Term and collocation extraction by means of complex linguistic web services

Ulrich Heid[1], Fabienne Fritzinger[1], Erhard Hinrichs[2], Marie Hinrichs[2], Thomas Zastrow[2]

[1]Universität Stuttgart
Institut für maschinelle Sprachverarbeitung
Azenbergstr. 12
D 70174 Stuttgart
[heid,fritzife]ims.uni-stuttgart.de

Universität Tübingen
Seminar für Sprachwissenschaft
Wilhelmstr. 19
D 72074 Tübingen
[eh,meh]sfs.uni-tuebingen.de

Abstract

We present a web service-based environment for the use of linguistic resources and tools to address issues of terminology and language varieties. We discuss the architecture, corpus representation formats, components and a chainer supporting the combination of tools into task-specific services. Integrated into this environment, single web services also become part of complex scenarios for web service use. Our web services take for example corpora of several million words as an input on which they perform preprocessing, such as tokenisation, tagging, lemmatisation and parsing, and corpus exploration, such as collocation extraction and corpus comparison. Here we present an example on extraction of single and multiword items typical of a specific domain or typical of a regional variety of German. We also give a critical review on needs and available functions from a user's point of view. The work presented here is part of ongoing experimentation in the D-SPIN project, the German national counterpart of CLARIN.

1. Introduction

This paper presents a web-based service environment for the use of linguistic resources and tools, focusing on early results of a part of the German D-SPIN project[1].

The scenario underlying the present study is targeted at linguists, philologists, terminologists and translators interested in an analysis of text resources they may have collected, with a view to lexis: single and multiword items typical of the specialised language of a given domain, or typical of a regional variety of German.

Identifying such lexical specificities is a complex task, which involves several computational linguistic processing steps and the combination of tools which not every user may have available. Thus, an objective of D-SPIN, as of the EU project CLARIN, is (i) to make the necessary tools available in a *service-oriented* architecture, (ii) to provide an environment which allows the user to apply them to his/her own texts (or to corpora made available for the purpose), and (iii) to combine them as needed.

In this paper, we present some exemplary corpus preprocessing tools, the retrieval of lexical collocations, and the comparison of data from two corpora. These functions are useful for finding specialised terminology and/or for identifying lexical regionalisms. Obviously, many more types of linguistic, terminological and philological analyses can in principle be supported by the tools.

In section 2., we sketch the application scenarios and illustrate the kinds of data to be extracted from texts. We distinguish between corpus preprocessing (tokenisation, tagging/lemmatisation and possibly parsing) and corpus exploration (data extraction). Section 3. is devoted to our preprocessing web services: architecture, formats used for text encoding, and in particular WebLicht, the web-based linguistic tool chainer (cf. (Hinrichs et al., 2009), (Hinrichs et al., 2010)) which allows the user to combine pipelines of linguistic processing modules without need for installing or adapting existing tools. We present its infrastructure function, its user interface and the range of preprocessing options at hand. In section 4., we turn to terminology and variety analyses, and discuss web service pipelines for collocation extraction and corpus comparison (e.g. Swiss vs. German news texts). Section 5. is a critical review of the current state, from the user's point of view: available functions vs. needs for future improvements.

2. Scenarios and targeted phenomena

Term candidate extraction targets both single word and multiword items (e.g. *Rechtsbeschwerde* 'appeal', *Rechtsnachfolger* 'legal successor', *Recht(e) geltend machen* 'assert one's rights'). Similarly, our comparison of regional varieties also targets single words and multiwords, in this case mainly collocations: in line with the pluricentricity hypothesis (cf. (Ammon, 2001)), we check Swiss (CH) and Austrian (AT) newspaper texts for region-specific lexis (not dialects). Examples of single words include AT *Krida* 'bankrupcy' for DE *Bankrott, Insolvenz*, or CH *Zustupf* '(financial) contribution' for DE *Zuschuss*. These specific items may be part of multiwords (*betrügerische Krida* 'fraudulent bankrupcy', *Zustupf leisten* 'give a contribution'). But also non-specific lexical items present in all varieties may occur in region-specific collocations (e.g. CH *markanter Anstieg* 'marked increase' for DE *deutlicher Anstieg*).

