Semantic Navigation of News

Walter Kasper, Jörg Steffen and Yajing Zhang*

DFKI GmbH, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany

Abstract. We present a system for browsing a news repository that is based on semantic similarity of documents. News documents get automatically annotated semantically using information extraction. Annotations are displayed to a user who can easily retrieve crosslingual semantically related documents by selecting interesting content items.

1 Introduction

In the MESH project (http://www.mesh-ip.eu) news from online sources get analyzed for automatic annotation of their content with respect to domain specific information about major reported events. News in English, German and Spanish from *Deutsche Welle* (http://www.dw-world.de) and *BBC* (http://news.bbc.co.uk) are used. The service enables professional users to search not only for news stories about specific events but also to get a comprehensive overview of the information available from different sources and supports intelligent navigation within related information and news about the same and similar events. The most basic information about such events that gets extracted is their time and location as well as involved persons and organizations.

The SENA system described here is based on a scenario like this: a professional user is searching a news archive or repository for information about some event. He already found some relevant document. Now he is interested to know whether there are other documents about the same event or similar events in the same region, etc. This is where SENA can help: when the user is viewing a document SENA also presents an overview of relevant content items of some semantic categories that the user might be interested in to find more information about. By selecting interesting content items the user can retrieve documents related and similar to the document he is viewing with respect to the selected items. As SENA is based on semantic annotations to the news documents and not just textual search, SENA supports crosslingual retrieval of documents in a straightforward manner for multilingual news repositories.

The rest of the paper is organized as follows. Section 2 gives an overview of the system environment and major components. SENA's concept of *semantic similarity* is discussed in section 3. Then the SENA user interface will be presented in section 4. Section 5 discusses related attempts to news annotation and navigation. Finally, we will give an outlook towards further development of SENA.

* This work was supported by the European Commission, under contract number FP6-027685 (MESH).

Kasper W., Steffen J. and Zhang Y. (2008).
Semantic Navigation of News.
In *Proceedings of the 5th International Workshop on Natural Language Processing and Cognitive Science*, pages 42-51
DOI: 10.5220/0001735700420051

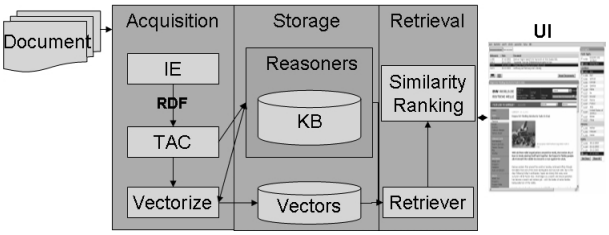

Fig. 1. SENA architecture.

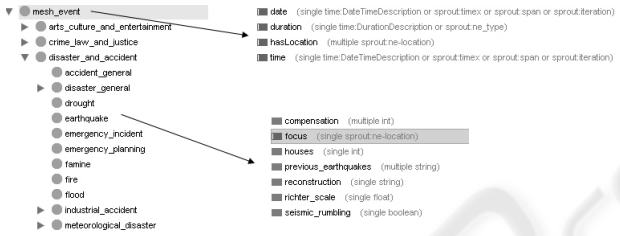

Fig. 2. Domain ontology.

2 System Environment

Figure 1 shows the overall architecture of the SENA system. Basically it consists of three layers: the *acquisition* layer where new documents get analyzed, the *storage* layer for storing the annotations and the *retrieval* layer that interacts with the user interface to retrieve documents from storage. The main components will be described in the next sections.

2.1 Knowledge Representation

Annotations about news documents are defined by OWL ontologies, the most important being a *domain ontology* derived from the IPTC subject code classification[1] and an ontology of news metadata that was derived from *NewsML*[2]. For multimedia documents a multimedia ontology based on the MPEG-7 standard is used.[3] Annotations are stored as instances of these ontologies in an RDF repository, linked to the document identifier or URL they belong to. Figure 2 shows parts of the domain ontology. The top-level class is *mesh_event*. Additionally, general properties of that class and some properties specific for the *earthquake* subclass are shown.

The ontologies provide the conceptual core of the RDF knowledge base and the basis for reasoning processes. In addition a repository of external knowledge is used to support semantic reasoning. At present, this external repository mainly contains information about locations, especially containment relations among them. Its current size

[1] http://www.iptc.org/NewsCodes/
[2] http://www.newsml.org
[3] We use the SmartWeb multimedia ontology: http://smartweb.dfki.de/ontology_en.html

is about 1.2 million RDF triples. In SENA this allows to retrieve documents concerning e.g. Bavaria even if the document only mentions some place that is located in that region.

