Ontology-based program shell for building and editing multilingual thesauri of subject domains

Yury Zagorulko*, Galina Zagorulko*
*A.P. Ershov Institute of Informatics Systems /Artificial Intelligence Laboratory, Novosibirsk, Russia
zagor@iis.nsk.su, gal@iis.nsk.su

Abstract— The paper presents a program shell for building and editing multilingual electronic thesauri, whose general structure, as well as composition of the thesaurus entries and the set of relations between the terms, meets the international and Russian standards. Using this shell, experts (i.e. specialists in a certain subject domain) can build multilingual thesauri for arbitrary languages and subject domains without the help of programmers and knowledge engineers. The shell is ontology-based, which provides not only the possibility of extending the set of relations between the terms and the integrity and consistency of the terminology system of the created thesaurus but also a convenient access to its content.

I. INTRODUCTION

To cope with a steady growth of the body of information presented in the textual form and in various languages, efficient facilities of automatic processing and retrieval of documents are required. Important components of such facilities are multilingual thesauri, which have recently been widely used both in document indexing and information retrieval [1, 2] including the Internet [3] and in automatic text processing [4]. Along with ontologies [5], thesauri play a critical part as a facility for describing a subject domain (SD) designed both for knowledge structuring and educational purposes.

In view of these reasons multilingual thesauri presenting the terminology of one or another knowledge area are in great demand. Though a large number of software tools for building thesauri have been developed by now [6, 7], there are no accessible, easy-to-use, flexible and at the same time sufficiently powerful and reliable tools.

Here by accessible tools we mean cheap or shareware tools, by easy-to-use tools we mean tools that can be easily used by experts in a subject domain without any training (extension work) or help of IT-specialists (programmers or knowledge engineers). By flexible tools we mean tools which allow one to easily adjust a thesaurus to a required SD, that is at any time add new properties to the description of terms and extend the set of relations between them. By sufficiently powerful tools we mean tools providing developers with all the necessary means for representation of all basic entities of a thesaurus and description of their properties. By reliable tools we mean tools automatically supporting the logical integrity of the thesaurus terminology system.

The paper presents a program shell intended for building multilingual electronic thesauri for any subject domain and designed according to the requirements described above. When designing the program shell, the authors were guided by the international and Russian standards regulating the development of information retrieval thesaurus (IRT), i.e. thesauri oriented to document indexing and information retrieval.

The paper is structured as follows. Section 2 presents and discusses the thesaurus conceptual scheme. Section 3 describes the architecture of the program shell and facilities for thesaurus editing and adjustment to a subject domain. Possibilities provided by the user interface of the program shell are considered in Section 4. Application of the program shell for development of the Russian-English thesaurus on computational linguistics is described in Section 5. Technical features of the program shell and its applications are outlined in the Conclusion.

II. A CONCEPTUAL SCHEME OF A THESAURUS

Before proceeding to a conceptual scheme of thesaurus, we shall specify what we mean by “thesaurus”.

Based on the definition from [8], thesaurus is meant as a specific kind of dictionaries of special and general lexicon where lexical units are linked by semantic relations (synonymy, hyperonymy, partonomy, etc.). In contrast to other kinds of dictionaries, in particular, an explanatory dictionary, thesaurus allows one to determine the meaning of a term (concept) by giving not only its definition but also the correlation of this term with other terms using the semantic relations between them. Thereby, all terms of thesaurus become linked in a single semantic network representing the system of knowledge about some subject domain.

Respectively, by a multilingual thesaurus we mean a dictionary whose basic units are terms presenting language expressions (words and phrases) for the concepts of the modeled subject domain in several languages, which are linked by semantic relations.

To make control of a thesaurus easier, as well as to have a possibility of using it for manual indexing and acquisition of knowledge about the modeled subject domain, it includes not only the definitions of the most important terms but also the descriptions of their sources, i.e. text documents or collections of text documents which contain these terms or their definitions.

A thesaurus will be a useful information resource if it will present a consistent system of concepts of an SD.