Such phenomena are among the data we assume that linguists, lexicographers, terminologists or translators may wish to extract from texts. This extraction requires a certain

[1]D-SPIN stands for Deutsche SPrachressourcen-INfrastruktur; the D-SPIN project is financed by the German Federal Ministry of Research and Education, BMBF; it is a national German complement to the EU-project CLARIN. See the URLs `http://www.sfs.uni-tuebingen.de/dspin` and `http://www.clarin.eu` for details.

amount of preprocessing of the texts, as well as additional steps: extraction by syntactic patterns (e.g. noun + adjective, verb + object noun, noun + genitive complement) – frequency counts and a calculation of cooccurrence significance – a frequency-based comparison of data from two corpora, e.g. specialised vs. general, or Swiss vs. German.
In the following sections, we discuss to what extent the tools needed for these tasks and the work flows resulting from their combination can be supported by linguistic web services.

3. Linguistic Web Services for German

Linguistic web services can be word-oriented or text-oriented. Word-oriented web services rely on lexical or corpus resources; users query the web service with a particular item and get back all data associated with the item in the respective resource (see e.g. Wortschatz[2]).
We will in the following sections concentrate on text-oriented web services: the user uploads a text (corpus) and the service calls one or more computational linguistic tools which are applied to the text (corpus) and deliver annotation results. As a variant, the tools may perform calculations (e.g. of frequency or significance) and deliver the results to the user.
Obviously, web services may be called interactively or via APIs, from tools. We focus on the former interaction, here, even though both are possible in our setup.
In our experiments, we implement an architecture consisting of four layers (cf. Fig. 1, from bottom to top): (i) tools, (ii) wrappers, services and converters, (iii) the web service infrastructure, as well as (iv) clients.

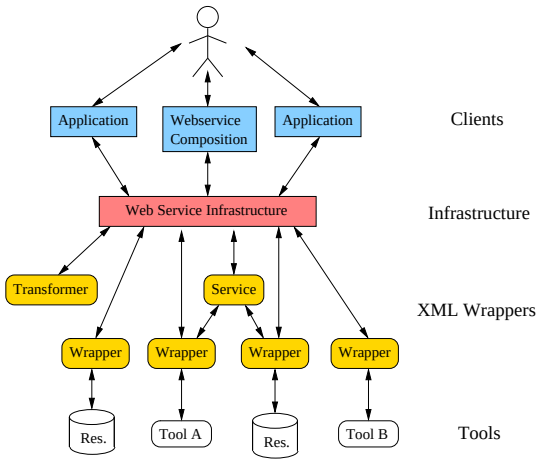


Figure 1: Layers of the web service architecture

Tools and resources are embedded each in a wrapper, which acts as an interface for their input and output and thereby supports communication between different tools in a pipeline. By using wrappers, we ensure that the tools and resources themselves need not be modified with respect to their stand-alone versions when being used in the web service setup. So far, we have integrated linguistic corpus processing tools provided by four German research institutes (see section 3.2.1., below). Wrappers may interact with service components: e.g. when a tool provided by one institution uses a resource provided by another one.
Components of the infrastructural layer are used, among others, to call converters, e.g. for transforming the D-SPIN-internal text corpus format into an exchange format (see below, section 3.1.). User interaction, as well as calls of the web services from applications (e.g. other linguistic processing tools) are directed to the infrastructure layer. In our setup, interactive calls are channeled through the WebLicht tool (Hinrichs et al., 2009), a linguistic tool chainer for composing task-specific web services. The chain builder function will be presented below, in section 3.2..
The implementation uses both Perl/Python and Java components; it is based on the REST architecture[3] (Richardson and Ruby, 2007), and on an Apache web server. As the whole setup is experimental, we have so far not worked in detail on authentification, access rights, billing, etc. The EU project CLARIN has an ongoing work package devoted to these issues, the results of which will be used, before our web services will be made available at large.

3.1. Formats used internally and for exchange

Internal communication. For communication between different web services, we use an XML-based text corpus format which is defined by an XML schema expressed in Relax NG. It uses additional constraints expressed in Schematron. A sample encoding taken from the current implementation is reproduced in Figure 2.
Basically, the format contains both a metadata section, to encode sources, tools, the language of the document, etc., and a multi-layered token and region annotation. Figure 2 shows a part of the internal encoding of the sentence *die zweite Studie lieferte ähnliche Ergebnisse*[4] parsed with the BitPar constituent parser (Schmid, 2004); we reproduce only some tokens and nodes of the parse tree, skipping metadata and lemmas (cf. the dependency tree in Figure 5, below).

