An Integrated Model of Lexical Chaining: Application, Resources and its Format

Ulli Waltinger, Alexander Mehler and Maik Stührenberg

Abstract. Lexical chaining has become an important part of many NLP tasks. However, the goodness of a chaining process and hence its annotation output depends on the quality of the chaining resource. Therefore, a framework for chaining is needed which integrates divergent resources in order to balance their deficits and to compare their strengths and weaknesses. In this paper we present an application that incorporates the framework of a meta model of lexical chaining exemplified on three resources and its generalized exchange format.

1 Introduction

Lexical chaining is the task of tracking semantically related tokens in texts. Thereby, semantic relatedness is modelled by means of lexical reference systems like *WordNet*. Path-based chaining, e.g., judges tokens to be related subject to the shortest path between their WordNet types. Chaining has become an important part of many NLP tasks ranging from *text segmentation* (Marcu 2000; Morris and Hirst 1991), *summarisation* (Barzilay and Elhadad 1997; Silber and McCoy 2002) and *skimming* (Teich and Fankhauser 2005) via *topic clustering & tracking* (Ferret 2002; Stokes 2004) to *hypertext authoring* (Green 1999), *text chaining* (Mehler 2005) and *spelling error detection* (Hirst and St-Onge 1998). The goodness of chaining depends on the quality of the chaining resource. This relates to the limited coverage of lexical reference systems: chaining ignores, e.g., words not covered by the operative reference system even if being central to the meaning of a text. On the other hand, co-occurrence networks may cover all types of a corpus by modeling corpus-specific word usages (Schütze 1998), but induce underspecified similarity judgements as they lack the type system of lexical reference systems. Thus, reliably solving any of the tasks enumerated above depends on the interoperability of alternative chaining resources (Li et al. 2003). That is, a framework for chaining is needed which integrates divergent resources in order to balance their deficits. In this paper we present such a framework. We describe a model for representing lexical networks as chaining resources: The model is *unified* as it maps a broad range of resources as terminological ontologies (e.g. *WordNet*), co-occurrence networks and social ontologies (e.g. the *Wikipedia*). The model provides *interoperability* as it includes an interface for integrating lexical resources which comply to a certain minimal representational standard (cf. Section 2). The model also provides a format for representing the input data as well

as resources used for lexical chains and the output of a chaining process. This is done to enable cascaded chaining algorithms whose stages operate on the output of the preceding one or to evaluate the goodness of different chaining resources on the same input data. The presented approach is implemented in a web application, providing users free access to the unified lexical chaining module, its resources and its exchange format.

In summary, this paper presents a unified framework for representing the input/output data of lexical chaining in support of interoperability of chaining resources and tools. The paper is organised as follows: Section 2 presents the framework for representing lexical networks as input to chaining. Section 3 describes the approach for representing the output as a generalized exchange format. Finally, Section 4 describes the software architecture of the actual lexical chaining application which combines both meta models.

2 A Meta Model of Lexical Chaining

Generally speaking, a chaining algorithm chains two tokens a, b of a text T complying to the following types of constraints: *Text external constraints* focus on the paths connecting the types x, y of a, b in the lexical network G (e.g. WordNet) which underlies chaining. *Text internal constraints* relate to the distance of a and b in T computed, e.g., in terms of units of the logical document structure in-between a and b. Text external constraints may be based, e.g., on the shortest path between x, y in G, on a measure of their semantic relatedness in G or on some patterns of allowable paths (Budanitsky and Hirst 2006). Note that measures of semantic relatedness usually suppose a bipartite graph structure as provided by words in relation to synsets. In our approach we abstract from the specifics of the lexical network G when specifying text external constraints. This is done by means of a *Generic Lexical Network Model* (GLNM) as input to a generalized chaining algorithm. The idea behind this approach is that concrete lexical networks (e.g. WordNet) can be modeled as instances of this graph model so that finally the GLNM is the single input format to the chaining algorithm. Following this approach, any new lexical network can be made input to chaining by simply mapping its constituents to the GLNM. In this sense, our approach provides interoperability of different chaining resources. The software architecture presented in this paper incorporates the GLNM by means of three network instances (cf. Table 1), namely *GermaNet* (i.e. a German pendant to WordNet (Lemnitzer and Kunze 2002)), the *Leipziger Wortschatz* (i.e. a very large German co-occurrence network (Biemann et al. 2004)), and the German release of the *Wikipedia* (Ponzetto and Strube 2007).