* The authors are grateful to the Russian Foundation for Basic Research (grant No. 13-07-00422) and the Presidium of the Russian Academy of Sciences (grant SB RAS No. 15/10) for financial support of this work.
Thereby, the program shell for building a thesaurus should provide not only the structures for representation of all its entities (both basic and auxiliary) and relations between them but also formal means for describing their structural and semantic properties which can be used for permanent control of the logical integrity and consistency of the thesaurus terminology system. Such structures and formal means are provided by the ontology included in the program shell. This ontology describes a conceptual scheme of the thesaurus and therefore it is called thesaurus representation ontology.

This ontology is based on international and Russian standards \[9, 10\] regulating the structure of monolingual and multilingual information retrieval thesauri, the set and properties of basic entities and relations between them, therefore it provides all necessary facilities for developing thesauri meeting the standards mentioned above.

Formally, the thesaurus representation ontology may be represented as follows:

\[ O_{th} = \{ C, R, T, D, At, P, Axt \} \]

where \( C = \{ T, S, S_K \} \) is a finite nonempty set of classes that represent the basic entities of a thesaurus; here \( T \) is a class of terms representing the concepts of the modeled subject domain, \( S_T \) is a class representing the sources of terms, \( S_K \) is a class representing a subarea of knowledge of the modeled SD.

\[ R = R^{TT} \cup R^{TST} \cup R^{TSK} \]

is a finite set of relations, where

\[ R^{TT} = \{ R_1^{TT}, ..., R_m^{TT} \} \subseteq T \times T \]

is a finite set of binary relations defined on terms,

\[ R^{TST} = \{ R_1^{TST}, ..., R_n^{TST} \} \subseteq T \times S_T \]

are binary relations linking the terms with their sources,

\[ R^{SKS} = \{ R^{SKT}, R^{SKS} \} \]

are binary relations serving for embedding a subarea of knowledge of the modeled SD in the thesaurus, where the relation \( R^{SKT} \subseteq T \times S_K \) links the thesaurus terms with the subareas of knowledge, and the relation \( R^{SKS} \subseteq S_K \times S_K \) defines the hierarchy of subareas of knowledge;

\( T \) is the set of standard data types;

\[ D = \{ d_1, ..., d_n \} \]

is the set of domains \( d_i = \{ s_1, ..., s_k \} \), where \( s_i \) is the value of the standard type string;

\[ At = \{ a_1, ..., a_m \} \]

is a finite set of attributes describing the declarative properties of the thesaurus basic entities and relations between them; the values of these properties are defined on the set \( T \cup D \); \n
\[ P = \{ P_1, ..., P_n \} \]

is the set of formal (mathematical) properties of relations \( R^{TT} \);

\( Axt \) is the set of axioms defining additional constraints on the links (relations) between terms.

Thus, the thesaurus representation ontology (see Fig. 1) includes classes describing its basic entities: terms, sources of terms (text documents and collections of text documents containing terms or their definitions) and subareas of knowledge related to terms. The ontology also represents relations that link the objects of classes listed above, descriptions of the properties of classes and relations, as well as the axioms defining additional semantics of relations. Besides, a set of domains \( D \), i.e. the sets of possible values of attributes of classes and relations is defined in the ontology. This allows us to reduce the number of errors while creating/editing a thesaurus.

Since the terms of IRT are usually divided into descriptors (preferred terms) and ascriptors (non-preferred terms or text entries which can be replaced by the corresponding descriptors during document indexing and retrieval), the class \( Tr \) ("Term") includes two subclasses \( Desc \) ("Descriptor") and \( Asc \) ("Ascriptor"), i.e. \( Tr = Asc \cup Desc \).

The main attributes of the class "Term" are \( Name, Language, Comments, Author of dictionary entry. Notice that Name of term can be specified by a word or word combination (phrase), Author of dictionary entry is given for the purpose of control of the process of a thesaurus collective development and can be excluded from its final version.

The classes "Descriptor" and "Ascriptor" being subclasses of the class "Term" inherit its attributes. Besides, the class "Descriptor" has the following additional attributes: \( Qualifier, Definition, Top term \).