2.2 Information Extraction

The core annotation and information extraction engine is provided by the SProUT platform ([1, 2]). SProUT combines finite state techniques with unification of typed feature structures (TFS). TFS provides a powerful device for representing and propagating information. Rules are expressed by regular expressions over input TFSs that get instantiated by the analysis. The uniform use of TFSs for input and output also allows for cascaded application of rule systems. For the MESH application the rule system is organized as

- a set of core named entity recognition (NER) grammars for date and time expressions, locations, persons and organizations
- domain specific annotation and extraction grammars based on the domain ontologies
- an upper level merging component that merges partial annotations and information into larger chunks, usually at paragraph level. It uses unifiability as main compatibility test.

TFSs support well information extraction (IE) by allowing to model OWL domain ontologies: basically, the OWL classes become types, the properties are treated as features.[4] As a result the domain ontology is represented in TFS by complex types that provide templates that need to be filled by IE. This approach also makes it easy to map the results of IE to instances of the domain ontology for further semantic processing and storage in the RDF repository.

So, IE in SENA goes beyond semantic tagging of phrases to detect co-occurrence of entities but attaches the information to event types. The analysis would not just detect that an *earthquake* event and that a location like *Iran* is mentioned but if possible would link the location to the event by a *hasLocation* property.

The result of IE is an RDF representation such as this for *earthquake in Iran*:[5]

```
mesh:id1   rdf:type  mesh:earthquake ;
           mesh:hasLocation mesh:id2 .
mesh:id2   rdf:type  mesh:country ;
           mesh:name "Iran" .
```

On average, IE generates about 300 (raw) RDF statements as annotations for a document.

[4] A similar idea is underlying the approach of [3].

[5] Actually UUIDs are used as unique instance identifiers.

2.3 Temporal Anchoring of Events

Often news texts talk not only about one major event but refer to other events and circumstances as well. Important clues to distinguish what belongs to the main topic of a news article and what is subsidiary information is given by temporal information. However, this information may not be given explicitly by some date or time expression but by relative references such as *the week before*. In order to distinguish and relate event information from different news sources, a *Time Anchoring Component* (TAC) was developed to resolve temporal expressions and anchor events to some date or time.

The MESH ontology uses *OWLTime* ([4, 5]) which provides classes for representing temporal instants, intervals and duration. The core date-time representation is the class *DateTimeDescription*.

The use of *OWLTime* presupposes to some extent that dates or times are completely specified. But it poses some problems for the representation of partial and underspecified temporal expressions as used in natural language text. The TAC component described here bridges the gap between temporal natural language expressions and *OWL-Time* representations.

We distinguish two types of temporal expressions:

- *Explicit expressions* refer to a specific point of time or period of time. It can be unambiguously identified in a calendar, for instance, *June 6th, 2006*.
- *Relative expressions* refer to a point of time or period of time that can only be unambiguously identified with the help of a reference time given by context. Examples include *yesterday*, *two hours later*, *in summer*, etc.

Different from the division made by [6], we do not distinguish deictic and relative expressions, since both of them require a contextually given reference time to anchor the expression correctly. The difference is only in the type of context.

The result temporal representation consists of a structure with features[6] ordered by their *granularity*: [*second* < *minute* < *hour* < *pofd* < *dofw* < *day* < *weeknumber* < *pofm* < *month* < *pofy* < *year*]. Among all these features, the feature *year* is obligatory which means each result at least must specify a value for *year*.

TAC includes a complete calendar model for date and time calculations as well as a temporal interval reasoner (cf. section 2.4 below) for handling time spans.

TAC's input consists of the RDF model from the IE system. The relevant temporal representations are extracted by SPARQL queries ([7]) on the model. The reference time is context-dependent. For news items it is often the publication or creation time. But the reference time is a dynamic concept: when another explicit time is mentioned in the text, it can become the new reference time for subsequent sentences in the same paragraph.

TAC also decides about the granularity level at which completion is necessary. The result inherits the granularity of the original incomplete expression. For instance, let the reference date be *October 24, 1997* (Friday). In example (1) the granularity of the original expression is *dofw* and is anchored to *Wednesday, October 15, 1997* while example (2) has the granularity of *minute* and will be anchored to *October 24, 1997, 4:20*.

[6] pofd: part-of-day, dofw: day-of-week, pofm: part-of-month, pofy: part-of-year

(1) The president flew to Berlin on *last Wednesday*.

(2) The earthquake happened at 04:20 *this morning*.