We have unified these three resources and made them input to a generalized chaining algorithm based on the GLNM. This algorithm allows to refer to network

Network	**Bipartite**	**Vertices**	**Typed**	$\lvert V \rvert$	**Edges**	**Typed**	$\lvert E \rvert$	**Weights**
GermaNet	+	words, synsets	+	≈ 0.1m	word & sense relations	+	≈ 0.3m	−
Wikipedia	+	articles, ...	+	≈ 1.5m	hyperlinks	+	≈ 19.0m	−
Leipzig	−	words	+	≈ 10.0m	co-occurrence relations	−	≈ 60.0m	+

Table 1. Lexical networks in relation to the generic lexical network model.

resources in isolation as well as in combination, where a fall-back strategy is used to select the resources.

3 A Generalized Format for Lexical Chains

Our goal is to provide a format which enables cascaded stages of chaining operating on the output of preceding stages. The format discussed (called *SGF-LC*) is an application of the *Sekimo Generic Format* (SGF) that is described in detail in Stührenberg and Goecke (2008) and which conforms to the following requirements:

- It uses a stand-off annotation format;
- it is as light-weight as possible to speed up processing;
- it can include other annotations or information if desired.

An SGF instance (and therefore an SGF-LC instance) can contain resources or references to resources stored elsewhere (both optional) and at least one `corpusData` element consisting of the *primary data* (i.e. the data that a resource is applied to) and its annotation (multiple annotation layers are supported). The `resources` element adds the ability to store the resource(s) (in our example GermaNet) used in a lexical chaining process. A `resource` element consists of optional metadata and the XML representation of the resource itself (or a reference to an external resource); in the example instance supplied in Listing 1.1 a GermaNet synset. The base layer that is provided by SGF uses the position in the character stream as boundaries for segments (e.g. a token or a part of a token) that take part in an annotation (in this case a lexical chain). These segments are stored in the `segment` elements and are identified via their respective `xml:id` attribute while the annotation of the lexical chains is stored underneath the `annotation` element, using a different XML namespace.

Listing 1.1. Lexical Chaining SGF Instance

```
<base:corpus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.text-technology.de/sekimo␣root.xsd"
  xmlns:base="http://www.text-technology.de/sekimo">
  <base:resources>
    <base:resource xml:id="gn">
      <base:content xmlns:gn="http://www.sfs.uni-tuebingen.de/lsd/">
        <gn:synsets>
          <gn:synset id="nBesitz.12" wordClass="nomen">
            <gn:lexUnit Eigenname="nein" artificial="nein" id="nBesitz.12.
                 Auktion" orthVar="nein" sense="1" stilMarkierung="nein">
              <gn:orthForm>Auktion</gn:orthForm>
            </gn:lexUnit>
            <!-- [...] -->
          </gn:synset>
        </gn:synsets>
      </base:content>
    </base:resource>
    <base:resourceRef xml:id="wiki" uri="wikipedia.dump"/>
  </base:resources>
  <base:corpusData xml:id="c1" type="text" sgfVersion="1.0">
    <base:primaryData start="0" end="100" xml:lang="de">
      <base:location uri="ex1.txt" mime-type="text/plain" encoding="UTF-8"/>
    </base:primaryData>
    <base:segments>
      <base:segment xml:id="i1" type="char" start="1" end="5"/>
      <base:segment xml:id="i2" type="char" start="6" end="10"/>
      <base:segment xml:id="i3" type="char" start="23" end="29"/>
      <base:segment xml:id="i4" type="char" start="33" end="45"/>
    </base:segments>
    <base:annotation>
      <base:level xml:id="lc" resourcesUsed="gn␣wiki">
        <base:meta><!-- [...] --></base:meta>
        <base:layer xmlns:lc="http://www.text-technology.de/lc"
          xsi:schemaLocation="http://www.text-technology.de/lc␣lc.xsd">
          <lc:link from="i4" to="i1" relation="hyperonymy" relationDirection="l"
            distance="0.848" distanceType="WordnetVektor" resourceUsed="gn"/>
          <lc:link from="i4" to="i3" relation="transitive" relationDirection="r"
           distance="-0.434" distanceType="NSS-Gwikipedia" resourceUsed="wiki"/>
          <lc:chains>
            <lc:chain id="lc_ch1" label="wirtschaft">
              <lc:ref base:segment="i1"/>
              <lc:ref base:segment="i3"/>
            </lc:chain>
          </lc:chains>
        </base:layer>
      </base:level>
    </base:annotation>
  </base:corpusData>
</base:corpus>
```

An SGF `corpusData` element can consists of several `annotation` elements (e.g. a logical document structure, the output of a parser/tagger). Each `annotation` element can contain optional metadata and the XML representation (as children of the `layer` element) of the annotation level.

