Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 493–500
ISBN 978-83-60810-27-9
ISSN 1896-7094

Tools for Syntactic Concordancing

Violeta Seretan and Eric Wehrli
LATL - Language Technology Laboratory
Department of Linguistics, University of Geneva
Email: {violeta.seretan, eric.wehrli}@unige.ch

***Abstract*—Concordancers are tools that display the immediate context for the occurrences of a given word in a corpus. Also called KWIC – Key Word in Context tools, they are essential in the work of lexicographers, corpus linguists, and translators alike. We present an enhanced type of concordancer, which relies on a syntactic parser and on statistical association measures in order to detect those words in the context that are syntactically related to the sought word and are the most relevant for it, because together they may participate in multi-word expressions (MWEs). Our syntax-based concordancer highlights the MWEs in a corpus, groups them into syntactically-homogeneous classes (e.g., verb-object, adjective-noun), ranks MWEs according to the strength of association with the given word, and for each MWE occurrence displays the whole source sentence as a context. In addition, parallel sentence alignment and MWE translation techniques are used to display the translation of the source sentence in another language, and to automatically find a translation for the identified MWEs. The tool also offers functionalities for building a MWE database, and is available both off-line and online for a number languages (among which English, French, Spanish, Italian, German, Greek and Romanian).**

I. Introduction

Knowledge of a word means knowledge of the relations that this word establishes with other words: "You shall know a word by the company it keeps!" [1, p. 179]. Hence, the study of words in context—in order to analyse how words are actually used and what their typical contexts are—is a major concern in any field dealing with language from diverse perspectives, no matter whether it is theoretically or practically motivated.

The advent of the computer era and the ever-increasing availability of texts in digital format allow for virtually unlimited exploration. Yet, this is at the same time one of the biggest issues that users presented with automatically detected contexts inevitably have to face. The information comes to them as huge amounts of unstructured data, characterised by a high degree of redundancy.

To help them overcome the problem of information overload, a new generation of concordancers have been developed that are able to pre-process textual data such that the most relevant contextual information comes first [2]. This is achieved using lexical association measures that quantify the degree of interdependence between words, by relying on statistical hypothesis tests, on concepts from information theory, on data mining techniques, or by making use of various other methods ([3], [4]).

A representative example of such a concordancer is the Sketch Engine [5]. It analyses a preexisting corpus of text in order to produce, for a given word, a one-page summary of its grammatical and collocational behaviour. In doing so, it first performs a shallow parsing of the corpus by relying on automatically assigned POS tags for words, then it applies an association measure derived from Pointwise Mutual Information [6]. For illustration, Figure 1 shows part of the "sketch" produced for the French word *atteindre* ("to reach, to attain"). By clicking on the links in the frequency column, users have the possibility to see the actual concordance line, with a left and right context for each instance found in the corpus.

Developed more or less simultaneously with the Sketch Engine, our concordancer *FipsCo* ([7], [8], [9]) shares several similarities with it, and was primarily designed with the specific goal of being integrated as a new type of tool in the workbench of translators from an international organisation. In this paper, we describe this tool, its underlying resources and methodology, its latest developments, and we present the manner in which it is currently integrated in the larger, evolving processing environment available in our laboratory.

The paper is organised as follows. Section II provides an overview of FipsCo. Section III presents the resources and methods on which it is founded. Section IV describes in greater detail its functionalities, then Section V introduces FipsCoWeb, its recently developed online version. Section VI discusses the manner in which FipsCo and FipsCoWeb are integrated into the larger language processing environment of LATL. The last section contains concluding remarks.

II. FipsCo: An Overview

In FipsCo, the system of syntax-based collocation extraction and concordancing developed in our laboratory, the input text is first syntactically analysed with a full parser, then it is processed with standard statistical methods which measure the strength of association between words.

Collocation, understood as "typical, specific and characteristic combination of two words" [10], is a generic term used here to encompass all syntactic word combinations found in a corpus that are relevant to the studied word, from a lexicographic point of view. As in [11], we consider that collocation refers, more generally, to "the way words combine in a language to produce natural-sounding speech and writing".

This concept is allowed to overlap with other types of MWEs, like compounds (e.g., *wheel chair*), phrasal verbs (e.g, *to ask [somebody] out*) or certain types of less figurative idioms (*to open the door [for smth]* "to allow [smth] to happen").

modifier	3650	1.0	objet	10314	4.6
gravement	77	40.88	paroxysme	77	45.26
mortellement	20	28.95	apogée	82	44.76
enfin	108	28.64	but	462	42.85
jamais	243	28.42	objectif	378	42.16
bientôt	65	25.56	sommet	209	39.25
rapidement	51	23.45	maturité	71	34.24
rarement	28	22.9	niveau	310	33.63
pas	671	21.78	âge	182	31.36
presque	53	21.62	degré	111	30.03
facilement	31	21.11	perfection	65	29.46
encore	121	19.9	limite	133	29.39
péniblement	12	19.68	stade	65	27.47
finalement	30	19.36	milliard	60	27.23
déjà	87	18.12	cible	62	26.81
parfois	38	17.86	maximum	65	26.45
grièvement	8	17.78	seuil	59	26.07
profondément	18	16.75	patient	63	26.03
directement	23	16.65	plénitude	26	23.46
plus	162	16.46	mètre	69	23.17
ne	470	16.22	summum	13	23.1
désormais	23	15.49	personne	294	22.79
maintenant	34	15.38	vitesse	65	22.28
ainsi	60	15.2	hauteur	51	22.26
même	73	14.98	million	65	21.59
près	19	14.61	tödlichen	5	21.36

Fig. 1. The Sketch Engine [5]: Sample (partial) output, showing collocates for the French verb *atteindre*, “to reach, to attain”.

