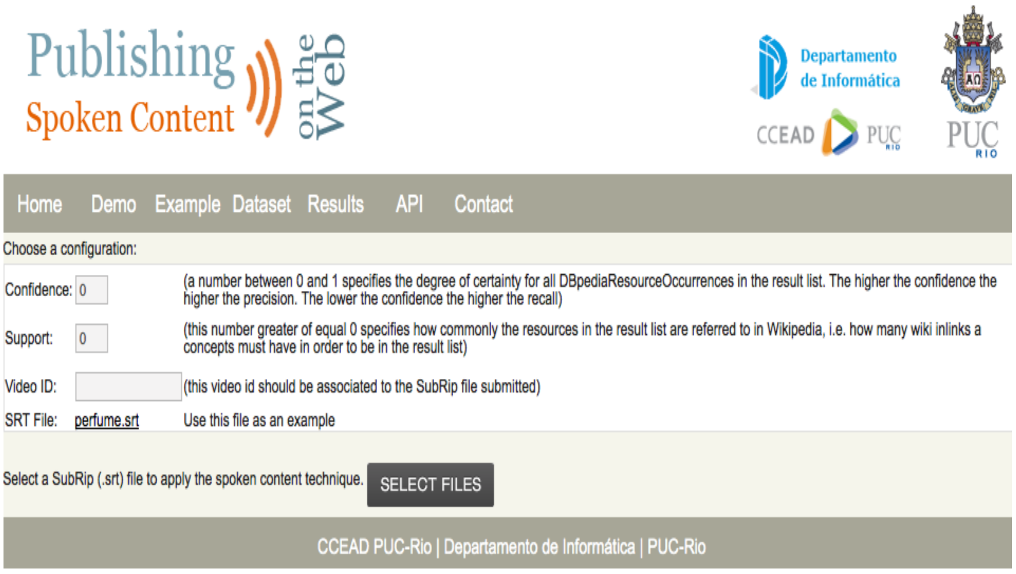
Finally, successive captions are separated from each other by blank marks. The set of captions forms the SubRip text file.

Figure 10 shows a fragment of a SubRip text file for the running example (For the complete SubRip text file, see Appendix A).

```
...

52
00:07:57,889 -> 00:08:00,439
The time it takes for the smell evaporates depends

53
00:08:00,439 -> 00:08:05,870
also external factors such as the degree of acidity and skin
oils.


54
00:08:05,870 -> 00:08:10,519
The temperature also has a direct relationship with the
volatility.
...
```

Figure 10 SubRip file text.

Figure 11 shows the Web accessible user interface (GUI), where some parameters need to be provided by the user before it begins processing. The *confidence* and *support* parameters are used in the annotation stage, and the *video id* parameter helps obtaining content-independent metadata for the HTML-ize stage. The definition of each parameter is explained in the following.


Publishing Spoken Content on the Web
Departamento de Informática
CCEAD PUC
PUC RIO
Home
Demo
Example
Dataset
Results
API
Contact
SELECT FILES
CCEAD PUC-Rio | Departamento de Informática | PUC-Rio

Choose a configuration:

Confidence:	0	(a number between 0 and 1 specifies the degree of certainty for all DBpediaResourceOccurrences in the result list. The higher the confidence the higher the precision. The lower the confidence the higher the recall)
Support:	0	(this number greater of equal 0 specifies how commonly the resources in the result list are referred to in Wikipedia, i.e. how many wiki inlinks a concepts must have in order to be in the result list)
Video ID:		(this video id should be associated to the SubRip file submitted)
SRT File:	perfume.srt	Use this file as an example

Select a SubRip (.srt) file to apply the spoken content technique.

Figure 11 Publishing tool interface.