In SGF-LC the actual storing of the lexical chains takes place underneath the `layer` element in line 32. The `link` element is used to represent relations between two or more items, e.g. which where applied in generating the chain. These relations are classified by the anchor (the `from` attribute), the target (the `to` attribute), the relation type and its direction. The distance between the segments that are referred to by the `from` and `to` attribute is described via the attributes `distance` and `distanceType`. The identification of the lexicographic resource used to calculate this distance is managed by the `resourceUsed` attribute. In the example listing two different resources have been used: GermaNet (which is stored inline, cf. lines 5–16 in Listing 1.1) and a Wikipedia dump (stored outside of the SGF instance, cf. line 17 in Listing 1.1). The effectiveness of each of these resources can easily be identified by traversing all `link` elements with the corresponding value of its `resourceUsed` attribute.

The `chain` element consists of at least one instance of `ref` elements referring to the before defined text spans by the `base:segment` attribute provided by the SGF base layer. Each chain has to be identified by a unique identifier and must bear a label.

Overall, SGF-LC allows for the cascaded application of chaining algorithms and, thus, supports interoperability for different chaining algorithms possibly based on different resources. In addition, not only the primary data used as input for chaining (and its respective output) can be stored, but the resources used in the chaining process as well. For this reason the described format can be employed as an exchange format and as basis for the evaluation of different lexical chaining resources and algorithms.

4 Framework of the Lexical Chaining Application

An aim of our project is to provide access to a unified lexical chaining application. There is already a numerous number of on- and offline lexical chaining applications. However, most of them are based on one single lexical resource (e.g. GermaNet or WordNet) and enables users to view different word senses or semantic relations connected to one token, focusing on experiments with manual annotation of lexical chains. Often, relatedness is delimited by its graph path and is visualized in a text-list style or through images. Other approaches such as the one described in Cramer and Finthammer (2008) measure semantic relatedness and similarity based on measurements (cf. Jiang and Conrath 1997; Leacock and Chodorow 1997; Li et al. 2003;

Lin 1998; Resnik 1995), offering the user not one but two possible related tokens as an input. However, computing lexical chaining on the basis of having an entire text or corpus to process, there are very few applications freely available. In addition, comparing different resources based on the same input is circuitous, because of the different implementations and export formats needed to be accessible and adjusted. As one result of our project, we developed an online application: *Scientific Workplace* (cf. Figure 1) which is freely available online[1] and which enables users to access the integrated divergent lexical resources as described in Section 2 and its generalized exchange format as described in Section 3. The framework of the application can be subdivided into three constitutive modules.

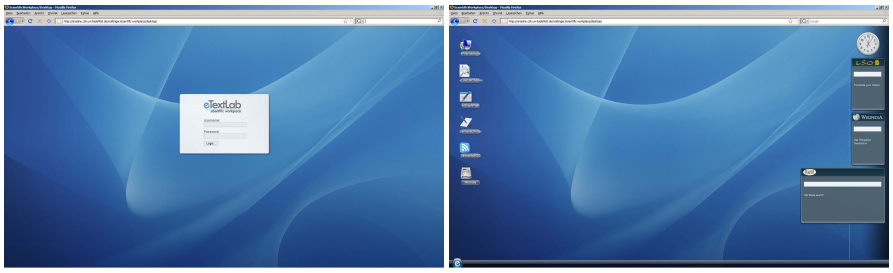

Figure 1. Scientific Workplace – Interface

4.1 Preprocessing

Since our aim was to enable users not only to compare different resources, but also to use different input formats, the software deployed needed to be quite comprehensive regarding the preprocessing of input texts. First, possible input formats such as plain text, pdf-, word documents and web-resources (e.g. a web page referred by its URL) need to be extracted. Referring to this, input parser and converter for the HTML-Stripping were implemented in the software architecture. The sentence separation is done by a token heuristic approach following Kiss and Strunk (2006). Second, single token have to be tagged in order to get the morphological information. The tagging architecture of the system integrates a trigram HMM-Tagger following Brants (2000) with an F-measure of 0.96. Training and evaluation is based upon the German Negra-Corpus (Uszkoreit Hans 2006). Named entity recognition is done by a simple rule-based module adopted from Cunningham et al. (2007). Third, most of the lexical resources are based on lemmata, while an input text consists mainly of word forms.