\( Qualifier \) is a special label (a word or group of words) used with a term to distinguish homonymous terms (homographs).

\( Definition \) explains meaning of descriptor in its language. (Note that the presence of term definitions in a thesaurus allows it to be used not only as a facility for manual or automatic indexing but also as a source of systematized knowledge about a modeled SD.)

The attribute \( Top term \) indicates that a descriptor is on the upper (top) level of the hierarchy of concepts.

The class \( S_T \) ("Source of terms") is described by the following attributes: \( Name, Reference, Language, Type, Abstract, and URL \). The values of the attribute \( Type \) are book, monograph, scientific paper, documentation, textbook, manual, dictionary, thesaurus, Internet resource, collection of text documents, etc. If a source of terms is a collection of text documents, then additional attributes can be given for it showing the number of documents contained in this collection and the total number of word usages.

The class \( S_K \) ("Subarea of knowledge") has such attributes as \( Name \) and \( Description \) of SD.
Three types of relations are presented in the ontology. They are relations between terms \( R^{TT} \), between terms and sources \( R^{TST} \) and relations used for embedding a subarea of knowledge of the modeled SD in the thesaurus \( R^{TSS} \).

Relations between terms \( R^{TT} \) play the leading role in the thesaurus. They define the place of each term in the thesaurus terminology system and thereby specify its meaning.

To represent semantic relations between concepts expressed by descriptors, the hierarchic and associative relations and the equivalence relation are used. Note that the first two relations link descriptors of one language, while the last relation link terms of different languages.

The ontology includes the following hierarchic relations: BT (“Broader Term”), BTG (“Broader Term (generic)”), BTP (“Broader Term (partitive)”), BTI (“Broader Term (instance)”). BT is an undifferentiated hierarchic relation between two descriptors. The BT relation is set between descriptors when the volume of a concept of a lower level descriptor is included in the volume of a concept of an upper level descriptor. The BTP relation is set between descriptors when a lower level descriptor represents a part of object designated by an upper level descriptor. The BTI relation serves for linking two descriptors when one of them represents some concept while another descriptor represents an instance of this concept.

The RT (“Related Term”) relation is used for determination of arbitrary associative links between descriptors. This relation is used in the case when terms are semantically or conceptually associated but this association is neither hierarchic nor equivalence relation. If it is required to represent a more rich set of relations specific for the modeled SD, such relations can be introduced into the ontology instead of the RT relation.

To determine equivalence of descriptors related to different languages, the relation “Equivalent to” is used. Two terms in different languages are considered as equivalent if they have the same or partly the same meaning. If it is required to take into account an equivalence degree of descriptors in different languages, either the relation “Equivalent to” can be supplied with an additional attribute describing this characteristic or this relation can be replaced by a set of relations corresponding to various equivalence degrees.

To link descriptors and ascriptors which are terms of the same language, various synonymy relations are used. In particular, the relation “Use” is set between a descriptor and an ascriptor in the case when the first can unambiguously replace the second in all contexts. In all other cases it is recommended to use other synonymy relations, for example “Use combination of terms” or “Use alternative terms”.

To establish the links between terms and sources of terms, the ontology provides three relations: “Included into”, “Included into part of document” and “Defined in”. The first relation links a term with a source when it occurs in an arbitrary place of the text document. The second relation links a term with a source if it occurs in a subject index or glossary. The third relation serves for linking a term with a source which contains a definition of the term.

To link term-descriptors with subareas of knowledge related to terms, the relation “Related Subarea” \( R^{SKT} \) is used. Hierarchies on subareas of knowledge can be defined using the relation “Include” \( R^{SRS} \).

For any relation from \( R^{TT} \), one or several noncontradictory mathematical properties from the following set – symmetry, reflexivity, transitivity, antireflexivity, and asymmetry – can be defined.