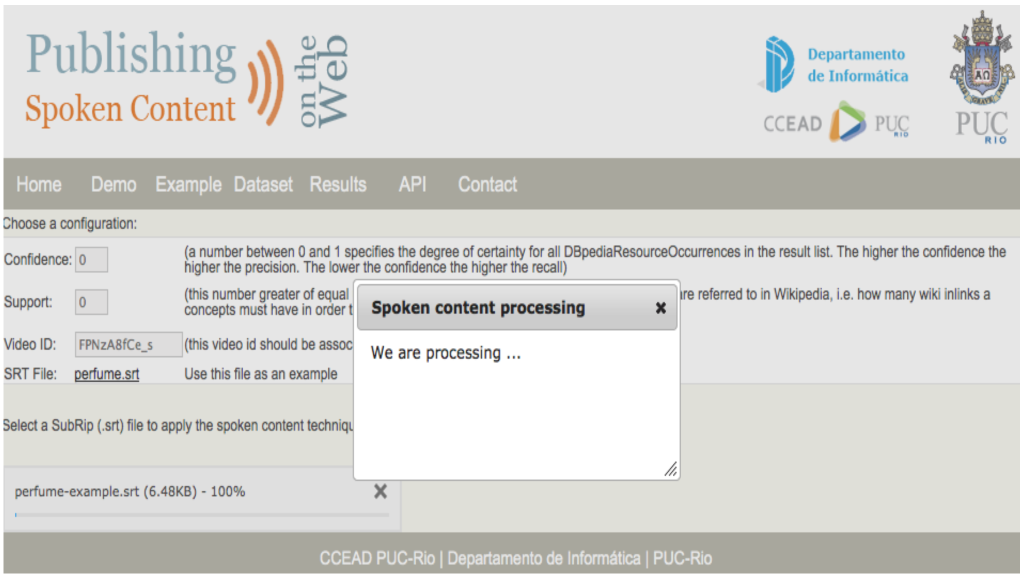
The *confidence* parameter specifies the degree of certainty for all DBpedia Resource Occurrences: the higher the confidence, the higher the precision; the lower the confidence, the higher the recall. This number should be a value between 0 and 1.

The *support* measure specifies how commonly the resources in the result list are referred to in Wikipedia, i.e., how many Wiki link concepts must have in order to be in the result list.

The last parameter is *the video ID,* which symbolizes the video identifier in the YouTube inventory. It is automatically assigned when the spoken content file is uploaded to the YouTube repository. Therefore, it is important to note that our tool considers that all the processed transcriptions have associated spoken content in the YouTube inventory.

Assume that the user provides to the GUI the following parameters:

- Confident: 0.6
- Support: 40
- Video id: FPNzA8fCe_s


Publishing Spoken Content on the Web
Departamento de Informática
CCEAD PUC RIO
PUC RIO
Home
Demo
Example
Dataset
Results
API
Contact
Spoken content processing

Choose a configuration:

Confidence: 0 (a number between 0 and 1 specifies the degree of certainty for all DBpediaResourceOccurrences in the result list. The higher the confidence the higher the precision. The lower the confidence the higher the recall)

Support: 0 (this number greater of equal
concepts must have in order t | re referred to in Wikipedia, i.e. how many wiki inlinks a

Video ID: FPNzA8fCe_s (this video id should be assoc

SRT File: perfume.srt Use this file as an example

Select a SubRip (.srt) file to apply the spoken content techniqu

perfume-example.srt (6.48KB) - 100%

We are processing ...

CCEAD PUC-Rio | Departamento de Informática | PUC-Rio

Figure 12 Processing a A/V asset.

Figure 12 illustrated how the tool starts the processing of the A/V asset. After receiving the transcription file, the tool parses the file and separates the three different sections of the SubRip text file for further processing.

If the caption text is available, it is passed to Google’s Machine Translation service. The tool sends an HTTP GET request with a URI of the following format:

```
GET https://www.googleapis.com/language/translate/v2?key=INSERT-
YOUR-KEY&source=language&target=language&q=text
```

Figure 13 URI format from Google translation API.

Three query parameters are required for each translation request:

- **Key**: identifies the calling application (i.e., the tool).
- **Source**: specifies the source language of the string
- **Target**: specifies the target language
- **q**: represents the source text string and identifies the string to translate.