The boundaries between the different kinds of MWEs are known to be particularly difficult to be drawn, as they are rather fuzzy [12]. From a practical point of view, all MWEs pose similar processing problems, regardless of any finer-grained classification. We will henceforth refer to the output of our system as to collocations (or collocation candidates), without making further, more elaborate distinctions.

As the Sketch Engine [5], our system outputs collocation candidates grouped by types (which conflate all the instances of a specific word combination detected in the corpus), partitioned into syntactically homogeneous classes, and ranked in the reverse order of the association strength. Thus, the user may easily consult a manageable amount of contextual data, consisting of the most relevant collocates. The data presented are organised and, to a certain extent, free of redundancy.[1]

Figure 2 shows some results obtained from a French corpus by using FipsCo. These results were filtered by the user so that they contain collocations for the sought word (in this case, the verb *atteindre*, “to reach, to attain”) in a specific syntactic configuration; the configuration retained here was verb-object. According to the association measure applied, the noun the most collocationally related to this word is *objectif* (“objective”). It was detected 271 times in the source corpus, and it is indeed a good collocate candidate, as one might easily agree that *atteindre un objectif* (“to reach a goal”) is a collocation in French.

The concordancer displays its 100th instance in the corpus, which, as can be seen, involves a rather complex syntactic context: the order of the items in the collocation is inverted, the items are inflected and not in the base word form, and there is additional material inserted in between. Due to a grammatical transformation (passivization), the original verb-object combination is realized, at the surface level, as a subject-verb combination. The identification of these type of complex cases, which are particularly difficult to handle by pattern-based shallow parsers, is possible in our system thanks to the deep analysis provided by the parser (cf. Section III).

Among the other combinations shown in Figure 2, one might find several other MWEs with the verb *atteindre*. The tool also presents the automatically retrieved translation of the context in another language, if parallel corpora are available. In a translation environment, such corpora are typically available from translation archives. Thus, when working on a new document, translators have the possibility to see how a given expression has previous been translated in various contexts.

Figure 2 also shows the buttons *Validate*, used for manually validating the automatically extracted results, and *Translate*, used for automatically detecting a translation for the selected collocations. The *Filter* button opens the interface in Figure 3, through which the user can control which corpus results to display.

FipsCo is freely available for research as an offline tool for Windows (cf. Section IV). One of the latest developments concerned the creation of a lighter-weight online version, which has already been made available to the public. A more detailed description of FipsCo and its Web version, FipsCoWeb, is provided in Section IV.

III. Underlying Resources and Methodology

This section provides details about the resources used by FipsCo and the method used to extract from text corpora the most relevant collocates for a given word.

A. Resources

FipsCo was built as an extension of Fips, a multilingual symbolic parser based on generative grammar concepts [13]. Fips can be characterised as a strong lexicalist, bottom-up, left-to-right parser. Given a sentence, it builds a rich structural representation combining *a*) the constituent structure; *b*) the interpretation of constituents in terms of arguments; *c*) the interpretation of elements like clitics, relative and interrogative pronouns in terms of intra-sentential antecedents; and *d*) co-indexation chains linking extraposed elements (e.g., fronted NPs and wh elements) to their canonical positions.

According to the theoretical stipulations on which Fips relies, some constituents of a sentence may move from their canonical “deep” position to surface positions, due to various grammatical transformations. For instance, in the case of

[1] All the corpus instances (tokens) of a same word combination are grouped under a single entry, the corresponding collocation type.