2.4 Reasoning

The knowledge base (KB) is embedded into a reasoning component (RC). All access to the KB has to go through the RC to prevent getting out of sync with the KB if that were directly modified behind RC's back. When the KB gets updated by analysis, the RC can apply forward chaining reasoning to add immediately some standard types of deductions.

SENA uses the Jena framework for RDF handling and storage[7]. Jena also provides a generic rule reasoner. In addition to pure rule reasoning, the Jena reasoner provides an extensible set of operators, e.g. for arithmetic operations. In SENA the base set of built-ins was extended to provide the core operators specified by the *Semantic Web Rule Language (SWRL)* standard.[8]

The RC consists of several reasoner modules: besides a generic OWL reasoner especially for inheritance and instance classification there are special purpose reasoners extending the Jena reasoner. These provide additional functionality that is difficult, inefficient or impossible to achieve by pure monotonic, open-world deductive reasoning. One example of such reasoning is TAC that was developed as it would be hard to deal with the necessary contextual and numeric calendar calculations in a strictly deductive manner.

One set of reasoners is applied when the KB gets updated by new analysis data. Specific reasoners provide the following functionalities:

- *Validation*: the RDF model from analysis is validated against the SENA ontologies for correctness. Also, the RDF model is purged with respect to intermediate analysis data, not part of the system ontologies.
- *Identity reasoner*: the RDF data from IE typically contains distinct instances for distinct occurrences of some entity. The identity reasoner checks whether entities from this set of fresh instances can be identified with each other and with entities already contained in the knowledge base. Basically, the reasoner derives a set of identity statements as *owl:same-as* statements, then merges all the information about the identical entities and links it to a base representative instance. It uses a mix of rule reasoning and heuristic reasoning, such as that person names might refer to the same person if their last name and the initials of their first names are equal. Typically, the identity reasoner reduces the annotation set by about 30-40% as persons, location etc. often can be identified with already known instances.
  If there are several possible instances some entity might be identified with, disambiguation is necessary. E.g., our KB knows about eleven places named *Paris*, most of them being in USA. For disambiguation, the reasoner then checks other locations in the model whether they can be related to common location "containers".

[7] http://jena.sourceforge.net

[8] The extensibility and adaptability of the Jena reasoning engine was the main reason for not using an OWL DL reasoner such as *Pellet* ([8]) as basis.

A second set of reasoners is mainly responsible for reasoning services at retrieval time. It helps to infer implicit information in response to queries to the KB. These reasoners are monotonic based on standard first-order logic implementing the OWL semantics. This set includes a temporal reasoner, based on an efficient implementation of Allen's interval logics ([9, 10]) as well as a location reasoner for reasoning on the location database.

3 Semantic Similarity of Documents

The most important information pieces for a document are contained in the KB as instances of the *mesh_event* class and its subclasses. The properties of an event specify its location, time and duration, etc. Given an unique document id (like its URL) a SPARQL query ([7]) is used to get all event instances for that document from the KB.

Each event instance can be interpreted as the root of a directed graph, with the properties as edges, instances and literals as nodes. We group all event instances of a document under a single artificial root node to get a single graph representing the events described in the document. We use these document graphs as a base for defining a similarity measure for documents. Numerous similarity measures for graphs exist. We use a similarity measure based on the maximal common subgraph of two graphs (e.g. [11]). This similarity measure has also the advantage that it is a metric. So for two documents D_1 and D_2 represented by the two document graphs G_1 and G_2, we define the maximum common subgraph similarity (*mcs*) as:

$$sim_{mcs}(D_1, D_2) = \frac{|mcs(G_1, G_2)|}{max(|G_1|, |G_2|)} \tag{1}$$

This results in a *mcs* document similarity measure in an interval [0,1], 1 if G_1 and G_2 are isomorphic and 0 if G_1 and G_2 are completely distinct.

Our implementation of *mcs* document similarity is based on the SimPack Java library ([12]). The SimPack accessor for Jena ontologies was extended to use instances of classes instead of just classes as we are interested in instance similarity, not in ontology similarity.

Besides the event annotations SENA creates additional annotations for information that cannot be related directly to the main events mentioned in the documents but can be of interest nevertheless such as persons or organizations mentioned. As we want to use all annotations to compare two documents, we define a second similarity measure that only considers these "unbound" instances.[9] We use a vector space model where the extracted instances are arranged in a vector where each instance corresponds to a dimension ([13]). The normalized frequency count of an instance c represents its weight in the vector:

$$weight(c) = \frac{|c|}{N_{type(c)}} \tag{2}$$

[9] These instances, of course, have also properties with literals as values and could also be treated as graphs, but the identity reasoner described in section 2.4 makes sure that different occurrences of the same person, organization and location are represented by a single unique instance in the knowledge base. This is why we consider these instances as single nodes.

where $N_{type(c)}$ is the total number of occurrences of instances of that type.