Exchange formats – long-term view. The current text corpus format was designed with a view to efficiency: in our tests, we processed among others the 10 million words EMEA corpus together with a 40 million words newspaper corpus (*Frankfurter Rundschau*, 1992/93) through one instance of the pipeline (see section 4.). The annotated results lead to rather large amounts of data which have to be transported through the web based pipeline. Thus, the definition of the D-SPIN text corpus format aims at keeping the XML overhead low (for details, see (Heid et al., 2010)).
Obviously, there exist standard proposals for corpus encoding formats, e.g. from the language resources standardisation work groups of ISO[5]. The ISO proposals cover a

[2] `http://wortschatz.uni-leipzig.de/axis/servlet/ServiceOverviewServlet`, an extensive linguistic database of the University of Leipzig.

[3] Implemented on Apache Web server and Tomcat application server.

[4] EN: the second study provided similar results.

[5] ISO TC 37/SC-4 `www.tc37sc4.org`

```
<D-Spin>
  <MetaData/>
  <TextCorpus lang="de" tokens="yes" parsing="TigerTB">
    <text>Die zweite Studie lieferte ähnliche Ergebnisse</text>
    <tokens>
      <token ID="t2">Die</token>
      <token ID="t3">zweite</token>
      <token ID="t4">Studie</token>
      ...
    </tokens>
    <parsing>
      <parse>
      <node cat="TOP">
        <node cat="S-TOP">
        <node cat="NP-SB">
          <node cat="ART*" tokID="t2">Die</node>
          <node cat="ADJA*" tokID="t3">zweite</node>
          <node cat="NN*" tokID="t4">Studie</node>
        </node>
        ...
      </node>
      </node>
      </parse>
    </parsing>
  </TextCorpus>
</D-Spin>
```

Figure 2: XML structure of constituent parsing

general graph-based metastandard for any kind of annotation LAF[6], (cf. (Ide and Romary, 2004)), as well as definitions of syntactic and morphosyntactic annotation formats, SynAF and MAF[7]. We have started to build converters that are able to translate the D-SPIN text corpus format into MAF and vice versa. A mapping of the tagset STTS (Schiller et al., 1995), which is a de facto standard for German, onto the data categories defined in `ISOcat`, according to ISO 12620[8], is under way. For unambiguous syntactic representations, a full mapping from the D-SPIN text corpus format to MAF/SynAF[9] will be provided, at least in an experimental way, by the end of 2009.

In the medium to long term, we expect the web services to be able to convert the D-SPIN internal format to the ISO proposals. The extent to which these can serve as an internal format themselves remains to be explored.

3.2. WebLicht: A Tool Chainer

The user interface layer and part of the infrastructure of our architecture is implemented through a tool chainer named WebLicht[10] (cf. (Hinrichs et al., 2009), (Hinrichs et al., 2010)). It is implemented as a web application so that there is no need for users to install any software on their own computers or to concern themselves with the technical details involved in building tool chains.

3.2.1. WebLicht: infrastructural aspects

WebLicht infrastructure functions. In its infrastructure functions, WebLicht allows the integration and use of distributed web services with standardised APIs. The nature of these open and standardised APIs makes it possible to access the web services from nearly any programming language, shell script or workflow engine (UIMA, Gate etc.).

WebLicht Preprocessing Tools. Currently, WebLicht offers linguistic resource and tool services that were developed independently at the Institut für Informatik, Abteilung Automatische Sprachverarbeitung of the University of Leipzig, at the Institut für Maschinelle Sprachverarbeitung at the University of Stuttgart, at the Berlin-Brandenburgische Akademie der Wissenschaften and at the Seminar für Sprachwissenschaft/Computerlinguistik of the University of Tübingen.

The tools mainly serve for preprocessing of corpora (tokenisers, taggers, parsers), but also for lexical-semantic annotation (GermaNet[11], synonym finder) and for frequency calculation. Although we only discuss tools for the German language, it should be noted that some tools are either language independent (e.g. the trainable tokeniser) or exist for other languages as well (e.g. taggers for EN, FR, IT, inter alia). Table 1 provides a listing of all currently implemented tools in WebLicht.