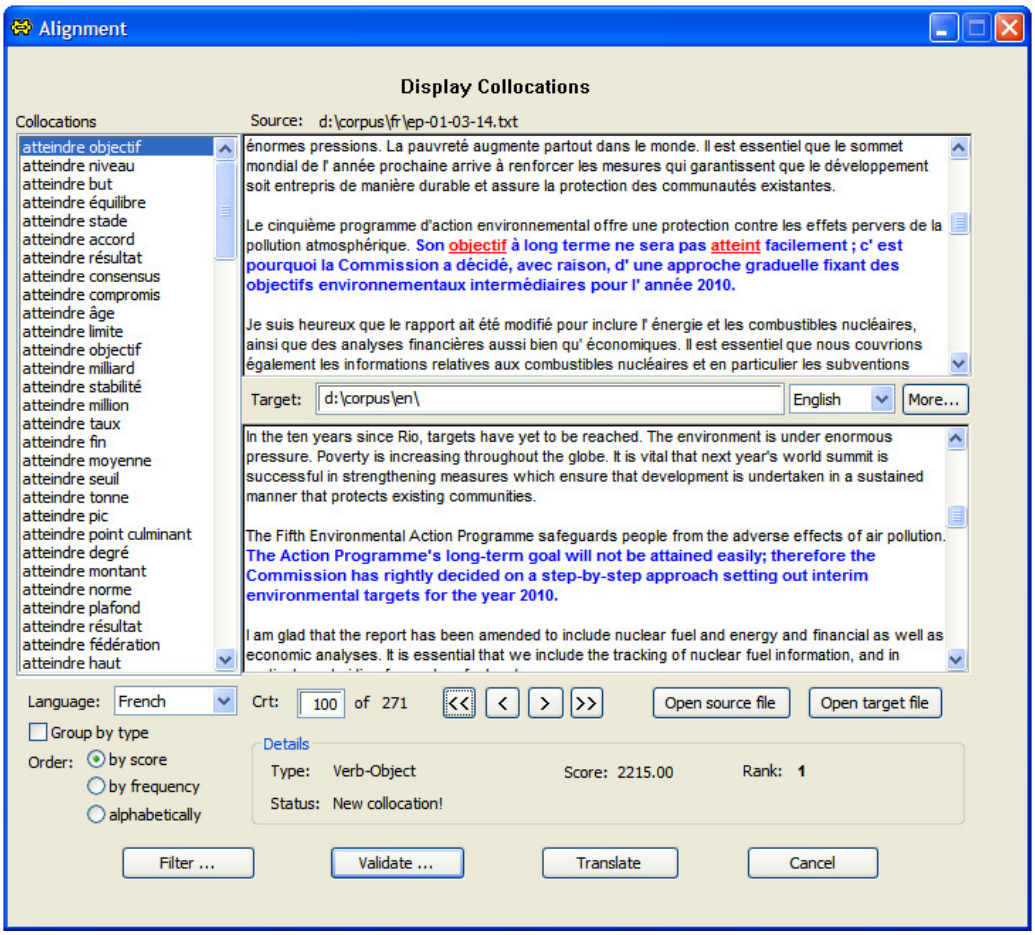

Fig. 2. FipsCo: Parallel concordancing interface, displaying filtered collocations for the French verb *atteindre* (“to reach, to attain”).

the French sentence shown in Figure 2, it is considered that the noun *objectif* moved from its original position of direct object into the surface position of subject due to a passivisation transformation. The parser keeps track of this movement by linking the (empty) object position of the verb *atteint* to the extraposed noun, *objectif*. In the normalised sentence representation it builds, the parser identifies this noun as the “deep” direct object. Consequently, the combination *atteindre–objectif* can successfully be identified from this sentence as a verb-object collocation, as the parser helps abstract away from the particular surface realization.

The parser is the most important resource on which syntactic concordancers like FipsCo rely in order to filter out “noise” and to return highly accurate extraction results.[2] However, parsers are only available for a handful of languages. Fips, in particular, relies on large resources (lexica and grammars) whose construction is time-consuming.

Currently available for English, French, Spanish, Italian, Greek and German, Fips is actually conceived as a generic parsing architecture, coupling a language-independent parsing engine with language-specific extensions. The language-independent part implements the parsing algorithm, based on three main types of operations: *Project* (assignment of constituent structures to lexical entries), *Merge* (combination of adjacent constituents into larger structures), and *Move* (creation of chains by linking surface positions of “moved” constituents to their corresponding canonical positions).

The language-specific part of the Fips parser consists of grammar rules of a given language and of a detailed lexicon for that language. In the formalism used by Fips, the role of most grammar rules is to specify the conditions under which two adjacent constituents may be merged into a larger constituent by a *Merge* operation. The construction of the lexicon is supported by a morphological generation tool that creates appropriate lexical entries corresponding to a specified inflection paradigm (when applicable). Unlike other parsers, Fips does not require POS-tagged data as input; the POS is assigned to words during the analysis, based on lexical information and on the parsing hypotheses.

Given the Fips architecture and the existing tools supporting the creation of lexical resources, we believe that the effort of extending Fips to a new language is comparable to the combined effort of building POS-taggers and developing

[2]An evaluation of FipsCo is presented in [14].

shallow parsers for the same language. Our recent work on Romanian [15] confirmed that a Fips parser version that can be satisfactorily be used for the purpose of collocation extraction can be built in a reasonable amount of time, of the order of several person-months.

B. Methodology

As mentioned in Section II, the extraction of collocations from text corpora is done by using a hybrid extraction method, which combines the syntactic information provided by the Fips parser with existing statistical methods for detecting typical lexical associations in corpora.[3]

Thus, in the first step, collocation candidates are identified as combinations of lexical items in predefined syntactic configurations (for instance, verb-object) from each sentence of the corpus, by traversing the parse structures returned by the parser. In the second step, the candidates obtained are ranked according to their probability to constitute collocations, as computed with the log-likelihood ratio association measure [16]. FipsCo actually implements a wide range of other measures that the user can choose for ranking collocation candidates; log-likelihood ratio is proposed by default as it is a well-established measure for collocation extraction.