[1] `http://www.scientific-workplace.org/`

Algorithm 1 Computing Lexical Chaining.

Require: T, G.

```
set LC
for each token t1 of the input text T do
    for for each token t2 within smax do
        Connect that v ∈ G where rm <= lmax.
    end for
end for
```

Our lemmatization module consists of a rule-based noun lemmatizer and a word form lexicon of around 4.9 million word-lemma pairs.

4.2 Resources and Chaining

The GLNM, as described in Section 2, is used as the template structure for the different resources. A single resource is though defined as a graph $G = (V,E)$ where V is the set of all lemmata of our lexical network and $E \subseteq V^2$ the corresponding set of edges. The lexical chain graph to be computed is defined as $LC = (V',E',\sigma)$ where V' is the set of all used lemmata of our input text T and $E' \subseteq V'^2$ the corresponding set of edges, computed by the lexical chaining algorithm. σ is the edge weight function. The generalized chaining algorithm, adopted of Hirst and St-Onge (1998), is computed by deploying a breadth-first search (BFS), which starts at a certain vertex V and then explores all neighbouring vertices in G. Therefore the complexity computing a single chain is $O(|E|+|V|)$, since each vertex and each edge of our graph G will be explored in the worst case. Since this approach calculates the length of a shortest path sp between two vertices $v1$, $v2$, our relatedness measure can be defined as $rm(v1,v2) = sp(v1,v2)$. Following this, we argue that there is a maximum allowed length $lmax$ between $V1$ and $V2$ in a lexical network which represents the borderline of relatedness. By that, if $rm > lmax$ we define $v1,v2$ as not related.

An input text T is further defined by the number i of tokens t. To this point we have not regarded the actual document structure of a text. Since a token of a text in the first paragraph tends to be more likely to be semantically related to a token in the same paragraph, than in the last paragraph, we have to introduce a second parameter $smax$. This parameter sets the maximum allowed paragraph window within the document structure of the text instance. For example, if we set $smax = 1$, only those tokens within the same paragraph will try to connect to each other. Having set both parameter $lmax$ and $smax$, we iterate over i and chain each $v1$ with $v2$ within $smax$. If $rm <= lmax$ we add e as e' to E' of LC. As a result we get a partitioned graph LC, representing the chained tokens, connected by E' with an σ of rm. It is

important to note that only those tokens are beard for the chaining, which are covered by the operative lexical reference system. Therefore, we deployed a fall-back strategy in selecting multiple resources. In this sense, we provide interoperability of different chaining resources, since the algorithm allows to refer to network resources in isolation as well as in combination. Figure 2 shows an example of a text about *Wimbledon* using GermaNet as the lexical resource. For an outline of the algorithm used for chaining cf. Algorithm 1.