Tool	Location(s)
Text2Dspin Converter	Berlin, Tübingen
TextCorpus to Lexicon Converter	Tübingen
Tokeniser	Berlin, Leipzig, Stuttgart
Sentence border detection	Leipzig
POS Tagger	Berlin, Stuttgart
POS Analyser	Tübingen
Base Form/Lemmatiser	Leipzig
Morphological Analyser	Stuttgart
Named Entity Recogniser	Berlin
Constituent Parser	Stuttgart
Semantic Annotator/GermaNet	Tübingen
Frequency Analyser	Leipzig
Co-occurrence Extractor	Leipzig
Similarity	Leipzig
Synonym Finder	Tübingen

Table 1: WebLicht tool overview

3.2.2. WebLicht Preprocessing Workflows

To get experience with multiple web service components, we have not only integrated several preprocessing and extraction steps which form a pipeline (tokenising, tagging, parsing, etc.): we also provide several alternative tools for some of the tasks, to choose from. Having different tools for the same purpose available makes sense when individual tools are known to differ with respect to the underlying philosophy, coverage, etc.

[6] ISO/DIS 24612, `http://www.tc37sc4.org/new_doc/ISO_TC_37_SC4_N311_Linguistic%20Annotation%20Framework.pdf`

[7] SynAF: ISO/DIS 24615, MAF: ISO/DIS 24611

[8] `www.isocat.org/`

[9] Note that SynAF does not yet address the issue of ambiguity; similarly, LAF also does not yet deal explicitly with ambiguity. For a proposal to include ambiguity handling into LAF, see (Kountz et al., 2008).

[10] `http://clarin.sfs.uni-tuebingen.de:8080/WebLicht0/`, developed at the University of Tuebingen.

[11] `http://www.sfs.uni-tuebingen.de/GermaNet/`, the German WordNet by the University of Tübingen.

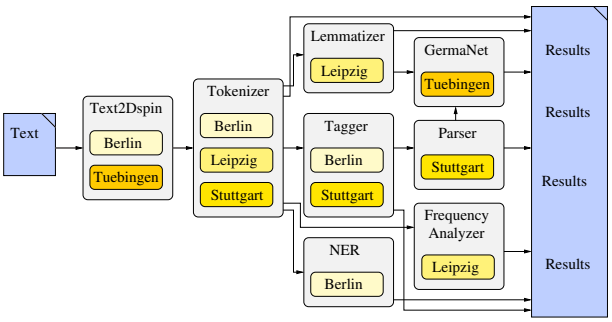


Figure 3: Extract of the WebLicht architecture: preprocessing components

Moreover, different workflows may require preprocessing of different depth: if a user only needs to extract adjective+noun pairs from his texts, there is no need for parsing, whereas for German verb+object pairs, results improve if parsed data are used (Ivanova et al., 2008). Figure 3 shows part of the preprocessing workflows currently supported by the WebLicht chainer ((Hinrichs et al., 2009), (Hinrichs et al., 2010)).

3.2.3. WebLicht as a user interface

The WebLicht platform is language-independent. The user is able to select tools and resources for a specific language by the choice options in a language selection field. This *Language* field currently allows the selection of e.g. German, English, Italian, French, etc.

Language input to the web services can be plain text to be inserted by the user in a plain text field or by uploading a plain text file whose source location can be specified. Alternatively, various format converters are offered to bring input text into the proper format used by WebLicht.

A *Selected Tools* field displays all web services that have already been entered into the web service chain. A *Next Tool Choises* field then offers the set of tools that can be entered into the chain next. The user often has a choice of alternative tools – sometimes a wide variety of services are offered as candidates. Obviously, only those tools are offered as options for further processing that are compatible with the results of the processing steps already completed.

For more information on the WebLicht user interface see (Hinrichs et al., 2010).

4. Complex scenarios for Web Services use

Above, the standard WebLicht setup has been described. It is characterised by its flexibility and by the possibility to process corpora of non-trivial size. Obviously, in such a case the processing may take some time, but users may prefer to wait several hours for a processing result over having their own computers blocked by ongoing processing, or having to install and adapt tools.

