Multidocument Concordances in Unitex

Nebojša Vasiljević

Ministry of Foreign and Internal Trade and Telecommunications
nebojsa.vasiljevic@mtt.gov.rs

Abstract. Unitex is a software system for the analysis of texts in natural language based on electronic dictionaries and local grammars [2]. It is designed to process a single text file at a time. For instance, Unitex can generate a concordance for a given text file and a given query, where the query can be a regular expression or a syntax graph. If a corpus consists of many documents, we need to join them in a single text file somehow, regardless of whether those documents are stored in a collection of text files in a folder or in a corpus management system. The simplest solution is just to concatenate several documents into a single text file, but then the document collection structure (what belongs to which document) will be lost. This paper will describe a software solution for multidocument concordances in Unitex that preserves document collection structure and present it properly in concordances. The solution is implemented as a Java program that relies on the standard Unitex *locate* and *concord* commands and do some additional pre- and post-processing. The final output is in the HTML format. The solution also supports overlapped concordances of two similar queries, particularly suitable when one query is looser than the other.

Keywords: Concordances, Corpus, Unitex

1 Introduction

Unitex is a software system for the analysis of texts in natural language [2]. Unitex is based on electronic dictionaries and local grammars and it has been developed at Laboratoire d'Automatique Documentaire et Linguistique (LADL) within the University of Paris-Est Marne-la-Vallée.

A text to be processed with Unitex should be stored in a single Unicode encoded[1] plain text file [4]. Before other operations, the source text should be pre-processed. Unitex pre-processing includes separator normalization, tokenization, sentence segmentation and dictionary application. The result of pre-processing is called tagged text.

One of the most common use cases of Unitex is to perform pattern matching queries on a tagged text and browse resulting concordance. A query can be specified as a regular expression or as a local grammar in the form of a syntax graph. An example of a syntax graph is shown on Figure 1.

[1] Both UTF-8 and UTF-16 encoding forms are supported.

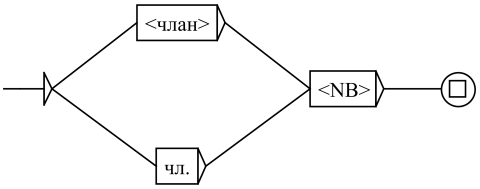

Fig. 1: A syntax graph in Unitex

The syntax graph matches any inflected form of the lemma "члан" (article) or the abbreviation "чл.", followed by a number. A morphological electronic dictionary is used to resolve inflected forms for a given lemma. Figure 2 shows a part of the concordance containing matches of the syntax graph in a legislative text[2].

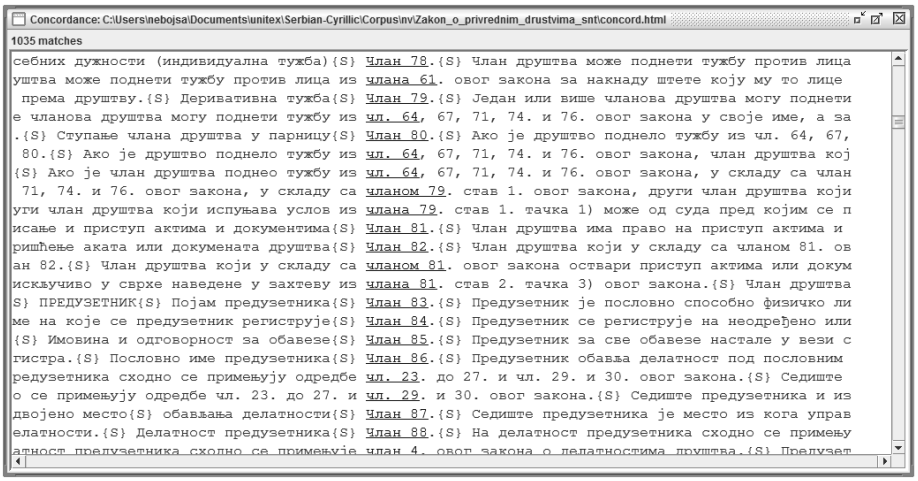

Fig. 2: Concordance for the graph from Figure 1

Let us now suppose that we need to find occurrences of the term "кривично дело" (criminal offence) in different laws. Since Unitex always processes a single text file at a time, we should concatenate selected laws into a single file. If we do so and build the appropriate concordance, the result will resemble the one shown in Figure 3. Criminal offences are usually mentioned in penalty provisions near the end of a legal act. When we browse this kind of concordances, besides the left and right context of matches, we need to know which document a match belongs to. Even if we concatenate multiple documents into a single text file,

[2] The Law on Business Companies (in Serbian) is used in this example.

Unitex cannot present in concordance what belongs to which document. That is why we developed a tool described below.

