

VoiceForge

1. BASIC INFORMATION (Overview and purpose of the tool)

1. Webservice name
VoiceForge Webservice (TTL2Ro)
2. Overview and purpose of the webservice
The VoiceForge Webservice provides a set of tools for Romanian text-to-speech (TTS) synthesis
3. A short description of the service
The service performs the following TTS specific task: text normalization, part-of-speech tagging, lexical stress assignment, phonetic transcription, syllabification and voice synthesis

2. TECHNICAL INFORMATION

1. WSDL code for invoking the web-service

The WSDL code for invoking the web service is accessible online at:

<http://nlptools.racai.ro:8080/tts/TTL2ro.asmx?WSDL>

Raw version given below:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="About">
        <s:complexType />
      </s:element>
      <s:element name="AboutResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="AboutResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ProcessTextToXML">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="text" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ProcessTextToXMLResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ProcessTextToXMLResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="ProcessText">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="text" type="s:string" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="ProcessTextResponse">
    <s:complexType>
        <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ProcessTextResult"
type="tns:TaggerResult" />
        </s:sequence>
    </s:complexType>
</s:element>
<s:complexType name="TaggerResult">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Paragraphs" type="tns:ArrayOfParagraph" />
    </s:sequence>
</s:complexType>
<s:complexType name="ArrayOfParagraph">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="Paragraph" nillable="true"
type="tns:Paragraph" />
    </s:sequence>
</s:complexType>
<s:complexType name="Paragraph">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Sentences" type="tns:ArrayOfSentence" />
    </s:sequence>
</s:complexType>
<s:complexType name="ArrayOfSentence">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="Sentence" nillable="true"
type="tns:Sentence" />
    </s:sequence>
</s:complexType>
<s:complexType name="Sentence">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Tokens" type="tns:ArrayOfToken" />
    </s:sequence>
</s:complexType>
<s:complexType name="ArrayOfToken">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="Token" nillable="true"
type="tns:Token" />
    </s:sequence>
</s:complexType>

```

```

    </s:sequence>
  </s:complexType>
  <s:complexType name="Token">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="Syllables" type="tns:ArrayOfSyllable" />
    </s:sequence>
    <s:attribute name="ChunkID" type="s:string" />
    <s:attribute name="ProcessedWord" type="s:string" />
    <s:attribute name="Word" type="s:string" />
    <s:attribute name="Lemma" type="s:string" />
    <s:attribute name="POS" type="s:string" />
    <s:attribute name="MSD" type="s:string" />
    <s:attribute name="PhoneticTranscription" type="s:string" />
    <s:attribute name="ToBIBreak" type="s:int" use="required" />
  </s:complexType>
  <s:complexType name="ArrayOfSyllable">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded" name="Syllable" nillable="true"
type="tns:Syllable" />
    </s:sequence>
  </s:complexType>
  <s:complexType name="Syllable">
    <s:attribute name="Lexical" type="s:string" />
    <s:attribute name="Phonetic" type="s:string" />
    <s:attribute name="ToBILabel" type="s:string" />
    <s:attribute name="LexicalStress" type="s:boolean" use="required" />
    <s:attribute name="DurationMod" type="s:double" use="required" />
    <s:attribute name="F0Mod" type="s:double" use="required" />
  </s:complexType>
  <s:element name="TTL2Raw">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="text" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="TTL2RawResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="TTL2RawResult" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="Synthesize">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="rawFormat" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>

```

```

        </s:complexType>
    </s:element>
    <s:element name="SynthesizeResponse">
        <s:complexType>
            <s:sequence>
                <s:element minOccurs="0" maxOccurs="1" name="SynthesizeResult" type="s:base64Binary"
/>
            </s:sequence>
        </s:complexType>
    </s:element>
</s:schema>
</wsdl:types>
<wsdl:message name="AboutSoapIn">
    <wsdl:part name="parameters" element="tns:About" />
</wsdl:message>
<wsdl:message name="AboutSoapOut">
    <wsdl:part name="parameters" element="tns:AboutResponse" />
</wsdl:message>
<wsdl:message name="ProcessTextToXMLSoapIn">
    <wsdl:part name="parameters" element="tns:ProcessTextToXML" />
</wsdl:message>
<wsdl:message name="ProcessTextToXMLSoapOut">
    <wsdl:part name="parameters" element="tns:ProcessTextToXMLResponse" />
</wsdl:message>
<wsdl:message name="ProcessTextSoapIn">
    <wsdl:part name="parameters" element="tns:ProcessText" />
</wsdl:message>
<wsdl:message name="ProcessTextSoapOut">
    <wsdl:part name="parameters" element="tns:ProcessTextResponse" />
</wsdl:message>
<wsdl:message name="TTL2RawSoapIn">
    <wsdl:part name="parameters" element="tns:TTL2Raw" />
</wsdl:message>
<wsdl:message name="TTL2RawSoapOut">
    <wsdl:part name="parameters" element="tns:TTL2RawResponse" />
</wsdl:message>
<wsdl:message name="SynthesizeSoapIn">
    <wsdl:part name="parameters" element="tns:Synthesize" />
</wsdl:message>
<wsdl:message name="SynthesizeSoapOut">
    <wsdl:part name="parameters" element="tns:SynthesizeResponse" />
</wsdl:message>
<wsdl:portType name="TTL2roSoap">
    <wsdl:operation name="About">
        <wsdl:input message="tns:AboutSoapIn" />
        <wsdl:output message="tns:AboutSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="ProcessTextToXML">

```