In the following, we turn to the experimental applications from terminology and language variety research sketched in section 2., now from a more technical perspective.

4.1. Scenarios revisited: tools

With the help of WebLicht services, we can extract frequency-wise prominent single words and their lexical collocations (e.g. in the sense of Bartsch (2004):76 (Bartsch, 2004)), from texts of a specialised domain, or from regional texts (cf. section 2.).

Collocation candidate extraction. Collocation extraction amounts to the extraction of all lemma pairs of a given syntactic pattern (e.g. adjective + noun, verb + object noun) and subsequent sorting of the candidates by their text-specific association value (for details, cf. (Weller and Heid, 2010)). The calculation of association values (e.g. Log Likelihood) is done by means of a (web service) call to Evert's UCS toolkit (Evert, 2005), which implements over thirty different association measures.

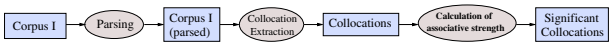
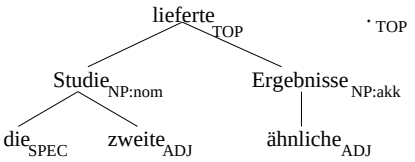


Figure 4: Collocation extraction procedure

Figure 4 schematises a simple collocation candidate extraction pipeline. Parsing, the extraction of lemma pairs and the calculation of associative strength are modules provided as web services, cf. (Fritzinger et al., 2009). These are three separate models, which we "package" into two web services, for reasons of practicality.

For identifying verb + object pairs, we rely on text parsed with the dependency parser FSPar (Schiehlen, 2003). Its output encodes grammatical relations between governor and governed items. As a sample, the analysis of *Die zweite Studie lieferte ähnliche Ergebnisse* is reproduced in Figure 5.



(a) Tree representation

0	Die	ART	d	|	2	SPEC
1	zweite	ADJA	2.	|	2	ADJ
2	Studie	NN	Studie	Nom:F:Sg	3	NP:1
3	lieferte	VVFIN	liefern	3:Sg:Past:Ind*	–1	TOP
4	ähnliche	ADJA	ähnlich	|	5	ADJ
5	Ergebnisse	NN	Ergebnis	Akk:N:Pl	3	NP:8
6	.	$.	.	|	–1	TOP

(b) Parsing output

Figure 5: Output of FSPAR: dependency analysis

FSPar internally encodes the dependency tree in linear form (Fig. 5 (b)), with several columns: the first column contains the position of the word form in the parsed sentence, the second the word form itself, followed by POS-tag and

(a) Only EMEA (not FR).

term candidates	f (abs.)
Durchstechflasche	5638
Injektionsstelle	3489
Pharmakokinetik	3426
Hämoglobinwert	3395
Fertigspritze	3271
Ribavirin	3234
Gebrauchsinformation	2801
Dosisanpassung	2580
Epoetin	2302
Hydrochlorothiazid	2128

(b) EMEA and FR

term candidates	weirdness	f (abs.)
Filmtablette	25522	6389
Injektionslösung	19854	4970
Packungsbeilage	14710	7365
Niereninsuffizienz	14233	3563
Verkehrstüchtigkeit	13558	3394
Leberfunktion	8385	2099
Hypoglykämie	8353	2091
Toxizität	7957	1992
Einnehmen	7035	7045
Hypotonie	6823	1708

Table 2: Single word term candidates – Top 10

lemma. The fifth column contains the morphosyntactic features of the word form, e.g. case, number, person, etc. Columns 6 and 7 are to be interpreted together: the number indicates the position of the word's governor, and the last column indicates the grammatical relation between the word and its governor. Thus, in Figure 5, the word form *Ergebnisse* is coded as plural accusative (line 5, col. 5) and as an accusative object (NP:8, col. 7) of the verb *liefern* (col. 7: value "3", pointing to *lieferte*, which is on line 3). The actual collocation extraction is implemented as a pattern-matching over FSPar-output, e.g. for verbs and their objects, (e.g. Perl/Python scripts).