The output of FipsCo is a so-called significance list, in which one finds at the top the candidates that are most likely to actually constitute collocations. A cut-off point can be applied by the user to the results, in order to retain only the candidates with higher scores. Typically, a frequency threshold is also employed to eliminate those combinations that only occur a few times in the corpus. This is because statistical measures are unreliable on low frequency data ($f < 5$). However, we opted for keeping all the candidate data (no frequency threshold), since relevant collocations may be found among combinations occurring only a few times in the corpus. Besides, a threshold can be applied by the user afterwards. The syntactic filter applied on the otherwise huge candidate data helps our system keep the statistical computation tractable. In the systems that do not use parsed data, high frequency cut-offs are often imposed only to reduce the amount of data to process.

IV. Detailed Description of FipsCo

FipsCo is implemented in Component Pascal under BlackBox Component Builder IDE,[4] just as the syntactic parser Fips, on which it relies. It makes an extensive use of the SQL database query language in order to store the extraction results, compute the collocation scores, filter the data that will be displayed, etc.

The system has, in principle, a pipeline architecture, as the typical execution flow follows the order in which the main components of the system are described below. However, there are no restrictions to the order in which the various components can be used, since the extracted and validated results can be stored and accessed later for visualisation.

A. File Selection

The source corpus used in an extraction session is specified by selecting the folder which contains the desired files and, optionally, by applying an automatic or manual filter on its content.

The automatic filter is based on:

- file location: inclusion or exclusion of the sub-folders; exclusion of sub-folders having a specific name;
- file name: this might be required to contain a given string of characters;
- file type: this must belong to a list of allowed types. The system supports all the file formats that can be currently imported by BlackBox, e.g., `odc` – Oberon document; `txt`, `htm`, and `html` – text; `rtf`, `doc` – rich text format; and `utf` – Unicode.
- file last modification date (from `date1` to `date2`; in the last `n` days).

In addition, the selection can be further narrowed manually, as the user may select or deselect items after the automatic filter applies. For instance, it is possible to choose items (files or folders) in the first level of the source folder with a mouse click, or by using standard selection commands (check all; uncheck all; invert selection).

B. Collocation Extraction

The collocation extractor is the main component of the system. It iteratively processes all the files in the selection. The number of files that can be processed is virtually unlimited. The collocation candidates identified from the parse trees are incrementally added to previous results until an extraction session ends. They are stored either in a database or in a single text file. As an option, they can also be stored file by file in a folder whose structure mirrors the structure of the source folder.

At the end of the extraction session, several processing statistics are computed for the source corpus that are derived from parsing information (e.g., the total number of tokens, sentences, sentences with a complete parse). Then, the candidates identified are ranked according to the chosen association measure (by default, log-likelihood ratio [16]).

C. Filtering

This component selects the results to be displayed in the concordancer, according to the parameters set by the user (see Figure 3). The extracted collocations can be filtered according to several criteria:

- syntactic type: the user can select one or more types from a list that is automatically built from the database containing the extracted collocations;
- collocation score: a range from `score1` to `score2`;[5]

[3]It is important to note that the method itself is not dependent on Fips or any of the specific theoretical assumptions made by Fips, but it can be used in conjunction with other parsers.

[4]BlackBox is developed by Oberon Microsystems (http://www.oberon.ch). A characteristic of this development environment is the ease of editing graphical user interfaces components, which turned into a big advantage for our system, in which visualisation plays a major role.

[5]The user is not required to know the actual maximal values; the corresponding fields can be left blank and these values will be retrieved by the system.

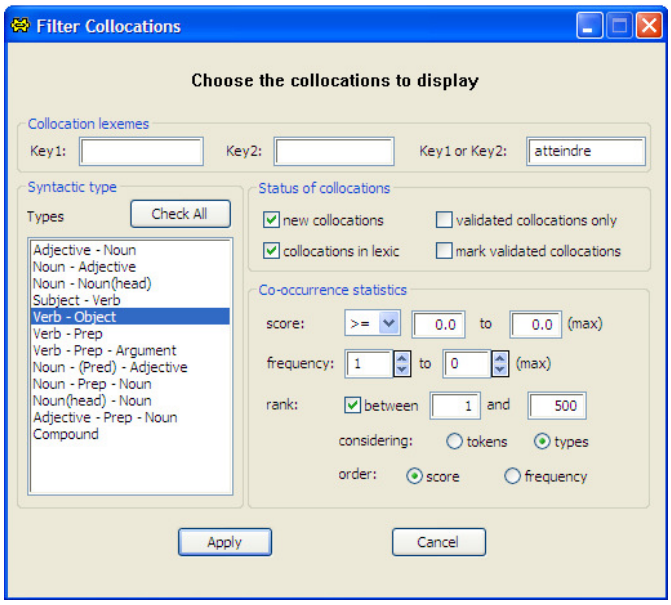

Fig. 3. FipsCo: Interface for filtering collocations.