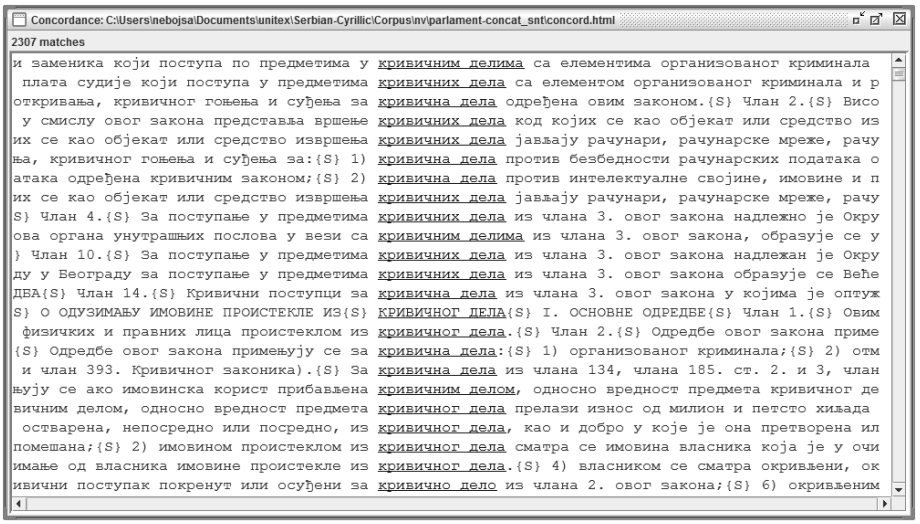

Fig. 3: Concordance for the term "кривично дело" (criminal offence)

2 Multidocument Corpora

Even though a corpus usually consists of a number of documents, Unitex does not have a built-in support to handle multidocument corpora. As we mentioned at the end of the previous section, documents should be concatenated prior to being processed with Unitex, but we also need to solve how to track source document identities. The solution consists of two parts:

1. inserting the documents' metadata when concatenating;
2. generating concordances with information on the documents to which matches belong.

Document's metadata is a line of text in the form of a Unitex's lexical tag:

{*<Title>*,.META+Title}

For instance:

{Закон о измени Закона о сточарству (2012),.META+Title}

Since a lexical tag is tokenized as a single token, parts of the document's metadata will not be confused with the regular text during the Unitex's processing.

Metadata of a document are inserted in a concatenated text file before the content of the document.

Implementation of the concatenation with the metadata insertion depends on how a corpus is stored (in a file system, corpus management system, etc.). This part of processing can also be done manually, using a plain text editor, if there are not too many documents in a corpus. A concatenated corpus should be pre-processed in Unitex like any other text.

A program in the Java programming language [1] is implemented to generate concordances with source document identities using a concatenated corpus described above. In the first phase, the program executes Unitex's commands *locate* and *concord* to produce a concordance in a usual way. In the second phase, the program does an additional processing to produce the resulting concordance in HTML [3]. The result is shown on Figure 4. It corresponds to the concordance from Figure 3.

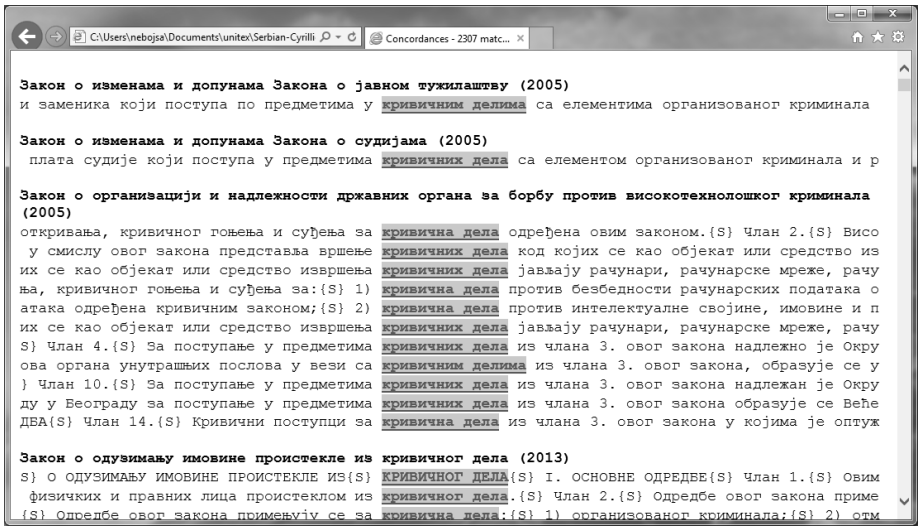

Fig. 4: Concordance for the term "кривично дело" (criminal offence)

Each match is represented as a hyperlink which leads to the appropriate place in the full text (Figure 5). Tags "{S}", that we can see on the figure, are the sentence delimiters inserted during the Unitex's pre-processing. So, the resulting HTML page, below the concordance (Figure 4), contains the full text of all documents (Figure 5). The advantage of the single HTML file approach with intra-page hyperlinks is that this kind of file is easy to exchange: you can just send a single HTML file and just open a received file in the default web browser. The disadvantage of the single HTML file approach is manifested when you have a big corpus. In this case a single HTML file becomes huge, sometimes too demanding for a web browser, when the browser becomes too slow or even

cannot open the page properly. That is why we are considering implementing an optional multi-page output, where the full text of each document will be stored in a separate HTML file. Another possible direction of the further development is using server-side technologies to generate HTML dynamically.