Extraction of prominent single word items. The identification of typical single word items, be it from specialised or regional texts, relies on an implementation of Ahmad et al. (1992)'s "weirdness" measure (Ahmad et al., 1992).
The intuition is that terms from the specialised language of a given domain are much more frequent in the domain specific text than elsewhere. Some specialised items do not show up at all in general language, others are found in both, but more frequently in the specialised text. A tool for identifying specialised items in this way simply has to determine the relative frequency of each item from the specialised text (RS), calculate its relative frequency in a general text used for comparison (RG), and determine the relationship between both (RS/RG). It produces two kinds of term candidate output: a list of words never found in the general texts (cf. Table 2 (a) for an example from a pharmaceutical corpus, we indicate absolute frequency in the specialized corpus), and a list of words found more often in the specialised than in the general texts (cf. Table 2 (b) for examples from the same corpus, with the weirdness figure RS/RG and absolute frequency in the specialized text).

4.2. Corpus data used for experimentation and sample results

Corpus data. For the terminology extraction work, we use the German part of the multilingual corpus of test reports for pharmaceuticals, made available by EMEA, the European Medicines Agency (ca. 10 million words). The data have been collected by Jörg Tiedemann and made available on his OPUS web site[12] This corpus is compared to newspaper text of *Frankfurter Rundschau*, consisting of ca. 40 million words, in order to identify specialised terminology.
In the experiments on the regional varieties of German, we use newspaper texts from the DeReKo corpus, *Deutsches Referenzkorpus*[13]. These are grouped according to countries. We use Austrian and Swiss material and compare it with newspaper texts from Germany, such as *Frankfurter Allgemeine Zeitung*, *Frankfurter Rundschau*, *Die Zeit*, *Stuttgarter Zeitung*, *Handelsblatt*.

Sample results. Table 2, above, contains a few typical single word candidates from the EMEA corpus: items only contained in EMEA (part (a)) and items significantly more frequent in EMEA than in the *Frankfurter Rundschau*, used as a "general language" text, for comparison purposes. The items are sorted by frequency (2(a)) and by the RS/RG quotient (2(b)), i.e. in decreasing order of relevance for the specialised text.
Table 3 contains a few verb + object collocations which are found in Swiss news texts but not in German news.

Abklärung	treffen	96
Abklärung	vornehmen	91
Anlaß	besuchen	73
Anlaß	durchführen	199
Anlaß	organisieren	367
Beschwerde	gutheißen	88
Bilanz	deponieren	82
Busse	aussprechen	72
Defizit	budgetieren	94
Einsitz	nehmen	295
Einsprache	erheben	262
Entscheid	fällen	79
Gegensteuer	geben	143
Gesuch	bewilligen	90

Table 3: Typical CH verb+object pairs

An adjective + noun case of the same type is *tiefer Preis* 'low price'. In German texts from Germany, we find this combination almost never; manual inspection shows that instead, German texts contain *niedriger Preis*. The tools provide the most significant collocations with *tief* in texts from Switzerland (Table 4 (a)) and those from texts from Germany (Table 4 (b)), with their absolute frequency in both corpora. While *tiefer Zins* 'low interest', *tiefer Preis*, *tiefe Inflation* are prominent in Swiss texts, these combinations are rare in texts from Germany. However, the (figurative) use of *tief* in texts from Germany, which is top ranked

[12] http://urd.let.rug.nl/tiedeman/OPUS/

[13] This corpus was collected jointly by *Institut für deutsche Sprache*, Mannheim, and by the Universities of Tübingen and Stuttgart, in the framework of a project financed by the Land Baden-Württemberg, http://www.ids-mannheim.de/projekte/dereko_I.

by frequency in our corpora (*tiefe Krise* 'deep crisis', *tiefer Einschnitt* 'deep cut'), is not absent from the Swiss data. The Swiss standard seems to have an additional sense of *tief*.

(a) Typical CH

Noun	f_{CH}	f_{DE}
Inflation	64	1
Preis	110	2
Zinsniveau	72	2
Zinssatz	26	1
Zins	252	12
Ölpreis	18	1

(b) Rather DE

Noun	f_{CH}	f_{DE}
Krise	108	307
Einschnitt	20	143
Mißtrauen	20	135
Spur	51	132
Loch	64	125
Graben	78	123

Table 4: Nouns with the collocate *tief* (equal size of corpora)

4.3. Web services for corpus comparison

The web services discussed in section 3. are all characterised by the fact that they take one input (file) and produce one output. For corpus comparison, as exploited in the experiments described in section 4.2., we need to be able to process two corpora. The user can provide these either separately or in one file. Internally, the two texts are kept separate, processed individually and then compared. Figure 6 symbolises a pipeline abstracted over both tasks, term and variant extraction.