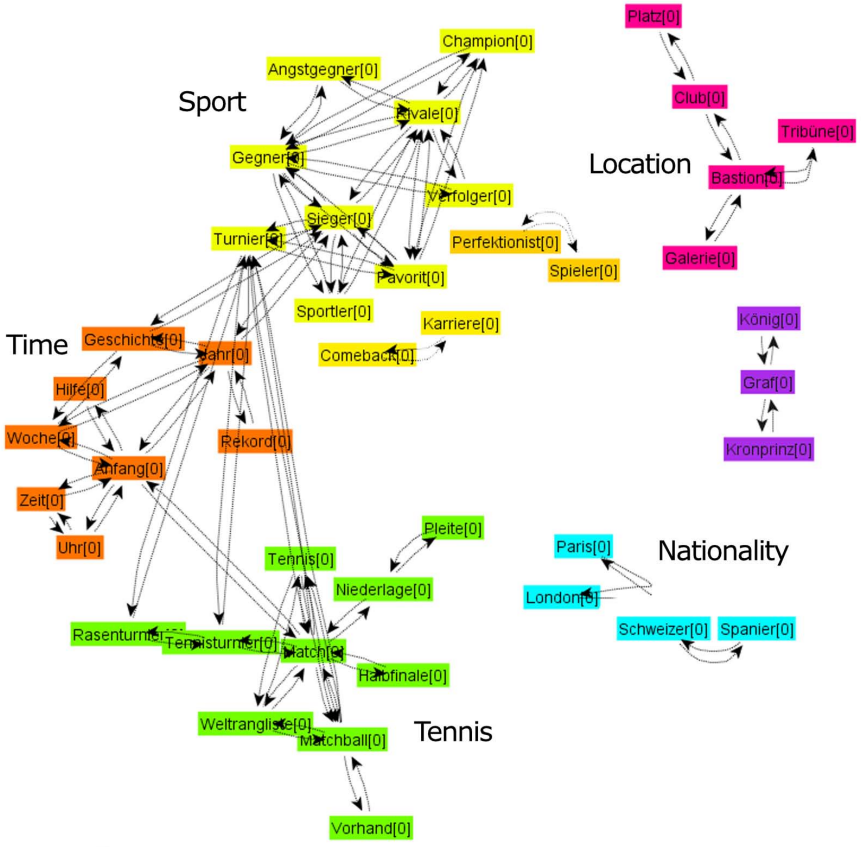

Figure 2. Lexical Chaining using GermaNet

4.3 Output and Depiction

After the chaining module is completed, the corresponding *LC* will be converted in the generalized format for lexical chains as described in Section 3. Since we focus on different resources, the same text can be repeatedly chained by different or combined lexical resources. The Graphical User Interface (GUI) of the *Scientific Workplace* consists of the main desktop area and the corresponding modules (cf. Figure 1). Each module is highlighted by an icon placed on the web desktop area. The entire web application incorporates a user as well as a file management module. In consideration of the lexical chaining an easy-to-use chaining component was deployed which allows users to drag-and-drop their files onto the chaining application (cf. Figure 3). Thus, having a generalized format for lexical chains there is still the need to visualize the resultant chained text. For this purpose, we have implemented a chain viewer which enables the user to select different chained components of the computed text by highlighting only connected tokens of the chained component (cf. Figure 4). As a visual depiction we produce a *token cloud* that highlights the labels (mostly frequent tokens within a chained component) of the document (cf. Figure 5). Additionally, a

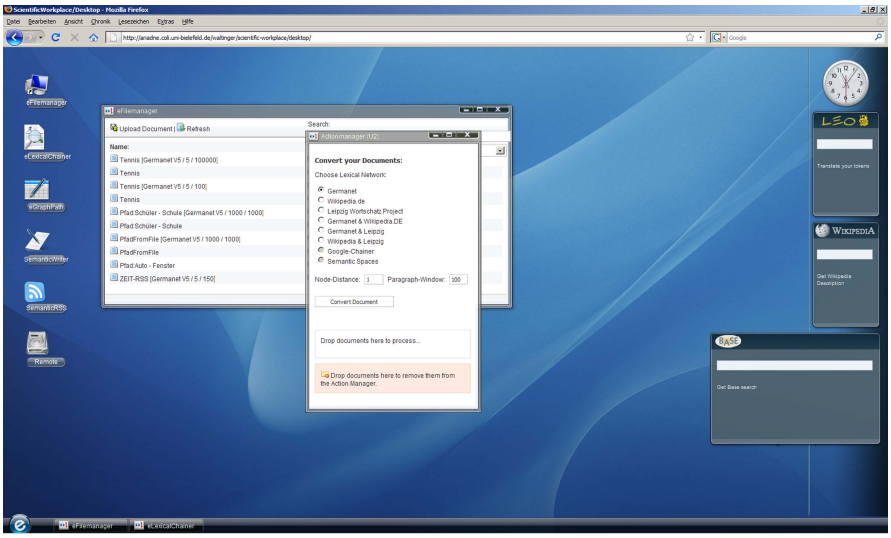

Figure 3. Scientific Workplace – Chainer

graph path module is included that allows to compute the shortest path between two tokens based on the preselected resource. As output either a SGF-LC file is rendered to download or a text file with the result can be viewed. Therefore, either token pairs or entire texts can be used in this online chaining application. The input and output