- corpus frequency: a range from `freq1` to `freq2`;
- collocation keywords: the user can search for collocations containing a specific word.

In addition, the user can specify the range of results to display (from `rank1` to `rank2`), according to the order given by the collocation score or by the corpus frequency. The range restrictions can be applied both to collocation types and to collocation instances (tokens).

D. Concordancing

This component is responsible for the visualisation of extraction results according to the selection made by the user. The (filtered) list of collocations is displayed on the left hand-side on the concordance interface, and can be ordered by score, by frequency in the corpus, or alphabetically. On the right hand-side, a text panel displays the context of the currently selected collocation in the source document. The whole content of the document is accessible, and is automatically scrolled to the current collocation; this collocation and the sentence in which it occurs are highlighted with different colors (cf. Figure 2).

Each item in the list represents a collocation type; its corresponding instances are read from the database when the user clicks on it. The right panel automatically displays the first instance, then the user has the possibility of navigating through all the instances by using the standard browsing arrows (`<< - first`, `< - previous`, `> - next`, `>> - last`), or to skip to a given instance by entering its order number.

The visualisation interface also displays information about the rank of the currently selected collocation, its syntactic type, its score, and its status relative to the parser's lexicon (new collocation, or collocation in lexicon). The user can easily switch to a different source language in order to load the collocations already extracted for that language, if these were stored in the same database.

E. Complex Collocations

By treating already extracted collocations as single lexical items, FipsCo is able to identify complex collocations that can be seen as structures containing embedded collocations: for instance, *atteindre point culminant* ("to be at the highest level") is a complex collocation of verb-object type, which contains an embedded noun-adjective collocation, *point culminant*.

The detection of such complex collocation is particularly useful when the resulting expression constitutes a non-decomposable compound, or when it contains a nested compound. In these cases, it is important to highlight the whole expression rather than nonsensical sub-parts. For instance, *genetically modified organisms* is a compound, and it will be desirable to output it as a whole rather than only the sub-part *modified organisms*. The expression *second world war* is more compositional, as *world war* is a collocation on its own. However, it is desirable to eliminate *second war* from the extraction results, if it only occurs in the corpus in the longer expression *second world war* .

Our method of detecting complex collocations is described in [17] and [18]. FipsCo includes a concordancing interface for displaying complex collocations, which is similar to the standard interface shown in Figure 2.

F. Sentence Alignment

When parallel corpora are available, the target sentence containing the counterpart of the source sentence can be detected and displayed in the alignment interface below the source sentence. The user selects the target language from a list of languages and specifies the path of the target corpus and the filename transformation rule needed to determine the filename of the target document (i.e., of the translation) from the filename of the source document. These rules assume that the source folder and the target folder have the same structure, and that the target filename can be obtained from the source filename by replacing the prefix and/or the suffix of the filename (which are assumed to be variable across languages), while keeping the middle part constant. For instance, `35.1.001E.txt` can be obtained from `35.1.001F.txt` by replacing the suffix `F` with `E`.

Once the target file has been found, the sentence that is likely to be the translation of the source sentence is identified using an in-house sentence alignment method ([7], [9]). The alignment component is operational both for binary collocations and for complex collocations.

G. Validation

This component provides functionalities that allow the user to create and maintain a list of manually validated collocations from the collocations visualised with the concordance and the alignment interfaces. An entry contains basic information about a collocation (such as the collocation keywords, lexeme

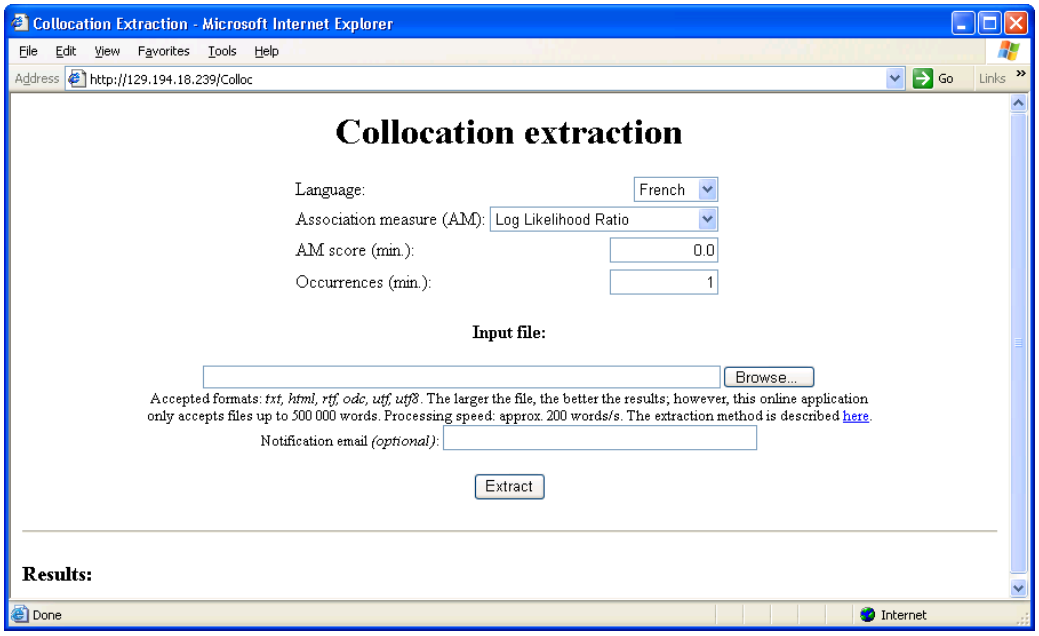

Fig. 4. FipsCoWeb: Interface (screen capture).