```

    <wsdl:input message="tns:ProcessTextToXMLSoapIn" />
    <wsdl:output message="tns:ProcessTextToXMLSoapOut" />
</wsdl:operation>
<wsdl:operation name="ProcessText">
    <wsdl:input message="tns:ProcessTextSoapIn" />
    <wsdl:output message="tns:ProcessTextSoapOut" />
</wsdl:operation>
<wsdl:operation name="TTL2Raw">
    <wsdl:input message="tns:TTL2RawSoapIn" />
    <wsdl:output message="tns:TTL2RawSoapOut" />
</wsdl:operation>
<wsdl:operation name="Synthetize">
    <wsdl:input message="tns:SynthetizeSoapIn" />
    <wsdl:output message="tns:SynthetizeSoapOut" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="TTL2roSoap" type="tns:TTL2roSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="About">
        <soap:operation soapAction="http://tempuri.org/About" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ProcessTextToXML">
        <soap:operation soapAction="http://tempuri.org/ProcessTextToXML" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ProcessText">
        <soap:operation soapAction="http://tempuri.org/ProcessText" style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="TTL2Raw">
        <soap:operation soapAction="http://tempuri.org/TTL2Raw" style="document" />
        <wsdl:input>

```

```
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
  <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="Synthesize">
  <soap:operation soapAction="http://tempuri.org/Synthesize" style="document" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="TTL2roSoap12" type="tns:TTL2roSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="About">
    <soap12:operation soapAction="http://tempuri.org/About" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ProcessTextToXML">
    <soap12:operation soapAction="http://tempuri.org/ProcessTextToXML" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ProcessText">
    <soap12:operation soapAction="http://tempuri.org/ProcessText" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="TTL2Raw">
    <soap12:operation soapAction="http://tempuri.org/TTL2Raw" style="document" />
    <wsdl:input>
```

```

    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="Synthetize">
  <soap12:operation soapAction="http://tempuri.org/Synthetize" style="document" />
  <wsdl:input>
    <soap12:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal" />
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="TTL2ro">
  <wsdl:port name="TTL2roSoap" binding="tns:TTL2roSoap">
    <soap:address location="http://nlptools.racai.ro:8080/tts/TTL2ro.asmx" />
  </wsdl:port>
  <wsdl:port name="TTL2roSoap12" binding="tns:TTL2roSoap12">
    <soap12:address location="http://nlptools.racai.ro:8080/tts/TTL2ro.asmx" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

2. Software dependencies and system requirements (if any)

Not applicable

3. Execution instructions

The Webservice provides the following methods:

- About
- ProcessText (string): TaggerResult– runs the text-processing frontend on the input string returning a TaggerResult object.
- ProcessTextToXML (string):string - runs the text-processing frontend on the input string returning a XML serialized TaggerResult object
- Synthetize (string): byte[] – synthetizes the voice given the raw input produced by the TTL2Raw method
- TTL2Raw (string):string – generates a raw encoding used by the voice synthesis module. The input string is the XML Serialization of the TaggerResult object returned by the ProcessTextToXML() method

4. Input/Output data formats

a. Input data formats

The initial input for the Web service is UTF8 encoded text.

b. Output data formats

The output of the Web service is a WAV file encoded as a raw byte array.

5. Integration with external tools

Not applicable

3. CONTENT INFORMATION

1. an usage example with associated data

A C# example is provided for invoking the WebService (presuming the SOAP client is instanced as the “tts” variable):

```
2. string s = tts.ProcessTextToXML("Aceasta este un test.");
3. string raw = tts.TTL2Raw(s);
4. byte[] wav = tts.Synthesize(raw);
5. FileStream fs = new FileStream("test.wav", FileMode.Create);
6. fs.Write(wav, 0, wav.Length);
7. fs.Close();
```

8. an example of the output data



9. Approximation of the time necessary to process the test input file.

The entire processing time (including network lags) on the input data was 5 seconds + 1 second for establishing the connection (total 6 seconds).

4. ADMINISTRATIVE INFORMATION

1. Contact

Tiberiu Boroş (tibi@racai.ro)

5. RELEVANT REFERENCES AND OTHER INFORMATION

1. <http://nlptools.racai.ro>
2. Boroş Tiberiu, Radu Ion, Dan Ştefănescu, Bermuda, a data-driven tool for phonetic transcription of words, Proceedings of Language Resources and Evaluation 2012, 978-2-9517408-7-7, 2012
3. Boroş Tiberiu, Blind Speech Segmentation applied to the Romanian Language, Proceedings of the 8th International Conference “Linguistic Resources and Tools of the Romanian Language”, 1843-911X, 2012
4. Tiberiu Boroş, Speech Segmentation for Collecting Non-Uniform Speech Units, Proceedings of Language & Technology Conference, Poznan, Poland, 25-27 November, 2011