In this example, the source language is English and the target languages are Portuguese and Spanish. Figure 14 and Figure 15 illustrate how the tool sends the requests to the Google service, using the caption number 52 in Figure 10.

```
GET https://www.googleapis.com/language/translate/v2?key=INSERT-
KEY&source=en&target=es&q=The%time%it%takes%for%the%smell%evapor
ates%depends
```

Figure 14 Request to translate English caption to Spanish.

```
GET https://www.googleapis.com/language/translate/v2?key=INSERT-
KEY&source=en&target=pt&q=The%time%it%takes%for%the%smell%evapor
ates%depends
```

Figure 15 Request to translate English caption to Portuguese.

Table 1 shows the result of the translations process, where the first column represents the Spanish version of the transcription and the second column shows the Portuguese version.

Spanish Version	Portuguese Version
...	...
52	52
00:07:57,889 -> 00:08:00,439	00:07:57,889 -> 00:08:00,439
El tiempo que lleva para la	O tempo que leva para a
evaporación del olor depende	evaporação do cheiro depende
53	
00:08:00,439 -> 00:08:05,870	53
ademas factores externos	00:08:00,439 -> 00:08:05,870
tales como el grado de	também factores externos,
acides en los aceites de la	tais como o grau de acidez de
piel.	óleos e da pele.
54	54
00:08:05,870 -> 00:08:10,519	00:08:05,870 -> 00:08:10,519
La temperatura ademas tiene	A temperatura também tem uma
una relación directa con la	relação direta com a
volatilidad.	volatilidade.
...	...

Table 1 Translated transcription in Spanish and Portuguese.

After translating the transcription, the tool creates the XHTML page for the transcription. We continue the example using only the transcription in English, assuming that the process is the same for the other transcriptions.

To create the XHTML page, the tool takes advantage of the metadata that resides in the inventory of YouTube. This database contains, for each YouTube video, among other metadata, the title and the description entered by the video owner. To extract the metadata, the tool uses the *video ID* parameter provided by the user; this represents a URI that unique and permanently identifies a feed or a feed entry. The tool extracts content-independent metadata from the feed entry by sending a request to its URI. The feed entry of the given *video id* is shown in the Figure 16.

```
http://gdata.youtube.com/feeds/api/videos/FPNzA8fCe_s
```

Figure 16 Extraction of content-independet Metadata.

Then, the tool adds the metadata extracted from the feed entry to the XHTML page: a *div section* contains the title and description of the A/V asset, inside a *h1* tag and a *p* tag, respectively. This is shown in Figure 17.

```
...
<div>
<h1>
     Everything becomes, Chemical Reactions, and Fritz
     Haber synthesis of ammonia
</h1>
<p>
     Audiovisual production produced by the PUC Rio in
     partnership with the Ministry of Education, Ministry
     of Science and Technology and the National Fund for
     Educational Development. Integrates a series of six
     programs (120 episodes) dedicated to supporting the
     teaching of chemistry in high school.
</p>
</div>
...
```

Figure 17 XHTML-ize process on the metadata.

Once the content-independent metadata is added to the Web page, the tool processes the transcription. Here an anchor tag represents each caption and the *href* attribute references the exact segment in the A/V asset where the speech occurs. To configure the *href* attribute, the tool uses section 2 of the SubRip text file, which contains the start time code of the caption. This is illustrated in Figure 18.

Observe that each static Web Page thus generated specifies the document type and the language of that content. In the example of Figure 18, the document type is XHTML (4.01 strict) and the language is English (xml:lang=”en”).

```
 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

 ...
<div>
<a href="URL?time=04m47s">Watch this excerpt</a>
<p>

       Against this backdrop comes Fritz Haber. The man who
       carried out the synthesis of ammonia in order to
       produce it on an industrial scale, from molecular
       hydrogen and nitrogen, abundantly available.

</p>
</div>
 ...
```

Figure 18 XHML-ize process on the transcription.