This pipeline also includes the collocation extraction pipeline from above (Fig. 4). A first comparison is carried out, when single words from corpus I (specialised, regional) are compared for relative frequency with their respective occurrences in corpus II (general). As we want to obtain collocations for exactly the relevant single word items from the specialised/regional corpus, the results of the first comparison are used to filter the collocation candidates obtained on the specialised/regional corpus: thereby those collocations which belong to both varieties are removed.

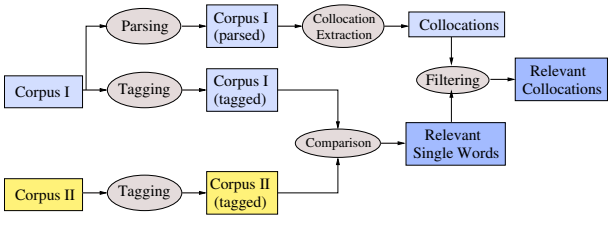


Figure 6: Extraction of specialised collocations

4.4. A format for multiword data

For the web-service-internal encoding of multiword data, as produced by the term candidate and the collocation extraction services, we devised an XML encoding which is inspired by the same principles as our corpus encoding format. It is also meant to be a "slim" XML format, which can in the long run be mapped onto LMF (ISO 24613:2008), the Lexical Markup Framework. An example of the encoding of two word pairs is displayed in figure 7.

```
<D-Spinversion="0.4">
    ...
    <Lexicon lang="de">
        <lemmas>
            <lemma ID="l1">Brise</lemma>
            <lemma ID="l2">halten</lemma>
            <lemma ID="l3">Rede</lemma>
            <lemma ID="l4">steif</lemma>
        </lemmas>
        <word-relations type="collocation">
            <word-relation func="Adj-Noun" freq="5">
                <sig measure="log-likelihood">3.5</sig>
                <sig measure="t-score">3.7</sig>
                <term lemID="l4">steife</term>
                <term lemID="l1">Brise</term>
            </word-relation>
            <word-relation func="Verb-Object" freq="17">
                <sig measure="log-likelihood">2.7</sig>
                <sig measure="t-score">2.8</sig>
                <term lemID="l2">halten</term>
                <term lemID="l3">Rede</term>
            </word-relation>
        </word-relations>
    </Lexicon>
</D-Spin>
```

Figure 7: Encoding of two word pairs

The example contains a lemma list (under `<lemmas>`), the type of relation (here: collocation, but the format would also accommodate, e.g. hypernyms, synonyms, etc.), the grammatical properties of the collocation (under `func`) and its frequency in the corpus, as well as one or more significance figures calculated by means of association measures.

5. Conclusion – New requirements

We have discussed a series of linguistic web services, a general architecture, formats and components, as well as a few non-trivial use cases. As of summer 2009, the implementation of the latter is still experimental. It seems obvious, however, that possibilities to chain several linguistic web services, as offered by WebLicht, for interactive work or as predefined complex services in the more complex use cases discussed in section 4., are vital for a broader use of web service-based linguistic analysis tools in realistic eHumanities applications.

Moreover, it seems necessary to be able to parametrise web services and to allow users to set parameters, before the processing chain is entered. In WebLicht, this is done by selecting a given service. But in the medium term, we would like to consider a setup where users rather select in terms of properties of the expected output than of the tools used. Taking up collocation extraction, Figure 8 symbolises a few of the parameters we envisage users may wish to set (from left to right): which grammar to use for parsing, which syntactic type or types of collocations to extract, which association measure(s) to use, how to package (e.g. by syntactic type) and how to sort and lexicographically display the results. We are still far from this flexibility, but the internal format used in our web services allows us to "transport" a parameter through the pipeline.