Besides, for any relation \( R \) from \( R^{TT} \), the inverse relation \( R^{-1} \) can be defined. Semantics of this property is described by the following axiom:

\[
\forall x, y \in Tr, x \neq y, \quad R, R^{-1} \in R^{TT}, \quad R^{-1} = \text{inverse}(R):
\]

\[
R(x, y) \rightarrow R^{-1}(y, x).
\]

In accordance with the properties of relations and axioms described in the ontology, the following is true: if the relation \( R \) possesses the symmetry property, then for any terms \( x \) and \( y \) existence of association \( R(x, y) \) induces existence of association \( R(y, x) \); if the relation \( R \) is asymmetric, then existence of association \( R(x, y) \) between a pair of terms \( x \) and \( y \) bans existence of association \( R(y, x) \); if the relation \( R \) possesses the antireflexivity property, then existence of association \( R(x, y) \) bans existence of associations \( R(x, x) \) and \( R(y, y) \); if the relation \( R_2 \) is an inverse with respect to the relation \( R_1 \), then for any terms \( x \) and \( y \) appearance of association \( R_2(x, y) \) induces existence of inverse association \( R_2(y, x) \).

Mathematical properties and inverse relations are defined for many relations of the ontology \( O^{n}_h \). So, for hierarchic relations BT, BTG, BTP, BTI, the transitivity and asymmetry properties are defined, as well as the corresponding inverse relations NT (“Narrower Term”), NTG (“Narrower Term (generic)”), NTI (“Narrower Term (instance)”), NTP (“Narrower Term (partitive)”), “Equivalent to” and “Related Term” relations are declared as symmetric and antireflexive. For the synonymy relation “Use”, an inverse relation “Used for” is defined.

III. THE PROGRAM SHELL ARCHITECTURE AND FACILITIES FOR THESAURUS CREATION AND MAINTENANCE

The program shell (see Fig. 2) consists of a user interface, data warehouse containing the thesaurus representation ontology and thesaurus content, and special editors providing facilities for filling the thesaurus content and its additional adjustment to a subject domain.

To insert concrete terms, descriptions of their sources and subareas of knowledge related to these terms into the thesaurus content, as well as to set links between all these entities, the content editor is used which is run by the thesaurus representation ontology. This editor provides experts with a convenient web-interface for management of the thesaurus content. After inserting or editing terms, descriptions of their sources and subareas of knowledge and relations between them, new information becomes immediately accessible for users through the user web-interface.
Integrity and consistency of the thesaurus terminology system is provided by special mechanisms of inference and control of logical integrity of the thesaurus which are embedded in the content editor and work on the basis of axioms and descriptions of properties of the relations and classes of terms defined in the ontology. In particular, on this basis relations between terms are set correctly and automatic addition and/or deletion of such relations is performed if necessary.

To automatize and accelerate the thesaurus development, facilities for extraction of a subject lexicon from texts related to the thesaurus subject domain are included in the program shell [11]. These facilities can process both a single text and corpus of texts. They not only extract potential terms for thesaurus from a text (text corpus) but also supply developers with information about the frequency of occurrence of these terms in the text (text corpus).

If a developer is not satisfied with a current set of entities or relations of a thesaurus, he/she can use the ontology editor to perform its additional adjustment. Using the ontology editor, a developer (an expert) can do the following: rename a relation or a class of terms; add a new relation (including description of its arguments, attributes and mathematical properties); edit the properties of a term and sources of terms; add a new class of entities, etc.

IV. USER INTERFACE

A convenient access to the terms of the thesaurus and its other entities is provided by the user web-interface. In this interface, the thesaurus content is presented to a user in the form of a network of interconnected information objects: the thesaurus terms, descriptions of subareas of knowledge related to them as well as descriptions of the sources of the terms and their definitions.

When navigating through thesaurus, a user can select the required term, view its detailed description (thesaurus entry) and descriptions of the sources where the term and/or its definition appears as well as subareas of knowledge related to it.

A user can indicate the type of information he is interested in, for example, all terms, only the descriptors, ascriptors or sources of terms. As a result, a user gets a full list of objects of the given type presented in the thesaurus. This list is alphabetically ordered and represented in a form of an html-page containing a set of hyperlinks to these objects. Information about a concrete object and its links is also represented in the form of an html-page (see Fig. 3). At the same time, the objects linked with the selected object are represented on its page by their hyperlinks. Using these hyperlinks, the user can proceed to a detailed description of these objects.