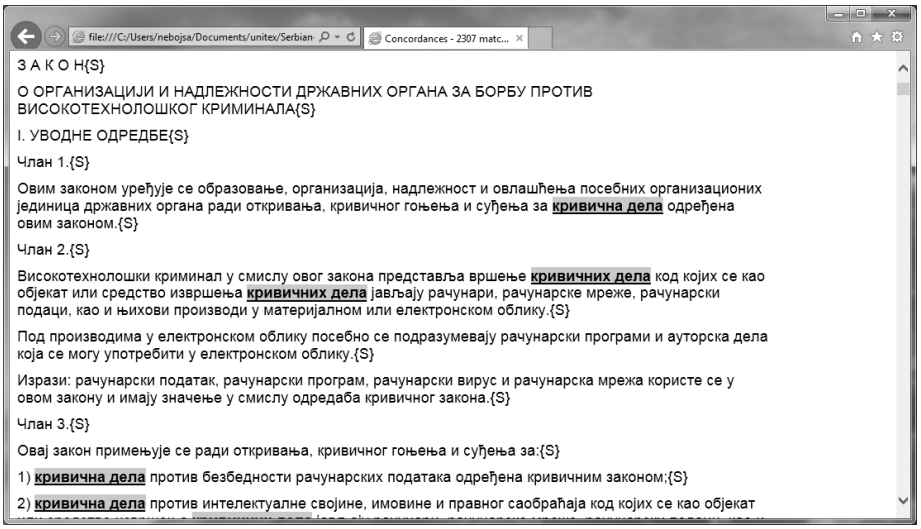

Fig. 5: Matches of the term "кривично дело" (criminal offence) in the legislation text

3 Overlapped Concordance

Another feature implemented in the concordance generating program is to display overlapped matches from two queries. The feature is particularly suitable when one query is looser than the other, where looser means that matches of the first query cover matches of the second query. Overlapped concordance helps us to analyse differences between a loose and a strict query in order to test syntax rules or to detect errors.

An example of overlapped concordances is shown on Figure 6, for a loose and a strict query for normative references. The light blue background is used for the strict query matches and the red background (darker in black and white print) is used for the parts of the loose query matches that are not covered by the strict query matches. In this example, all cases where you can see red (darker) background are the result of a broken rule of the references syntax: comma instead of "и" (and), dot after a number's suffix (a dot should be avoided after suffixed numbers) and incompatible starting levels of references in a list (the last shown case).

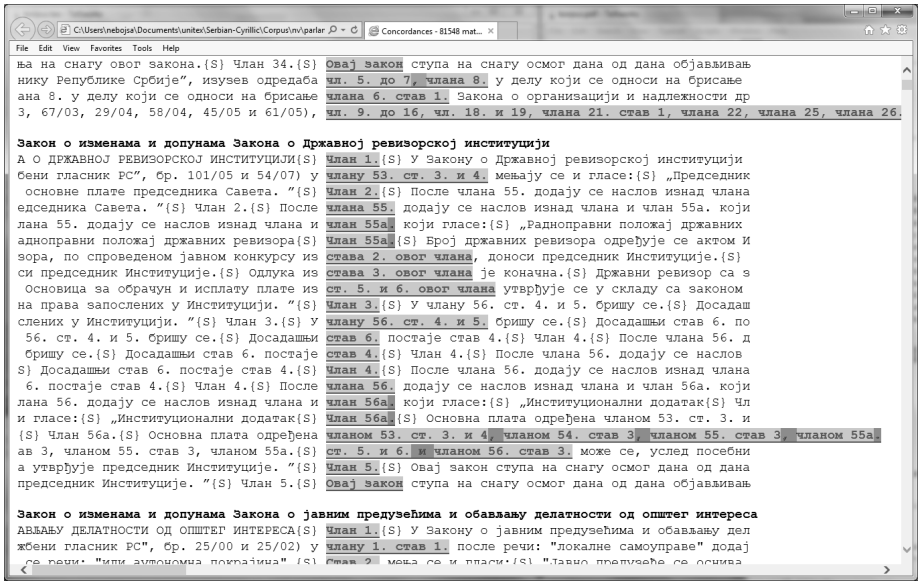

Fig. 6: Overlapped concordance

4 Conclusions

Unitex does not have a built-in multidocument concordances support, but it is possible to implement an appropriate custom software solution based on the Unitex's command line interface and the well documented output file formats. Besides the multidocument support, there are other possible additional concordance features like overlapped concordance, which is successfully demonstrated by the concordance generating software described in this paper.

References

1. Ken Arnold, James Gosling, and David Holmes. *The Java Programming Language*. Java (Prentice Hall). Prentice Hall, 2012.
2. Sébastien Paumier. *Unitex 3.1 User Manual*. Available at `http://www-igm.univ-mlv.fr/~unitex/UnitexManual3.1.pdf`.
3. Dave Raggett, Arnaud Le Hors, Ian Jacobs, et al. HTML 4.01 Specification. *W3C recommendation*, 24, 1999.
4. The Unicode Consortium. *The Unicode Standard, Version 6.2.0*, 2012.