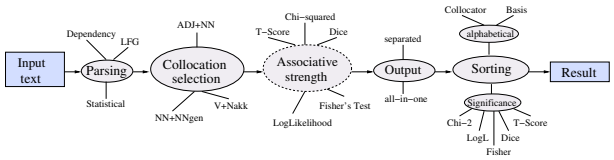


Figure 8: Possible points for user interaction

Finally, we have shown the need for and the usefulness of web services that can take more than one input. Future work will elaborate on the challenges mentioned here, besides the inclusion of additional tools, work towards full standards compatibility and more extensive experimentation with web service-based linguistic processing chains.

Acknowledgements

We gratefully acknowledge the support of the BMBF for the D-SPIN project. We should also like to thank our colleagues from Leipzig and Berlin for making their tools available for integration into WebLicht, and for discussing related issues with us. Special thanks to Helmut Schmid and Max Kisselew (IMS Stuttgart) for their helpful comments on components and procedures described in sections 3 and 4.

6. References

Khurshid Ahmad, Andrea Davies, Heather Fulford, and Margaret Rogers. 1992. What is a term? the semi-automatic extraction of terms from text. In Mary Snell-Hornby, Franz Pöchhacker, and Klaus Kaindl, editors, *Translation Studies - an interdiscipline.* John Benjamins, Amsterdam/Philadelphia.

Ulrich Ammon. 2001. Die Plurizentrizität des Deutschen, oder: Wer sagt, was gutes Deutsch ist? In Kurt Egger and Franz Lanthaler, editors, *Die deutsche Sprache in Südtirol: Einheitssprache oder regionale Vielfalt*, pages 11–26. Folio, Wien/Bozen.

Sabine Bartsch. 2004. Structural and functional properties of collocations in English. In *A corpus study of lexical and pragmatic constraints on lexical co-occurrence*. Narr, Tübingen.

Stefan Evert. 2005. *The statistics of word cooccurrences: word pairs and collocations*. Ph.D. thesis, Universität Stuttgart.

Fabienne Fritzinger, Max Kisselew, Ulrich Heid, Andreas Madsack, and Helmut Schmid. 2009. Werkzeuge zur extraktion von signifikanten wortpaaren als web service. In *GSCL-Symposium Sprachtechnologie und eHumanities, Technischer Bericht Nr.2009-01, Universität Duisburg Essen*, Duisburg.

Ulrich Heid, Helmut Schmid, Kerstin Eckart, and Erhard Hinrichs. 2010. A Corpus Representation Format for Linguistic Web Services: the D-SPIN Text Corpus Format and its Relationship with ISO Standards. In *Proceedings of LREC 2010*, Malta.

Erhard Hinrichs, Marie Hinrichs, Thomas Zastrow, Gerhard Heyer, Volker Boehlke, Uwe Quasthoff, Helmut Schmid, Ulrich Heid, Fabienne Fritzinger, Alexander Siebert, and Jörg Didakowski. 2009. Weblicht: Web-based LRT services for German. In *Workshop on linguistic processing pipelines, GSCL Jahrestagung 2009*, Potsdam.

Marie Hinrichs, Thomas Zastrow, and Erhard Hinrichs. 2010. WebLicht: Web-based LRT Services in a Distributed eScience Infrastructure. In *Proceedings of LREC 2010*, Malta.

Nancy Ide and Laurent Romary. 2004. International standard for a linguistic annotation framework. *Journal of Natural Language Engineering*, 10:3–4.

Kremena Ivanova, Ulrich Heid, Sabine Schulte im Walde, Adam Kilgarriff, and Jan Pomikálek. 2008. Evaluating a German Sketch Grammar: A case study on noun phrase case. In *Proceedings of the VIth International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakesh, Morocco. CD-ROM.

Manuel Kountz, Ulrich Heid, and Kerstin Eckart. 2008. A LAF/GrAF based encoding scheme for underspecified representations of dependency structures. In *Proceedings of the VIth International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakesh, Morocco. CD-ROM.

Leonard Richardson and Sam Ruby. 2007. *RESTful Web Services*. O'Reilly.

Michael Schiehlen. 2003. A cascaded finite-state parser for German. In *Proceedings of the Research Note Sessions of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, April.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1995. Vorläufige Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung, and Seminar für Sprachwissenschaft, Universität Tübingen.

Helmut Schmid. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

Marion Weller and Ulrich Heid. 2010. Extraction of German multiword expressions from parsed corpora using context features. In *Proceedings of LREC 2010*, Malta.