Further navigation through the thesaurus content is a process of transition from some element of the thesaurus to others along the links presenting the semantic (between terms) and bibliographic (between terms and sources) relations.

For example, when we look through information on the term machine aided translation from the Russian-English thesaurus on computational linguistics (to be dwelt on in the next section), we can see both the values of its attributes and its connections with other terms and the sources of the terms (see Fig. 3). In particular, we are able to look through all definitions of the term and find out the name of the author (developer) of this dictionary entry. Using the presented connections as elements of navigation, we can proceed to browsing its synonyms, its equivalent in the other language, concepts (descriptors) located in the hierarchy above and below this term, as well as descriptions of the sources of its definitions.

In addition to navigation through the thesaurus content, the system provides a simple search of terms and descriptions of sources of terms using keywords and an advanced search taking into account the type and properties of a term (a source of terms), as well as filtration of terms (sources of terms) with respect to certain properties such as correlation with a subarea of knowledge, authorship of a certain expert (person), occurrence in certain source of terms, etc.

V. AN EXAMPLE OF THE PROGRAM SHELL APPLICATION

The program shell has been successfully applied to the development of an electronic Russian-English thesaurus on computational linguistics (CL) [12].

The thesaurus developed is an integral and consistent system of computational linguistics concepts which are linked together by semantic relations reflecting the position of each concept within this system.

The concepts contained in the thesaurus are represented by Russian and English terms.

All terms and relations of this thesaurus are instances of classes and relations of the thesaurus representation ontology. In accordance with this ontology, the thesaurus includes two types of terms: descriptors (preferred terms) and ascriptors (non-preferred terms, or text entries).

All descriptors of the thesaurus are supplied with definitions. Each descriptor is linked with its equivalent in the other language, its synonyms (asccriptors), relevant subareas of computational linguistics and allied sciences as well as with sources of terms, i.e. text documents or collections of text documents where this descriptor occurs or is defined.

The thesaurus contains two hierarchies of concepts: taxonomy and partonomy. In addition, descriptors are linked together by the relation of symmetric association.
To make the thesaurus easier to use, two relations, “DescriptorSource” and “AscriptorSource,” were inserted into the thesaurus representation ontology instead of the “Included into” relation. The former is intended for linking the source of terms with descriptors, and the latter, for linking the source of terms with ascriptors. Besides, instead of the “ Included into part of document” relation, two new relations were introduced into the ontology. They are “SourcePartDescriptor” and “SourcePartAscriptor,” which link, correspondingly, terms-descriptors and terms-ascriptors with the sources of terms where they occur in the index part or in the glossary.

The thesaurus covers with its terms all the key subareas of computational linguistics: theoretical computational linguistics, automatic text processing, speech technologies, corpus linguistics, linguistic resources (including computational lexiconology and lexicography), and CL applications (including information retrieval, Machine translation, information extraction, and text data mining).

For the purpose of systematization of the thesaurus terms, a hierarchy of CL subareas has been developed as well as of the areas allied with CL to which the thesaurus terms are related.

These subareas are ordered with the help of the “Include” relation.

The hierarchy of CL subareas looks as follows:

1. Theoretical CL
   1.1. Computational semantics
   1.2. Generative linguistics
   1.3. Speech signal processing
      1.3.1. Digital encoding, transmission and compression of a speech signal
         1.3.2. Computerized speech signal processing
      2. Natural language processing
         2.1. Automatic text processing
            2.1.1. Text segmentation
            2.1.2. Morphological analysis/synthesis
            2.1.3. Syntactic analysis
            2.1.4. Word-sense disambiguation
            2.1.5. Anaphora resolution
            2.1.6. Natural language generation
      2.2. Speech technologies
      2.2.1. Automatic speech recognition
2.2.2. Automatic speech synthesis
2.2.3. Voice biometrics
2.2.4. Voice control
2.3.1. Machine translation
2.3.2. Information retrieval
2.3.3. Information extraction
2.3.4. Summarization
2.3.5. Terminology extraction
2.3.6. Automatic text classification
2.3.7. Human-computer interaction
2.3.8. Question answering
2.3.9. Text data mining
2.4. Corpus linguistics
2.4.1. Text corpora
2.4.2. Speech corpora
2.5. Systems and methods performance evaluation
3. Linguistic resources
3.1. Computational lexicology and lexicography
3.2. Ontologies and thesauri
3.3. Information systems and resources on linguistics
3.3.1. Language documentation