The tool starts the enrichment process by annotating the content as in Figure 17, where the Dublin core metadata schema is used (Dublin Core offers fifteen properties to describe resources [41]). The effects of enriching the content can be seen in Figure 19.

```
...
<div xmlns:dc="http://purl.org/dc/elements/1.1/">
<div about="http://example.com/spokenContent">
<h1 property="dc:title">
     Everything becomes, Chemical Reactions, and Fritz
     Haber synthesis of ammonia
</h1>
<p property="dc:description">
     Audiovisual production produced by the PUC Rio in
     partnership with the Ministry of Education, Ministry
     of Science and Technology and the National Fund for
     Educational Development. Integrates a series of six
     programs (120 episodes) dedicated to supporting the
     teaching of chemistry in high school.

</p>
</div>
</div>
...
```

Figure 19 Dublin Core enrichment.

We can represent the annotations in Figure 20 with the help of a diagram that shows the connecting URLs properties.

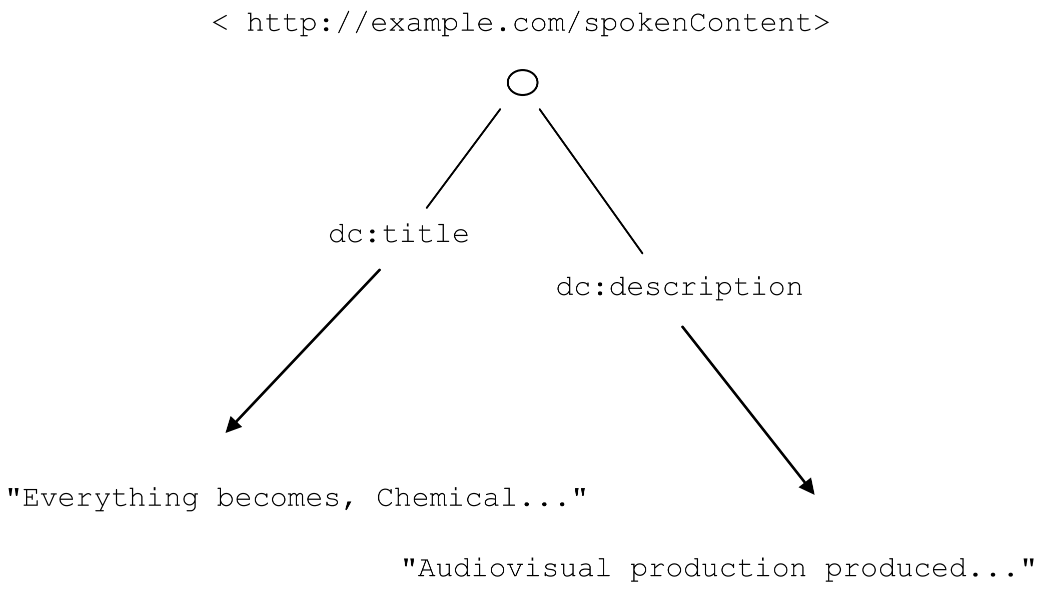

< http://example.com/spokenContent>
dc:title
dc:description
"Everything becomes, Chemical..."
"Audiovisual production produced..."

Figure 20 Diagram representation of the annotation in Figure 19.

The next step of the enrichment process is to add machine-readable data to the div sections that contains the captions. This proceeds by passing to the DBpedia Spotlight service endpoint as input the two parameters provided by the user (confidence and support). The service automatically annotates references to DBpedia resources in the Web page, thereby linking unstructured information sources to the Linked Open Data cloud through DBpedia. This is performed by locating entities in the plain text through named entity extraction, including entity detection and name resolution.

We will show in Figure 21 how to consume the Dbpedia endpoint using the given parameters. Assume the following text caption will be enriched.

- Text caption:"Against this backdrop comes Fritz Haber. The man who carried out the synthesis of ammonia in order to produce it on an industrial scale, from molecular hydrogen and nitrogen, abundantly available."

```
http://spotlight.dbpedia.org
/rest/annotate?text=Against%20this%20backdrop%20comes%20Fritz%2
0Haber.%20The%20man%20who%20carried%20out%20the%20synthesis%20o
f%20ammonia%20in%20order%20to%20produce%20it%20on%20an%20indust
rial%20scale,%20from%20molecular%20hydrogen%20and%20nitrogen,%2
0abundantly%20available &confident=0.6&support=40
```

Figure 21 Request to the SpotLight service endpoint.