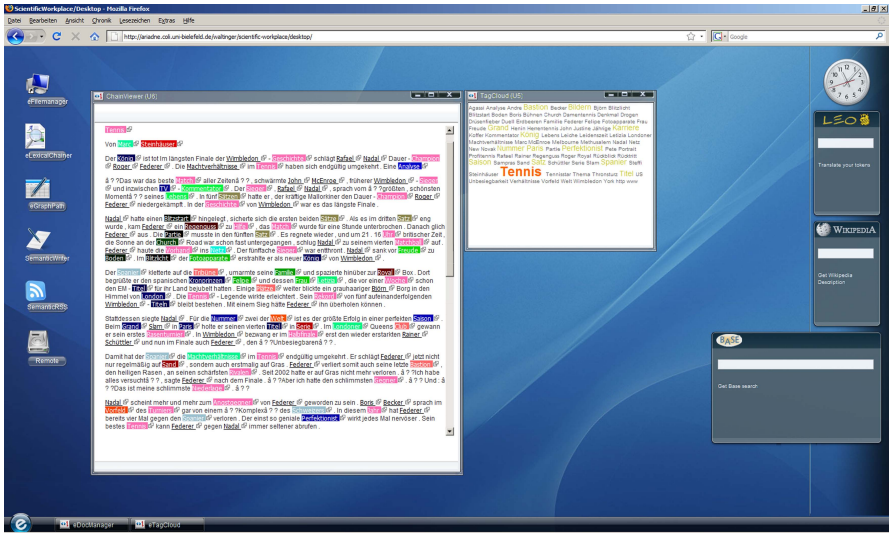

Figure 4. Scientific Workplace – ChainViewer

is stored in a uniform schema in a MySQL database. The web-based chaining tool is running on an Apache web server, parser, converter and the actual algorithm is developed in C++. GUI and actions are coded with PHP and Javascript. Therefore, the only requirement for running the application is a web-browser (Javascript enabled) and a registered account at our website. We are currently working on extending the unified resource package by a semantic space module, a translation of WordNet and a web search engine (*yahoo.com*) chaining module. Requests on implementing special resources of other research projects are welcome.

5 Conclusion

In this paper, we presented a framework for lexical chaining that integrates divergent resources. We developed a Generic Lexical Network Model (GLNM), an approach to model concrete lexical networks as instances of a single input format. Furthermore, we outlined a generalized format for lexical chains (SGF-LC), which enables cascaded stages of chaining operating on the output of preceding stages. Additionally, we developed an online application, called *Scientific Workplace*[2], which enables

[2] `http://www.scientific-workplace.org/`

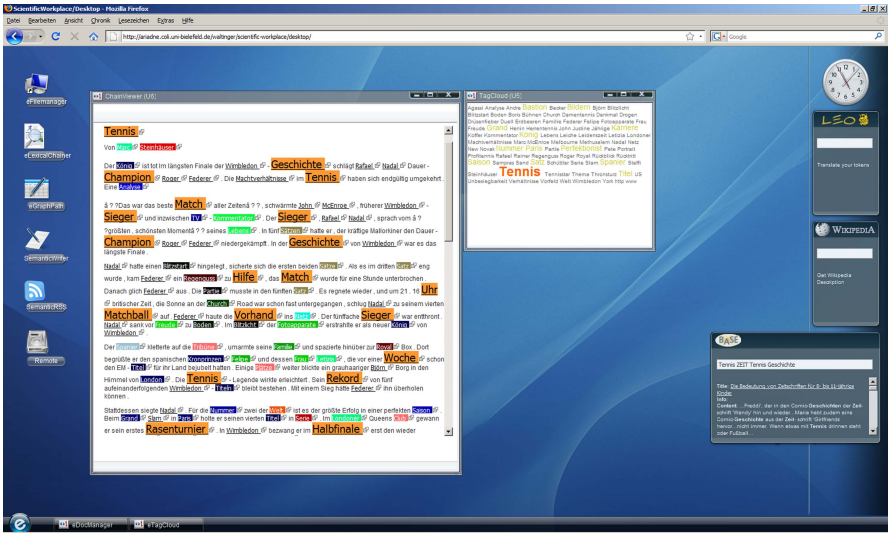

Figure 5. Scientific Workplace – TokenCloud

users to access the integrated divergent lexical resources as well as the generalized format. In a nutshell, this paper presented a web application that incorporates the framework of a meta model of lexical chaining and its generalized exchange format.