The weight of instances from the headline of a document is boosted to accommodate the fact that the headline often holds the most important concepts of the document. Such instances are treated as if they appeared 5 times as often in the document. A similar mechanism is used for re-ranking user selected concepts on retrieval (cf. section 4).

For vector space modeling also a number of similarity measures exist. We use the cosine similarity. This measure quantifies the similarity between the two vectors as the cosine of the angle between the two vectors. The similarity of two documents D_1 and D_2 with the document vectors $\overrightarrow{D_1}$ and $\overrightarrow{D_2}$ is then defined as:

$$sim_{cos}(D_1, D_2) = \frac{\sum_{i=1}^{n} \overrightarrow{D_{1_i}} \times \overrightarrow{D_{2_i}}}{\sqrt{\sum_{i=1}^{n} \overrightarrow{D_{1_i}}^2} \times \sqrt{\sum_{i=1}^{n} \overrightarrow{D_{2_i}}^2}} \tag{3}$$

The resulting measure is also a value in the interval [0, 1].

Both similarity measures are combined to get a total similarity value for documents. For each similarity measure sim_i of a set of measures we define a weight w_i to control its influence on the total similarity value. The total similarity value is normalized to the interval [0,1]:

$$sim(D_1, D_2) = \sum_{i=1}^{n} \frac{w_i \times sim_i(D_1, D_2)}{\sum_{i=1}^{n} w_i} \tag{4}$$

In the current setup we use equal weights for w_{mcs} and w_{cos}.

In a similar way, additional similarity measure could get integrated into this framework, e.g. text based document similarity measures such as TFIDF.

4 Navigating Documents

SENA is realized as a web-service. So only a browser is needed to access and use the system. Figure 3 shows the main window of SENA's user interface, the *Document Selection* tab. A second tab (*New Document*) allows users to add new documents to the repository by entering their URL. The new document is analyzed immediately and added persistently to the repository including the extracted annotations. In the *Document Selection* view the new documents are immediately available on a par with the old documents.

The *Document Selection* view is made up of three major areas:

- the list of selected documents, the *Selection*
- the *Document* area displaying the actual document.
- the *Navigator* area to the right.

The selection list displays the selected documents with their title. The similarity or relevance of the documents with respect to a start document the user had been viewing before creating the selection is indicated by a numerical value. This start document obviously always is the "most similar" selected document with a similarity value of 1. A second column shows the date, usually the publication or creation date of the document. The ordering of documents can be changed by clicking on the *Relevance*

Fig. 3. SENA user interface for navigation.

or *Date* header to sort according to that dimension. Sorting by date corresponds to a timeline view. The *language* buttons allow the user to exclude or include documents in that language.

Clicking into the document list displays the corresponding document in the *Document* area. For web documents, only the URL with the document's annotations are stored in the repository, not the document itself. So the document is retrieved directly by its URL from the web when requested. The selected document becomes the start document for further navigation.

The *Navigator* on the right side of the screen provides the main navigation device of SENA. Parts of the semantic annotations of the displayed document are shown there, according to semantic categories such as *event type*, *locations*, *persons* and *dates* mentioned. The date list contains not just the dates explicitly mentioned but also those indirectly referenced that have been resolved by TAC. Each item is accompanied by a relevance measure, based on the frequency, and a checkbox. The checkboxes allow the user to select a combination of interesting items that he would like to get further information about. Additionally, a time range for interesting documents and events can be specified, e.g. by using the date list that is indicative of related events. The *GetDocs* button then retrieves the documents according to his selection, computes their similarity to the viewed document (the *start document*) with respect to the selected items. The results of the retrieval then are displayed in the Selection area. From there the user can select other documents. This way the user easily can navigate within a large set of documents by going from one document to the next along a path defined by his selected concepts and the document similarity. Logically, the user selection corresponds to a Boolean *AND* query for the set of documents satisfying all the constraints.

In this first version of SENA, the content items are displayed as flat lists, giving an incomplete view on the extracted information, as it does not exhibit the semantic

relations between the entities. E.g. that the earthquake event actually had been in Bam (and not in Malta, also mentioned in the text) is not obvious from that flat representation. Better ways of representing the semantic information, maintaining the simplicity of use, are under investigation.