indexes for the participating items, syntactic type, score and corpus frequency). A monolingual entry may also contain the source sentence of the currently visualised instance, which provides a naturally-occurring usage sample for the collocation. A bilingual entry stores, in addition, the target sentence found via alignment and the translation proposed for the collocation: the translation can be manually retrieved by the user from the target sentence.

Additional information related to the currently visualized collocation instance is stored (namely, the name of the source and target file, the file position of the collocation's items in the source and target files, and the file position of the source and target sentences). Most of this information is automatically filled in by the system. The entries in the list of collocations validated in a session can be updated, deleted, or saved—completely or in part—by the user in a monolingual and in a bilingual database.

H. Translation

This component attempts to detect a translation equivalent for the collocations visualised in the concordancer, by scanning the existing translations and using a strategy briefly described below.

First, a limited number of corpus sentences (50 in our current experiments) in the source language is retrieved for the source collocation, based on the corpus instances detected during extraction. The alignment component is then used for finding, for each source sentence, the corresponding target sentence in the desired target language, for which a parallel corpus is available.

The target mini-corpus thus obtained is parsed, and collocations are extracted from it using the same method that was applied to the source corpus. Finally, a process of collocation matching takes place, which tries to find, among the extracted collocations, the one that is likely to represent a translation for the source collocation. The matching is performed by applying a series of filters on the extracted pairs that gradually reduce their number until a single item is retained, which will be proposed as translation. An updated description of the translation method can be found in [19].

V. Online Version: FipsCoWeb

FipsCoWeb, which is introduced for the first time in this paper, is the online version of the FipsCo system. Its current interface is shown in Figure 4. FipsCoWeb allows the user to upload a file and to set the initial processing and visualization parameters (e.g., association measure, cut-off score, frequency threshold). After the processing is done on the server side, the user is presented with the results, as shown in Figure 5. The user has then the possibility to apply different parameters, to apply a syntactic filter, and to see the actual occurrences of a collocation by clicking on the corresponding link. The words in the collocation will be presented in the sentence context, and highlighted for readability (cf. Figure 6).

FipsCoWeb currently allows users to upload files containing up to 0.5 million words. While this is a reasonable size for online corpus exploration, the processing, which is performed at an average of 200 tokens/second, might take a while to complete. Depending on the file size, users might only be able to see the results after a few minutes or a longer lapse of time (typically, half an hour). For this reason, FipsCoWeb gives users the possibility to enter the e-mail address at which the link to results is sent when the server-side computation is completed. Results are stored on the server and can be