As we can see, the confident and support are sent as parameters in the HTTP request. The output of the Web service can be requested in different content types, such as: «text/html», «application/xhtml+xml», «text/xml», «application/json». In our application, we use the «application/xhtml+xml» type that comes with embedded RDFa.

Figure 22 shows the XHTML code and the effects of the process of enrichment with RDFa, as applied to the document in Figure 18.

```
...
     Against this backdrop comes <a
     about="http://dbpedia.org/resource/Fritz_Haber"
     typeof="http://dbpedia.org/ontology/Scientist"
     href="http://dbpedia.org/resource/Fritz_Haber"
     title="http://dbpedia.org/resource/Fritz_Haber">Fritz
     Haber</a>. The man who carried out the <a
     about="http://dbpedia.org/resource/Chemical_synthesis"
     href="http://dbpedia.org/resource/Chemical_synthesis"
     title="http://dbpedia.org/resource/Chemical_synthesis">
     synthesis</a> of <a
     about="http://dbpedia.org/resource/Ammonia"
     typeof="http://dbpedia.org/ontology/ChemicalCompound"
     href="http://dbpedia.org/resource/Ammonia"
     title="http://dbpedia.org/resource/Ammonia">ammonia</a>
     in order to produce it on an industrial scale, from <a
     about="http://dbpedia.org/resource/Hydrogen"
     href="http://dbpedia.org/resource/Hydrogen"
     title="http://dbpedia.org/resource/Hydrogen">molecular
     hydrogen</a> and <a
     about="http://dbpedia.org/resource/Hydrogen"
     href="http://dbpedia.org/resource/Hydrogen"
     title="http://dbpedia.org/resource/Hydrogen">nitrogen</
     a>, abundantly available.
...
```

Figure 22 Example of an enriched Web document.

The first identified entity was annotate with the @typeof attribute that indicates a new data item “scientist”. That is, it asserts that “Fritz Haber” is of type “scientist”. Figure 23 represents this scenario as a graph connecting URLs.

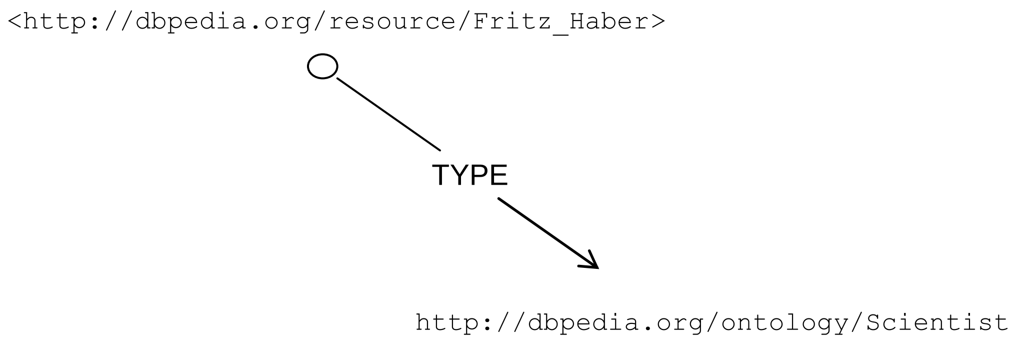

<http://dbpedia.org/resource/Fritz_Haber>
TYPE
http://dbpedia.org/ontology/Scientist

Figure 23 Graph representation of the annotated content in Figure 22.

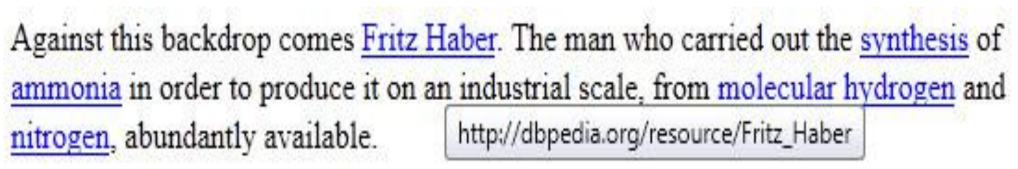
The triple that this graph represents is written, using the Notation3 syntax [N3], as follows:

```
<http://dbpedia.org/resource/Fritz_Haber>
http://www.w3.org/2000/01/rdf-schema#Datatype>
<http://dbpedia.org/ontology/Scientist>;
```

Figure 24 [N3] syntax of the Graph in Figure 23.