5 Related Work

A large number of news services exist on the web that allow searching news archives. Not only newspapers, journals or TV stations operate web portals for searching their content, but there are also numerous cross-site news aggregators, e.g. from Google or Yahoo. Usually they are based on some text indexing, classification or clustering techniques that define textual document similarity measures though these usually are fixed and not dependent on users' choices. But there are also attempts to employ more semantic based techniques. Most of the approaches appear to use more shallow term based semantic tagging techniques, such as annotating documents with *WordNet* senses ([14]), e.g. [15]. Recent examples of approaches employing semantic web ontologies and being more similar to what we are investigating in SENA are the NEWS project ([16]) and especially the CALAIS project by Reuters ([17]) that provides a web based automatic annotation service for news that can be integrated into applications. But, besides also providing NER, their ontology framework seems to be more shallow than that of SENA, resembling more classification schemes rather than defining complex domain specific semantic objects that get instantiated by information extraction. In that, SENA's approach is close to the research done in the field of *Topic Detection and Tracking* (TDT; [18]).

Interesting navigation functionality beyond search is provided by the *Europe Media Monitor* with its *NewsExplorer* ([19, 20]). But technically it seems to be closer to standard text based news services rather than using a semantic web ontology framework. Similar to SENA it offers special navigation for some classes of named entities but does not allow selection of multiple items.

6 Conclusions and Outlook

We presented a system that provides users with easy to use navigation clues for retrieving related documents based on his choice of points of interest. In contrast to most other news services SENA's retrieval is based on semantic similarity with respect to domain specific ontologies that get instantiated by information extraction technologies and supports also crosslingual retrieval in a straightforward manner. First feedback from user evaluations with news professionals in the MESH project indicates high interest in navigation aids as provided by SENA.

SENA is at an early stage of development. For the future, improvements to the user interface such as better visualization including the relations between entities from the underlying RDF graph are planned. Also, additional similarity measures such as text based measures that also support textual search will be included. An evaluation of the similarity measures and retrieval quality is another point on the agenda.

References

1. Becker, M., Drożdżyński, W., Krieger, H.U., Piskorski, J., Schäfer, U., Xu, F.: SProUT-shallow processing with unification and typed feature structures. In: International conference on NLP, Mumbai, India (2002)
2. Drozdzynski, W., Krieger, H.U., Piskorski, J., Schäfer, U., Xu, F.: Shallow processing with unification and typed feature structures — foundations and applications. Künstliche Intelligenz **1** (2004) 17–23
3. Pérez, G., Amores, G., Manchón, P., Gonzáles, D.: Generating multilingual grammars from OWL ontologies. Research in Computing Science **18** (2006) 3–14
4. Hobbs, J.R., Pan, F.: An ontology of time for the Semantic Web. ACM Transactions on Asian Language Information Processing (TALIP) **3** (2004) 66–85
5. Hobbs, J.R., Pan, F.: Time Ontology in OWL. W3C Working Draft 27 September 2006. http://www.w3.org/TR/2006/WD-owl-time-20060927/ (2006)
6. Han, B., Gates, D., Levin, L.: From language to time: A temporal expression anchorer. Proceedings of the Thirteenth International Symposium on Temporal Representation and Reasoning (TIME'06) (2006) 196 – 203
7. Prudhommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. http://www.w3.org/TR/2004/WD-rdf-sparql-query-20041012/ (2004)
8. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Journal of Web Semantics (2006)
9. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM **26** (1983) 832–843
10. Koomen, J.A.: The TIMELOGIC temporal reasoning system. Technical Report Technical Report 231, Computer Science Dept., University of Rochester (1989)
11. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters **19** (1998) 255–259
12. Bernstein, A., Kaufmann, E., Kiefer, C., Bürki, C.: SimPack: A Generic Java Library for Similiarity Measures in Ontologies. Technical report, Department of Informatics, University of Zurich (2005)
13. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. Communications of the ACM (1975) 613–620
14. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. Cambridge/Mass.: MIT Press (1998)
15. Oleshchuk, V., Pedersen, A.: Ontology based semantic similarity comparison of documents. In: Proc. 14th International Workshop on Database and Expert Systems Applications. (2003) 735–738
16. Bernardi, A.: News engine web services. http://www.dfki.uni-kl.de/ berhardi/News (2004)
17. OpenCalais: Calais. http://www.opencalais.com (2008)
18. Allan, J., ed.: Topic Detection and Tracking: Event-based Information Organization. Springer (2002)
19. EMM: Europe news monitor: News brief. http://emm.jrc.it (2007)
20. EMM: Europe media monitor: News explorer. http://press.jrc.it/NewsExplorer (2007)