Bibliography

Barzilay, Regina and Michael Elhadad (1997). Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, Madrid, Spain*.

Biemann, Christian, Stefan Bordag, Gerhard Heyer, Uwe Quasthoff, and Christian Wolff (2004). Language-independent methods for compiling monolingual lexical data. In Alexander F. Gelbukh (ed.), *CICLing*, volume 2945, 217–228, Springer.

Brants, Thorsten (2000). Tnt - a statistical part-of-speech tagger. In *Proceedings of theSixth Applied Natural Language Processing Conference ANLP-2000, Seattle, WA.*, URL `http://www.coli.uni-saarland.de/~thorsten/tnt/`.

Budanitsky, Alexander and Graeme Hirst (2006). Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics* 32(1):13–47.

Cramer, Irene and Marc Finthammer (2008). An evaluation procedure for word net based lexical chaining: Methods and issues. In *Proceedings of Global WordNet Conference 2008, Szeged, Hungary*.

Cunningham, Hamish, Kalina Bontcheva, Valentin Tablan, and Diana Maynard (2007). Gate - a general architecture for text engineering. URL `http://gate.ac.uk/`.

Ferret, Olivier (2002). Using collocations for topic segmentation and link detection. In *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002, Taipei*, 260–266, Morgan Kaufmann.

Green, Stephen J. (1999). Building hypertext links by computing semantic similarity. *IEEE Transaction on Knowledge and Data Engineering* 11(5):713–730.

Hirst, Graeme and David St-Onge (1998). Lexical chains as representations of context for the detection and correction of malaproprisms. In Christiane Fellbaum (ed.), *WordNet – An Electronic Lexical Database*, Cambridge, Massachusetts: MIT Press.

Jiang, J.J and D.W. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *ROCLING X (1997), Taiwan, 1997.*

Kiss, Tibor and Jan Strunk (2006). Unsupervised multilingual sentence boundary detection. In *Computational Linguistics*, 485–525.

Leacock, Claudia and Martin Chodorow (1997). Combining local context and wordnet similarity for word sense identification. In *Fellbaum 1997.*, 265–283.

Lemnitzer, Lothar and Claudia Kunze (2002). Adapting GermaNet for the Web. In *Proceedings of the First Global Wordnet Conference*, 174–181, Central Institute of Indian Languages, Mysore, India.

Li, Yuhua, Zuhair A. Bandar, and David McLean (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering* 15(4):871–882.

Lin, Dekang (1998). And information-theoretic definition of similarity. In *5th International Conference on Machine Learning, Madisin, WI.*

Marcu, Daniel (2000). *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, Massachusetts: MIT Press.

Mehler, Alexander (2005). Lexical chaining as a source of text chaining. In Jon Patrick and Christian Matthiessen (eds.), *Proceedings of the 1st Computational Systemic Functional Grammar Conference, University of Sydney, Australia*, 12–21.

Morris, J. and G. Hirst (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17(1):21–48.

Ponzetto, Simone Paolo and Michael Strube (2007). Deriving a large scale taxonomy from wikipedia. In *Proceedings of AAAI '07*.

Resnik, Philip (1995). Computation and language. In *14th International Joint Conference on Artificial Intelligence.*

Schütze, Hinrich (1998). Automatic word sense discrimination. *Computational Linguistics* 24(1):97–123.

Silber, H. Gregory and Kathleen F. McCoy (2002). Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics* 28(4):487–496.

Stokes, Nicola (2004). *Application of Lexical Cohesion Analysis in the Topic Detection and Tracking Domain*. Ph.D. thesis, National University of Ireland, Dublin.

Stührenberg, Maik and Daniela Goecke (2008). Sgf – an integrated model for multiple annotations and its application in a linguistic domain. In *Proceedings of Balisage – The Markup Conference*.

Teich, Elke and Peter Fankhauser (2005). Exploring Lexical Patterns in Text: Lexical Cohesion Analysis with WordNet. In *Interdisciplinary studies on information structure*, 129–145, SFB 632, Universität Potsdam.

Uszkoreit Hans, Brants Sabine Foeldesi Christine, Brants Thorsten (2006). Negra corpus. URL `http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/`.