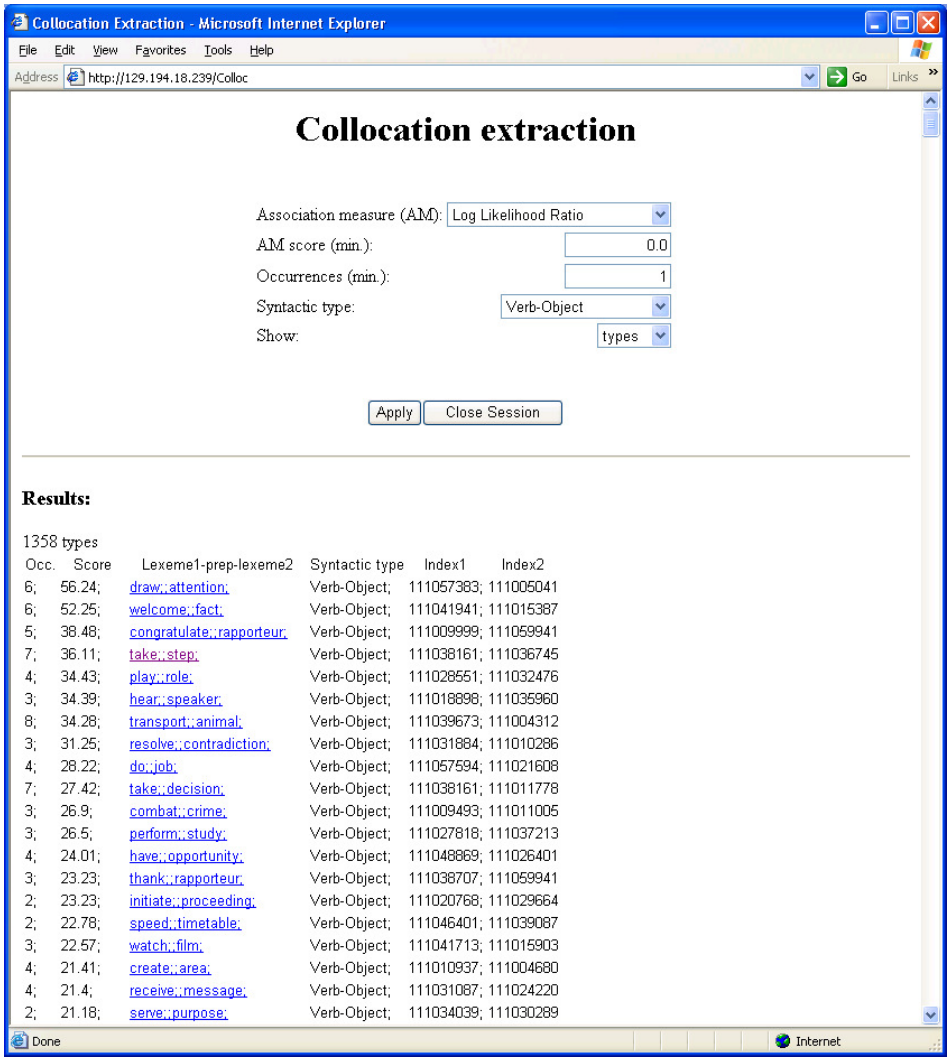

Fig. 5. FipsCoWeb: Sample results (screen capture).

consulted later, until users explicitly decide to clear them, by clicking on the *Close Session* button. A feature that is currently unavailable in the system, but can be easily implemented, is the search for collocations with a given word.[6]

The Web version has been implemented in BlackBox (see Section II), and the Web server itself[7] runs as a BlackBox program. This made the integration between the involved software modules easier. However, since it runs as a unique Windows process, it cannot be efficiently used for the parallel processing of large files. A solution is currently being worked on to circumvent this problem. Future work will focus on implementing FipsCoWeb as a Web service.

VI. Integration in the NLP Environment of LATL

LATL develops a range of NLP tools in several areas. FipsCo (and its online version, FipsCoWeb) are not isolated tools, but are part of a larger processing framework specifically dealing with MWEs, from different practical perspectives.

As a matter of fact, the corpus-based study of words and their collocates was not, in our case, a goal in itself. The collocations that lexicographers manually validate are entered into the lexical database of the parser Fips, and are used to guide future analyses performed by Fips [20]. Their translations (either manually or automatically obtained) are used to populate the bilingual lexicon of a rule-based machine translation based on Fips. The collocations added in the lexicon are further used in two applications of terminology assistance, Twic and TwicPen [21], which look up the lexicon and propose a translation for a given word that is compatible with the grammatical context. If the selected word is part of a MWE, these systems output the translation of the whole MWE, rather than a translation for the word in isolation. Work is under way to augment the MWE resources for all the languages supported by the Fips parser.

[6]Note that the online version does not aim to re-implement all the functionalities of FipsCo.

[7]O_3-WAF (Web-Application-Framework); http://o3-software.de/

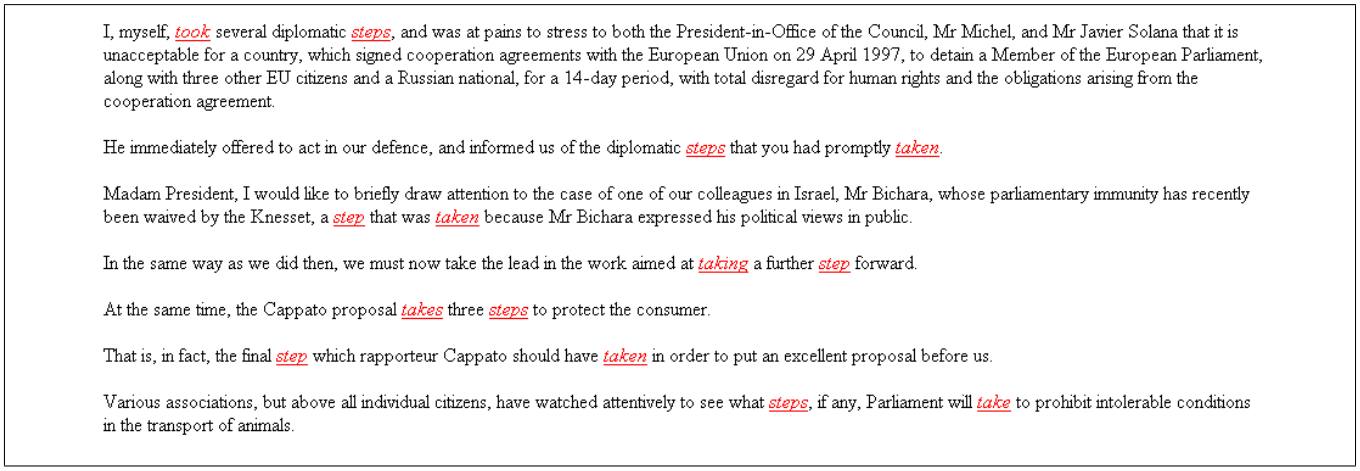

Fig. 6: FipsCoWeb: Collocation instances in context (screen capture).