The Russian terms were largely taken from a collection of papers presented at the “Dialogue” International conferences on computational linguistics [13] held from 2000 to 2010 since it is an acknowledged source of Russian terms in real use. Included in the program shell were facilities for the extraction of subject lexicon (see Section 3) used for obtaining, from the collection of papers, a list of words and collocations statistically significant for this subject domain. After that these words and collocations, considered as candidates in terms of a given SD, were processed (filtered) by CL experts and inserted into the thesaurus with the help of the content editor (see Fig. 2).

Also used for obtaining Russian terms were subject indexes, glossaries and texts of Russian textbooks and monographs, a collection of academic texts on corpus linguistics as well as encyclopedias, encyclopedic dictionaries and Internet resources relating to computational linguistics.

For the English part of the thesaurus, English equivalents were selected from available English-language sources on CL. Substantially, indexes of several modern and most reliable English-language reviews and textbooks, such as the handbook [14], were used for these purposes.

In order to prove the relevance of the terms inserted into the thesaurus and to acquaint users with the practice of their use, the links of each term with its sources, i.e. text documents or collections of text documents where this term occurs, are presented.

An example description of a source is given in Fig. 4. Presented in the description are the following features of the source: its short name (Handbook Mitkov R. (ed) 2003), type (book), language (English), reference (The Oxford handbook of computational linguistics (Ruslan Mitkov ed.) N.Y.: Oxford university press, 2003), and a brief summary (abstract). In addition, the source is linked with terms occurring in it by the relations “DescriptorSource” and “AscriptorSource,” which indicate the frequency of the term occurrence in the source, and by “SourcePartDescriptor” and “SourcePartAscriptor”, showing the term occurrence in index or glossary of the source, which is indicative of the importance of the term and increases the degree of belief in it. Terms-descriptors supplied with definitions-interpretations are linked with a definition source by means of the relation “Defined in”. The number of the descriptor definition in its description is given as an additional attribute of this relation.

At present the thesaurus contains about 1700 terms linked by about 8000 semantic relations; it also includes the descriptions of about 190 term sources and 50 CL subareas.

VI. CONCLUSION

The paper has presented a program shell for building and editing multilingual electronic thesauri, whose general structure as well as composition of the thesaurus entries and the set of relations between the terms meet the international and Russian standards. The shell is ontology-based, which provides not only the possibility of extending the set of relations between the terms, integrity and consistency of the terminology system of the created thesaurus but also a convenient access to its content.

The shell allows one to build dual-purpose thesauri, i.e. thesauri intended not only for document indexing and information retrieval but also for using by people (for example, students) who just want to turn to the system of concepts of some subject domain.

The program shell provides developers with advanced facilities of adjustment based on the ontology, so it can be applied to building multilingual thesauri for any languages and subject domains.

To support the use of a thesaurus by third party applications, the program shell includes an application programming interface (API) providing access to descriptions of terms and other entities of the thesaurus. In particular, for any term-descriptor one can get its dictionary translation (its equivalents in other language), a list of term-descriptors linked with it by hierarchical relations, a list of synonyms, a list of subareas of knowledge related to it, and a list of sources of terms including this term. Besides, the program shell supports import of a thesaurus in standard formats such as XML and RDF.

The program shell has been successfully used in the development of the Russian-English thesaurus on computational linguistics representing integral and consistent system of the concepts of the computational linguistics domain linked by semantic relations that specify for each concept its position in this system.

REFERENCES