In

Figure 25 can be seen the HTML presentation of the code in Figure 22.



Against this backdrop comes Fritz Haber. The man who carried out the synthesis of ammonia in order to produce it on an industrial scale, from molecular hydrogen and nitrogen, abundantly available.

http://dbpedia.org/resource/Fritz_Haber

Figure 25 HTML representation of the annotation stage.

To make content navigation easier, the tool embeds in the Web page a YouTube player that is implemented using the YouTube Javascript player API. Thus, the user can make calls to pause, play, seek to a certain time in a video, set the volume, and mute the player.

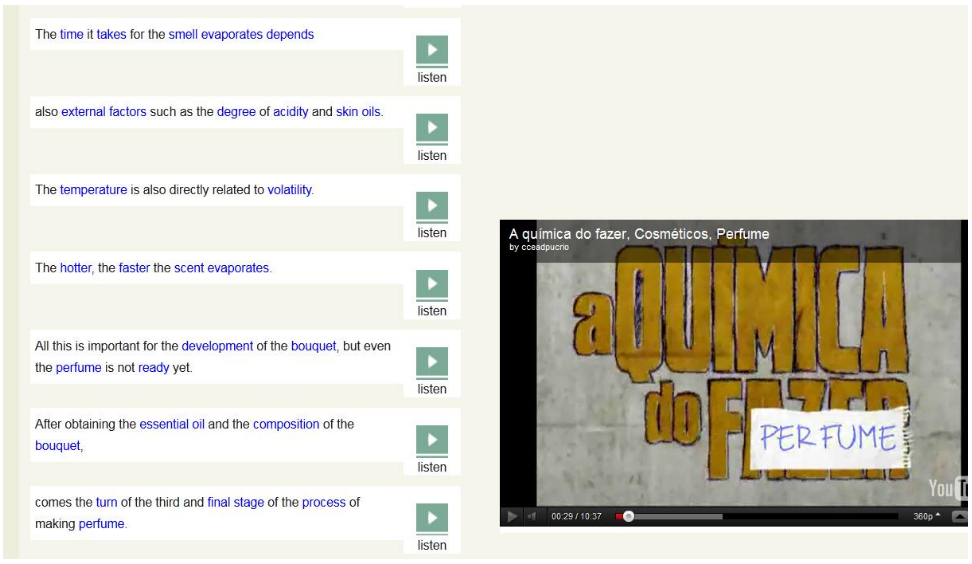
The user interface lets the user start to playback a video from any desire paragraph. The seek function changes the start time, as illustrated below:

player.seekTo(seconds:Number, allowSeekAhead:Boolean):Void

- The first parameter identifies the time to which the player should advance to. The player will advance to the closest keyframe before that time, unless the player has already downloaded the portion of the video to which the user want to jump to. In that case, the player will advance to the closest keyframe.
- The allowSeekAhead parameter determines whether the player will make a new request to the server if the seconds parameter specifies a time outside of the currently buffered video data.

Figure 26 shows the generated XHTML page, where each paragraph is displayed together a “listen” button, which the user can activate to playback the video from the point corresponding to the paragraph. This UI is very useful because it improves the browsing experience and helps decrease the time to locate a particular point in a video.

The tool ends processing an input video by returning a link that points to the generated XHTML page, which can be displayed in the same browser or downloaded to be published in another Web server.



The time it takes for the smell evaporates depends

listen

also external factors such as the degree of acidity and skin oils.

listen

The temperature is also directly related to volatility.

listen

The hotter, the faster the scent evaporates.

listen

All this is important for the development of the bouquet, but even the perfume is not ready yet.

listen

After obtaining the essential oil and the composition of the bouquet,

listen

comes the turn of the third and final stage of the process of making perfume.

listen

Figure 26 Spoken Content page.