VII. Conclusion

In this article, we provided an updated description of FipsCo, a tool for extracting collocations (and multi-word expressions more generally) from corpora, which has been developed at LATL in the last several years. Since FipsCo is based on parsing and offers multiple visualisation functionalities, it can be seen as a tool for syntax-based corpus exploration, or syntactic concordancing.

Also, we introduced FipsCoWeb, the online version of this tool, recently developed and already functional. This version can be used to upload a user's own text corpus as a file and to consult the retrieved collocations. The two tools are part of a larger processing framework dedicated to MWEs, and are being used to provide resources for the two main long-term NLP projects pursued in our laboratory, namely, a multilingual symbolic parser and a machine translation system based on it.

Acknowledgement

The work reported in this paper has been supported by the Swiss National Science Foundation (grant no. 100012-117944). We would like to thank the four anonymous reviewers of an earlier version of this paper for useful comments and suggestions.

References

[1] J. R. Firth, *Papers in Linguistics 1934-1951*. Oxford: Oxford Univ. Press, 1957.
[2] G. Barnbrook, *Language and Computers: A Practical Introduction to the Computer Analysis of Language*. Edinburgh: Edinburgh University Press, 1996.
[3] S. Evert, "The statistics of word cooccurrences: Word pairs and collocations," Ph.D. dissertation, University of Stuttgart, 2004.
[4] P. Pecina, "Lexical association measures: Collocation extraction," Ph.D. dissertation, Charles University in Prague, 2008.
[5] A. Kilgarriff, P. Rychly, P. Smrz, and D. Tugwell, "The Sketch Engine," in *Proceedings of the Eleventh EURALEX International Congress*, Lorient, France, 2004, pp. 105–116.
[6] K. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational Linguistics*, vol. 16, no. 1, pp. 22–29, 1990.
[7] L. Nerima, V. Seretan, and E. Wehrli, "Creating a multilingual collocation dictionary from large text corpora," in *Companion Volume to the Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, Hungary, 2003, pp. 131–134.
[8] V. Seretan, L. Nerima, and E. Wehrli, "A tool for multi-word collocation extraction and visualization in multilingual corpora," in *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004*, Lorient, France, 2004, pp. 755–766.
[9] V. Seretan, "Collocation extraction based on syntactic parsing," Ph.D. dissertation, University of Geneva, 2008.
[10] F. J. Hausmann, "Kollokationen im deutschen Wörterbuch. Ein Beitrag zur Theorie des lexikographischen Beispiels," in *Lexikographie und Grammatik. Akten des Essener Kolloquiums zur Grammatik im Wörterbuch*, ser. Lexicographica. Series Major 3, H. Bergenholtz and J. Mugdan, Eds., 1985, pp. 118–129.
[11] D. Lea and M. Runcie, Eds., *Oxford collocations dictionary for students of English*. Oxford: Oxford University Press, 2002.
[12] K. R. McKeown and D. R. Radev, "Collocations," in *A Handbook of Natural Language Processing*, R. Dale, H. Moisl, and H. Somers, Eds. New York, USA: Marcel Dekker, 2000, pp. 507–523.
[13] E. Wehrli, "Fips, a "deep" linguistic multilingual parser," in *ACL 2007 Workshop on Deep Linguistic Processing*, Prague, Czech Republic, 2007, pp. 120–127.
[14] V. Seretan and E. Wehrli, "Multilingual collocation extraction with a syntactic parser," *Language Resources and Evaluation*, vol. 43, no. 1, pp. 71–85, 2009.
[15] V. Seretan, E. Wehrli, L. Nerima, and G. Soare, "FipsRomanian: Towards a Romanian version of the Fips syntactic parser," in *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010.
[16] T. Dunning, "Accurate methods for the statistics of surprise and coincidence," *Computational Linguistics*, vol. 19, no. 1, pp. 61–74, 1993.
[17] V. Seretan, L. Nerima, and E. Wehrli, "Extraction of multi-word collocations using syntactic bigram composition," in *Proceedings of the Fourth International Conference on Recent Advances in NLP (RANLP-2003)*, 2003, pp. 424–431.
[18] L. Nerima, E. Wehrli, and V. Seretan, "A recursive treatment of collocations," in *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010.
[19] V. Seretan, "Extraction de collocations et leurs équivalents de traduction à partir de corpus parallèles," *TAL*, vol. 50, no. 1, pp. 305–332, 2009.
[20] E. Wehrli, V. Seretan, and L. Nerima, "Sentence analysis and collocation identification," in *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*, Beijing, China, 2010, pp. 27–35.
[21] E. Wehrli, L. Nerima, V. Seretan, and Y. Scherrer, "On-line and off-line translation aids for non-native readers," in *Proceedings of the International Multiconference on Computer Science and Information Technology*, Mragowo, Poland, 2009, pp. 299–303.